# 7 Programming

The following programming information pertains to the 700 Series Color Mobile Computer:

- Creating CAB Files (page 224)
- Customization and Lockdown (page 241)
- FTP Server (page 242)
- Kernel I/O Control Functions (page 250)
- Network Selection APIs (page 266)
- Notifications (page 289)
- Reboot Functions (page 291)
- Remapping the Keypad (page 292)



Note: "700 Color" pertains to 740, 741, 750, 751, 760, and 761 Computers unless otherwise noted.

# **Creating CAB Files**

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

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

Use the CAB Wizard application (CABWIZ.EXE) to generate a .CAB file for your application.

# **Creating Device-Specific CAB Files**

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

- 1 Create an .INF file with Windows CE-specific modifications (page 224).
- **2** *Optional* Create a SETUP.DLL file to provide custom control of the installation process (page 236).
- **3** Use the CAB Wizard to create the .CAB file, using the .INF file, the optional SETUP.DLL file, and the device-specific application files as parameters (*page 239*).

# **Creating an .INF File**

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

# [Version]

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

## Required? Yes

- Signature: "signature\_name" "\$Windows NT\$"
- **Provider**: "*INF\_creator*" The company name of the application, such as "Microsoft."
- CESignature "\$Windows CE\$"

# **Example**

## [Version]

```
Signature = "$Windows NT$"
Provider = "Intermec"
CESignature = "$Windows CE$"
```

# [CEStrings]

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

# Required? Yes

• AppName: app\_name

Name of the application. Other instances of %AppName% in the .INF file are replaced with this string value, such as RP32.

• InstallDir: default\_install\_dir

Default installation directory on the device. Other instances of %Install-Dir% in the .INF file are replaced with this string value. Example: \SDMMC\_Disk\%AppName%

# **Example**

[CEStrings]

AppName="Game Pack"
InstallDir=%CE1%\%AppName%

# [Strings]

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

# Required? No

• string\_key: value

String consisting of letters, digits, or other printable characters. Enclose *value* in double quotation marks """ if the corresponding string key is used in an item that requires double quotation marks. No string\_keys is okay.

# **Example**

[Strings]

reg\_path = Software\Intermec\My Test App

# [CEDevice]

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

# Required? Yes

- **ProcessorType**: processor\_type
  The value that is returned by **SYSTEMINFO**.dwProcessorType.For example, the value for the ARM CPU is 2577
- UnsupportedPlatforms: platform\_family\_name
  This lists known unsupported platform family names. If the name specified in the [CEDevice.xxx] section is different from that in the [CEDevice] section, both platform\_family\_name values are unsupported for the microprocessor specified by xxx. That is, the list of unsupported platform family names is appended to the previous list of unsupported names. Application Manager will not display the application for an unsupported platform. Also, a user will be warned during the setup process if the .CAB file is copied to an unsupported device.

# **Example**

[CEDevice]

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

- VersionMin: minor version
  - Numeric value returned by **OSVERSIONINFO**.dwVersionMinor. The .CAB file is valid for the currently connected device if the version of this device is greater than or equal to **VersionMin**.
- VersionMax: major\_version

  Numeric value returned by OSVERSIONINFO.dwVersionMajor. The

  .CAB file is valid for the currently connected device if the version of this device is less than or equal to VersionMax.
- BuildMin: build\_number

  Numeric value returned by OSVERSIONINFO.dwBuildNumber. The

  .CAB file is valid for the currently connected device if the version of this device is greater than or equal to BuildMin.
- BuildMax: build\_number
   Numeric value returned by OSVERSIONINFO.dwBuildNumber. The
   .CAB file is valid for the currently connected device if the version of this device is less than or equal to BuildMax.

# **Example**

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

```
[CEDevice]
                                ; A "template" for all platforms
UnsupportedPlatforms = pltfrm1
                                ; Does not support pltfrm1
; The following specifies version 1.0 devices only.
VersionMin = 1.0
VersionMax = 1.0
[CEDevice.ARM]
                            ; Inherits all [CEDevice] settings
; This will create a .CAB file specific to ARM devices.
ProcessorType = 2577 ; ARM .cab file is valid for ARM microprocessors.
UnsupportedPlatforms =
                            ; pltfrm1 is still unsupported
; The following overrides the version settings so that no version checking is
performed.
VersionMin =
VersionMax =
[CEDevice.MIPS]
                              ; Inherits all [CEDevice] settings
; This will create a .CAB file specific to "MIPS" devices.
ProcessorType = 4000 ; MIPS .CAB file is valid for MIPS
microprocessor.
UnsupportedPlatforms =pltfrm2 ; pltfrm1, pltfrm2 unsupported for MIPs .CAB
file.
```



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

# [DefaultInstall]

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

# Required? Yes

- Copyfiles: copyfile\_list\_section
   Maps to files defined later in the .INF file, such as Files.App, Files.Font, and Files.Bitmaps.
- AddReg: add\_registry\_section Example: RegSettings.All
- CEShortcuts: shortcut\_list\_section
  String that identifies one more section that defines shortcuts to a file, as defined in the [CEShortcuts] section.
- CESetupDLL: setup\_DLL
  Optimal string that specifies a SETUP.DLL file. It is written by the Independent Software Vendor (ISV) and contains customized functions for operations during installation and removal of the application. The file must be specified in the [SourceDisksFiles] section.
- CESelfRegister: self\_reg\_DLL\_filename
  String that identifies files that self-register by exporting the DllRegisterServer and DllUnregisterServer Component Object Model (COM)
  functions. Specify these files in the [SourceDiskFiles] section. During
  installation, if installation on the device fails to call the file's exported
  DllRegisterServer function, the file's exported DllUnregisterServer
  function will not be called during removal.

## Example

## [DefaultInstall]

AddReg = RegSettings.All CEShortcuts = Shortcuts.All

# [SourceDiskNames]

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

## Required? Yes

- disk\_ordinal: disk\_label,,path
  1=,"App files", C:\Appsoft\RP32\...
  2=,"Font files",,C:\RpTools\...
  3=,"CE Tools",,C:\windows ce tools...
- CESignature: "\$Windows CE\$"

## Example

# [SourceDiskFiles]

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

# Required? Yes

• filename: disk\_number[,subdir]
RPM.EXE = 1,c:\appsoft\...
WCESTART.INI = 1
RPMCE212.INI = 1
TAHOMA.TTF = 2



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

## Example

#### 

sample.exe = 2 ; Uses the SourceDisksNames.MIPS identification of 2.

# [DestinationDirs]

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

# Required? Yes

• file\_list\_section: *0,subdir*String that identifies the destination directory. The following list shows the string substitutions supported by Windows CE. Use these only for

the beginning of the path. \ %CE1% \Program Files \Windows %CE2% %CE3% \My Documents %CE4% \Windows\Startup %CE5% \My Documents %CE6% \Program Files\Accessories %CE7% \Program Files\Communication %CE8% \Program Files\Games %CE9% \Program Files\Pocket Outlook %CE10% \Program Files\Office %CE11% \Windows\Start Menu\Programs %CE12% \Windows\Start Menu\Programs\Accessories %CE13% \Windows\Start Menu\Programs\Communications %CE14% \Windows\Start Menu\Programs\Games %CE15% \Windows\Fonts

%CE15% \Windows\Fonts %CE16% \Windows\Recent %CE17% \Windows\Start Menu %InstallDir%

Contains the path to the target directory selected during installation. It is declared in the [CEStrings] section

%AppName%

Contains the application name defined in the [CEStrings] section.

# **Example**

## [DestinationDirs]

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

# [CopyFiles]

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

# Required? Yes

- copyfile\_list\_section: destination\_filename,[source\_filename]
  The source\_filename parameter is optional if it is the same as destination\_filename.
- copyfile\_list\_section: flags
  The numeric value that specifies an action to be done while copying files. The following table shows values supported by Windows CE.

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

# Example

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

# [AddReg]

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

# Required? No

- add\_registry\_section: registry\_root\_string
  String that specifies the registry root location. The following list shows the values supported by Windows CE.
  - HKCR Same as HKEY\_CLASSES\_ROOT
  - HKCU Same as HKEY\_CURRENT\_USER
  - HKLM Same as HKEY LOCAL MACHINE
- add\_registry\_section: value\_name Registry value name. If empty, the "default" registry value name is used.
- add\_registry\_section: *flags*Numeric value that specifies information about the registry key. The following table shows the values that are supported by Window CE.

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

## Example

AddReg = RegSettings.All

## [RegSettings.All]

```
HKLM, %reg_path%, .0x000000001,3 ; <default> = "alpha" 
HKLM, %reg_path%, test, 0x00010001,3 ; Test = 3 
HKLM, %reg_path%\new, another, 0x00010001,6 ; New\another = 6
```

# [CEShortCuts]

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

# Required? No

- **shortcut\_list\_section**: *shortcut\_filename*String that identifies the shortcut name. It does not require the .LNK extension.
- shortcut\_list\_section: shortcut\_type\_flag

  Numeric value. Zero or empty represents a shortcut to a file; any nonzero numeric value represents a shortcut to a folder.
- **shortcut\_list\_section**: *target\_file\_path*String value that specifies the destination location. Use the target file name for a file, such as MyApp.exe, that must be defined in a file copy list. For a path, use a *file\_list\_section* name defined in the [**Destination-Dirs**] section, such as *DefaultDestDir*, or the *%InstallDir*% string.
- shortcut\_list\_section: standard\_destination\_path
  Optional string value. A standard %CEx% path or %InstallDir%. If no
  value is specified, the shortcut\_list\_section name of the current section or
  the DefaultDestDir value from the [DestinationDirs] section is used.

