Intermec

User’s Guide

WWAN Toolkit

Intermec

User’s Guide

WWAN Toolkit

Intermec Technologies Corporation

Worldwide Headquarters
6001 36th Ave.W.
Everett, WA 98203
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 noticed and do not represent a commitment on the part of Intermec Technologies Corporation.

Document Change Record

This page records changes to this document. The document was originally released as version -001.

Version Number	Date	Description of Change
003	10/2006	Revised to support software release 5.2. Added references to the CN3 mobile computer.

Contents

Before You Begin

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

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.

Services	Description	In the USA and Canada call 1-800-755-5505 and choose this option
Factory Repair and On-site Repair	Request a return authorization number for authorized service center repair, or request an on-site repair technician.	1
Technical Support	Get technical support on your Intermec product.	2
Service Contract Status	Inquire about an existing contract, renew a contract, or ask invoicing questions.	3
Schedule Site Surveys or Installations	Schedule a site survey, or request a product or system installation.	4
Ordering Products	Talk to sales administration, place an order, or check the status of your order.	5

Outside the U.S.A. and Canada, contact your local Intermec representative. To search for your local representative, from the Intermec web site, click **Contact**.

Who Should Read This Manual

This *User's Guide* is for the person who is responsible for installing, configuring, and using the Intermec WWAN Toolkit.

This *User's Guide* provides you with information about the features of the WWAN Toolkit, and how to install, configure, operate, maintain, and troubleshoot it.

Before you work with the WWAN Toolkit, 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:

Document Title	Part Number
WWAN Toolkit C++ API Programmer's Reference Manual	*1-960651-XX*
700 Color with Windows Mobile 2003 Quick Start Guide	*962-054-XXX*
700 Color CDMA Radio Setup Quick Start Guide	*962-054-XXX*
700 Series Mobile Computer User's Manual (Windows Mobile 2003)	*961-054-XXX*
CN3 Mobile Computer User's Manual	*935-003-XXX*

The Intermec web site at www.intermec.com contains our documents (as PDF files) that you can download for free.

To download documents

1. Visit the Intermec web site at 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.

1 Introduction

This chapter contains the following:

- Product Overview
- The WWAN Toolkit Core API
- The WWAN Toolkit Connectivity API

Product Overview

The Intermec WWAN Toolkit is a powerful Software Development Kit with which it is easy to build mobile applications that facilitate communication across cellular networks. It gives the developer the freedom to focus on the core design task at hand without having to consider issues in the periphery, such as choosing the correct communication standard or keeping the application up-to-date with recent OS upgrades.

The WWAN Toolkit supports both the GSM and CDMA standards, and thereby allows for the development of applications in a global context. It works on all Intermec Mobile Computers equipped with a WWAN radio. It also incorporates LAN and WLAN functionality, helping developers to make their mobile applications even more versatile.

TheWWAN Toolkit consists of two Application Programming Interfaces (APIs). The WWAN Toolkit Core API is targeted for applications that require only WWAN communication. The WWAN Toolkit Connectivity API on the other hand, has a smaller subset of WWAN methods but adds the capability to automatically or manually switch between WWAN/ WLAN and LAN communication.

The two APIs cannot be used in parallel, so the choice of which API to use must be made early in the development process. As a general rule, you should utilize the Core API when you need a fully-featured WWAN API, and choose the Connectivity API when you need network switching capabilities.

Note: Even though the WWAN Toolkit may in some cases operate together with other applications or SDKs that interact with the WWAN radio Intermec does not recommend this as it might cause unpredictable results..

WWAN Toolkit Core API

The WWAN Toolkit Core API offers a rich API for WWAN interaction. At the same time it simplifies the development procedure by being less complex than the majority of low level APIs currently available on the market. The user does not need to understand complex AT protocols and the toolkit works across various operating system platforms. The various functions of the API relate to the following main categories:

- Power Management
- RAS and Connection Manager Data connectivity
- SMS
- Voice Calls
- SIM Card Management
- Phone Book
- Network Information

- Radio Information

Note: The Core API does not contain any functionality for WLAN interaction but this can be achieved by utilizing other Intermec APIs such as 80211api.dll and 80211pm.dll in conjunction with the Core API. The WLAN APIs are described in the 700 series manual and the IDL documentation. For more information, visit **www.intermec.com/idl.**

WWAN Toolkit Connectivity API

The Connectivity API is a library built upon the WWAN Toolkit Core and various other Intermec libraries. The API, which has been cut down to a small number of methods, does not replace the WWAN Toolkit but is merely an alternative way of getting access to your networks. Despite its relative simplicity, it still supports the following functionality:

- Automatic or Manual Switching between WWAN/WLAN/LAN
- Profile and Connection Information
- Device Information
- Event Handling
- Telephony
- Messaging (SMS)

API documentation is distributed as separate manuals for each programming language.

2 Getting Started

This chapter covers the following topics:

- Installing the WWAN Toolkit on a PC
- Installing the WWAN Toolkit on your Mobile Computer
- Developing Applications in Different Environments
- Running the Toolkit in your Development Environment (Emulator)
- Guidelines for Using the Core API

Installing the WWAN Toolkit on a PC

To install the WWAN Toolkit on your development PC, simply insert the product CD in the CD drive and follow the on-screen instructions. If the installation does not start automatically, open your browser and execute the Setup.exe file located on the disk.

The installation requires 30 MB of free disk space on your “Windows” drive. Typically, program files will be located in:

C:/Program Files/Intermec/WWAN Toolkit V5.2

Note: Throughout this document, this folder will be referred to as [INSTALLDIR]. For a complete list and descriptions of all the files installed, refer to Appendix A.

Also note the following:

- Upgrading from older versions of the WWAN Toolkit requires a previously installed version of the software.
- The toolkit only supports the **Windows XP** and **Windows 2000** platforms.

Installing the WWAN Toolkit on Your Mobile Computer

Please note that the WWAN Toolkit currently supports the following platforms:

- 760 GSM/CDMA running Pocket PC 2003 or Windows Mobile 2003.
- 761A GSM/CDMA running Pocket PC 2003 or Windows Mobile 2003.
- 761A GSM/CDMA running Windows Mobile 2003 Phone Edition.
- 761B GSM/CDMA running Windows Mobile 2003 Phone Edition.
- CN3 GSM/CDMA running Windows Mobile 5.0 Phone Edition.

To install the WWAN Toolkit on your mobile computer

1. Copy the WWANToolkit5_2.ARM.cab file located in [INSTALLDIR]\Cab Files\ to your mobile computer. If you are planning to develop applications for data communications on the 760 GSM unit or if you wish to use the WWAN Toolkit Connectivity API, copy Modems_MC45_All.cab as well. Refer to Appendix A for a complete list and descriptions of all the files installed.
2. Execute the file(s) on your mobile computer.
3. Please observe additional installation steps required for the different development environments in the sections ahead.
4. Warm-boot the device.

Developing Applications in Different Environments

This section gives a brief description of how to get started developing WWAN applications in different programming environments.

If you are planning on building a data communications application you will need to start by obtaining certain network information:

- Contact your GSM/CDMA network operator and request the data you need to be able to connect to the WWAN network. On GSM this is typically an Access Point Name (APN). For instance, CDMA often requires a user name and a password.
- If you are planning to use the Connectivity API and WLAN, contact your IT department to obtain SSID and security preferences for your local WLAN.

Since each API and programming language has its own reference manual, this guide does not contain any API listings. Please refer to the main page of the documentation (which can be located via the Start Menu on your PC) for more information.

Source code samples can be found in [INSTALLDIR]\Samples\[API]\[Language] and are described in the "Samples" section of this guide.

C++ Development

The WWAN Toolkit supports development in Embedded Visual C++ 4 and Visual Studio 2005 (CN3 devices only).

Core API Development

1. Add a link reference to the WWANToolkit.lib file located in [INSTALLDIR]\WWANToolkitCore\C++\lib.

   To add the reference, select **Project** > **Settings** in the menu bar. Click the **Link** tab, and in the Category drop-down menu, select **Input**. Add the name of the .lib file to the Object/Library Modules field and point out the path to the .lib file in the Additional Library Path field.

2. Click the **C++** tab, and select **Preprocessor** from the Category drop-down menu. Point out the header files by adding the API directory to the Additional Include Directories field.

Connectivity API Development

1. Add a link reference to the ConnectivityApi.lib file located in the [INSTALLDIR]\Connectivity API\C++\lib directory.

   To add the reference, select **Project** > **Settings** in the menu bar. Click the **Link** tab, and in the Category drop-down menu, select **Input**. Add the name of the .lib file to the Object/Library Modules field and point out the path to the .lib file in the Additional Library Path field.

2. Click the **C++** tab, and select **Preprocessor** from the Category drop-down menu. Point out the header files by adding the API directory to the Additional Include Directories field.

3. Add the header files to your project. The header files are also located in the API/include directory. The following header files should be included:

ConnectivityAPI.h	Methods
ConnectivityErrorCodes.h	Error code definitions
ConnectivityObjects.h	Objects and general definitions

.NET Development (C# and VB.NET)

The WWAN Toolkit supports development with Compact Framework V1.0 and V1.1.

Core API Development

1. Include the .NET wrapper dll in your application directory on the device. This is called Intermec.Communications.WWAN.dll and is located in [INSTALLDIR]\Core\DotNet\lib.

2. To develop a .NET application using the Core API you need to add a reference to the wrapper dll. The reference is added by opening up the references dialog box and selecting “Intermec WWAN Toolkit” from the drop-down list.

Note: C# developers can also use the Visual Studio 2003 plugin as described in Chapter 4, “Tools.”

Connectivity API Development

1. Include the .NET wrapper dll in your application directory on the device. The Connectivity API dll is called Intermec.Communications.ConnectivityDotNetAPI.dll and can be found in [INSTALLDIR]\Connectivity\DotNet\lib.

2. To develop a .NET application using the Connectivity API you need to add a reference to the wrapper dll. The reference is added by opening up the references dialog box and selecting “Intermec.Communications.ConnectivityDotNetAPI” from the drop-down list.

Java Development

Java development for Windows Mobile does not require a specific development environment, which is why the setup procedure is not described in detail here.

Core API Development

1. Copy WWANjni.dll to the Windows folder of your device, the dll is located in [INSTALLDIR]\WWANToolkitCore\Java\lib.
2. Remember to include the wwantk.jar file in your application folder on the device and to add the file to the class path when using the Core API. Add a reference to the wwantk.jar file located in [INSTALLDIR]\WWANToolkitCore\Java\lib. Add an import statement to your source pointing to the com.intermec.wwantoolkit package.

Note: Users who have developed applications with older WWAN Toolkit versions can still use the com.intermec.c700.wwantoolkit.jni package since this is still supported. However, the wwantoolkit package adds object-oriented Java classes and is easier to use.

The com.intermec.c700.jni package is not documented in this release.

Connectivity API Development

1. Copy ConnectivityJNI.dll to the Windows folder of your device, the dll is located in [INSTALLDIR]\ConnectivityAPI\Java\lib.
2. Add a reference to the connectivity.jar file located in [INSTALLDIR]\ConnectivityAPI\Java\lib. Add an import statement to your source pointing to the com.intermec.wwantoolkit.connectivity package. Remember to add the .jar file to your class path as well as copying it to the device.

Running the Toolkit Core in your Development Environment (Emulator)

The WWAN Toolkit adds the feature to test your applications in your development environment as if they ran on a mobile computer (all methods return TRUE without performing any actual radio interaction).

The emulator .dlls are located in [INSTALLDIR]/WWANToolkitCore/Emulator.

Embedded Visual C++ 4.0

1. Add the headers (found under [INSTALLDIR]/WWANToolkitCore/C++/Include) and lib ([INSTALLDIR]/WWANToolkitCore/Emulator) to your project.
2. Start the Windows CE Remote File Viewer (cefilevw.exe is normally located in the folder C:\Program Files\Windows CE Tools\Common\Platman\bin\). Connect to Pocket PC 2003 Emulator.
3. Open up the Windows folder.
4. Select **File** > **Export File**, browse to the emulator dll and select it.
5. Start your project with the emulator running in the background.

Visual Studio .NET 2003

1. Add a reference to the WWAN Toolkit .NET wrapper.
2. Click the right mouse button in the Solution explorer of your project and select **Add** > **Add Existing Item**.
3. Browse to the emulator dll, select the dll.
4. Make sure that the build action is set to “Content” for the dll in the properties window.
5. Start/Run your project and select “Pocket PC 2003 Emulator” as your target when deploying.

Java Environments (WSDD)

1. Add a reference to WWANToolkit.jar.
2. Add a system property called “emulate” that is set to ON. With “emulate” the JNI DLL is never accessed and the JNI access methods return canned responses. -Demulate=off will turn emulation off. Or, just remove -Demulate from the JVM startup. To set a system property in command prompt, add “`-Demulate=ON`” to the argument list.

Guidelines for Using the Core API

When the WWAN Toolkit is used to interact with the modem, use the following as a guide to correctly set up the modem and the Toolkit Core.

Initiating the Toolkit and Radio

Power on the radio

The radio must be powered on before it can be used, which is done by calling the `WWANRadioPowerOn()` method. This method will take approximately 20 seconds to execute and when it returns the radio is powered and ready.

Set up the toolkit

To set up the internal states and variables, the WWAN Toolkit must be set up. To use the cross-platform feature of the toolkit, it should be set up with `SetupWWANToolKit()`. This method will automatically detect the carrier and radio type, and set it up so it can be used. This will also set up multiplexing on some radios, providing the application with the means to communicate with the modem during a WWAN connection.

Ensure that the modem is responding

Use the `IsModemReady()` function.

Check if the PIN code is required – GSM only

SIM cards that are used in GSM network can be locked, and can only be used once they have been unlocked. To check if the SIM card is ready to be used, call the `CheckSimPin()` method . This method can return:

- `PIN_READY`, the SIM card is unlocked and ready to be used
- `PIN_SIM_PIN`, the SIM card is locked, and can be unlocked by calling `EnterSimPin()`
- `PIN_SIM_PUK`, the SIM card is locked, and can only be unlocked by calling `EnterPukNumber()`

Follow up with `CheckSimPin()`, to ensure that the PIN or PUK entry was successful. If not, use the entry functions above. For more information about PIN codes, please see the Glossary chapter or the mentioned API methods.

Note: Entering the incorrect PIN code more than three times will lock the device. In this event, a PUK code is required to unlock it.

Select network operator – GSM only

Use the `AutomaticNetworkRegister()` method to let the radio select which network to use. The choice is based on information in the SIM card provided by your network operator. It may also be required to select the operator. If this is the case, see the "Network Selection Functionality" section in the *WWAN Toolkit C++ API Programmer's Reference Manual*.

Setup event reporting

All applications that require events from the toolkit, must "register their interest" through `SetModemEventReporting()`. This method will make sure that the application can receive events concerning the network, the connection process and incoming voice calls and SMSs.

Error handling

Errors reported by methods in the Toolkit can be retrieved through two methods: The `GetLastWWANErrorCode()` will return the last known error code and `GetLastWWANErrorString()` will return a string representation of the last error code. Using these methods will help applications handle the problems and notify the user.

Shutting Down the Toolkit and the Radio

Before powering off the radio, the toolkit must be cleanly exited. The following describes the advised steps to do this.

Shut down the Toolkit

To shut down the toolkit, call the `ShutDownToolKit()` method. This will close down all internal variables and also make a clean exit from the network.

Power off the radio

Use the `WWANRadioPowerOff()` to power of the radio. After this method has been called, no other Toolkit method should be called except for `WWANRadioPowerOn().`

About Data WWAN Connections

There are two types of data connections and two different methods to establish them.

Types:

- Packet Switched Data – This is done through the GPRS and 1xRTT networks that exist within the framework of most GSM and CDMA networks. This is most often billed based on the amount of data transmitted and not the length of the transmission.
- Circuit Switched Data – This is done through a normal dial-up connection using the GSM and CDMA networks. This is most often billed based on the duration of the transmission (call) and not the amount of data transmitted.

Methods:

- RAS – This method is the core communication module for Windows. This will establish a TCP/IP stack that is capable of communicating with the network or the Internet. After a connection has been established, applications can in their turn open sockets to servers that are available on the network. The RAS API will not display any pop-up windows about the current state of the connection process.
- Connection Manager – This exists on top of the RAS API. Applications that are “Connection Manager aware” can use the same connection instead of establishing an entirely new one. One example of such applications is the Pocket Internet Explorer. Using Connection Manager will also display a bubble on top of all applications showing the connection process, note that this bubble can be misused by certain users to enter faulty settings and disrupt the application.

Creating a WWAN Connection Entry

The toolkit can create connection entries but it can also use settings configured by the user. The steps below show how to create a PSD connection entry:

Create a dialling location

Use `CreateDiallingLocation()` to create the location that you are dialling from. This is commonly a “g,g,g” entry within the PocketPC dialling location pages.

Create a RAS entry

Use `CreateRasEntry()` to create the RAS entry. This requires all the parameters for the PSD network. If you are unsure of those parameters, please contact your network representative. In most cases, it is not necessary to set up all parameters and most of them can be left blank. An application should only have to set up the connection parameters once during its lifetime, but it is also possible to create them before and removing them afterwards. There is one major difference between connecting to a PSD network in GSM and CDMA:

- When using GPRS, the phone number should be *99***1# or *99#. The GPRS APN can be set up using the extra dial commands. Please refer to the complete method description in the API manual for more information.
- When using 1xRTT, the phone number should be #777.

Acknowledge this when creating a Circuit Switched connection:

- When using GSM or CDMA, the phone number should be the number of the dialup server.

Link the new RAS entry to Connection Manager

This is only required if the connection will be using Connection Manager instead of RAS, as explained at the start of this section.

Use `AddEntryToConnMgr()` to link the RAS entry so it can be used by the Connection Manager. Linking it will also display it under "Connections" in the Settings menu.

Removing a WWAN connection

The toolkit will not clean up the Windows registry when it is shut down, so connections that have been created will remain when the toolkit is set up later on.

Remove the RAS entry link from Connection Manager

This is only required if the `AddEntryToConnMgr()` method has been used. Use `RemoveEntryFromConnMgr()` to remove the link. This will also remove the entry under "Connections" in the Settings menu.

Deleting the RAS entry

To delete the RAS entry, use `DeleteRasEntry()`.

Deleting the dialling location

Use `DeleteDiallingLocation()` to delete it from the dialling location list.

Establishing the WWAN connection

Once the connection parameters have been set up, the connection process can be started.

Checking for signal strength

Always make sure that there is a cellular network available before attempting to connect. Use `SignalStrength()` to determine if the signal is strong enough to successfully establish a connection. A signal strength of 0 means that there is no network available and all connection attempts will fail.

Checking for PSD network

This is only needed if the connection is a Packet Switched data connection. The `GetGPRSAttach()` method will confirm the availability of a GPRS or 1xRTT network. In GSM networks, the radio must attach to the GPRS network before it can be used; `SetGPRSAttach()` will attach it to the GPRS network. In CDMA networks, the radio is automatically attached when started so there is no need to call `SetGPRSAttach()`. After attaching to the PSD network, check the attach status through `GetGPRSAttach()`. If there is no PSD network available, all PSD connection attempts will fail.

Setting up GPRS APN (GSM ONLY)

This is only needed if the connection is a PSD connection used in a GPRS network. The APN is set up through the `GPRSSetupContext()` method. The APN can be set up using that method or adding it as a extra dial command in a RAS entry.

Make the WWAN Connection

Establishing connections can take a long time, and therefore all of the methods included in this process are *asynchronous*, which means that they will start the connection process but the connection is not yet active when the method returns. It is important to remember that the connection attempt can fail even if the initial method call is successful.

To establish a Connection Manager connection, use the `DoDialUpConnect()` method; to establish a RAS connection, use `DoRASConnect()`. Those methods will start the connection process, and will only return FALSE if there is a major problem with the connection parameters. The Toolkit will send events until the connection has either been established or dropped, and applications must await the appropriate connection event before they can start communicating with servers. To ensure that the connection has been fully established, it is considered best practice to check the toolkit event call-backs for the RAS event RASCS_Connected and/or the Connection Manager event CONNMGR_STATUS_CONNECTED.

If the connection has been up for a while it is best to make sure that it is still functioning as it should. The Toolkit has a method to send an ICMP ping to any server, to make sure that the connection is up. Use `ActiveWwanAvailable()` to determine the WWAN connection status.

Dropping the WWAN connection

To terminate a Connection Manager connection, use `DoDialUpHangup()`. To terminate a RAS connection use `DoRASHangup()`.

3 Using the Connectivity API

This chapter contains the following topics:

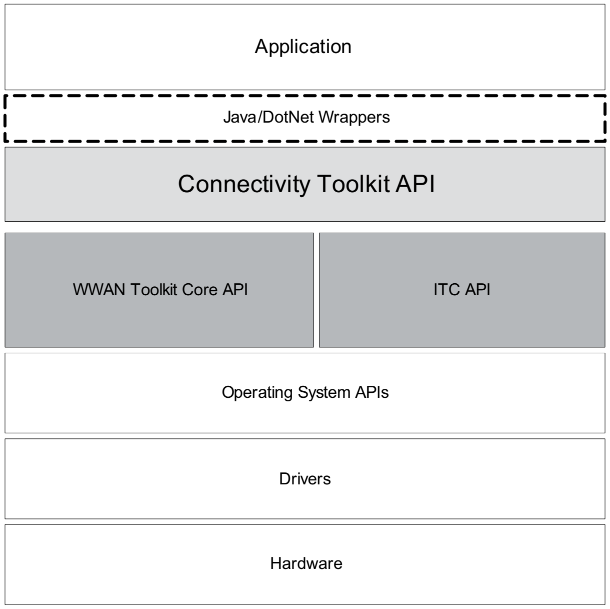
- Introduction to the Connectivity API
- Use Case Scenarios
- Guidelines for Using the Connectivity API

Introduction to the Connectivity API

The Connectivity API provides the user with a simple way of automatically or manually switching between PSD/CSD/WLAN/LAN networks. IP addresses are not persistent, so whenever the carrier is switched the user receives a notification via the event handler. This works quite well with a lot of queuing systems as they use what is available at the moment.

The API is intelligent enough to always select the cheapest media available (for automatic switching): LAN will always be preferred to WLAN and WLAN over WWAN. To keep the API easy to use the number of methods has been kept at a minimum. Connectivity parameters are defined in XML documents (known as Profiles and Connections) and can be reconfigured without recompiling the source code.

The Connectivity API is built upon existing Intermec components such as the WWAN Toolkit Core, the 80211 APIs and NetTools:



The Connectivity Toolkit API and Adjacent Layers.

The Connectivity API has been designed to correspond to the different scenarios that an application can go through during a workday or work session. These different scenarios are described as “profiles” in the API. Each profile can contain a number of connections; a connection in its turn can be made through a LAN, WLAN or WWAN.

The API will read two XML files containing this information and use it to switch between the different networks. The API will take care of all internal checking, such as checking the availability of a network, and relies on parameters to know when to switch to a different network. All the application has to do, besides initializing the API, is to control when a different profile should be used.

Use Case Scenarios

In order to accurately map out the different profiles an application will be using, the user is recommended to construct case scenarios for it. That way, it is easier to see when a specific network is needed and when it is not.

Example Scenario

Let us imagine the daily routines of an employee at a package delivery company:

Start of the day

– The messenger comes into the office and picks up the mobile computer from the docking station. The unit is fully charged and has been updated with the latest delivery information and software during the night.

In the depot

– The messenger scans each package.

– The package ID is sent to the database through WLAN.

On the road

– In his truck, the messenger keeps in touch with his employer via voice calls and SMSs.

At the customer's site

– The customer signs off each package electronically.

– The mobile application updates the status of each delivered package on the server through WWAN.

– The application downloads new delivery and pickup information.

Back at the office/depot

– The messenger heads out for another round of deliveries and pickups, or docks the mobile computer and goes home for the day.

Based on this scenario, it seems appropriate to set up the following profiles for the application: "Depot" (WLAN/LAN), "On the road" (WWAN) and "Customer site" (WLAN/WWAN).

Note that WWAN has been excluded from the "Depot" profile; this is to avoid unnecessary costs. Once the application has confirmed that all packages have been scanned, the messenger is ready to move out, and the application tells the API to switch to the "On the road" profile. This profile allows only for WWAN connections and will not try to connect to any WLAN that will coincidentally come into range. When the messenger has arrived at the customer's site, the application loads the appropriate profile and the messenger can once again send and receive data from the HQ server via WLAN, if this is available.

For more information on setting up profiles, refer to the section describing the Connectivity Profile Manager in the “Tools” chapter.

Guidelines for Using the Connectivity API

This section describes how to implement some typical features of a WWAN application using the Connectivity API. It is assumed that you are already familiar with the Profile Manager Tool and know how to deploy profiles to your mobile device. If this is not the case, refer to the “Tools” chapter in which the functionality of Profile Manager is thoroughly described.

Initializing the API

Before anything else can be done you need to initialize the API. This is done by calling the `Initialize()` method.

Automatically Switching Between Connections

Automatic switching is performed by simply calling the `SetProfile(L“MyProfile”)` method, where `“MyProfile”` represents the name of the connection you want to use. After making the SetProfile call you need to monitor the events from the switcher: a CONNECTIVITY_EVENT_SWITCHER_SUCCESS event indicates that the switcher has found an available connection and connected through it. The CONNECTIVITY_EVENT_CONNECTION_CONNECTED event that precedes the “switcher success” event will tell you the index of the connection being used. This index can be used with the `GetConnection`() method to obtain the name of the connection.

Locking a Connection

When you want to force the switcher to stay connected to a particular connection (e.g. when transferring data) you should call the `Lock(NULL`) method. By supplying a null value as the parameter you tell the API that you want to use whatever connection is used at the moment until `Unlock()` is called. The call will fail if there is no current connection available.

Unlocking a Connection

After calling the `Lock(null)` method you must call the `Unlock()` method to release the lock. If you do not, the switcher will stay connected to the same connection until it loses the signal. When the signal is lost the switcher will not try to connect to a new connection until the old one has been unlocked.

Manually Selecting a Connection

The `Lock()` method can not only be used to lock the currently used connection, it can also be used to force the switcher to try to connect to a specific connection.

Locking to a specific connection is done by entering the name of the connection; `Lock(L"MyPSD Connection")`.

The selected connection is released by calling `Unlock()`.

Powering on Radios Without Connecting

When creating a profile using the Profile Manager tool you have the ability to select the option “Disable automatic switching” and the power management option set to “Always On”. In this state when `SetProfile(L"MyProfile")` is invoked, the switcher will power on radios used by the connections included in the profile, but it will not connect to any connections when they become available. This is useful when you want to keep switching manual (with the Lock/Unlock methods) and want to minimize the time required to establish a connection. Remember that this will of course increase power consumption.