# **Example**

```
CEShortcuts = Shortcuts.All
[Shortcuts.All]
Sample App, 0, sample.exe
                                  ; Uses the path in DestinationDirs. Sample
App, 0, sample.exe, %InstallDir%; The path is explicitly specified.
                     Sample .INF File
                     ; Required section
[Version]
Signature = "$Windows NT$"
Provider = "Intermec Technologies Corporation"
CESignature = "$Windows CE$"
;[CEDevice]
;ProcessorType =
[DefaultInstall]
                    ; Required section
CopyFiles = Files.App, Files.Fonts, Files.BitMaps, Files.Intl,
Files.TelecomNcsCE, Files.Windows, Files.Import, Files.Export, Files.Work,
Files.Database, Files.WinCE AddReg = RegSettings.All ; CEShortcuts =
Shortcuts.All
[SourceDisksNames] ; Required section
1 = ,"App files" ,,c:\appsoft\...
2 = ,"Font files" ,,c:\WinNT\Fonts
3 = "CE Tools", c:\windows ce tools\wce400\700ie\mfc\lib\x86
[SourceDisksFiles]
                    ; Required section
rpm.exe = 1,C:\Appsoft\program\wce400\WCEX86Rel700
wcestart.ini = 1
```

#### Chapter 7 — Programming

```
rpmce212.ini = 1
intermec.bmp = 1
rpmlogo.bmp = 1
rpmname.bmp = 1
import.bmp = 1
export.bmp = 1
clock.bmp = 1
printer.bmp = 1
filecopy.bmp = 1
readme.txt = 1
lang_eng.bin = 1
rpmdata.dbd = 1,database\wce1
tahoma.ttf = 2
mfcce212.dll = 3
olece212.dll = 3
olece211.dll = 1,c:\windows ce tools\wce400\NMSD61102.11\mfc\lib\x86
rdm45wce.dll = 1,c:\rptools\rdm45wce\4_50\lib\wce400\wcex86rel
picfmt.dll = 1,c:\rptools\picfmt\1_00\wce400\wcex86rel6110
fmtctrl.dll = 1,c:\rptools\fmtctrl\1_00\wce400\wcex86rel6110
ugrid.dll = 1,c:\rptools\ugrid\1_00\wcex86rel6110
simple.dll = 1,c:\rptools\pspbm0c\1_00\wce400\wcex86rel
psink.dll = 1,c:\rptools\psink\1_00\wce400\WCEX86RelMinDependency
pslpwce.dll =1,c:\rptools\pslpm0c\1_00\wce400\WCEX86RelMinDependency
npcpport.dll = 1,c:\rptools\cedk\212_03\installable drivers\printer\npcp
idexcom.dll = 1,c:\ptools\psdxm0c\1_00\x86
ncsce.exe = 1,c:\rptools\ncsce\1_04
nrinet.dll = 1,c:\rptools\ncsce\1_04
[DestinationDirs] ; Required section
;Shortcuts.All = 0,%CE3% ; \Windows\Desktop
Files.App = 0,%InstallDir%\DataBase
Files.BitMaps = 0,%InstallDir%\Bitmaps
Files.Fonts = 0,%InstallDir%\Fonts
Files.Intl = 0,%InstallDir%\Intl
Files.TelecomNcsCE = 0,%InstallDir%\Telecom\NcsCE
Files.Windows = 0,%InstallDir%\Windows
[CEStrings]
                  ; Required section
AppName = Rp32
InstallDir = \storage_card\%AppName%
[Strings]
                  ; Optional section
;[Shortcuts.All]
;Sample App, 0, sample.exe
                                        ; Uses the path in DestinationDirs.
;Sample App, 0, sample.exe, %InstallDir%; The path is explicitly specified.
[Files.App]
rpm.exe,,,0
rpm.ini,rpmce212.ini,,0
mfcce212.dll,,,0
olece212.dll,,,0
olece211.dll,,,0
rdm45wce.dll,,,0
picfmt.dll,,,0
```

```
fmtctrl.dll,,,0
ugrid.dll,,,0
simple.dll,,,0
psink.dll,,,0
pslpwce.dll,,,0
npcpport.dll,,,0
;dexcom.dll,,,0
[Files.DataBase]
rpmdata.dbd,,,0
[Files.Fonts]
tahoma.ttf,,,0
[Files.BitMaps]
intermec.bmp,,,0
rpmlogo.bmp,,,0
rpmname.bmp,,,0
import.bmp,,,0
export.bmp,,,0
clock.bmp,,,0
printer.bmp,,,0
filecopy.bmp,,,0
[Files.Intl]
lang_eng.bin,,,0
[Files.TelecomNcsCE]
ncsce.exe,,,0
nrinet.dll,,,0
[Files.Windows]
readme.txt,,,0
[Files.Import]
readme.txt,,,0
[Files.Export]
readme.txt,,,0
[Files.Work]
readme.txt,,,0
[Files.WinCE]
wcestart.ini,,,0
[RegSettings.All]
HKLM, "SOFTWARE\Microsoft\Shell\AutoHide",,0x00010001,1
; Autohide the taskbar HKLM, "SOFTWARE\Microsoft\Shell\OnTop",,0x00010001,0
 ; Shell is not on top
HKLM, "SOFTWARE\Microsoft\Clock", SHOW_CLOCK, 0x00010001, 0
  ; Clock is not on taskbar
```

# **Using Installation Functions in SETUP.DLL**

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

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



Note; Use [DefaultInstall] > CESelfRegister (page 228) in the .INF file to point to SETUP.DLL.

# After the CAB File Extraction

Cab files that need to cause a warm reset after cab extraction will need to create the \_\_RESETMEPLEASE\_\_.TXT file in the "\Windows" directory. The preferred method to create this file is within the DllMain portion of the SETUP.DLL file. It looks like this:

```
#include <windows.h>
#include <Tlhelp32.h>
#include <winioctl.h>
#include <ce_setup.h> // in the public SDK dir
#define IOCTL_TERMINAL_RESET CTL_CODE (FILE_DEVICE_UNKNOWN,FILE_ANY_ACCESS,
2050, METHOD_NEITHER)
BOOL APIENTRY DllMain( HANDLE h, DWORD reason, LPVOID lpReserved )
 return TRUE;
} // DllMain
//***************************
// $DOCBEGIN$
// BOOL IsProcessRunning( TCHAR * pname );
// Description: Get process table snapshot, look for pname running.
//
// Arguments: pname - pointer to name of program to look for.
// for example, app.exe.
// Returns: TRUE - process is running.
          FALSE - process is not running.
//
// $DOCEND$
//**********************
BOOL IsProcessRunning( TCHAR * pname )
 HANDLE hProcList;
```

```
PROCESSENTRY32 peProcess;
  DWORD thDeviceProcessID;
  TCHAR lpname[MAX_PATH];
  if ( !pname | | !*pname ) return FALSE;
  _tcscpy( lpname, pname );
  _tcslwr( lpname );
  hProcList = CreateToolhelp32Snapshot( TH32CS_SNAPPROCESS, 0 );
  if ( hProcList == INVALID_HANDLE_VALUE ) {
    return FALSE;
  } // end if
  memset( &peProcess, 0, sizeof(peProcess) );
  peProcess.dwSize = sizeof(peProcess);
  if ( !Process32First( hProcList, &peProcess ) ) {
    CloseToolhelp32Snapshot( hProcList );
    return FALSE;
  } // end if
  thDeviceProcessID = 0;
  do {
    _tcslwr( peProcess.szExeFile );
    if ( _tcsstr( peProcess.szExeFile, lpname ) ) {
      thDeviceProcessID = peProcess.th32ProcessID;
      break;
    } // end if
  } while ( Process32Next( hProcList, &peProcess ) );
  if ( ( GetLastError() == ERROR_NO_MORE_FILES ) && ( thDeviceProcessID == 0
) ) {
    CloseToolhelp32Snapshot( hProcList );
    return FALSE;
  } // end if
  CloseToolhelp32Snapshot( hProcList );
  return TRUE;
} // IsProcessRunning
codeINSTALL_INIT Install_Init(
  HWND hwndParent,
  BOOL fFirstCall,
  BOOL fPreviouslyInstalled,
  LPCTSTR pszInstallDir )
  return codeINSTALL_INIT_CONTINUE;
codeINSTALL_EXIT Install_Exit (
  HWND hwndParent,
  LPCTSTR pszInstallDir,
  WORD cFailedDirs,
  WORD cFailedFiles,
  WORD cFailedRegKeys,
```

}

```
WORD cFailedRegVals,
  WORD cFailedShortcuts )
  HANDLE h;
  TCHAR srcfile[MAX_PATH];
  TCHAR dstfile[MAX_PATH];
  if (cFailedDirs || cFailedFiles || cFailedRegKeys ||
    cFailedRegVals || cFailedShortcuts)
    return codeINSTALL_EXIT_UNINSTALL;
  if ( IsProcessRunning( L"autocab.exe" ) )
    h = CreateFile( L"\\Windows\\ resetmeplease .txt",
       (GENERIC_READ | GENERIC_WRITE), 0, NULL, CREATE_ALWAYS,
      FILE_ATTRIBUTE_HIDDEN, NULL );
    if ( h != INVALID_HANDLE_VALUE )
       CloseHandle( h );
    else
       // Couldn't create the file. If it failed because the file already
exists, it is not fatal.
       // Otherwise, notify user of the inability to reset the device and they
will have to
       // perform it manually after all of the installations are complete.
    } // end if
  else
    DWORD dret;
    h = CreateFile( L"SYI1:",
       (GENERIC_WRITE | GENERIC_READ), 0, NULL, OPEN_EXISTING,
       FILE_ATTRIBUTE_NORMAL, NULL );
    // Force a warm start NOW.
    if ( h != INVALID_HANDLE_VALUE )
      DeviceIoControl( h, IOCTL TERMINAL RESET, NULL, 0, NULL, 0, &dret,
NULL);
       // Won't return, but we'll show clean up anyway
      CloseHandle( h );
    }
    else
      // Couldn't access SYSIO. Notify user.
    } // end if
  } // end if
  return codeINSTALL_EXIT_DONE;
codeUNINSTALL INIT
Uninstall_Init(
  HWND hwndParent,
  LPCTSTR pszInstallDir ) {
```