Using SMS and Voice Functionality

SMS and voice functionality can only be used when the WWAN radio is powered on. The WWAN radio is powered on whenever you load a profile that contains a WWAN connection and the profile hasn’t got the “Suppress SMS/Suppress Voice” flags set. You can also load a connection (Lock) that utilizes the WWAN radio to turn on the radio.

By monitoring the CONNECTIVITY_EVENT_HARDWARE_PHONE_AVAILABILITY event you will be notified when the phone is ready to be used.

SIM PIN Security

The Connectivity API can only unlock a SIM card, it cannot change the PIN code or disable the PIN. Whenever the Connectivity API sends the event CONNECTIVITY_EVENT_HARDWARE_WWAN_PIN_REQUEST you have to call the `Authenticate(…)` method or the powering on of the WWAN radio will be halted until a valid PIN/PUK code has been entered.

Powering Off All Radios

By calling `SetProfile(NULL)` you tell the API to shut down all active radios. This is particularly useful when entering a hospital, airplane or any other vehicle or building that requires broadcasting electronics to be turned off. Under normal circumstances you should switch to a profile that only uses radios that are allowed in the particular environment that you are working in. Radios are switched on again when `SetProfile(L“MyProfile”)`or `Lock(L”MyConnection”)` is called.

Battery Levels

By monitoring the battery status events your application can keep track of the main battery status. It is recommended that you switch off all radios whenever you receive a “LOW” level indication from the API and charge the batteries as soon as possible.

External Vs Battery Power Management

The Connectivity API will signal whenever it switches from battery to external power. However, this is not used to detect Ethernet availability since the Connectivity API does not know whether the power comes from a docking station or from an external power cord.

Deinitializing the API

Before you close down your application it is important that you call the `Deinitialize()` method. .NET users should avoid putting this call in the `Dispose()` method as this might prevent it from being called when required.

4 Tools

The WWAN Toolkit comes with a number of useful development tools, that are described in this chapter.

- Device Info
- Connectivity Profile Manager
- WWAN Wizard – Visual Studio 2003 Plugin for C# .NET

Device Info

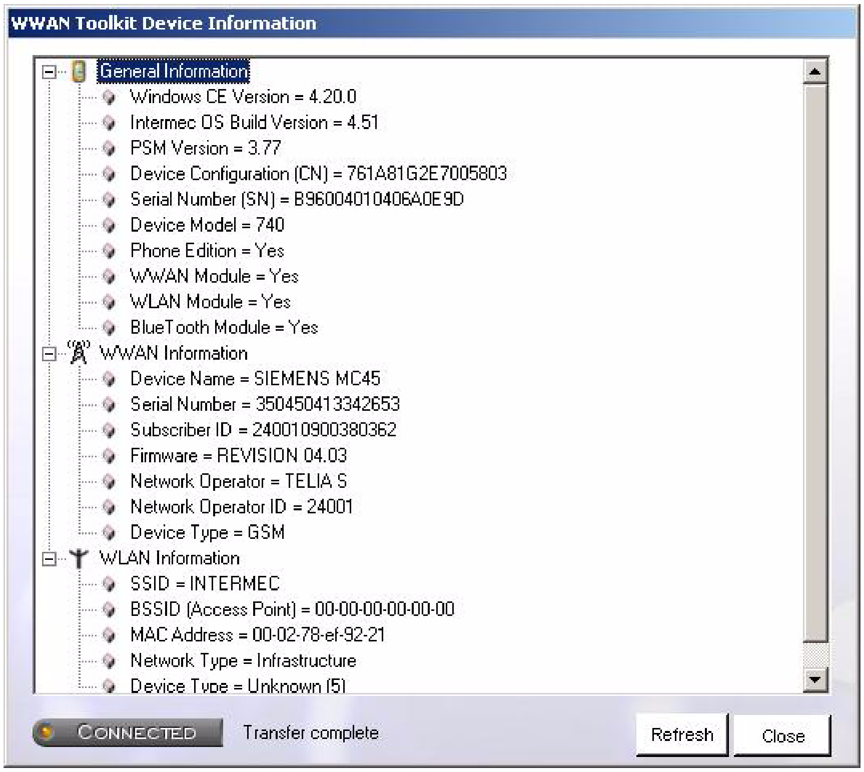
This tool is used to interrogate the device about its current configuration.

The tool can be used in situations where the user (or Intermec support) requires additional information about a device’s configuration. It obtains this information by using the “GetXXXInfo” methods in the Connectivity API.

To obtain information about your device

1. Start the tool via the **Start Menu** > **WWAN Toolkit** > **Tools** > **Device Information**.
2. Put the device in the cradle and make sure it is connected via Active Sync (the icon located in the bottom left corner of the application should read “Connected”).
3. Press the **Refresh** button. The process takes 60 seconds to execute, do not remove the device from the cradle during this time.
4. When the process is complete, you can expand the tree view to see detailed information of your system.

Note: The configuration information is also stored as a file on your PC (“DeviceInfo.txt”, located in the Tools folder). Include this file in support errands, as it will allow Intermec support staff to resolve issues more efficiently.



- General Information
  - Windows CE Version = 4.20.0
  - Intermec OS Build Version = 4.51
  - PSM Version = 3.77
  - Device Configuration (CN) = 761A81G2E7005803
  - Serial Number (SN) = B96004010406A0E9D
  - Device Model = 740
  - Phone Edition = Yes
  - WWAN Module = Yes
  - WLAN Module = Yes
  - BlueTooth Module = Yes
- WWAN Information
  - Device Name = SIEMENS MC45
  - Serial Number = 350450413342653
  - Subscriber ID = 240010900380362
  - Firmware = REVISION 04.03
  - Network Operator = TELIA S
  - Network Operator ID = 24001
  - Device Type = GSM
- WLAN Information
  - SSID = INTERMEC
  - BSSID (Access Point) = 00-00-00-00-00-00
  - MAC Address = 00-02-78-ef-92-21
  - Network Type = Infrastructure
  - Device Type = Unknown (5)

CONNECTED Transfer complete

Connectivity Profile Manager

The Connectivity API is configured through XML files. These files are created with a tool called the Connectivity Profile Manager. A “profile” determines the overall behavior of the API and contains one or more connections. A connection contains data about a specific network (PSD/CSD/WLAN/LAN) such as APN, SSID etc.

Profiles are loaded using the `SetProfile ( string` *`name`* `)` method and once a profile has been loaded the API determines what connections are available and automatically selects the cheapest media.

To start the tool

- Select **Start Menu** > **Programs** > **WWAN Toolkit** > **Tools** > **Profile Manager**

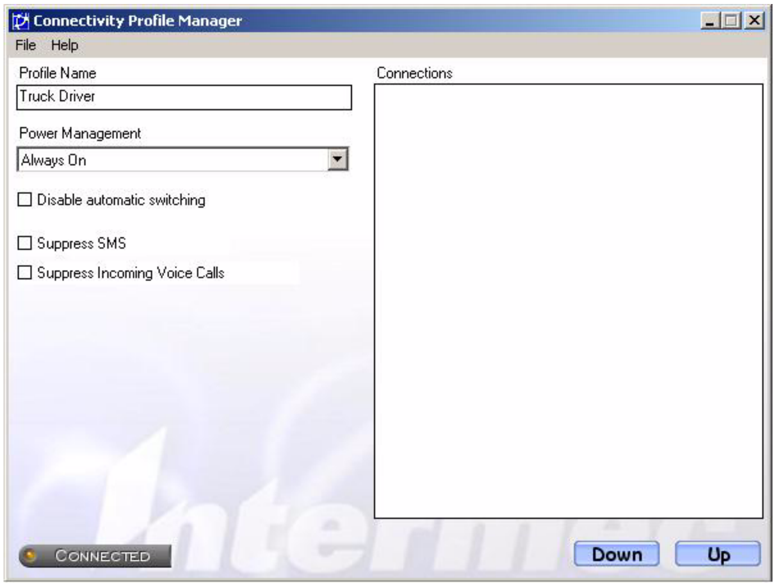
The tool can also be started from the Tools menu in Visual Studio.

Creating a New Profile

The first thing you will see when you start the application is the Profile view.

To create a new profile

1 Select **File > New.**



Connectivity Profile Manager

Profile Name: Truck Driver

Power Management: Always On

□ Disable automatic switching

□ Suppress SMS

□ Suppress Incoming Voice Calls

2 Enter a name for your profile. This would typically be a name that describes when and where the profile should be used, for example “Depot Profile”.

3 Select **Power Management**

Power Mgmt Modes	Description
Always On	Keeps all radios on at all times. This provides the fastest switching but consumes the most battery power.
Rough Switching	Turns on the radios after a connection has been lost. This saves power compared to the “Always On” mode as all radios are not switched when there is an active connection, but it also gives the longest delay when trying to connect to a different media after the current connection has been lost.
Smooth Switching	This power mode turns on the radios when the signal of the currently loaded connection is getting weak. This demands more power than rough switching, but decreases the delay times.

4 If you do not want a ring signal or event whenever an incoming voice call is received, check the **Suppress Incoming Voice Call** button. If you do not want to receive an SMS event whenever an incoming SMS is received, check the **Suppress SMS** button.

5 Leave **Disable Automatic Switching** unchecked for now. This feature is further described in the Connectivity usage guidelines section.

6 Select **File** > **Save** to save your profile.

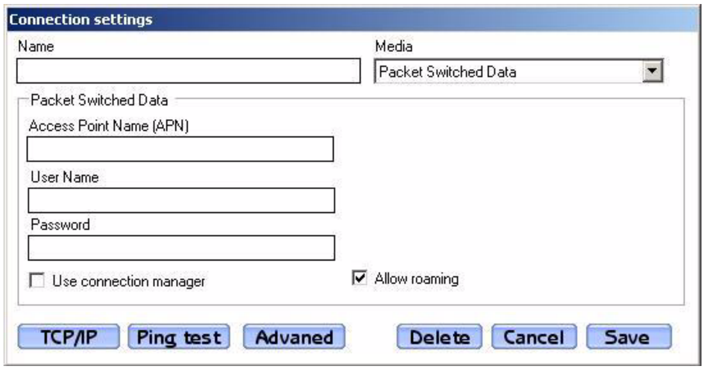
The next step is to create connections that can be used in the profile.

Creating a New Connection

Connections consist of data that describe what media should be used and the settings required to establish a connection.

To create a new connection

1 Select **File** > **New Connection.** A new screen appears:



Media: Packet Switched Data

☐ Use connection manager

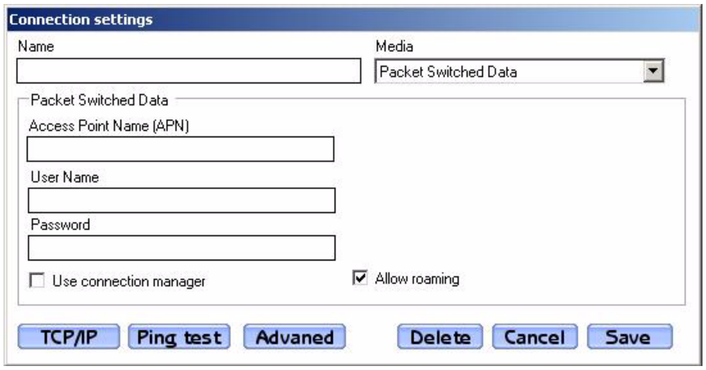
☑ Allow roaming

2 Enter a unique name for your connection that describes the connection type and the carrier.

3 Select the media you want to use. An overview of the different media types is provided in the table below;

Media	Description
Packet Switched (PSD)	GPRS or 1xRTT dialup connection
Circuit Switched (CSD)	GSM or CDMA dialup connection, expensive (paid per minute as opposed to per packet)
Wireless LAN	Using the built in WLAN card (802.11)
LAN	Ethernet (via dock). To be able to use Ethernet you have to add a LAN connection to your profile, otherwise the switcher will disregard any available LAN connection. USB cable cannot be inserted in the dock when using LAN.

"Packet Switched" (GPRS/1xRTT) Connection Settings


Connection settings

Name

Media: Packet Switched Data

Packet Switched Data

Access Point Name (APN)

User Name

Password

☐ Use connection manager

☑ Allow roaming

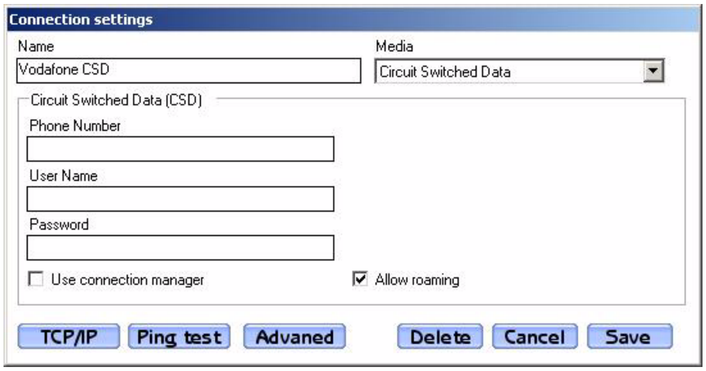
Mandatory parameters for a packet switched connection vary between different networks. Typically, you will need to retrieve the APN and your user name and password from the network operator. If your network does not utilize DHCP, press the **TCP/IP** button and fill out the form according to the network preferences.

The Connectivity API can either use RAS or MS Connection Manager to connect to the Internet. By checking the **Use connection manager** checkbox your connection will be registered with MS Connection Manager. MS Connection manager allows applications such as Pocket Internet Explorer to use your connection. The drawback is that it also prompts notification dialog boxes about the current connection manager status and permits the user to terminate a connection manually.

By checking the **Allow Roaming** checkbox your connection will be able to connect even if the WWAN radio is currently roaming through a different network than your home network. Remember that roaming can be costly.

Pressing the **Advanced** button reveals some additional parameters. A complete list of parameters for the packet switched connection mode is provided in Appendix B.

"Circuit Switched" (GSM/CDMA) Connection Settings



Connection settings

Name: Vodafone CSD

Media: Circuit Switched Data

Circuit Switched Data (CSD)

Phone Number:

User Name:

Password:

☐ Use connection manager

☑ Allow roaming

The only required parameter for a circuit switched connection is a valid phone number to connect through. If your network does not utilize DHCP, press the **TCP/IP** button and fill out the form according to the networks preferences.

The Connectivity API can either use RAS or MS Connection Manager to connect to the Internet. By checking the **Use connection manager** checkbox your connection will be registered with MS Connection Manager. MS Connection manager allows applications such as Pocket Internet Explorer to use your connection. The drawback is that it also prompts notification dialog boxes about the current connection manager status and permits the user to terminate a connection manually.

By checking the **Allow Roaming** checkbox your connection will be able to connect if the WWAN radio is currently roaming through a different network than your home network. Remember that roaming can be costly.

Pressing the **Advanced** button reveals some additional parameters. A complete list of parameters for the circuit switched connection mode is provided in Appendix B.

Wireless LAN (80211) Connection Settings

There are two ways to configure an 802.11/WLAN connection. You can either specify all parameters in the Connectivity Profile Manager which gives you full control of the parameters for all your media with a single tool, or you can use the settings already set up on the device under "Wireless Settings profiles". This second choice works well in combination with Intermec Smart Systems as you can let the Smart Systems console be responsible for setting up the parameters.

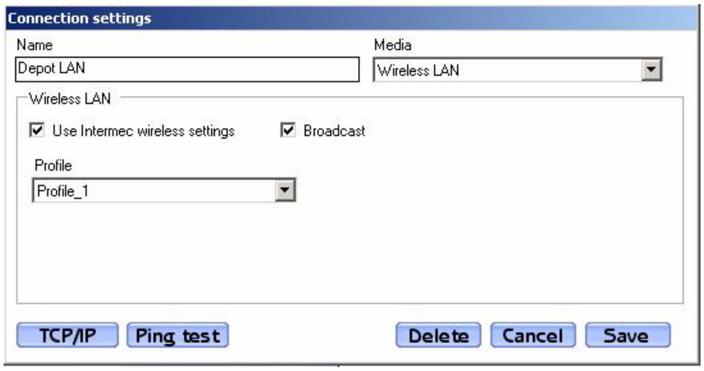
If your network does not utilize DHCP, press the **TCP/IP** button and fill out the form according to the networks preferences.

The **Broadcast** checkbox indicates that the SSID is broadcasted. This tells the Connectivity API that it should only try to connect to a network if it finds an SSID that matches the same SSID that you have specified.

If the checkbox is unchecked, SSID will not be broadcasted and the Connectivity API will only attempt to connect when it finds an unnamed SSID with sufficient signal strength. Having SSIDs that are not broadcasted consumes more power and is a lot slower than having broadcasted ones. We strongly recommend that you use only broadcasted SSIDs.

SSIDs that are not broadcasted offer a low level of security, since they can easily be “sniffed”.

Selecting the **Use Intermec wireless settings** checkbox tells the Connectivity API to utilize a pre-configured wireless profile found on the device (note that this is not a connectivity profile, but a wireless settings profile found on the device).



Connection settings

Name: Depot LAN

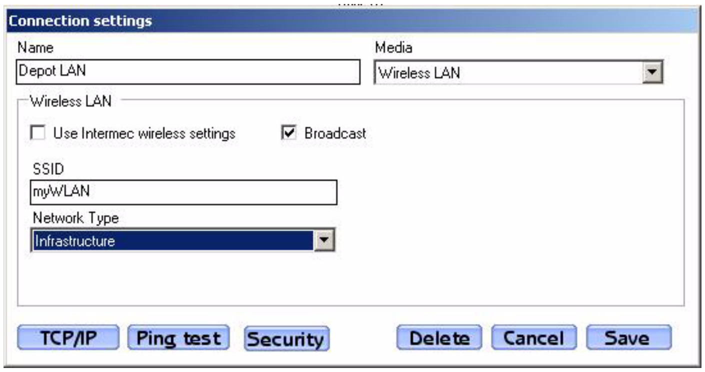
Media: Wireless LAN

Wireless LAN

☑ Use Intermec wireless settings ☑ Broadcast

Profile: Profile_1

Deselecting **Use Intermec wireless settings** allows you to specify all your WLAN parameters through the Connectivity Profile Manager. Enter your SSID and network type (Ad-Hoc also requires you to specify the channel used).



Connection settings

Name: Depot LAN

Media: Wireless LAN

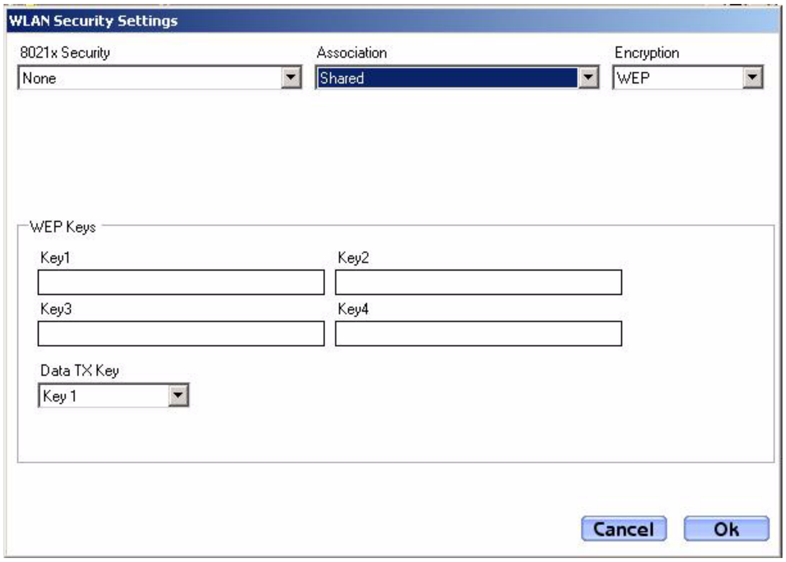
Wireless LAN

☐ Use Intermec wireless settings ☑ Broadcast

SSID: myWLAN

Network Type: Infrastructure

You can also set up your security by pressing the **Security** button. These settings are not described in-depth in this guide- please refer to the documentation of your mobile device.



WLAN Security Settings

8021x Security: None

Association: Shared

Encryption: WEP

WEP Keys

Key1:

Key2:

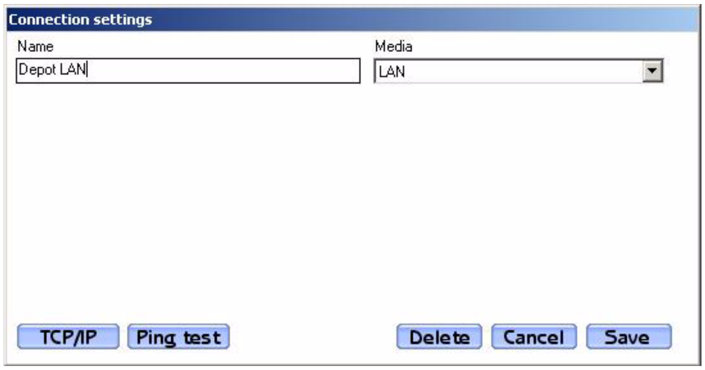
Key3:

Key4:

Data TX Key: Key 1

Note: The Connectivity API currently only supports WEP, WPA, LEAP and PEAP. TLS/TTLS is not supported through the Profile Manager. You can however still utilize TLS/TTLS by applying "Intermec wireless settings". Microsoft Zero Config is not supported.

LAN Connection Settings



Connection settings

Name: Depot LAN

Media: LAN

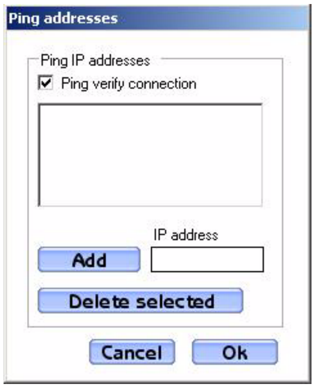
LAN only has settings for TCP/IP. If you do not create a LAN connection the switcher will assume that you do not wish to utilize a LAN even if it becomes available.

Typically, the only thing you need to do is to create a connection, name it, select LAN as media type and save the connection (DHCP is set as default).

If your network does not use DHCP, press the **TCP/IP** button and fill out the form according to the networks preferences.

Ping Test

Clicking the **Ping Test** button opens up a dialog box:



☑ Ping verify connection

Ping Testing against IP addresses is a way for the Connectivity API to verify that a connection is maintained. The Connectivity API will not issue the connected event if not all addresses are reachable.

Check the box in the upper left corner, enter an IP address and press the **Add** button. Press **Ok** when you are done, and remember to save your connection.

Delete an IP address by selecting it from the list and press **Delete Selected**.

Ping can also be achieved manually by issuing the command `Ping(....)` in your code.

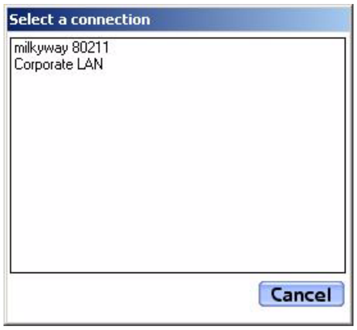
Remember that ping testing can be costly when connected through GPRS/ 1xRTT (Packet Switched connections) depending on the terms stated in the contract with your network provider.

Adding a Connection to a Profile

When you have closed the connection screen (and saved the connection), you can add the connection to your profile.

To add a connection to a profile

1 If your profile is still active, right click the Connections listbox on the right side of the screen and select Add.



milkyway 80211
Corporate LAN

2 Select the connections you want to add to the profile by clicking in the listbox.

3 The connection has now been added to your profile. Repeat the process if you wish to add more connections to the profile. When you are done, select **File** > **Save**.

(The procedure described above can also be done by clicking the right mouse button in the list box on the profile screen and selecting “New And Add”. The connection screen will appear and when you press “Save” the connection will be automatically added to your active profile.)

You can automatically deploy to the device by selecting **Save And Deploy** from the menu. The device must be connected to your PC via ActiveSync (see indicator at the bottom left corner of Profile Manager).

You have now completed the process of creating a profile, and can proceed to implement the necessary API calls in your project. Do not forget that the complete documentation for the Connectivity API can be conveniently accessed via **Start Menu** > **WWAN Toolkit** > Documentation.

Deleting a connection from a profile

Select the item to be deleted (single-click the left mouse button). Right click on the list box and select “Delete”.

Deleting / editing a connection

Note: You cannot delete a connection that is used in a profile. If you want to delete the connection completely you have to remove the connection from all profiles first.

To delete or edit a connection

1 Select **File** > **Edit Connection**

2 Select the connection that you want to delete/edit.

3 Press the Delete button located at the left to delete or edit the connection. Then press Save.

Loading/Modifying a profile

1 Select **File** > **Load**

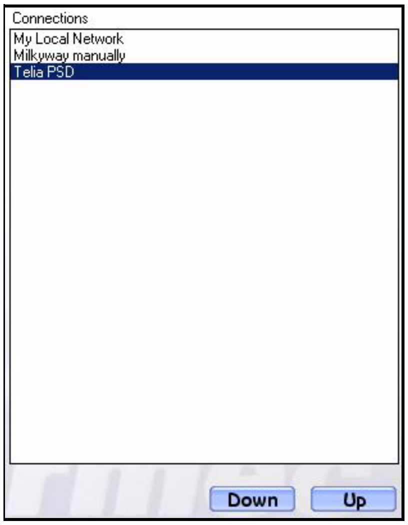
2 Select the profile from the list and make your changes as needed.

3 Select **Save And Deploy** from the menu when you are done.

Ordering Connections Within the Profile

Connections are automatically ordered within the profile based on their relative cost. For example, a LAN connection will receive a higher rank than a WLAN connection, and a PSD connection will be placed above a CSD ditto.

This hierarchy can be disabled by manually re-ordering the connections: Any connection can be selected and then moved up or down in the list with the **Up** and **Down** buttons. The Connectivity API will always try to use the connection on top of the list first, provided this is available.



Connections

My Local Network
Milkyway manually
Telia PSD

Copying Profiles From a Device

You can copy all profiles and connections directly from a device.

To copy all profiles from a device

1. First, make sure the device is properly docked (see indicator) in the cradle.
2. Select **File** > **Copy From Device**.

Your current Profiles are being saved as a backup (you cannot merge your current profiles with the ones copied from the device).

Note: In case you wish to restore the profiles you had on your PC before importing, select **Restore from Backup**. The profiles copied from the device will now be replaced by the backed-up profiles.

WWAN Wizard – Visual Studio 2003 Plugin for C# .NET

The WWAN Toolkit comes with a plugin that integrates with Visual Studio 2003 and greatly simplifies the development process of WWAN applications. It currently supports automatic code generation (C#) for the WWAN Toolkit Core API.