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

codeUNINSTALL_EXIT
Uninstall_Exit(HWND hwndParent) {
   // TODO: Perform the reverse of INSTALL_EXIT here
   return codeUNINSTALL_EXIT_DONE;
}
```

The system software looks for the following directory structure and files on the installed media card whether it be an SD card or CF card or embedded flash file system. No other folders need exist.

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

# **Creating CAB Files with CAB Wizard**

After you create the .INF file and the optional SETUP.DLL file, use the CAB Wizard to create the .CAB file. The command-line syntax for the CAB Wizard is as follows:

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

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

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

## Example

This example creates .CAB files for the ARM and MIPS microprocessors, assuming the Win32 SETUP.INF file contains the ARM and MIPS tags: cabwiz.exe "c:\myfile.inf" /err myfile.err /cpu arm mips



**Note:** CABWIZ.EXE, MAKECAB.EXE, and CABWIZ.DDF (Windows CE files available on the Windows CE Toolkit) must be installed in the same directory on the desktop computer. Call CABWIZ.EXE using its full path for the CAB Wizard application to run correctly.

# **Troubleshooting the CAB Wizard**

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

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

# **Customization and Lockdown**

Pocket PC (Windows Mobile) is a hardware specification created by Microsoft Corporation. Devices that wish to carry the Pocket PC logo must meet the minimum hardware requirements set in the Pocket PC specification. Manufacturers are free to add extra hardware functionality.

Pocket PC devices also use a specialized version of the CE operating system. This operating system is built from Windows CE 4.2 but contains customizations, most notably the lack of a desktop and the addition of the Today Screen.

To carry the Pocket PC logo, all devices must be tested at an Independent Test Laboratory. The ITL testing is done based on Microsoft requirements. The test lab then reports the findings back to Microsoft Corporation and Intermec Technologies. If the 700 Color Computer passed all tests, Intermec is allowed to ship the device with the Pocket PC logo. Each time the operating system is modified, Intermec must resubmit to ITL testing.

This means we cannot change the operating system much and still be a Pocket PC device. For example, if we remove Word from the Start menu, the device would fail ITL testing and we would not be able to ship devices with the Pocket PC logo.

Although many customers want a Pocket PC device, some customers would prefer that their users not have access to all of the Pocket PC features. Intermec cannot customize the operating system in any way but a custom application can:

Delete items from the Start menu, and Programs folder. These items are just shortcuts in the file system so the application is not really being deleted. Cold booting the device will bring these items back so the application will need to be run on every cold boot.

Use the RegFlushKey() API to save a copy of the registry to a storage device. See the 700 Color Management Tools portion for more information on how to do this. Saving a copy of the registry restores most system settings in a cold boot situation.

Use the SHFullScreen() API in conjunction with other APIs to make the application take up the entire display and prevent the start menu from being available.

Remap keys and disable keys on the keypad.

Create a custom SIP.

Make changes to the registry to configure the device.

Should you want your 700 Color Computer to display a full screen, keep in mind that your computer is Pocket-PC certified by Microsoft Corporation. Check out resources on programming for the Pocket PC, using the following links. These give full instructions on how to display full screen.

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

# **FTP Server**

FTP support is provided through the FTP Server application FTPDCE.EXE (MS Windows CE Versions) which is provided as part the base system.

FTPDCE is the Internet File Transfer Protocol (FTP) server process. The server can be invoked from an application or command line. Besides servicing FTP client requests the FTP Server also send a "network announcement" to notify prospective clients of server availability.



**Note**: You should consult the RFC959 specification for proper use of some of these commands at the following URL:

- http://www.ietf.org/rfc/rfc959.txt for the text version, or
- http://www.w3.org/Protocols/rfc959/ for an html version

Do the following to send commands:

- 1 Start an FTP client and connect to the device FTP server.
- **2** Log in with "intermec" as the user name and "cr52401" for the password.
- **3** From the FTP client, send the command.
- **4** Wait for a response.

# **Synopsis**

ftpdce [ options ]

# **Options**

–Aaddr	(where <i>addr</i> is in the form of a.b.c.d) Sets the single target address to which to send the network announcement. <i>Default is broadcast</i> .
-Bbyte	Sets the FTP data block size. Smaller sizes may be useful over slower links. Default is 65536.
-Cname	Sets the device name. Used by Intermec management software.
–Fvalue	Disables the default Intermec account. A value of "0" disables the account. <i>Default is "1"</i> . Note that disabling the default account without providing a working access control list on the server will result in a device that will not accept any FTP connections.
–Hsec	Sets the interval between network announcements in seconds. A value of "0" turns the network announcement off. <i>Default is 30 seconds</i> .
–Iaddr	(where addr is in the form of a.b.c.d) Sets the preferred 6920 Communications Server (optional).
-Llog	(where log is either "0" or "1") Sets the state of logging. Default is 0 (disabled).
-Nsec	Specifies the number of seconds to wait before initially starting FTP server services.
-Pport	Sets the UDP port on which the network announcement will be sent. Default port is 52401.
-Qport	Sets the port on which the FTP Server will listen for connections. Default port is 21.
–Rdir	Sets the FTP mount point to this directory. Default is the root folder of the object store.
-Tscrip	Sets the script name for the 6920 Communications Server to process.
-Uurl	Sets the default URL for this device.
–Z"parms"	Sets extended parameters to be included in the network announcement.

# **Configurable Parameters Via the Registry Editor**

The following parameters receive default values during the installation of the Intermec FTP Server components. A few of the parameters are visible in the registry by default, but most must be created in order to modify the default behavior of the FTP server.

## **BlockSize**

Setting this parameter configures the Intermec FTP Server to transmit and receive Ethernet packets using the specified data block size. By default, the FTP server transmits and receives data using a 64K data block size. Adjusting this value may be useful in certain wireless TCP/IP installations.

Key	HKLM\Software\Intermec\IFTP
Value Type	REG_DWORD - data block size, in bytes.
Valid Range	0x100-0x10000 (256-65536 decimal).
Default	65536

## DeviceName

This parameter configures the Intermec FTP Server to include the specified device name in the Intermec Device Network Announcement (IDNA). Adjusting this value may be useful in assigning a symbolic name to this device for asset tracking.

Key	HKLM\Software\Intermec\IFTP
Value Type	REG_SZ
Valid Range	None.
Default	None.

## **DeviceURL**

This parameter configures the Intermec FTP Server to transmit the specified URL in the IDNA. This can be used by Intermec management software for asset management.

Key	HKLM\Software\Intermec\IFTP
Value Type	REG_SZ
Valid Range	None.
Default	None.

# **IDNATarget**

This parameter configures the Intermec FTP Server to transmit the IDNA to a specific destination instead of a general UDP broadcast. This parameter is useful on networks that do not allow UDP broadcasts to be routed between subnets. The use of this parameter restricts the reception of the IDNA to the target destination only.

Key	HKLM\Software\Intermec\IFTP
Value Type	REG_SZ
Valid Range	None.
Default	None.

## ManifestName

This parameter configures the Intermec FTP Server to transmit the specified manifest name in the IDNA. This parameter is used by the Intermec 6920 Communications Server for communication transactions. See the 6920 Communications Server documentation for proper use of this parameter.

Key	HKLM\Software\Intermec\IFTP
Value Type	REG_SZ
Valid Range	None.
Default	iftp.ini

# **PauseAtStartup**

This parameter configures the Intermec FTP Server to sleep for the specified number of seconds before making the FTP service available on the device.

Key	HKLM\Software\Intermec\IFTP
Value Type	REG_DWORD - stored in seconds.
Valid Range	None.
Default	0

#### Root

This parameter configures the Intermec FTP Server to set the root of the FTP mount point to the specified value. *Note that this must map to an existing directory or you will not be able to log into the FTP Server.* 

Key	HKLM\Software\Intermec\IFTP
Value Type	REG_SZ
Valid Range	None.
Default	· ·

# **Transferring Files Over TCP/IP Networks**

The File Transfer Protocol (FTP) server transfers files over TCP/IP networks. The FTPDCE.EXE program is a version that does not display a window, but can run in the background.

FTPDCE is the Internet File Transfer Protocol (FTP) server process. The server can be invoked from an application or command line. Besides servicing FTP client requests, the FTP Server also sends a "network announcement" to notify prospective clients of server availability.

## **Remarks**

The FTP Server currently supports the following FTP requests:

CDUP	Changes to the parent directory of the current working directory.	
CWD	Changes working directory.	
DELE	Deletes a file.	
HELP	Gives help information.	
LIST	(This FTP request is the same as the ls -lgA command). Gives list files in a directory.	
MKD	Makes a directory.	
MODE	(Always Uses Binary). Specifies data transfer mode.	
NLST	(Not supported) Gives a name list of files in directory (this FTP request is the same as the ls command).	
NOOP	Does nothing.	
PASS	Specifies a password.	
PWD	Prints the current working directory.	
QUIT	Terminates session.	
RETR	Retrieves a file.	
RMD	Removes a directory.	
RNFR	Specifies rename-from file name.	
RNTO	Specifies rename-to file name.	
STOR	Stores a file.	
SYST	Shows the operating system type of server system.	
TYPE	(Binary transfers only.) Specifies the data transfer type with the Type parameter.	
USER	Specifies user name.	
XCUP	(Not Normally Used) Changes the parent directory of the current working directory.	
XCWD	(Not Normally Used) Changes the current directory.	
XMKD	(Not Normally Used) Creates a directory.	
XPWD	(Not Normally Used) Prints the current working directory.	
XRMD	(Not Normally Used) Removes a directory.	

SITE	The following extended OEM commands are supported by the SITE request. For Microsoft FTP clients, you can send site commands by preceding the command with "quote" such as "quote site status			
	ATTRIB	Gets or sets the attributes of a given file. (SITE ATTRIB)		
		Usage  QUOTE SITE ATTRIB [+R   -R] [+A   -A ] [+S   -S] [+H   -H] [[path] filename]  + Sets an attribute.  - Clears an attribute.  R Read-only file attribute.  A Archive file attribute.  S System file attribute.  H Hidden file attribute.		
		To retrieve the attributes of a file, only specify the file. The server response will be: 200-AD SHRCEIX filename		
		If the flag exists in its position shown above, it is set. Also, in addition to the values defined above, there is also defined:  C Compressed file attribute.  E Encrypted file attribute.  I INROM file attribute.  X XIP file attribute (execute in ROM, not shadowed in RAM).		
	ВООТ	Reboots the server OS. This will cause the system on which the server is executing to reboot. The FTP Server will shut down cleanly before reboot. All client connections will be terminated. Cold boot is default except for the PocketPC build in which the default is warm boot. (SITE BOOT)		
		Usage: QUOTE SITE BOOT [WARM   COLD]		
	COPY	Copies a file from one location to another. (SITE COPY)		
		Usage: QUOTE SITE COPY [source] [destination]		
		<pre>Example: QUOTE SITE COPY \\Storage Card\one.dat' \\Stor- age Card\two.dat'</pre>		
	EXIT	Exits the FTP Server. This command will shut down the FTP Server thus terminating all client connections. (SITE EXIT)		
		Usage: QUOTE SITE EXIT		
	HELP	Gives site command help information. (SITE HELP)		
		Usage: QUOTE SITE HELP [command]		
	KILL	Terminates a running program. (SITE KILL)		
		Usage: QUOTE SITE KILL [program   pid]		
	LOG	Opens or closes the program log. (SITE LOG)		
		Usage: QUOTE SITE LOG [open [filename]   close]		
	PLIST	Lists the running processes (SITE PLIST)		
		Usage: QUOTE SITE PLIST		
	RUN	Starts a program running. If the program to run has spaces in path or filename, wrapping the name with single quotes is required.		
		Usage: QUOTE SITE RUN [program]		
		Example: QUOTE SITE RUN '\Storage Card\app.exe'		

STATUS	Returns the current settings of the FTP Server. MAC, serial number, model, IP address, network announcement information as well as OS memory usage are returned. (SITE STATUS)  Usage: QUOTE SITE STATUS	
TIMEOUT	Toggles idle timeout between 120 to 1200 seconds (2 to 20 minutes). If this timer expires with no activity between the client and the server, the client connection will be disconnected. If the optional seconds argument is supplied, the server will set the connection timeout to the number of seconds specified. <i>Default is 120 seconds or 2 minutes.</i> (SITE TIMEOUT)	
	Usage: QUOTE SITE TIMEOUT [seconds]	
EKEY	Gives site command electronic key information. (SITE HELP)	
	Usage: QUOTE SITE EKEY [command]	
EVAL	Gives site command electronic value information. (SITE HELP)	
	Usage: QUOTE SITE EVAL [command]	
GVAL	Gives site command general value information. (SITE HELP)	
	Usage: QUOTE SITE GVAL [command]	
PVAL	Gives site command value information. (SITE HELP)	
	Usage: QUOTE SITE PVAL [command]	

The remaining FTP requests specified in RFC 959 are recognized, but not implemented.

The banner returned in the parenthetical portion of its greeting shows the version number of the FTP Server as well as the MAC address, serial number and operating system of the machine hosting the server.

The FTP Server supports browsing from the latest Netscape and Microsoft web browsers. Drag-and-drop capability is available using this environment.

The FTPDCMDS subdirectory contains commands to use from the web browser.

- Click EXITME.BIN to execute a SITE EXIT command.
- Click REBOOTME.BIN to execute SITE BOOT command.
- Use the GET command on these files to have the FTP Server execute these commands.

## • Security:

A customer configurable access control list may be installed on the 700 Color Computer. This list will allow customers to restrict access via the FTP Server to users they wish and is in addition to default Intermec accounts that are disabled using the -F0 option at runtime.

The access control list is named FTPDCE.TXT and is placed in the same directory on the 700 Color Computer as the FTPDCE.EXE server. The FTP Server encrypts this file to keep the information safe from unauthorized users. This file is encrypted when the FTP Server is started so a file that is placed onto the 700 Color Computer after the FTP Server starts will require a restart of the FTP Server to take effect.

The format of the FTPDCE.TXT is as follows:

FTPDCE:user1!passwd1<cr><lf>user2!passwd2<cr><lf>user3
!passwd3<cr><lf>...



Note: The user accounts and passwords are case sensitive. Once the access control list is encrypted on the 700 Color Computer, the FTP Server hides this file from users. Once an access control list is installed on the 700 Color Computer, a new one is not accepted by the FTP Server until the previous one is removed. Encrypted access control lists are not portable between 700 Color Computers.

# Stopping the FTP Server from Your Application

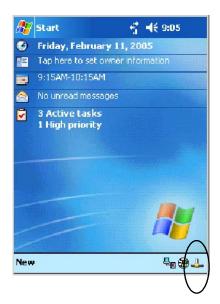
To allow application programmers the ability to programmatically shut down the FTP Server, the FTP Server periodically tests to see if a named event is signaled. The name for this event is "ITC\_IFTP\_STOP" (no quotes).

For examples on how to use events, consult the Microsoft Developer Network Library at www.msdn.com. The MSDN Library is an essential resource for developers using Microsoft tools, products, and technologies. It contains a bounty of technical programming information, including sample code, documentation, technical articles, and reference guides.

# **Autostart FTP**



This automatically starts the FTP Server (FTPDCE.EXE) when the 700 Color Computer is powered on. This is provided with the NDISTRAY program (the Network Driver Interface Specification tray application), which displays the popup menu that currently allows you to load and unload the network drivers. Tap the antenna icon in the System Tray of the Today screen (a sample antenna icon is circled below) for this pop-up menu.