You can either add WWAN enabled forms to your existing solution or create a new solution with WWAN enabled forms with a few simple clicks.

If you choose to install the plugin (see WWAN Toolkit installation instructions) you should restart your development PC to load the plug-in (alternatively, issue the devenv/setup command).

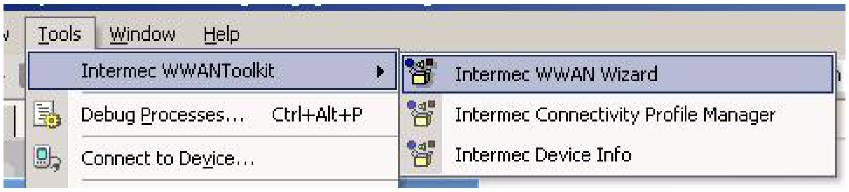
The plug-in can easily add WWAN enabled source code into your existing projects or create a brand new "smart device" solution containing WWAN enabled source code.

You can open up any of the source code files/forms and modify them as you wish. Take the opportunity to add your own "look and feel" to the forms and make them appear less generic. Some of the forms might require additional code to work in a real life situation, but as the wizard cannot dictate how you design your projects, the choice is up to you.

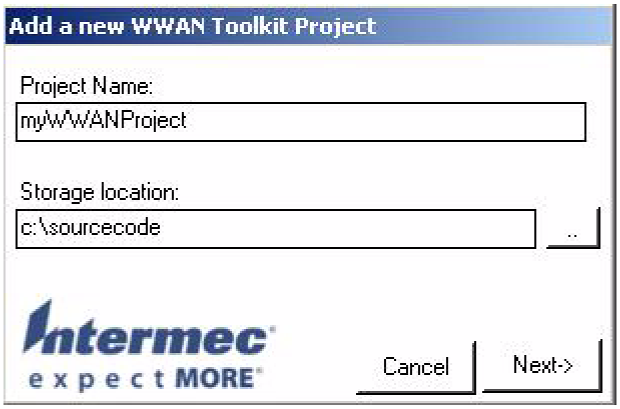
Using the Wizard

Note: If you wish to add source code to an existing project you should open up the project in the development environment before proceeding.

1 In Visual Studio, the plugin is started under **Tools > Intermec WWAN Toolkit > Intermec WWAN Wizard**.



2 If you are creating a new project, give it a name and tell the wizard where the project should be saved. Click **Next**.

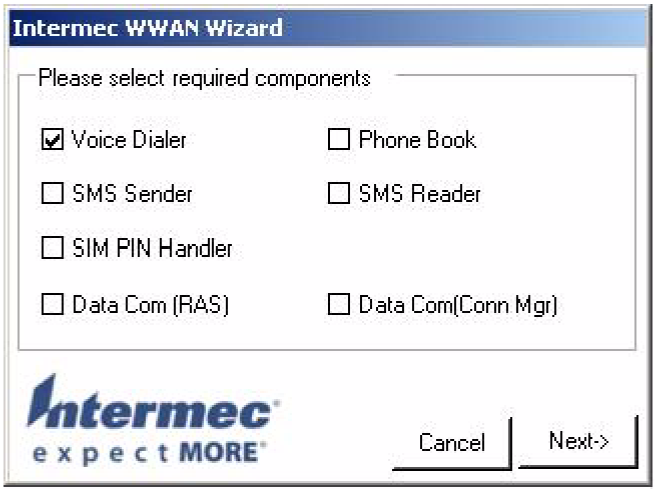


Add a new WWAN Toolkit Project

Project Name: myWWANProject

Storage location: c:\sourcecode

3 In the new dialog box, select what type of components (forms) you wish to include in your project (descriptions can be found in the table below, thorough descriptions follow in the next section). When you are finished, click **Next**.



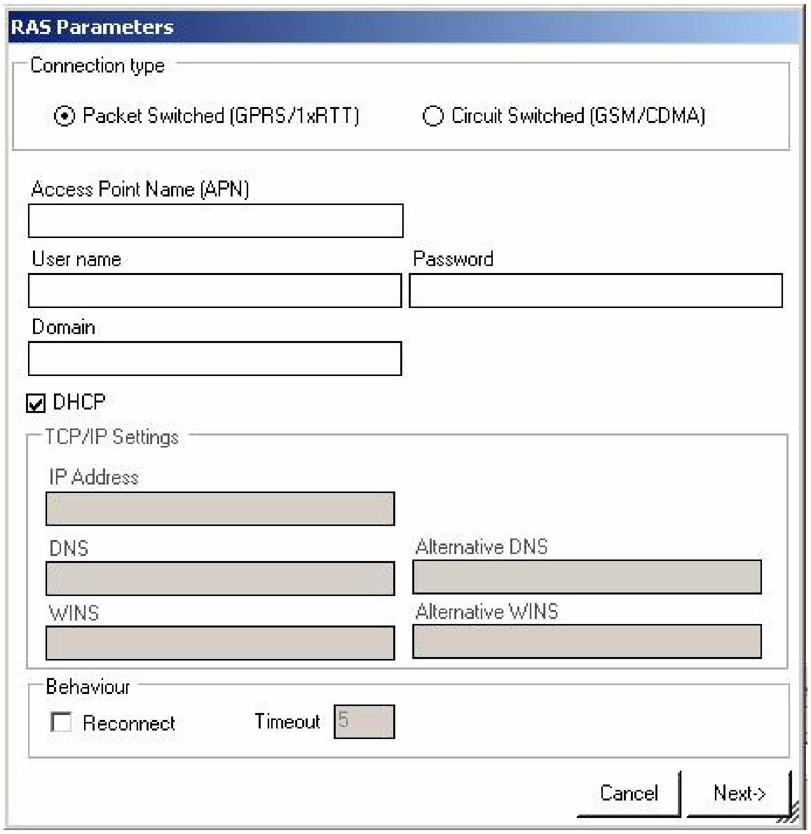
Intermec WWAN Wizard

Please select required components

☑ Voice Dialer
□ Phone Book
□ SMS Sender
□ SMS Reader
□ SIM PIN Handler
□ Data Com (RAS)
□ Data Com(Conn Mgr)

Component	Description
Voice Dialer	Simple voice dialer that can dial voice calls, handle incoming calls and send DTMF tones.
Phone Book	Read, write and delete items from the built in phone book
SMS Sender	Send SMS messages, receive SMS messages.
SMS Reader	Read and delete stored SMS messages.
SIM PIN Handler	Manage SIM PIN security
Data Com (RAS)	Data communication using RAS.
Data Com (Conn Mgr)	Data communication using MS Connection Manager

4 If you have selected any of the “DataCom” components in the previous step, you will be asked to enter additional information. Most of this is network-specific and therefore has to be obtained via your network operator.



RAS Parameters

Connection type

● Packet Switched (GPRS/1xRTT) ○ Circuit Switched (GSM/CDMA)

Access Point Name (APN)

User name

Password

Domain

☑ DHCP

TCP/IP Settings

IP Address

DNS

Alternative DNS

WINS

Alternative WINS

Behaviour

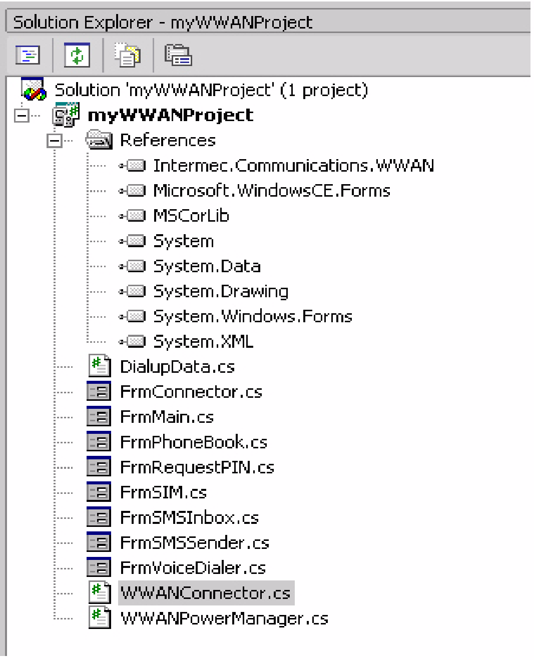
□ Reconnect

Timeout 5

The “**Behaviour”** parameter tells the application whether or not to reconnect when a connection has been dropped. The Timeout parameter specifies the duration (in minutes) of the reconnection attempt.

Click **Next** when all data has been entered.

5 The source code is being added to your project! Note that the References folder contains a reference to the WWAN Toolkit .NET wrapper.



```
Solution Explorer - myWWANProject
Solution 'myWWANProject' (1 project)
  myWWANProject
    References
      Intermec.Communications.WWAN
      Microsoft.WindowsCE.Forms
      MSCorLib
      System
      System.Data
      System.Drawing
      System.Windows.Forms
      System.XML
    DialupData.cs
    FrmConnector.cs
    FrmMain.cs
    FrmPhoneBook.cs
    FrmRequestPIN.cs
    FrmSIM.cs
    FrmSMSInbox.cs
    FrmSMSSender.cs
    FrmVoiceDialer.cs
    WWANConnector.cs
    WWANPowerManager.cs
```

Common Components

The following components are always added to your project.

Power Manager – WWANPowerManager.cs

WWANPowerManager contains methods to control the power of the radio and to start and shut down the toolkit. This class is instantiated and controlled from the FrmMain.cs form.

PIN Query – FrmRequestPin.cs

FrmRequestPin is instantiated and visualized by the power manager whenever a PIN code is required by the SIM card.

Start Form – FrmMain.cs

Simple start form that powers the radio (through WWANPowerManager). Typically, you should move any event handlers that are “application global” to this class (incoming voice call, incoming SMS) and handle them accordingly. The form also retrieves some parameters from the toolkit to notify you about the current status of the radio.

Typically you should add buttons/source to instantiate the other forms.

You could remove this form completely and replace it with your own start form.

Selectable Components

The following classes and forms are selected by the user and added to the project.

Voice Dialler – FrmVoiceDialer.cs

This is a rudimentary voice dialler that handles both incoming and outgoing voice calls. DTMF tones are also supported. In a complete solution it is wise to move the event handling to the main form, since keeping it in the voice dialler form requires that the form is activated in order to receive incoming voice calls. This is however depending on implementation and design choices made by the user.

Phone Book – FrmPhoneBook.cs

FrmPhoneBook contains a form that allows you to read/write/delete entries from the toolkit's built-in phone book. On the 76x and CN3 GSM devices this is synonymous to the SIM phone book. The SIM phone book is very limited and we strongly recommend you to create your own phone book using a file or database.

SIM PIN Handler – FrmSim.cs

FrmSim lets you lock and unlock your SIM card, change the PIN code and unlock it using PIN & PUK codes.

SMS Inbox – FrmSMSInbox.cs

FrmSMSInbox has a simple GUI that allows you to read and delete SMS messages (stored on the SIM card). Rather than using this built-in function, which uses the SIM card for storage, we recommend you to implement your own storage solution. This is to prevent the SIM card memory from being overloaded.

SMS Sender – FrmSMSSender.cs

FrmSMSSender allows you to send SMS messages. It also notifies the user whenever a new SMS is received.

In a complete, real life solution it would be wise to move the event handling (incoming SMS) to the main form since keeping it in the SMS inbox form requires the form to be activated to receive incoming SMSs. This is however depending on implementation and design choices made by the user. It is also wise to delete the message instantly upon arrival and store it in a database (or file) to make sure that the SIM card is not swamped with messages.

Data Com (RAS/Conn Mgr) – DialupData.cs

Holds the parameters for the connection, this is typically where you would change your parameters if they where incorrect or if the operator had changed the network requirements. The class is instantiated by WWANConnector.cs.

If you want to change a RAS connection to MS Connection Manager or vice versa you can easily change the parameter according to your likings:

```
public bool UseConnectionManager = FALSE
```

Data Com (RAS/Conn Mgr) – WWANConnector.cs

The WWANConnector class handles the logic behind the connection process. It is responsible for setting up, connecting and disconnecting the connections.

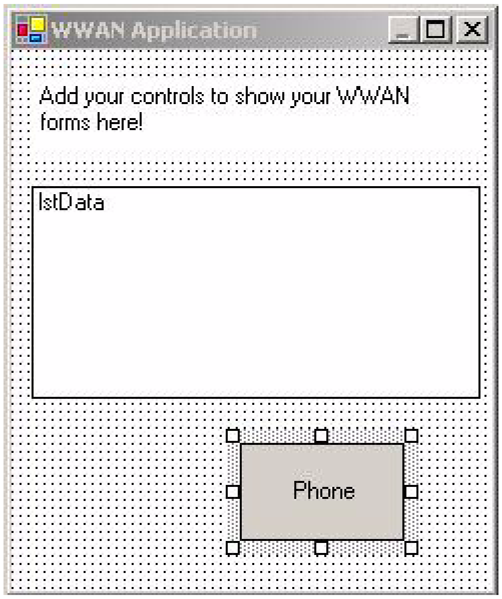
Data Com (RAS/Conn Mgr) – FrmConnector.cs

The simple FrmConnector form utilizes the WWANConnector class to connect and disconnect with RAS or MS Connection Manager (depending on the settings)

Putting Together a Working Application – Example

Here is how you can create a simple, working voice dialler :

1 After you have selected your components, you need to create an instance of the forms in your start form (FrmMain). Add a button to FrmMain:



Add your controls to show your WWAN forms here!