The default is to start the FTP Server at boot time, unless the following registry entry is defined and set to "0" which disables AutoFTP. "1" enables the AutoFTP. The entry can be set from the NDISTRAY pop-up menu by selecting either AutoFTP On or AutoFTP Off.

HKEY\_LOCAL\_MACHINE\Software\Intermec\Ndistray\StartupIFTP

These new entries are located below the selections to load the network drivers. If the StartupIFTP registry key is not defined, the FTP Server is loaded by default, to provide "out-of-the-box" capability for customers who want to begin loading files to the 700 Color Computer without any prior configuration.

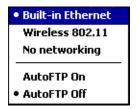


**Note**: If a network driver is unloaded using the NDISTRAY popup menu, and the FTP Server is running, the FTP Server is stopped.

On a resume, if AutoFTP is enabled and the FTP Server is running, it is stopped and restarted. NDISTRAY uses a helper application named RESE-TIFTP to implement the restart on resume feature.

## To do an AutoFTP Installation Check:

- **1** Ensure the FTP Server is running "out-of-the-box" the first time.
- 2 Tap Start > Today to access the Today screen, then tap the antenna icon in the System Tray to bring up the NDISTRAY pop-up menu. Select AutoFTP Off to disable AutoFTP. Do a warm boot and confirm the FTP Server is not running.



**3** Tap **Start** > **Today** to access the Today screen, then tap the antenna icon in the System Tray to bring up the NDISTRAY pop-up menu. Select **AutoFTP On** to enable AutoFTP, reboot, confirm it is running.



- **4** Unload the network driver when the FTP Server is running and confirm that it is not running any more.
- **5** Load the FTP Server, establish a connection, then suspend and resume. The server should still run, but the FTP connection to the client should be dropped.

# **Kernel I/O Controls**

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

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

# IOCTL\_HAL\_GET\_DEVICE\_INFO

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

# **Syntax**

BOOL KernelIoControl( IOCTL\_HAL\_GET\_DEVICE\_INFO, LPVOID lpInBuf, DWORD nInBufSize, LPVOID lpOutBuf, DWORD nOutBufSize, LPDWORD lpBytesReturned );

## **Parameters**

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

## **Return Values**

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

# IOCTL\_HAL\_ITC\_READ\_PARM

# Usage

#include "oemioctl.h"

# **Syntax**

BOOL KernelIoControl( IOCTL\_HAL\_ITC\_READ\_PARM,LPVOID lpInBuf,DWORD nInBufSize,LPVOID lpOutBuf,DWORD nOutBufSize,LPDWORD lpBytesReturned );

#### **Parameters**

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

## **Return Values**

Returns TRUE if function succeeds. Returns FALSE if the function fails. GetLastError() may be used to get the error value. Either ERROR\_INVALID\_PARAMETER or

ERROR\_INSUFFICIENT\_BUFFER may be returned when this function is used to get the error.

## **ID Field Values**

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

## **ID Field Values**

#### ITC NVPARM ETHERNET ID

This IOCTL returns the Ethernet 802.11b or 802.11b/g MAC Address. Six bytes are returned in the buffer pointed to by the *lpOutBuffer* parameter.

## ITC NVPARM SERIAL NUM

This IOCTL returns the serial number of the device in BCD format. Six bytes are returned in the buffer pointed to by the *lpOutBuffer* parameter.

## ITC NVPARM MANF DATE

This IOCTL returns the device date of manufacture in the BCD YYYY/MM/DD format. Four bytes are returned in the buffer pointed to by the *lpOutBuffer* parameter.

## ITC\_NVPARM\_SERVICE\_DATE

This IOCTL returns the device's date of last service in BCD YYYY/MM/DD format. Four bytes are returned in the buffer pointed to by the *lpOutBuffer* parameter.

## **ID Field Values (continued)**

## ITC\_NVPARM\_DISPLAY\_TYPE

This IOCTL returns the device's display type. One byte is returned in the buffer pointed to by the *lpOutBuffer* parameter.

## ITC\_NVPARM\_EDG\_IP

This IOCTL returns the device Ethernet debug IP address. Four bytes are returned in the buffer pointed to by the *lpOutBuffer* parameter.

## ITC\_NVPARM\_EDBG\_SUBNET

This IOCTL returns the device Ethernet debug subnet mask. Four bytes are returned in the buffer pointed to by the *lpOutBuffer* parameter.

## ITC\_NVPARM ECN

This IOCTL returns ECNs applied to the device in a bit array format. Four bytes are returned in the buffer pointed to by the *lpOutBuffer* parameter.

#### ITC NVPARM CONTRAST

This IOCTL returns the device default contrast setting. Two bytes are returned in the buffer pointed to by the *lpOutBuffer* parameter.

## ITC NVPARM MCODE

This IOCTL returns the manufacturing configuration code for the device. Sixteen bytes are returned in the buffer pointed to by the *lpOutBuffer* parameter.

## ITC\_NVPARM\_VERSION\_NUMBER

This IOCTL returns the firmware version for various system components. These values for the *ClassId* field of the PARMS structure are allowed when ITC\_NVPARM\_VERSION\_NUMBER is used in the *id* field:

- VN\_CLASS\_KBD Returns a five-byte string, including null terminator, that contains an ASCII value which represents the keypad microprocessor version in the system. The format of the string is *x.xx* with a terminating null character
- VN\_CLASS\_ASIC Returns a five-byte string, including null terminator, that contains an ASCII value which represents the version of the FPGA firmware in the system. The format of the string is *x.xx* with a terminating null character.
- VN\_CLASS\_BOOTSTRAP Returns a five-byte string, including null terminator, that contains an ASCII value which represents the version of the Bootstrap Loader firmware in the system. The format of the string is *x.xx* with a terminating null character.

## ITC NVPARM INTERMEC SOFTWARE CONTENT

This IOCTL reads the manufacturing flag bits from the non-volatile data store that dictates certain software parameters. A BOOLEAN DWORD is returned in the buffer pointed to by *lpOutBuffer* that indicates if Intermec Content is enabled in the XIP regions. TRUE indicates that it is enabled. FALSE indicates that it is not enabled.

## ITC\_NVPARM\_ANTENNA\_DIVERSITY

This IOCTL reads the state of the antenna diversity flag. A BOOLEAN DWORD is returned in the buffer pointed to by *lpOutBuffer* that indicates if there is a diversity antenna installed. TRUE indicates that it is installed. FALSE indicates that it is not installed.

## ITC\_NVPARM\_WAN\_RI

This IOCTL reads the state of the WAN ring indicator flag. A BOOLEAN DWORD is returned in the buffer pointed to by *lpOutBuffer* that indicates the polarity of the WAN RI signal. TRUE indicates active high. FALSE indicates active low.

## **ID Field Values (continued)**

## ITC\_NVPARM\_RTC\_RESTORE

This IOCTL reads the state of the real-time clock restore flag. A BOOLEAN DWORD is returned in the buffer pointed to by *lpOutBuffer*. TRUE indicates that the RTC is restored upon a cold boot. FALSE indicates that the RTC is not restored.

## ITC\_NVPARM\_INTERMEC\_DATACOLLECTION\_SW

This IOCTL reads the state of the data collection software enabled flag. A BOOLEAN DWORD is returned in the buffer pointer to by *lpOutBuffer* that indicates the data collection software is to install at boot time. FALSE indicates the data collection software should not install.

## ITC\_NVPARM\_INTERMEC\_DATACOLLECTION\_HW

This IOCTL reads the data collection hardware flags. A BYTE is returned in the buffer pointer to by *lpOutBuffer* that indicates the type of data collection hardware installed. The maximum possible value returned is ITC\_DEVID\_SCANHW\_MAX.

- ITC\_DEVID\_SCANHW\_NONE No scanner hardware is installed.
- ITC\_DEVID\_OEM2D\_IMAGER OEM 2D imager is installed.
- ITC\_DEVID\_INTERMEC2D\_IMAGER Intermec 2D imager is installed.
- ITC DEVID SE900 LASER SE900 laser is installed.
- ITC\_DEVID\_SE900HS\_LASER SE900HS laser is installed.
- ITC\_DEVID\_INTERMEC\_EVIO EVIO linear imager is installed.

The high bit indicates whether the S6 scanning engine is installed. The bit mask for this is ITC\_DEVID\_S6ENGINE\_MASK. A non-zero value indicates that the S6 scanning engine is installed.

## ITC\_NVPARM\_WAN\_INSTALLED

This IOCTL reads the state of the WAN radio installed flag. A BOOLEAN DWORD is returned in the buffer pointed to by *lpOutBuffer*. TRUE indicates that the WAN radio is installed. FALSE indicates that no WAN radio is installed.

## ITC\_NVPARM\_WAN\_FREQUENCY

This IOCTL reads the state of the WAN radio frequency flag. A BOOLEAN DWORD is returned in the buffer pointed to by *lpOutBuffer*. TRUE indicates that the WAN radio frequency is United States. FALSE indicates that the WAN radio frequency is European.

## ITC\_NVPARM\_WAN\_RADIOTYPE

This IOCTL reads the WAN radio ID installed by manufacturing. A BYTE is returned in the buffer pointer to by *lpOutBuffer* which indicates the type of WAN radio hardware installed. The maximum possible value returned is ITC\_DEVID\_WANRADIO\_MAX. The current definitions are:

- ITC DEVID WANRADIO NONE No WAN radio installed.
- ITC\_DEVID\_WANRADIO\_SIERRA\_SB555 CDMA Sierra Wireless radio.
- ITC\_DEVID\_WANRADIO\_XIRCOM\_GEM3503 GSM/GPRS Intel (Xircom) radio.
- ITC\_DEVID\_WANRADIO\_SIEMENS\_MC45 GSM/GPRS Siemens radio.
- ITC\_DEVID\_WANRADIO\_SIEMENS\_MC46 GSM/GPRS Siemens radio.

## ITC\_NVPARM\_80211\_INSTALLED

This IOCTL reads the state of the 802.11b or 802.11b/g radio installed flag. A BOOLEAN DWORD is returned in the buffer pointed to by *lpOutBuffer*. TRUE indicates that the 802.11b or 802.11b/g radio is installed. FALSE indicates that no 802.11b or 802.11b/g radio is installed.

#### ITC NVPARM 80211 RADIOTYPE

This IOCTL reads the 802.11b or 802.11b/g radio ID installed by manufacturing. A BYTE is returned in the buffer pointer to by *lpOutBuffer* that indicates the type of 802.11b or 802.11b/g radio hardware installed. The maximum possible value returned is ITC\_DEVID\_80211RADIO\_MAX. The current definitions are:

- ITC\_DEVID\_80211RADIO\_NONE No 802.11b or 802.11b/g radio installed.
- ITC\_DEVID\_80211RADIO\_INTEL\_2011B Intel 2011B radio installed.

## **ID Field Values (continued)**

## ITC\_NVPARM\_BLUETOOTH\_INSTALLED

This IOCTL reads the state of the Bluetooth radio installed flag. A BOOLEAN DWORD is returned in the buffer pointed to by *lpOutBuffer*. TRUE indicates that the Bluetooth radio is installed. FALSE indicates that no Bluetooth radio is installed.

#### ITC\_NVPARM\_SERIAL2\_INSTALLED

This IOCTL reads the state of the serial 2 (COM2) device installed flag. A BOOLEAN DWORD is returned in the buffer pointed to by *lpOutBuffer*. TRUE indicates that the serial 2 device is installed. FALSE indicates that no serial 2 device is installed.

#### ITC\_NVPARM\_VIBRATE\_INSTALLED

This IOCTL reads the state of the vibrate device installed flag. A BOOLEAN DWORD is returned in the buffer pointed to by *lpOutBuffer*. TRUE indicates that the vibrate device is installed. FALSE indicates that no vibrate device is installed.

## ITC\_NVPARM\_LAN9000\_INSTALLED

This IOCTL reads the state of the Ethernet device installed flag. A BOOLEAN DWORD is returned in the buffer pointed to by *lpOutBuffer*. TRUE indicates that the Ethernet device is installed. FALSE indicates that no Ethernet device is installed.

## ITC\_NVPARM\_SIM\_PROTECT\_HW\_INSTALLED

This IOCTL reads the state of the SIM card protection hardware installed flag. A BOOLEAN DWORD is returned in the buffer pointed to by *lpOutBuffer*. TRUE indicates that the SIM card protection hardware is installed. FALSE indicates that no SIM card protection hardware is installed.

#### ITC NVPARM SIM PROTECT SW INSTALLED

This IOCTL reads the state of the SIM card protection software installed flag. A BOOLEAN DWORD is returned in the buffer pointed to by *lpOutBuffer*. TRUE indicates that the SIM card protection software is installed. FALSE indicates that no SIM card protection software is installed.

#### ITC\_NVPARM\_SIM\_PROTECT\_SW\_INSTALLED

This IOCTL reads the state of the SIM card protection software installed flag. A BOOLEAN DWORD is returned in the buffer pointed to by *lpOutBuffer*. TRUE indicates that the SIM card protection software is installed. FALSE indicates that no SIM card protection software is installed.

# **IOCTL HAL ITC WRITE SYSPARM**

Describes and enables the registry save location.

# Usage

#include "oemioctl.h"

# **Syntax**

BOOL KernelIoControl( IOCTL\_HAL\_ITC\_WRITE\_SYSPARM,LPVOID lpInBuf,DWORD nInBufSize, LPVOID lpOutBuf, DWORD nOutBufSize, LPDWORD lpBytesReturned );

## **Parameters**

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

# **Return Values**

Returns TRUE if function succeeds. Returns FALSE if the function fails. GetLastError() may be used to get the error value. Either ERROR\_INVALID\_PARAMETER or ERROR\_INSUFFICIENT\_BUFFER may be returned when this function is used to get the error.

## **ID Field Values**

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

#### **ID Field Values**

## ITC\_REGISTRY\_SAVE\_ENABLE

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

## ITC DOCK SWITCH

This IOCTL sets a position of the dock switch. The dock switch may be set to either "modem" or "serial" positions. 

### IpOutBuf must point to a buffer that contains a byte value of either DOCK\_MODEM or DOCK\_SERIAL as defined in OEMIOCTL.H; the value specifies the position the switch is to be set. The call appears as follows: 

### // port = DOCK\_MODEM or DOCK\_SERIAL as defined in oemioctl.h

### BOOL SetDockSwitch( BYTE port)

### DWORD cmd = ITC\_DOCK\_SWITCH;

### DWORD cbRet;

### return KernelIoControl(IOCTL\_HAL\_ITC\_WRITE\_SYSPARM, &cmd, sizeof(cmd), &port, sizeof(port), &cbRet)

#### ITC WAKEUP MASK

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

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

## ITC\_AMBIENT\_KEYBOARD (does not apply to the 730 Computer)

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

## ITC\_AMBIENT\_FRONTLIGHT (does not apply to the 730 Computer)

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

# **IOCTL HAL GET DEVICEID**

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

# Usage

#include "pkfuncs.h" #include "deviceid.h"

# **Syntax**

BOOL KernelioControl( IOCTL\_HAL\_GET\_DEVICEID, LPVOID lpInBuf, DWORD nInBufSize, LPVOID lpOutBuf, DWORD nOutBufSize, LPDWORD lpBytesReturned );

## **Parameters**

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

# **Return Values**

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

# IOCTL\_HAL\_GET\_OAL\_VERINFO

Returns the HAL version information of the Pocket PC image.

# Usage

#include "oemioctl.h"

# **Syntax**

BOOL KernelIoControl( IOCTL\_HAL\_GET\_OAL\_VERINFO,LPVOID lpInBuf,DWORD nInBufSize,LPVOID lpOutBuf,DWORD nOutBufSize,LPDWORD lpBytesReturned );

## **Parameters**

lpInBuf	Should be set to NULL.		
lpInBufSize	Should be set to zero.		
lpOutBuf	*	NINFO structure as defined by ds should have these values: sizeof (tagOemVerInfo);  "ITC\0"  'N' ""  SeaRay  Current build version number "Intel\0" "PXA255\0"; ""  Build time Build date	
nOutBufSize	The size of VERSIONINFO in bytes.		
lpBytesReturned	Returns sizeof(PVERSIONINFO).		

# **Return Values**

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

## IOCTL\_HAL\_GET\_BOOTLOADER\_VERINFO

Returns the HAL version information of the Pocket PC image.

#### Usage

#include "oemioctl.h"

#### **Syntax**

BOOL KernelioControl( IOCTL\_HAL\_GET\_OAL\_VERINFO,LPVOID lpInBuf, DWORD nInBufSize,LPVOID lpOutBuf,DWORD nOutBufSize,LPDWORD lpBytesReturned );

#### **Parameters**

lpInBuf	Should be set to NULL.	
nInBufSize	Should be set to zero.	
lpOutBuf	Must point to a VERSIONINFO structure as defined by OEMIOCTL.H. The fields should have these values:  • cboemverinfo Sizeof (tagOemVerInfo);  • verinfover 1  • sig; "ITC\0"  • id; "B'  • tgtcustomer ""	
	• tgtplat • tgtplatversion	SeaRay Current build version number of the bootstrap loader
	<ul><li>tgtcputype[8];</li><li>tgtcpu</li><li>tgtcoreversion</li><li>date</li></ul>	"Intel\0"; "PXA255\0" "" Build time
• time Build date		Build date
nOutBufSize	The size of VERSIONINFO in bytes.	
lpBytesReturned	The number of bytes returned to lpOutBuf.	

#### **Return Values**

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

### **IOCTL HAL WARMBOOT**

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

#### Usage

#include "oemioctl.h"

#### **Syntax**

BOOL KernelIoControl( IOCTL\_HAL\_WARMBOOT,LPVOID lpInBuf,DWORD nInBufSize,LPVOID lpOutBuf,DWORD nOutBufSize,LPDWORD lpBytesReturned );

#### **Parameters**

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

#### **Return Values**

None.

### IOCTL\_HAL\_COLDBOOT

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

#### Usage

#include "oemioctl.h"

#### **Syntax**

BOOL KernelIoControl( IOCTL\_HAL\_COLDBOOT,LPVOID lpInBuf,DWORD nInBufSize,LPVOID lpOutBuf,DWORD nOutBufSize,LPDWORD lpBytesReturned );

#### **Parameters**

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

#### **Return Values**

None.

### IOCTL\_HAL\_GET\_RESET\_INFO

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

#### Usage

#include "oemioctl.h"

#### **Syntax**

BOOL KernelIoControl( IOCTL\_HAL\_GET\_RESET\_INFO,LPVOID lpInBuf,DWORD nInBufSize,LPVOID lpOutBuf,DWORD nOutBufSize,LPDWORD lpBytesReturned );

#### **Parameters**

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

#### **Return Values**

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

### Sample