2 In the ‘click event handler’, add the following code (sample shows the voice dialler);

```
private void button1_Click(object sender,
    System.EventArgs e)
{
    FrmVoiceDialer dialer = new FrmVoiceDialer();
    dialer.ShowDialog();
}
```

5 Samples

This chapter contains descriptions of the sample code.

About the Samples

The sample source code is located in the [INSTALLDIR]\Samples folder. Each API has its dedicated subfolder (Core and Connectivity) located in the samples folder. Each language in turn has their dedicated subfolder (C++, Java, C# and VB) located under each API folder.

To test the applications make sure you follow the installations and preparation steps described in previous chapters.

Core API Samples

The following samples can be found for each language in the WWAN Toolkit Core folder:

Core API Samples

Sample	Description
DataCom	RAS and Connection Manager sample that supports packet switched connections on both CDMA and GSM devices.
VoiceDialer	Simple voice dialler for incoming and outgoing calls. Also supports audio level adjustment.
SIMPin	Manage the security settings of the SIM card (GSM only).
PhoneBookManager	Manage the phone book
SMS	Send and manage SMSs.
Info	Generic information application (network status, radio information)

DataCom

This example demonstrates how to create a dial-up connection with RAS (Remote Access Server) or the MS Connection manager.

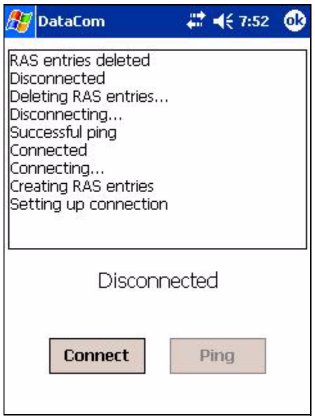
When the application is started, that is when the radio has been powered on and the status of the SIM card has been checked, the user can click the connect button. Based on the settings, an attempt is made to create a dialup connection. If a connection is established, another click on the button will make the application disconnect.

The application contains “reconnection logic”. This is applied when the connection has been broken unintentionally, for example due to weak signal strength.

The main difference between the languages is the event models. In Java, observable interfaces are used to notify the GUI about events. In C++, a reference to the GUI class is passed on to the WWANConnector and in C# and VB.Net events are raised to notify the GUI. In C++ the events from the WWANToolkit are first handled in the GUI and then passed on to the connector class while in the other languages the connector itself listens to the toolkit events.

Filename	Language	Description
DataCom.java	Java	Main class where GUI and application is initialized.
FrmDataCom.cs	C#	
FrmDataCom.vb	VB .NET	
DataComDlg.cpp	C++	

Filename	Language	Description
WWANConnector.java	Java	Class that holds connection logic and reports events back to the GUI. All connection parameters are stored in this class.
WWANConnector.cs	C#	
WWANConnector.vb	VB .NET	
WWANConnector.cpp	C++	



RAS entries deleted
Disconnected
Deleting RAS entries...
Disconnecting...
Successful ping
Connected
Connecting...
Creating RAS entries
Setting up connection

Disconnected

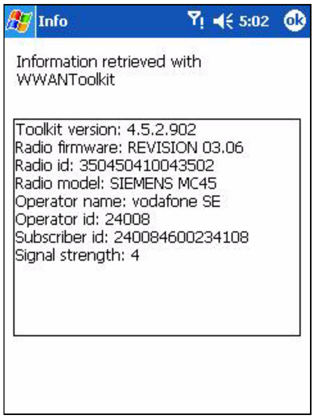
The Datacom sample

Info

This example shows how to start up the toolkit and retrieve information from it.

When the application is started, it tries to power on the radio and then moves on to checking the status of the SIM card and retrieves information from the toolkit.

Filename	Language	Description
Info.java	Java	Main class where all relevant application events take place.
FrmInfo.cs	C#	
FrmInfo.vb	VB .NET	
InfoDlg.cpp	C++	



Information retrieved with WWANToolkit

```
Toolkit version: 4.5.2.902
Radio firmware: REVISION 03.06
Radio id: 350450410043502
Radio model: SIEMENS MC45
Operator name: vodafone SE
Operator id: 24008
Subscriber id: 240084600234108
Signal strength: 4
```

The Info sample

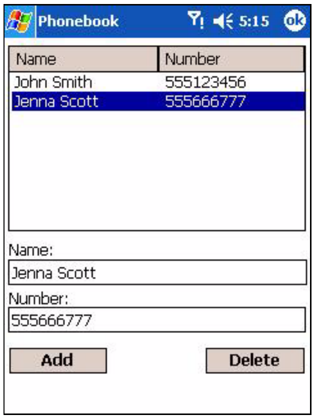
Phone Book

This example demonstrates how to use the phone book.

At startup the application powers on the radio, checks the status of the SIM card, and populates the phone book list. The phone book entries are read either from the SIM card (GSM devices) or 'from file' (CDMA devices).

Since AWT (Abstract Window Toolkit) is used in Java, it is not possible to add hidden data to a list item here. Hence the index of the phone book entry is added to each list item in the rightmost position of a pipe ('|') separated string.

Filename	Language	Description
PhoneBook.java	Java	Main class where all relevant application events take place.
FrmPhoneBook.cs	C#	
FrmPhoneBook.vb	VB .NET	
PhoneBookDlg.cpp	C++	



Name	Number
John Smith	555123456
Jenna Scott	555666777

Name:
Jenna Scott

Number:
555666777

The Phonebook sample

SimPin

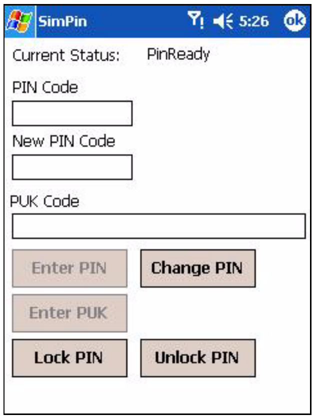
This example demonstrates how to use the SIM (Subscriber Identity Module) security functionality.

During application startup, the radio is powered on and the status of the SIM card is read. Based on the current status, buttons for each available option are enabled.

Example: If the SIM card is locked and requires the PUK (Personal Unblocking Key) code to be unlocked, only the button entitled “Enter PUK” will be enabled.

The available functions are: entering the PIN (Personal Identification Number) and PUK number if the card is locked, changing the PIN number and enable/disable the PIN lock on the SIM card.

Filename	Language	Description
SimPin.java	Java	Main class where all relevant application events take place.
FrmSimPin.cs	C#	
FrmSimPin.vb	VB .NET	
SimPinDlg.cpp	C++	



Current Status: PinReady

The SimPin sample

SMS

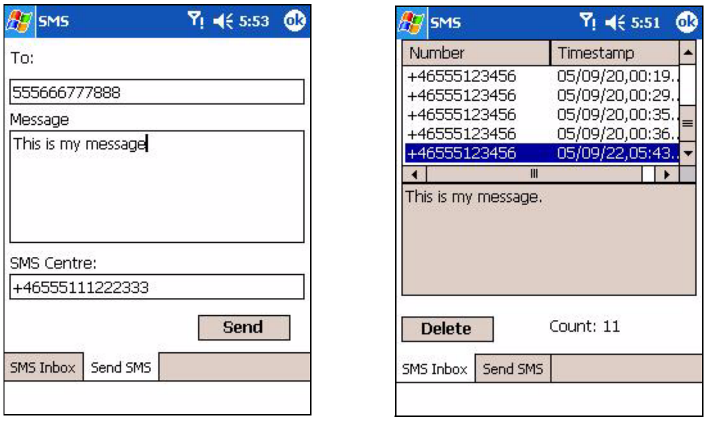
This example demonstrates how to send SMS (Short Message Service) messages and display received messages.

At startup, the application powers on the radio, checks the status of the SIM card and then tries to populate the message inbox with the received messages currently stored on the SIM card. Under the “SMS Inbox” tab the user can select a message from a list and read its contents. It is also possible to erase a selected message. If the user switches to the “Send SMS” tab an SMS message can be created and sent.

In this case, the C++ implementation represents the biggest deviation. To create tab pages in C++ it is recommended to “subclass” the CTabCtrl once for each respective tab. Each tab class then takes care of its own events (i.e. TabInbox takes care of inbox events and TabSendSms takes care of incoming SMS). The other languages handle all application events in the main class.

When AWT (Abstract Window Toolkit) is used in Java development, tab pages are not supported. This is solved by using one panel for the inbox and another for sending SMSs. The panels are either hidden or displayed based on which button is pressed at the bottom of the screen.

Filename	**Language**	**Description**
Sms.java	Java	Main class where all relevant application events take place.
FrmSms.cs	C#	
FrmSms.vb	VB .NET	
SmsDlg.cpp	C++	Startup and initialization class. Listens to all toolkit events and delegates them to one of the tab classes.
TabSendSms.cpp	C++	Subclass of CTabCtrl that handles SMS sending.
TabInbox.cpp	C++	Subclass of CTabCtrl that handles the SMS inbox.



To:
555666777888

Message
This is my message

SMS Centre:
+46555111222333

Number	Timestamp
+46555123456	05/09/20,00:19..
+46555123456	05/09/20,00:29..
+46555123456	05/09/20,00:35..
+46555123456	05/09/20,00:36..
+46555123456	05/09/22,05:43..

This is my message.

Count: 11

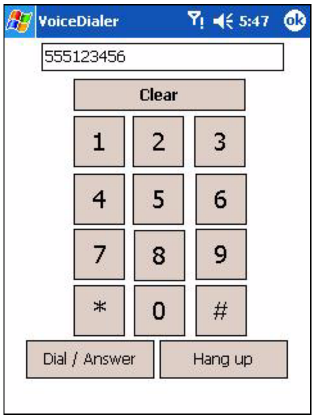
The SMS sample: *'Send' window and SMS inbox.*

VoiceDialer

This example demonstrates how to create a voice dialling application.

When the application is started, that is when the radio is powered on and the status of the SIM card is checked, the user can make and receive phone calls. DTMF (Dual Tones Multi Frequncy) tones will be sent during ongoing calls if a number, star or hash button is pressed.

Filename	Language	Description
VoiceDialler.java	Java	Main class where all relevant application events take place.
FrmVoiceDialler.cs	C#	
FrmVoiceDialler.vb	VB .NET	
VoiceDialler.cpp	C++	



555123456

The Voicedialer sample

Connectivity API Sample

The following sample can be found for each language in the Connectivity folder:

Sample	Description
Connectivity	Automatic and manual switching application. Requires a profile and connections to be created with the Connectivity Profile Manager tool before execution. Please refer to the ‘Tools’ section of this manual for further information. Also shows all events received from the Connectivity API in a listbox.

This example shows how to take advantage of the Connectivity API.

Before the application can be used you need to create XML files containing your profiles and connection settings. This is done in the desktop application “Connectivity Profile Manager” (see the “Tools” section for more information).

During start-up, the GUI is displayed and the profiles and connections available in the XML-files (located in the “\Windows” directory on the handheld device), are read and populate the drop-down boxes.

The application consists of three forms: the “Connectivity”, “Ping” and “Eventlog” forms.

In the “Connectivity” form the user can choose to set a profile or lock (manually connect) on to a specific connection. When a profile is selected and the **Set** button pressed, the switching functionality of the Connectivity API is used to see which of the connections for the specified profile are available and which one is ranked the highest. (Since LAN is normally cheaper and faster than WWAN it should be given higher priority.) The switcher will use the selected connection as long as it exists and no higher ranked alternative becomes available.

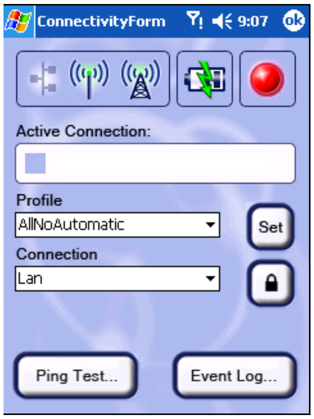
With the Lock button you can force the Connectivity Toolkit into using a specified connection, and thereby prevent switching.

At the top left corner of the screen the statuses of the different connection types (LAN, WLAN and WWAN) are displayed. The possible options are: not powered (crossed over), powered but unavailable (dim grey) and available (coloured)[1].

[1] Please note that due to 80211 architecture, an “available” state does not necessarily mean that the particular network you want to use is available, it merely indicates that the device has detected one or more 80211 networks with sufficient network signal strength.

The other two icons displayed at the top indicate battery status and connection status. The connection status indicator is grey when the switcher is idle, yellow when it is working, blue when a connection is established and red if the switcher is unable to establish a connection.

When a connection is established the type and name of the connection is displayed in the "Active Connection" field.

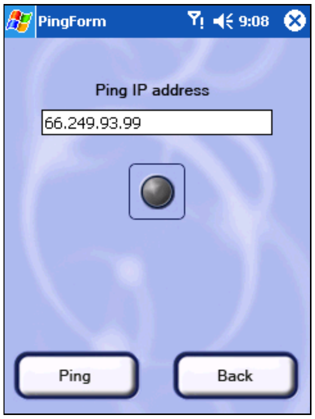


Profile: AllNoAutomatic

Connection: Lan

C# .NET Connectivity Sample: *Establishing a connection*

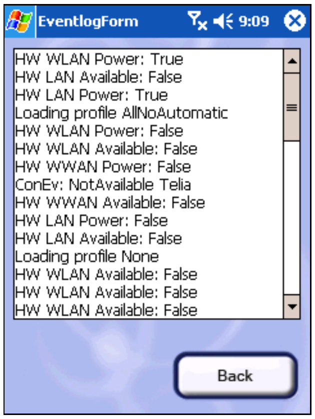
Press the **Ping Test…** button to verify the connection. This launches another page where an IP address can be entered. When the **Ping** button is pressed, an attempt is made to ping the given IP address with the current connection. The result of the ping operation is displayed as a status LED at the centre of the screen. Red indicates that the target could not be reached, while blue indicates success.



Ping IP address: 66.249.93.99

C# .NET Connectivity Sample: *Verifying a connection*

Plenty of messages are sent from the Connectivity Toolkit to the application, and not all of them are suitable to display graphically. Pressing the **Eventlog...** button launches a page where the user can view all log messages.



```
HW WLAN Power: True
HW LAN Available: False
HW LAN Power: True
Loading profile AllNoAutomatic
HW WLAN Power: False
HW WLAN Available: False
HW WWAN Power: False
ConEv: NotAvailable Telia
HW WWAN Available: False
HW LAN Power: False
HW LAN Available: False
Loading profile None
HW WLAN Available: False
HW WLAN Available: False
HW WLAN Available: False
```

C# .NET Connectivity Sample: *Displaying the eventlog*

The visual presentation layer is more elaborate for the .NET languages, C# and VB than for Java and C++. They all share similar functionality, but the latter two use no graphic components.

Other than that, the event handling feature constitues the greatest difference between the languages:

Language	Event handling procedure
Java	A listener adapter is extended or a listener is implemented.
C#	Listeners are added for each event.
VB .NET	The Connectivity Toolkit is referenced using the “with events” attribute.
C++	All window events sent from the toolkit are acknowledged and redirected to the appropriate handler method.

For simplicity the application is developed with CTabCtrl in C++ instead of using separate forms.

Filename	Language	Description
ConnectivityFrame.java	Java	Main class where all relevant application events take place.
ConnectivityForm.cs	C#	
ConnectivityForm.vb	VB .NET	
ConnectivityDlg.cpp	C++	

6 Frequently Asked Questions

This chapter addresses some of the issues that may come up when the WWAN Toolkit is used to develop software applications.

A Network Issues

1 Common

1. **Q: My application fails to enable modem event reporting. Why is this?**

   A: Typical sequence: PowerOn, Setup, EnableModemEventReporting. Make sure you have a SIM card. On the 760, the toolkit will not report an error unless event reporting has been enabled.

2. **Q: What characters are supported when sending/receiving SMSs?**

   A: On the 760, the PDU 7-bit alphabet is used. Refer to http://www.dreamfabric.com/sms/default_alphabet.html

2 CDMA

1. **Q: Why does SetGPRSAttach always return FALSE on CDMA?**

   A: You can only perform GetGPRSAttach on CDMA. This verifies that the 1xRTT network is available; there is no need to attach to it as on GPRS devices. Distinguish between GPRS and CDMA devices via an “If” statement.

2. **Q: Can I send SMS messages on a CDMA device?**

   A: On the 761 and CN3 running Phone Edition OS you can. On 760 running Premium you cannot.

3. **Q: Why do I not get a roaming event on CDMA devices?**

   A: There is a possibility that you will receive signal strength from another network when you have lost the connection to your home network. Your connection will be terminated and you must cater for these events yourself. The toolkit does not support roaming events on CDMA radios. GetGPRSAttach should tell you if 1xRTT is available for any connections to be made.

3 GSM

1. **Q: Where can I learn more about GSM/GPRS technology?**

   A: Cisco has released a “design guide” on the subject. This is available at:

   http://www.cisco.com/en/US/netsol/ns341/ns396/ns177/ns278/networking_solutions_design_guide_chapter09186a00801219ac.html

   Bio Enable’s web page is not as comprehensive but covers most topics:

   http://www.bioenabletech.com/technical_introduction_to_gsm_modem_technology.htm

4 WWAN

1. **Q: Why is there no ring signal when I get a voice call?**

   A: The “incoming call” event is received but no sound is played. Make sure the ring signal sound file is properly installed in the windows folder. The WAV file is automatically installed with the WWAN Toolkit cab file. You can also rename any WAV file to WWANTKRing.wav and copy it to the windows folder on your device.

2. **Q: Why is there no sound when I pick up an incoming voice call? When I receive an incoming voice call (or dial a number) and pick up the phone (answer), I can hear that there is a connection but the microphone and speaker on the device seem “dead”?**

A: If you have upgraded your OS/firmware the speaker and microphone volume levels are sometimes set to zero by default. Use SetSpeakerVol and SetMicVol to change the levels as preferred.

3. Q: How do I retrieve the phone number of the device?

A: Use the GetPhoneNumber () method.

5 LAN

1. Q: The Connectivity API cannot detect LAN. What's wrong?

A: Make sure there is no serial and/or USB cable connected to the dock. Ethernet is automatically disabled whenever the device is connected via USB. Make sure you have a LAN connection defined and that it is added to your profile. If there is no LAN connection defined the switcher will discard the availability of Ethernet.

B Development Issues

1 Common

1. Q: **Can I run an application which utilizes COM port 4 or COM port 5 at the same time as the WWAN Toolkit?**

   A: No. The WWAN toolkit requires access and control of both COM4 and COM5 at all times on the Intermec 760. If you choose to do so this might cause unpredictable results. On Phone Edition units however, COM5 is not used by the WWAN Toolkit.

2. Q: **Can I combine Cellcore (TAPI, ExTapi...) with the WWAN Toolkit?**

   A: On Phone Edition units you could possibly use a small subset of the APIs, but you cannot use SMS or Voice functionality from Cellcore while running the toolkit. The 760 does not support Cellcore APIs.

3. Q: **I have previously written an application in Embedded Visual Basic. Now that the the WWAN Toolkit no longer supports the COM wrapper, what do I do?**

   A: Migrate your application to Java, C++, VB.Net or C# (Compact Framework). The following Microsoft web page may be helpful:

   http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnppcgen/html/fromemb.asp

2 .NET

1. Q: **What does “MissingMethodException” (.NET) mean?**

   A: This means that the .NET wrapper cannot find the native method or dll. Make sure wwantoolkit.dll is located in the windows directory of your device. Make sure the wrapper is matching the dll you are using.

C Connection Problems

1 Common

2 RAS

1. Q: What does RAS Error 119 (the port is disconnected) mean?

A: This error typically occurs when you have an APN name that does not match your network operator’s APN (GSM only). Make sure you enter the correct APN for the SIM you are currently using.

2. Q: Where can I find information about the RAS error codes?

A: http://support.microsoft.com/kb/q163111/

3. Q: What baud rate should I use when I create a RAS Connection?

A: 115200 bps. This parameter has no significance on 7x1 or CN3 devices running Phone Edition on 7x0 machines and machines running Premium. 115200 is the maximum speed.

4. Q: I get RAS Error #619 “The Port Is Disconnected”. What does this mean?

A: Something is wrong with your network settings. Make sure you have the correct APN, username and password (and any other parameters required by your operator).

Most common source of error: incorrect APN.

3 Connection Manager

1. Q: Why do I receive a "No path to destination" message when I am trying to establish a ConnectionManager connection?

A: Verify that your connection details are correct. Also make sure that the ConnectionName in the DoDialUpConnect() method call uses "Internet Settings" when running in Pocket PC 2002 and "My ISP" when it is running in PPC 2003 and later versions.

2. Q: Can I use the Microsoft Connection Manager API/GUI to create my connections while using the toolkit for powering on the radio?

A: We do not recommend it even though it may work on some devices. The WWAN Toolkit supplies a sufficient set of commands to control the Connection Manager API through the toolkit. Alternatively, you can use RAS to remove the notification bubbles and gain even more control.

4 GPRS

1. Q: Why does the toolkit not let me know when I have lost my (packet switched) connection?

A: The PSD networks usually take somewhere between 20 minutes – 2 hours to verify that there is no connection available, so even if you do not have sufficient signal strength you will not receive an event telling you the connection is missing. Ping using the ActiveWWANAvailable method to make sure your server is available.

2. Q: I need to use a very long string in the “extra commands string” from the GPRS connection: ^SGAUTH=1+CGDCONT=1,"IP","mylongapn.inter.net". This string is longer than 39 characters, thereby exceeding the character limit of this command. How can I introduce this string?

A: Use SetGPRSContext. This command does not have the same limitations that the extra dial command has. In code (C#):

SetGprsContextFlags f =
SetGprsContextFlags.SetApn|SetGprsContextFlags.SetDataCompression|SetGprsCont

extFlags.SetHeaderCompression; myToolkitRef.Connections.SetGprsContext(flags,1,"
mylongapn.inter.net","",false,false);

Then use phone number "*99***1#" when setting up the RAS connection. The CAB file can be found in the toolkit installation folder on your PC. Remember to warm-boot the device after installing the mc45 CAB file, otherwise the device will not find the "wana...." modem type.

You need to have the MODEMS_MC45_All.cab file installed when you use SetGPRSContext. You should also use modem name "WANA on COM4: (No modem reset)" when setting up the RAS connection.

For more information on this issue, please refer to the method CreateRAS(Connection con) found in the WWAN Communicator source code (included in the WWAN Toolkit C# samples).

D Operating System

1 Windows Mobile 2003

1. Q: My modem (WWAN Application) has stopped responding after I reflashed my 761 to WM2003 Premium, What could be the reason for this?

A: If you have a 761 that was previously running Phone Edition, and you reflash it to run Premium Edition instead, you will need to reset the user-defined profile stored in the MC45 to factory defaults. This is because Phone Edition changes the result code format (ATVx) in the user-defined profile to be numeric, whereas most modem software written for Premium Edition expect text results such as OK and ERROR. To reset the user-defined profile in the MC45 to factory defaults, you need to send two AT commands. AT&F will set all current parameters to manufacturer defaults. AT&W will then save those parameters to the user-defined profile in non-volatile memory..

2 Windows Mobile 2003 Phone Edition

1. Q: I am using the WWAN Toolkit on a 700 CXL device and I want my SMSs to end up in my inbox without being “grabbed” by the toolkit. How do I accomplish this? Why does SetGPRSAttach always return FALSE on CDMA?

A: The SMS is handled by the Toolkit by default. If you want to turn this feature off and let your inbox handle all SMSs (not recommended) you can add a key to the registry:

HKEY_LOCAL_MACHINE\SOFTWARE\Intermec\WWANToolkit\SMSPassthrough

Value and type is of no importance, the key should just be there for the toolkit to pass on the SMSs to the inbox.

2. Q: How do I make SMSs cold boot resilient on a 700 CXL?

A: By default, all SMSs are stored in the file WINDOWS\\sms_storage.dat If you want to change this location (for cold boot resilience etc.), modify the registry key:

HKEY_LOCAL_MACHINE \SOFTWARE\Intermec\WWANToolkit\SmsFileName

3. Q: The toolbar in Phone Edition signals "No service" when running a WWAN Toolkit application. What is wrong?

A: As the toolkit takes control of all functionality that was previously handled by the operating system, you can no longer rely on any tools included in the OS. The WWAN Toolkit will still be able to query network status and perform its tasks as usual.

4. Q: Our application previously used "Siemens MC45 on COM4: (CHAP)" / "Siemens MC45 on COM4: (PAP)" when creating a RAS entry. Now that modems_mc45_all.cab is no longer supported on the 761, how do I do I use CHAP or PAP?

A: You use it the same way you always have, since the toolkit can distinguish between the different modem types. You can also use it in the “extra dial string” (SGAUTH) as in previous versions of the toolkit.

The appropriate modem names for PAP/CHAP are:

"Siemens MC45 on COM4: (PAP)"

"Siemens MC45 on COM4: (CHAP)"

"Siemens MC45 on COM4: (PAP/CHAP)"

5. Q: I can no longer start the phone application (Phone Edition) once I have started my WWAN Toolkit application. What is wrong?

A: If you warm-booted the device or your application crashed just before you tried to use the phone application you need to restart the WWAN application and make sure it performs a power on/setup/shutdown/power off of the toolkit to restore the phone application.

6. Q: Can I use the Phone Edition applications (Phone Application, Inbox etc.) at the same time as I am using the WWAN Toolkit?

A: No, you cannot. The toolkit needs to be in full control.

7. Q: Why can I not uninstall the WWAN Toolkit on the device (Phone Edition)?

A: Warm-boot the device to release the OS handle to the dlls. Uninstall the toolkit.

8. Q: On my Phone Edition unit I want to be able to control voice calls using the built-in dialer. How do I do this?

A: By adding the registry key SOFTWARE\Intermec\WWANToolkit\PhoneApplication. Note that events will still be sent to the toolkit, while all incoming calls are handled by the OS phone application.

9. Q: Can I enable the signal strength event on Premium OS?

A: Yes, by adding a DWORD registry key under SOFTWARE\Intermec\WWANToolkit\EnableSignalStrength where the value indicates poll time (in seconds).

7 WWAN Glossary

This chapter provides a list with explanations of important terms and concepts related to wireless technology.

1xRTT

Short for *single carrier (1x) Radio Transmission Technology*, a 3G wireless technology based on the CDMA platform. 1xRTT has the capability of providing speeds of up to 144 kbps.1xRTT is also referred to as CDMA2000.

700C Models supported

The following 700 models are supported by the toolkit:

- 700C (a.k.a.760)
- 700CXL (a.k.a. 761)

700C comes with two different radio options:

- Siemens MC45 (GSM)
- Sierra Wireless SB555 (CDMA)

700CXL comes with two different radio options:

- Siemens MC45 (GSM)
- Sierra Wireless EM3420 (CDMA)

APN

Short for *Access Point Name*.