```
typedef struct {
  DWORD ResetReason;
                                            // most recent reset type
  DWORD ObjectStoreState;
                                             // state of object store
} HAL_RESET_INFO, * PHAL_RESET_INFO;
// Reset reason types
#define HAL_RESET_TYPE_UNKNOWN
#define HAL_RESET_REASON_HARDWARE
                                         1 // cold
#define HAL_RESET_REASON_SOFTWARE
                                         2 // suspend
#define HAL_RESET_REASON_WATCHDOG
#define HAL_RESET_BATT_FAULT
                                        8 // power fail
#define HAL_RESET_VDD_FAULT
                                         16 // warm boot
// Object store state flags
#define HAL_OBJECT_STORE_STATE_UNKNOWN
                                          0
#define HAL_OBJECT_STORE_STATE_CLEAR
```

## IOCTL\_HAL\_GET\_BOOT\_DEVICE

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

#### Usage

#include "oemioctl.h"

#### **Syntax**

BOOL KernelIoControl( IOCTL\_HAL\_GET\_BOOT\_DEVICE,LPVOID lpInBuf,DWORD nInBufSize,LPVOID lpOutBuf,DWORD nOutBufSize,LPDWORD lpBytesReturned );

#### **Parameters**

lpInBuf	Should be set to NULL.	
lpInBufSize	Should be set to zero.	
lpOutBuf	Must point to a buffer large enough to hold a DWORD (4 bytes) that contains the boot device. The following boot devices are supported:  #define HAL_BOOT_DEVICE_UNKNOWN 0  #define HAL_BOOT_DEVICE_ROM_XIP 1  #define HAL_BOOT_DEVICE_ROM 2  #define HAL_BOOT_DEVICE_PCMCIA_ATA 3  #define HAL_BOOT_DEVICE_PCMCIA_LINEAR 4  #define HAL_BOOT_DEVICE_IDE_ATA 5	
nOutBufSize	The size of <i>lpOutBuf</i> in bytes (4).	
lpBytesReturned	The number of bytes returned by the function.	

#### **Return Values**

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

## IOCTL\_HAL\_REBOOT

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

### Usage

#include "oemioctl.h"

#### **Syntax**

BOOL KernelIoControl( IOCTL\_HAL\_REBOOT,LPVOID lpInBuf,DWORD nInBufSize,LPVOID lpOutBuf,DWORD nOutBufSize,LPDWORD lpBytesReturned );

#### **Parameters**

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

### **Return Values**

None.

### **IOCTL PROCESSOR INFORMATION**

Returns processor information.

#### Usage

#include "pkfuncs.h"

#### **Syntax**

BOOL KernelIoControl( IOCTL\_PROCESSOR\_INFORMATION,LPVOID lpInBuf,DWORD nInBufSize,LPVOID lpOutBuf,DWORD nOutBufSize,LPDWORD lpBytesReturned );

#### **Parameters**

```
lpInBuf
             Should be set to NULL.
nInBufSize
             Should be set to zero.
lpOutBuf
             Should be a pointer to the PROCESSOR_INFO structure. The
             PROCESSOR_INFO structure stores information that describes
             the CPU more descriptively.
typedef __PROCESSOR_INFO {
WORD wVersion;
                             // Set to value 1
WCHAR szProcessorCore[40]; // "ARM\0"
                             // 4
WORD wCoreRevision;
WCHAR szProcessorName[40]; // "PXA255\0"
WORD wProcessorRevision; // 0
WCAHR szCatalogNumber[100]; // 0
// 0
DWORD dwInstructionSet;
DWORD dwClockSpeed;
                             // 400
nOutBufSize
             Should be set to sizeof(PROCESSOR_INFO) in bytes.
lpBytesReturned Returns sizeof(PROCESSOR_INFO);
```

#### **Return Values**

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

## IOCTL\_GET\_CPU\_ID

Returns Xscale processor ID.

#### Usage

#include "oemioctl.h"

#### **Syntax**

BOOL KernelIoControl( IOCTL\_GET\_CPU\_ID,LPVOID lpInBuf, DWORD nInBufSize,LPVOID lpOutBuf,DWORD nOutBufSize,LPDWORD lpBytesReturned );

#### **Parameters**

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

#### **Return Values**

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

## **Network Selection APIs**

The Network Selection APIs change the network adapter configuration programmatically. Both drivers support the same IOCTL function numbers for loading and unloading the drivers.

Loading and unloading of the 802.11b or 802.11b/g driver is performed by the FWL1: device in the system by performing DeviceIOControl() calls to the driver.

Loading and unloading of the driver for the built-in Ethernet adapter is performed by the SYI1: device in the system by performing DeviceIOControl() calls to the driver.

- For loading an NDIS driver associated with an adapter, the IOCTL is IOCTL\_LOAD\_NDIS\_MINIPORT.
- For unloading NDIS drivers associated with an adapter the IOCTL is IOCTL\_UNLOAD\_NDIS\_MINIPORT.

#### **Example**