An access point is:

- An IP network to which a mobile can be connected
- A set of settings which are used for that connection
- A particular option in a set of settings in a mobile phone

When a GPRS mobile phone sets up a context, the access point is selected by identifying the APN. This access point is then used in a DNS query to a private DNS network. This process (called APN resolution) finally gives the IP address of the GGSN which should serve the access point. At this point a PDP context can be activated.

Circuit Switching

A circuit-switched network is a type of network in which a physical path is obtained for and dedicated to a single connection between two end-points in the network for the duration of the connection. Ordinary voice phone service is circuit-switched. The telephone company reserves a specific physical path to the number you are calling for the duration of your call. During that time, no one else can use the physical lines involved. Used in GSM and CDMA networks.

CDMA

Short for *Code-Division Multiple Access,* a digital cellular technology that uses spread-spectrum techniques. Unlike competing systems, such as GSM, that use TDMA, CDMA does not assign a specific frequency to every user. Instead, every channel uses the full available spectrum.

CHAP

CHAP (Challenge-Handshake Authentication Protocol)is a more secure procedure for connecting to a system than the Password Authentication Procedure (PAP). CHAP works like this: Once the link has been set up, the server sends a challenge message to the connection requestor. The requestor responds with a value obtained by using a one-way hash function. The server checks the response by comparing it its own calculation of the expected hash value and if the values match, the authentication is acknowledged. If this is not the case, the connection is usually terminated.

At any time, the server can request the connected party to send a new challenge message. Since CHAP identifiers are changed frequently and authentication can be requested by the server at any time, CHAP provides more security than PAP. RFC1334 defines both CHAP and PAP.

CSD

Short for *Circuit Switched Data*. See also Circuit Switching.

DTMF

Dual Tone Multi Frequency is the signal to the phone company that you generate when you press the touch keys of an ordinary telephone. With DTMF, each key you press on your phone generates two tones of specific frequencies.

ESN

Electronic Serial Number. Used on CDMA networks to identify the unit.

GGSN

Short for *Gateway GPRS Support Node*. The GGSN in GPRS is equivalent to the Home Agent in Mobile IP. It is a router which detunnels user data from GTP and transmits normal user data IP packets.

GPRS

Short for *General Packet Radio Service*, a standard for wireless communications which run at speeds up to 115 kbps, compared with the 9.6 kbps of current GSM systems.

GPRS, which supports a wide range of bandwidths, is an efficient use of limited bandwidth and is particularly suited for sending and receiving small bursts of data, such as e-mail and Web browsing, as well as large volumes of data.

GSM

Short for *Global System for Mobile Communications,* one of the leading digital cellular systems. GSM uses narrowband TDMA, which allows eight simultaneous calls on the same radio frequency.

IMEI

International Mobile Equipment Identity. A unique number given to every single mobile phone. The number consists of four groups that looks like this: nnnnnn--nn-nnnnnn-n The first set of numbers is the type approval code (TAC). The first two digits represent the country code. The rest make up the final assembly code. The second group of numbers identifies the manufacturer: 01 and 02 = AEG 07 and 40 = Motorola 10 and 20 = Nokia 41and 44 = Siemens 51= Sony, Siemens, Ericsson The third set is the serial number and the last single digit is an additional number (usually 0).

IMSI

International Mobile Subscriber Identity. Identifier used by GSM networks to identify different cellular devices.

MMS

Short for *Multimedia Messaging Service*. With an MMS compatible handset, you can send and receive a combination of text, images, sound and video.

Multiplexing

Multiplexing means the sending of multiple signals or streams of information on a carrier at the same time in the form of a single, complex signal and then recovering the separate signals at the receiving end.

Throughout the WWAN Toolkit manual we are referring to the toolkit's ability to create and handle several virtual COM ports to support modem interaction when connected via RAS or via the Connection Manager.

Network Operator

See *Service Provider*

PAP

PAP (or Password Authentication Protocol) is the most basic form of authentication, in which a user's name and password are transmitted over a network and compared to a table of name-password pairs. Typically, the passwords stored in the table are encrypted. The Basic Authentication feature built into the HTTP protocol uses PAP.

PDP

The PDP context is a data structure present on both the SGSN and the GGSN which contains the subscriber's session information when the subscriber has a session active. When a mobile device wants to use GPRS, it must first attach and then activate a PDP context. This allocates a PDP context data structure in the SGSN that the subscriber is currently visiting and the GGSN serving the subscriber's access point.

The recorded data includes:

- The subscriber's IP address
- The subscriber's IMSI
- The subscriber's

- Tunnel ID (TEID) at the GGSN
- Tunnel ID (TEID) at the SGSN

The tunnel ID (TEID) is a number allocated by the GSN which identifies the tunnelled data related to a particular PDP context.

Packet Switching

Packet switching is a method of switching data in a network where individual packets of a set size and format are accepted by the network and delivered to their destinations. The sequence of the packets is maintained and the destination established by the exchange of control information (also contained in the packets) between the sending terminal and the network before the transmission starts. The network is open to all users, all the time, with packets from the various nodes being interleaved throughout the network. The packets can be sent in any order, as the control information sent at the beginning of the transmission ensures they are interpreted in the correct order at the receiving end. Because each packet carries its own control instructions, it can use any route to reach its destination. As used in GPRS and 1xRTT networks.

PIN

Personal Identification Number. A code used for all GSM based phones to establish authorization for access to certain functions or information. The PIN code is delivered together with your network subscription.

PSD

Short for *Packet Switched Data*. See also *Packet Switching*.

PUK

Personal Unblocking Code. You will need to get this from your service provider if you enter the wrong PIN code 3 times in a row.

RAS

Short for *Remote Access Service*. RAS makes it possible to connect a remote client computer to a network server over a WAN link or a VPN. The remote computer then functions on the server's LAN as if the remote computer was connected to the LAN directly.

Roaming

Roaming within your home network means that your mobile phone automatically sets up communication procedures with different radio base stations as you are moving. International roaming means that you can use networks other than your own when travelling abroad.

Service Provider

The *service provider* is the company that owns your mobile phone’s signature on a contract and bills you for your calls. Also known as *Network Operator*.

SGSN

Short for *Serving GPRS Support Node*. The SGSN is the node which in a sense carries out the same function as the Local Agent in Mobile IP. However, an SGSN is considerably more complex since it also performs a full set of interaction with the connected radio network.

SIM

Short for *Subscriber Identity Module*. A SIM-card is a smart card containing information that is inserted into the 700C with a built in GSM module to facilitate connection to a network operator. The card can also store miscellaneous data used by the cell phone, such as address books, SMSs etc.

SMS

Short Message Service is the transmission of short text messages to and from a mobile phone, fax machine and/or IP address. Messages must not be longer than 160 alpha-numeric characters and contain no images or graphics.

TDMA

Short for *Time Division Multiple Access,* a technology for delivering digital wireless services using time-division multiplexing. TDMA works by dividing a radio frequency into time slots and then allocating slots to multiple calls. In this way, a single frequency can support multiple data channels simultaneously.

UMTS

Short for Universal Mobile Telecommunications System, a 3G mobile technology that is capable of delivering information at speeds up to 2Mbps.

WWAN

Short for *Wide Wireless Area Network*. A common term to describe cellular networks like GPRS, CDMA etc.

A Toolkit Files

This chapter lists all the files that are installed with the toolkit, along with brief descriptions of each file.

WWAN Toolkit: Files installed

[Cab Files]

Modems_MC45_All.cab	Modem definitions for 760 GSM
WWANToolkit5_2.ARM.CAB	Toolkit cab file, to be installed on device
[Cab Files\Signed]	
WWANToolkit5_2.ARM.CAB	Toolkit cab file, to be installed on a secure CN3

[Connectivity API]

[Connectivity API\C++\include]	
StructUtil.cpp	Utility class to be used when extracting information (e.g. GetConnection)
StructUtil.h	Header for the above cpp
ConnectivityErrorCodes.h	Error code definitions
ConnectivityObjects.h	Events, structs and other generic definitions
ConnectivityToolkit.h	Main header, method definitions
[Connectivity API\C++\lib]	
ConnectivityApi.dll	Connectivity Binary
ConnectivityApi.lib	Library file
[Connectivity API\DotNet\doc]	
Files used during installation	
[Connectivity API\DotNet\lib]	
ConnectivityApi.dll	Connectivity binary
Intermec.Communications.WWAN.ConnectivityApi.dll	.NET wrapper binary
[Connectivity API\Java\lib]	
ConnectivityApi.dll	Connectivity binary
ConnectivityJNI.dll	JNI wrapper binary
connectivity.jar	Library file

[Documentation]

All documentation can be accessed through **Start Menu** -> **WWAN Toolkit** -> **Documentation**

[Plugins]

[Plugins\DotNet]
C# Plugin/Wizard installed in Visual Studio, see Tools section of the User Guide for more information.

[Samples]

Sample source code assembled according to API. Please refer to the "samples" section of the User Guide for more information.

[Tools]

[Tools\DeviceInfo]
Device information tool. Please refer to the tools section of the User Guide for more information.
[Tools\Profile Manager]
Connectivity Profile manager tool, please refer to the tools section of the User's Guide for more information.

[WWAN Toolkit Core API]

[WWAN Toolkit Core API\C++\include]	
WWANErrorCodes.h	Error code definitions
WWANToolkit.h	Main header file
[WWAN Toolkit Core API\C++\lib]	
WWANToolkit.dll	Main Core binary
wwantoolkit_mapirule.dll	MAPI rule used by Phone Edition to direct SMS messages. COM dll, must be registered on device.
WWANToolkit.lib	Library file
[WWAN Toolkit Core API\Dotnet\doc\userguide]	
Files used during installation, produces help integrated into Visual Studio.	
[WWAN Toolkit Core API\Dotnet\lib]	
Intermec.Communications.WWAN.dll	Core .NET wrapper

WWANToolkit.dll	Main Core binary
wwantoolkit_mapirule.dll	MAPI rule used by Phone Edition to direct SMS messages. COM dll, must be registered on device.
[WWAN Toolkit Core API\Emulator]	
wwantoolkit.dll	Emulator dll compiled for x86 processors, read more in the emulator section of the User Guide.
wwantoolkit.lib	Library file.
[WWAN Toolkit Core API\Java\lib]	
WWANjni.dll	JNI wrapper for WWAN Toolkit Core API
WWANToolkit.dll	WWAN Toolkit core binary
wwantoolkit_mapirule.dll	MAPI rule used by Phone Edition to direct SMS messages. COM dll, must be registered on device.
wwantk.jar	Library file

B Connection Types

This appendix provides detailed descriptions of each connection type that can be used when you are working with WWAN Toolkit Connectivity development.

Packet Switched Settings (Basic)

Parameter	Typical Value	Description
APN (Access Point Name)	Internet.vodafone.net	GPRS networks only. APN is obtained from your network operator. Some networks do not require an APN.
User Name	XXXX	User name obtained from network operator
Password	YYYY	Password obtained from network operator
Use Connection Manager	Unchecked (FALSE)	This checkbox lets your connection be registered with MS Connection Manager. MS Connection manager allows applications such as Pocket Internet Explorer to use your connection. The drawback is that it also prompts notification dialog boxes about the current connection manager status and permits the user to terminate a connection manually.
Allow Roaming	Checked (TRUE)	If this checkbox is checked the connection will be able to connect even if the WWAN radio is currently roaming through a different network than your home network. Remember that roaming can be costly.

Packet Switched Settings (Advanced)

Parameter	Typical Value	Description
PDP Address	192.199.199.2	Use a specific PDP address
Command String	SGAUTH=2	AT command to execute before connection is attempted.
Disable CHAP	FALSE	Some networks require you to disable CHAP when connecting with PAP authentication.
Disable PAP	FALSE	Some networks requires you to disable PAP when connecting with CHAP authentication.

Circuit Switched Settings (Basic)

Parameter	Typical Value	Description
Phone Number	+46811112222	Phone number supplied by your network operator to connect to their network.
User Name	XXXX	User name obtained from network operator
Password	YYYY	Password obtained from network operator
Use Connection Manager	Unchecked (FALSE)	This checkbox lets your connection be registered with MS Connection Manager. MS Connection manager allows applications such as Pocket Internet Explorer to use your connection. The drawback is that it also prompts notification dialog boxes about the current connection manager status and permits the user to terminate a connection manually.
Allow Roaming	Checked (TRUE)	If this checkbox is checked the connection will be able to connect even if the WWAN radio is currently roaming through a different network than your home network. Remember that roaming can be costly.

Circuit Switched Settings (Advanced)

Parameter	Typical Value	Description
Command String	+CBST=71,0,1	AT command to execute before connection is attempted.

Wireless LAN Settings

Parameter	Typical Value	Description
Use Intermec Settings	Unchecked (FALSE)	If unchecked all parameters are configured through the Profile Manager Tool. If checked, parameters on the device will be used according to the selected wireless profile.
Profile	Profile_1 – Profile_4	Only when “Use Intermec Settings” is checked.
SSID	myWLAN	The SSID of your wireless network.
Network Type	Infrastructure	Infrastructure = switcher based network. Ad-Hoc = Device to device network
Channel	1-11	Only when Ad-Hoc has been selected.

Worldwide Headquarters
6001 36th Avenue West
Everett, Washington 98203
U.S.A.

tel 425.348.2600

fax 425.355.9551

www.intermec.com

WWAN Toolkit User's Guide

P/N 1-960641-03