```
#include <winioctl.h>
#include "sysio.h"
void DoLoad(int nDevice) {
  LPTSTR devs[] = { _T("SYI1:"), _T("FWL1:") };
  HANDLE hLoaderDev;
  DWORD bytesReturned;
  hLoaderDev = CreateFile(devs[nDevice], GENERIC READ GENERIC WRITE, 0,
  NULL, OPEN EXISTING, 0, NULL);
  if (hLoaderDev != INVALID_HANDLE_VALUE) {
     if (!DeviceIoControl( hLoaderDev, IOCTL_LOAD_NDIS_MINIPORT, NULL, -1, NULL, 0,
     &bytesReturned, NULL)){
        MessageBox(NULL, TEXT("SYSIO IoControl Failed"), TEXT("Network
        loader"),MB ICONHAND);
        if (hLoaderDev!=INVALID_HANDLE_VALUE) CloseHandle(hLoaderDev);
        hLoaderDev = INVALID_HANDLE_VALUE; // bad handle
     }else {
        CloseHandle(hLoaderDev);
     }
  }
}
void DoUnload(int nDevice) {
  LPTSTR devs[] = { _T("SYI1:"), _T("FWL1:") };
  HANDLE hLoaderDev;
  DWORD bytesReturned;
  hLoaderDev = CreateFile(devs[nDevice], GENERIC_READ|GENERIC_WRITE, 0,
  NULL, OPEN EXISTING, 0, NULL);
  if (hLoaderDev != INVALID_HANDLE_VALUE) {
     if (!DeviceIoControl( hLoaderDev, IOCTL_UNLOAD_NDIS_MINIPORT, NULL, -1, NULL, 0,
     &bytesReturned, NULL)){
        MessageBox(NULL, TEXT("SYSIO IoControl Failed"), TEXT("Network
        loader"),MB ICONHAND);
        if (hLoaderDev!=INVALID_HANDLE_VALUE) CloseHandle(hLoaderDev);
        hLoaderDev = INVALID_HANDLE_VALUE; // bad handle
     }else {
        CloseHandle(hLoaderDev);
  }
}
```

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

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

NETWLAN.DLL PRISMNDS.DLL	This file is the 802.11b or 802.11b/g driver. It is present in all 700 CE loads that use the 802.11b or 802.11b/g network interface card.	
80211API.DLL	This file is an Intermec authored file that provides the programmer with a set of API calls to configure or monitor status of the 802.11b or 802.11b/g network.	
MOD80211.DLL	The CORE module for the 802.11 NIC. It provides the 802.11b or 802.11b/g status information displayed when the CORE application is running.	
80211CONF.EXE	This is the "Control Panel" for configuring the 802.11b or 802.11b/g network parameters. Note that it is an EXE file and is actually called by CPL802.CPL (see below). It is also called by the CORE application when the "Configuration" button is pressed.	
CPL802.CPL	A control panel application that does nothing but call 80211CONF.EXE.	
80211SCAN.EXE	Internally manages the Scan List activity.	
802PM.DLL	This handles profile management for radio configurable values.	
URODDSVC.EXE	This handles radio configuration and security authentication based on a selected profile.	

The Profile Manager supports up to four radio configuration profiles. These profiles are the same as those set by the Wireless Network control panel applet that runs on the Windows CE unit. You can configure different 802.11b or 802.11b/g profiles and switch between them using the 802.11 API. See the ConfigureProfile() function on page 283 for more information.

### **Basic Connect/Disconnect Functions**

Below are functions available for the 700 Color Computer when enabled with the 802.11b or 802.11b/g radio module.

#### RadioConnect()

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

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

### RadioDisconnect()

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

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

#### RadioDisassociate()

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

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

# **Query Information Functions**

### **GetAssociationStatus()**

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

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

### **GetAuthenticationMode()**

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

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

## GetBSSID()

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

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

### **GetDiversity()**

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

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

#### GetLinkSpeed()

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

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

### GetMac()

Call this function to get the MAC address of the 802.11b or 802.11b/g radio.

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



**Note:** Be sure to call RadioConnect() *before* calling this function for this function to work properly.

#### GetNetworkMode()

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

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

### GetNetworkType()

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

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

### GetSSID()

Call this function to get the desired SSID of the 802.11b or 802.11b/g radio.

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



**Note:** Call RadioConnect() *before* this function for this function to work properly.

### GetPowerMode()

Call this function to get the current power savings mode of the radio.

Crentary	HITME Cot Downwood ( HI ONG C )	
Syntax	UINT GetPowerMode( ULONG & );	
Parameters	NDIS_RADIO_POWER_MODE_CAM	Continuous Access Mode (ie: always on).
	NDIS_RADIO_POWER_MODE_PSP	Power Saving Mode.
	NDIS_RADIO_POWER_UNKNOWN	Unknown power mode.
	NDIS_RADIO_POWER_AUTO	Auto. (Available for 730 Mobile Computers)
	NDIS_RADIO_POWER_MODE_FAST_PSP	Fast PSP, good savings, fast
Return Values	ERROR_SUCCESS when successful, ERR_QUERY_FAILED when the query failed, or ERR_CONNECT_FAILED if a connection with the radio failed.	
Remarks	If ERROR_SUCCESS is returned, your ULONG reference is populated with one of the parameters listed above.	
Definitions	<pre>#ifdef DYNAMIC_LOADING typedef UINT (*PFN_GetPowerMode)(ULONG &amp;); #else UINT GetPowerMode(ULONG &amp;); #endif</pre>	



Note: Do not use Automatic Switching mode at this time.

### GetRSSI()

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

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

## GetTXPower()

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

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

## **GetWepStatus()**

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

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

### **GetRadiolpAddress()**

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

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

### **GetCCXStatus()**

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

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

### **Set Information Functions**

#### AddWep()

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



Note: Add the default transmit key last.

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

### EnableWep()

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

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

## **EncryptionStatus()**

Call this function to set the desired encryption status.

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

### SetAuthenticationMode()

Call this function to set the desired authentication mode.

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

## SetChannel()

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

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

### SetNetworkMode()

Call this function to set the desired Network Mode.

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

### SetPowerMode()

Call this function to set the desired power mode.

Syntax	UINT SetPowerMode( ULONG mode );	
Parameters	NDIS_RADIO_POWER_MODE_CAM	Continuous Access Mode (ie: always on).
	NDIS_RADIO_POWER_MODE_PSP	Power Saving Mode.
	NDIS_RADIO_POWER_UNKNOWN	Unknown power mode.
	NDIS_RADIO_POWER_AUTO	Auto. (Available for 730 Computers)
	NDIS_RADIO_POWER_MODE_FAST_PSP	Fast PSP, good savings, fast
Return Values	ERROR_SUCCESS when successful, ERR_QUERY_FAILED when the query failed, or ERR_CONNECT_FAILED if a connection with the radio failed.	
Remarks	None.	
Definitions	<pre>#ifdef DYNAMIC_LOADING typedef UINT (*PFN_SetPowerMode) #else UINT SetPowerMode(ULONG mode); #endif</pre>	(ULONG mode);

### SetSSID()

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

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

### SetCCXStatus()

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

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

### SetMixedCellMode()

Call this function to set the desired mixed cell mode.

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

### RemoveWep()

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

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

# **Helper Functions**

### **ConfigureProfile()**

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

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

### EnableZeroConfig()

This enables or disables the Wireless Zero Configuration Wizard from Microsoft. After calling this function, a warm-boot is required for the change to take effect.



**Note:** Enabling this function effectively disables all the SET commands in this API.

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

### isZeroConfigEnabled()

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

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

### isOrinoco()

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

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

### isSupplicantRunning()

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

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

#### StartScanList()

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

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

### StartSupplicant()

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

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

### StopSupplicant()

Call this function to stop the supplicant service.

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

### isDHCPEnabled()

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

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

## RenewDHCP()

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

Syntax	UINT RenewDHCP( );
Parameters	None.
Return Values	ERROR_SUCCESS when successful.
Remarks	You should not have to call this function on Microsoft PocketPC 2003 or Microsoft Windows CE 4.2 .NET and later devices.
Definitions	<pre>#ifdef DYNAMIC_LOADING typedef UINT (*PFN_RenewDHCP)(); #else UINT RenewDHCP(); #endif</pre>

#### **GetCurrentDriverName()**

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

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

### ResetRadioToSystemSave()

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

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

### EnableSuppLogging()

Call this function to set the desired supplicant logging mode.

Syntax	UINT EnableSuppLogging( ULONG );	
Parameters	NDIS_SUPP_LOGGING_ON Supplicant I	ogging Enabled
	NDIS_SUPP_LOGGING_OFF Supplicant I	Logging Disabled
Return Values	ERROR_SUCCESS when successful.	
Remarks	None.	
Definitions	<pre>#ifdef DYNAMIC_LOADING typedef UINT (*PFN_EnableSuppLogging)(ULONG); #else UINT EnableSuppLogging(ULONG); #endif</pre>	

### SwitchPacketDriver()

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

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

## **Deprecated Functions**

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

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

## **Notifications**

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

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

Regarding an applications interface to NLED.DLL, LEDs must be available for use by applications. This is possible via two functions exported by the COREDLL.DLL file. To use the LED functions, declare these as extern "C" as follows:

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

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

- Notification LED 0
- Radio On LED 1 (does not apply to the 730 Computer)
- Alpha Lock LED 2
- Scanner LED 3
- Low Battery 4
- Vibrator 5 (does not apply to the 730 Computer)

### **NLEDGetDeviceInfo**

### Usage

#include "nled.h"

### **Syntax**

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

#### **Parameters**

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

### **NLEDSetDevice**

## Usage

#include "nled.h"

### **Syntax**

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

#### **Parameters**

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

## **Reboot Functions**

There are several methods, via Kernel I/O Control functions, that an application program can use to force the 700 Color Computer to reboot.

### **IOCTL\_HAL\_REBOOT**

IOCTL\_HAL\_REBOOT performs a warm-boot. See page 263.

### **IOCTL HAL COLDBOOT**

Invoking the KernelIOControl function with IOCTL\_HAL\_COLDBOOT forces a cold reboot. This resets the 700 Color Computer and reloads Windows CE as if a power-up was performed. The contents of the Windows CE RAM-based object store are discarded. See page 260.

### IOCTL\_HAL\_WARMBOOT

This function is supported on 700 Color Computers. It performs a warm boot of the system, preserving the object store. See page 260.

# Remapping the Keypad



**Note**: Use caution when remapping the keypad. Improper remapping may render the keypad unusable. Data within the 700 Color Computer could also be lost, should any problems occur.

Applications have the ability to remap keys on the 700 Color Numeric Keypad and 700 Color Alphanumeric Keypad. This will allow applications to enable keys that would otherwise not be available, such as the [F1] function key. Also, to disable keys that should not be available, such as the alpha key because no alpha entry is required. Care should be exercised when attempting to remap the keypad because improper remapping may cause the keypad to become unusable. This can be corrected by cold booting the device which will cause the default keymap to be loaded again.

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

There are three "planes" supported for the 700 Color Numeric Keypad and Alphanumeric Keypad. Keys that are to be used in more than one shift plane must be described in each plane.

#### **Unshifted Plane**

The unshifted plane contains values from the keypad when not pressed with other keys, such as the following:

Press the Keys		
Numeric Keypad	Alphanumeric Keypad	To Enter This
1 Caps Send	M1	1
5 <sub>A3</sub>	<b>(75</b> )	5
<i>9</i> wxy2	<b>Y9</b>	9

#### **Gold Plane**

The gold plane contains values from the keypad when a key is simultaneously pressed with the [Gold] key on the numeric keypad or the [Gold/White] key on the alphanumeric keypad, such as the following:

Press the Keys		
Numeric Keypad	Alphanumeric Keypad	To Enter This
[Gold] © (Taps)	[Gold/White] (Esc Sond)	Send
[Gold] © 5,KL	[Gold/White] 🕏 😘	A3
[Gold] © 9WXYZ	[Gold/White] 🔊 🤌	PgDn

## Alpha (Blue) Plane

The alpha plane contains values from the keypad when the keypad has been placed in alpha mode by pressing the blue alpha key, such as the following:

Press the Keys			
Numeric Keypad	Alphanumeric Keypad	To Enter This	
[Alpha] (Alpha) (1 Caps Send)	[Alpha] (Cap) Lock	Caps	
[Alpha] (Alpha) (5,JKL)	[Alpha] Poul	j	
[Alpha] (Alpha) (9WXYZ)	[Alpha] w7	W	

## **Key Values**

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

#### **Numeric Keypad**

For the 700 Color Numeric Keypad, the following registry keys contain the plane mappings:

- The unshifted plane mapping can be found in the registry at:

  HKEY\_LOCAL\_MACHINE\HARDWARE\DEVICEMAP\KEYBD\Vkey
- The gold plane mapping can be found in the registry at:
   HKEY\_LOCAL\_MACHINE\HARDWARE\DEVICEMAP\KEYBD\VkeyGold
- The alpha plane mapping can be found in the registry at:

  HKEY\_LOCAL\_MACHINE\HARDWARE\DEVICEMAP\KEYBD\VkeyAlpha

#### Alphanumeric Keypad

For the 700 Color Alphanumeric Keypad, the following registry keys contain the plane mappings:

- The unshifted plane mapping can be found in the registry at:
   hkey\_local\_machine\hardware\devicemap\keybd\alpha\vkey
- The gold plane mapping can be found in the registry at:
   HKEY\_LOCAL\_MACHINE\HARDWARE\DEVICEMAP\KEYBD\ALPHA\VkeyGold
- The alpha plane mapping can be found in the registry at:

  HKEY\_LOCAL\_MACHINE\HARDWARE\DEVICEMAP\KEYBD\ALPHA\VkeyAlpha

### **How Key Values Are Stored in Registry**

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

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

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

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



Note: Do not remap scan codes 0x01, 0x41, 0x42, 0x43, 0x44. Remapping these scan codes could render your 700 Color Computer unusable until a cold-boot is performed.

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

### **Change Notification**

Just changing the registry keys will not immediately change the key mappings. To notify the keypad driver that the registry has been updated, signal the "ITC\_KEYBOARD\_CHANGE" named event using the CreateEvent() API.

### Advanced Keypad Remapping

It is also possible to map multiple key presses to one button and to map named system events to a button. The multiple key press option could be useful to cut down on the number of keys needed to press in a given situation or to remap which key behaves like the action key. Mapping events to a button could be useful to change which buttons will fire the scanner, control volume, and allow for suspending and resuming the device. If you need help performing one of these advanced topics please contact Intermec Technical Support.

# **Scan Codes**

At the lowest driver level, the 700 Color Numeric Keypad and the 700 Color Alphanumeric Keypad identifies keys as scan codes. These scan codes are sent via the keypad microcontroller, and cannot be changed without modifying the keypad firmware.

# **Numeric Keypad**

The following scan codes pertain to the 700 Color Numeric keypad:

# **Numeric Keypad Scan Codes and Meanings**

Press this Key	Meaning	ScanCode
	Reserved	0x00
1/0	I/O button	0x01
	Scanner Handle Trigger	0x02
	Scanner Left	0x03
	Scanner Right	0x04
4 GHI	4/GHI/A2	0x06
	None	0x07
(2)	Left arrow/Back Tab	0x08
	None	0x09
Bissp	BkSp// (forward slash)	0x0A
	[Gold] key	0x0B
	None	0x0C
Esc	Esc/- (minus sign)	0x0D
(A)	Down arrow/Volume decrease	0x0E
1 Caps Send	1/Caps/Send	0x0F
(ZPORS)	7/PQRS/PgUp	0x10
Alpha	[Alpha] key	0x11
	None	0x12
٥٠	Up arrow/Volume increase	0x13
40	Right arrow/Tab	0x14
2 ABC End	2/ABC/End	0x15
8 <sup>nuv</sup>	8/TUV/* (asterisk)	0x16
(O <sub>Win</sub> )	0/Win	0x17
(5 <sub>A3</sub> )	5/JKL/A3	0x18
	None	0x19
Action	Action/+ (plus symbol)	0x1A
3 DEF	3/DEF/backlight	0x1B
<u>Э</u> мхүг	9/WXYZ/PgDn	0x1C

# Numeric Keypad Scan Codes and Meanings (continued)

Press this Key	Meaning	ScanCode
Enter @	Enter/@ (at symbol)	0x1D
6 MNO	6/MNO/A4	0x1E
	None	0x1F-0x40
	Charge Detect	0x41
<b>√</b> ○	LCD frontlight	0x42
	Ambient light	0x42
	Threshold crossed	0x42
	Headset detected	0x43
	Keypad Backlight	0x44
	Ambient Light	0x44
	Threshold Crossed	0x44

# **Alphanumeric Keypad**

The following scan codes pertain to the 700 Color Alphanumeric keypad:

# **Alphanumeric Keypad Scan Codes and Meanings**

Press this Key	Meaning	ScanCode
	Reserved	0x00
	I/O button	0x01
	Scanner Handle Trigger	0x02
	Scanner Left	0x03
	Scanner Right	0x04
(AAT)	A/A1 key	0x05
(BA2)	B/A2 key	0x06
Esc Send	Escape/Send	0x07
2	Left arrow/Back Tab	0x08
<b>A</b>	Up arrow/Volume increase	0x09
<b>*</b>	Down arrow/Volume decrease	0x0A
<b>3</b>	Right arrow/Tab	0x0B
Action End	Action/End	0x0C
<b>E</b> Win	E/Win	0x0D
F=	F/= (equal sign)	0x0E
<b>€</b>	G/* (asterisk)	0x0F
(CA3)	C/A3	0x10
HI	H// (forward slash)	0x11
DA4	D/A4	0x12

# Alphanumeric Keypad Scan Codes and Meanings (continued)

Press this Key	Meaning	ScanCode
AgUp	J/PgUp	0x13
Ke	K/@ (as symbol)	0x14
(L-)	L/– (minus sign)	0x15
M1	M/1	0x16
<u>N2</u>	N/2	0x17
(*)	I/backlight	0x18
P <sub>RyDn</sub>	P/PgDn	0x19
(Q)	Q/, (comma)	0x1A
R+	R/+ (plus sign)	0x1B
<b>S4</b> )	S/4	0x1C
75	T/5	0x1D
03	O/3	0x1E
Cap	Caps/Lock	0x1F
BKSP	BkSp	0x20
<b>v</b> •	V/. (period)	0x21
<b>W7</b>	W/7	0x22
<b>(88)</b>	X/8	0x23
<u>U6</u>	U/6	0x24
	Gold/White	0x25
	NumLock	0x26
Space	Space	0x27
20	Z/0	0x28
(Enter)	Enter	0x29
<b>(Y9</b> )	Y/9	0x2A
	None	0x2B-0x40
	Charge Detect	0x41
<b>√</b> ◎	LCD frontlight	0x42
	Ambient light	0x42
	Threshold crossed	0x42
	Headset detected	0x43
	Keypad Backlight	0x44
	Ambient Light	0x44
	Threshold Crossed	0x44

# Sample View of Registry Keys

The following is a sample view of the current default key mapping for the 700 Color Numeric Keypad. See the registry on your device for the latest key mappings.

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

# **Configurable Settings**

This appendix contains information about the Data Collection, Intermec Settings, SNMP, Unit Information, Utilities, and Wireless Network control panel applets that may be on the 700 Series Color Mobile Computer.



Note: "700 Color" pertains to 740, 741, 750, 751, 760, and 761 Computers unless otherwise noted.

SNMP, Intermec Settings, and Data Collection settings that can appear under **Settings** are dependent on what hardware configuration is done for each 700 Color Computer at the time of shipment. These settings currently only appear if a scanner or an imager option is present.

Likewise, other control panel applets that are specifically related to the 802.11b or 802.11b/g radio module will appear when a 802.11b or 802.11b/g radio module is installed in a 700 Color Computer. Control panel applets that are specific for Wireless Printing, CDMA/1xRTT, and GSM/GPRS radio modules will only appear when each respective hardware configuration is done on the 700 Color Computer. See Chapter 4, "Network Support," for more information about the radio modules or the wireless printing.

Information about using reader commands and configuration bar codes to configure some of your settings is also in this appendix.



**Note**: Information about the settings you can configure with the Intermec Settings control panel applet is described in the *Intermec Computer Command Reference Manual* (P/N: 073529). The online manual is available from the Intermec web site at www.intermec.com.

# **Configuration Parameters**

A configuration parameter changes the way the 700 Color Computer operates, such as configuring a parameter to have the 700 Color Computer emit a very loud beep in a noisy environment. Use any of the following methods to execute configuration parameters:

- Change Data Collection and SNMP parameters via control panel applets later in this appendix.
- Send parameters from an SNMP management station. See "SNMP Configuration on the 700 Color Computer" starting on page 195.
- Scan EasySet bar codes. You can use the EasySet bar code creation software from Intermec Technologies Corporation to print configuration labels. Scan the labels to change the scanner configuration and data transfer settings.

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

# **Changing a Parameter Setting**

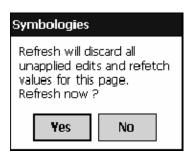
Menus of available parameters for each group are listed. Use the scroll bars to go through the list. Expand each menu (+) to view its parameter settings. Tap a parameter to select, or expand a parameter (+) to view its subparameters.

Note that each parameter or subparameter is shown with its default setting or current setting in (< >) brackets. Tap a parameter or subparameter to select that parameter, then do any of the following to change its setting: Tap **Apply** to apply any changes. *Note that these illustrations are from a Symbologies parameter.* 

- Typing a new value in an entry field.
- Choosing a new value from the drop-down list.
- Selecting a different option. The selected option contains a bullet.
- Tap **Defaults**, then **Apply** to restore factory-default settings. Tap **Yes** when you are prompted to verify this action.



• Tap Refresh to discard changes and start again. Tap Yes when you are prompted to verify this action.



# **About Configuration Parameters**

You can find this information about each configuration parameter:

- Name and Purpose:
   Describes the parameter and its function.
- Action:
   Describes what to do with a parameter once that parameter is selected.
- SNMP OID: Lists the SNMP OID for the parameter.
- Syntax or Options:
  Syntax lists the two-character code for the parameter, if the parameter is configurable by scanning a bar code or by sending parameters through a network. Both Syntax and Options list acceptable values for the parameter.

# **Data Collection Control Panel Applet**



**Note:** This applet is not available in units with PSM Build 3.00 or newer. To determine your PSM Build version, tap **Start** > **Programs** > **File Explorer** > the **PSMinfo** text file.



If your unit has PSM Build 3.00 or newer, then you may have the Intermec Settings control panel applet in place of the Data Collection applet. Information about the settings you can configure with the Intermec Settings applet is described in the Intermec Computer Command Reference Manual. The online manual is available from the Intermec web site at www.intermec.com.

See the Data Collection Resource Kit in the Intermec Developer Library (IDL) for information about data collection functions. Contact your Intermec representative for more information.



To access the settings from the 700 Color Computer, tap **Start** > **Settings** > the **System** tab > **Data Collection** to access its control panel applet.

Use the left and right arrows to scroll through the tabs along the bottom of the control panel applet, then tap a tab to access its menus. These tabs represent the following groups of settings or parameters:

- Symbologies (starting on page 303)
- Symbology Options (starting on page 324)
- Beeper/LED (starting on page 332)
- Imager (starting on page 338)
- Virtual Wedge (starting on page 343)

# **Symbologies**

You can change bar code symbology parameter settings in your 700 Color Computer via the **Data Collection** control panel applet. The following parameters are for bar code symbologies. Additional information about the more common bar code symbologies are in Appendix B, "Bar Codes." Note that these parameters are listed in the order of their appearance within this tab.

Most of these symbologies apply to both the imager and the laser scanner tools. However, when using an imager, the Macro PDF (page 314), Micro PDF417 (page 316), Matrix 2 of 5 (page 318), Telepen (page 319), and Code 11 (page 320) symbologies are not supported. Likewise, when using a laser scanner, the QR Code (page 321), Data Matrix (page 322), and MaxiCode (page 323) symbologies are not supported.



**Note:** The 730 Computer uses the EV10 APS linear imager which supports 1D symbologies.

The following table shows which bar code symbologies are supported by an imager, a laser scanner, or the EV10 APS Linear Imager

Bar Code Symbology	lmager	Laser Scanner	EV10 APS Linear Imager
Code 39	X	X	X
Interleaved 2 of 5	X	X	X
Standard 2 of 5	X	X	X
Matrix 2 of 5		X	X
Code 128	X	X	X
Code 93	X	X	X
Codabar	X	X	X
MSI		X	X
Plessey		X	X
UPC	X	X	X
EAN/EAN 128	X	X	X
Code 11		X	X
PDF417	X	X	X
Micro PDF417		X	X
Telepen		X	X
Data Matrix	X		
QR Code	X		
MaxiCode	X		

#### Code 39

Code 39 is a discrete, self-checking, variable length symbology. The character set is uppercase A–Z, 0–9, dollar sign (\$), period (.), slash (/), percent (%), space (), plus (+), and minus (-).

#### Action

Tap (+) to expand the Code 39 parameter, select the setting to be changed, then tap an option to change this setting or select an option from the drop-down list.

#### **SNMP OID**

1.3.6.1.4.1.1963.15.3.3.1.1.3.1

## **Options**

Decoding	0	Not active Active (default)
Format	0 1	Standard 43 characters (default) Full ASCII
Start/Stop	0 1	Not transmitted (default) Transmitted
Start/Stop characters (Not supported when us- ing an imager)	0 1 2	\$ (dollar sign) only * (asterisk) only (default) \$ and * (dollar sign and asterisk)
Check digit	0 1 2 3 4 5 6	Not used (default) Mod 43 transmitted Mod 43 not transmitted French CIP transmitted French CIP not transmitted Italian CPI transmitted Italian CPI not transmitted
Bar code length	0 1	Any length (default) Minimum length
Minimum length	001-254	Minimum length 1–254 (default is 6)



Note: If Bar code length = "1" then Minimum length is entered.



#### Standard 2 of 5

Standard 2 of 5 is a discrete and self-checking symbology that uses the bars to encode information and the spaces to separate the individual bars.

#### **Action**

Tap (+) to expand the **Standard 2 of 5** parameter, select the setting to be changed, then tap an option to change this setting or select an option from the drop-down list.

#### **SNMP OID**

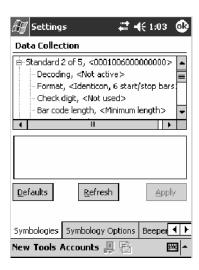
1.3.6.1.4.1.1963.15.3.3.1.1.4.1

# **Options**

•		
Decoding	0 1	Not active (default) Active
Format	0 1	Identicon, 6 start/stop bars (default) Computer Identics, 4 start/stop bars
Check digit	0 1 2	Not used (default) Mod 10 transmitted Mod 10 not transmitted
Bar code length	0 1 2	Any length Minimum length (default) Fixed lengths
Minimum length	001-254	Minimum length 1–254 (default is 6)
Fixed length 1	000-254	Fixed bar code length 0-254 (default is 0)
Fixed length 2	000-254	Fixed bar code length 0-254 (default is 0)
Fixed length 3	000–254	Fixed bar code length 0-254 (default is 0)



Note: If Bar code length = "1" then Minimum length is entered. If Bar code length = "2" then Fixed length 1, Fixed length 2, or Fixed length 3 is entered.



#### Codabar

Codabar is a self-checking, discrete symbology.

#### Action

Tap (+) to expand the **Codabar** parameter, select a setting to be changed, then select an option from the drop-down list to change this setting.

#### **SNMP OID**

1.3.6.1.4.1.1963.15.3.3.1.1.5.1

## **Options**

-		
Decoding	0	Not active (default) Active
Start/Stop	0 1 2 3 4	Not transmitted (default) abcd transmitted ABCD transmitted abcd/tn*e transmitted DC1–DC4 transmitted
CLSI library system (Not supported when using an imager)	0	Not active (default) Active
Check digit	0 1 2	Not used (default) Transmitted Not transmitted
Bar code length	0 1 2	Any length Minimum length (default) Fixed lengths
Minimum length	003–254	Minimum length 3–254 (default is 6)
Fixed length 1	000–254	Fixed bar code length 0-254 (default is 0)
Fixed length 2	000–254	Fixed bar code length 0-254 (default is 0)
Fixed length 3	000-254	Fixed bar code length 0-254 (default is 0)



Note: If Bar code length = "1" then Minimum length is entered. If Bar code length = "2" then Fixed length 1, Fixed length 2, or Fixed length 3 is entered.



## **UPC/EAN**

UPC/EAN are fixed-length, numeric, continuous symbologies that use four element widths.

#### **Action**

Tap (+) to expand the UPC/EAN parameter, select the setting to be changed, then select an option to change this setting.

#### **SNMP OID**

1.3.6.1.4.1.1963.15.3.3.1.1.6.1

UPC A	0 1	Not Active Active (default)
UPC E	0 1	Not Active Active (default)
EAN 8	0 1	Not Active Active (default)
EAN 13	0 1	Not Active Active (default)
Add-on digits	0 1	Not required (default) Required
Add-on 2 digits	0 1	Not active (default) Active
Add-on 5 digits (Not supported when using an imager)	0	Not active (default) Active
UPC A check digit	0 1	Not transmitted Transmitted (default)
UPC E check digit	0 1	Not transmitted Transmitted (default)
EAN 8 check digit	0 1	Not transmitted Transmitted (default)
EAN 13 check digit	0 1	Not transmitted Transmitted (default)
UPC A number system	0 1	Not transmitted Transmitted (default)
UPC E number system	0 1	Not transmitted Transmitted (default)
UPC A re-encoding	0 1	UPC A transmitted as UPC A UPC A transmitted as EAN 13 (default)
UPC E re-encoding	0 1	UPC E transmitted as UPC E (default) UPC E transmitted as UPC A
EAN 8 re-encoding	0 1	EAN 8 transmitted as EAN 8 (default) EAN 8 transmitted as EAN 13



#### Code 93

Code 93 is a variable length, continuous symbology that uses four element widths.

#### **Action**

Tap the Code 93 parameter, then select an option to change this parameter setting. Tap (+) to access the Code 93 Lengths parameter.

#### **SNMP OID**

1.3.6.1.4.1.1963.15.3.3.1.1.7.1

# **Options**

- 0 Not active (default)
- 1 Active

# Code 93 Length

Sets the Code 93 bar code length.

#### Action

Tap (+) to expand the Code 93 parameter, then tap (+) to expand the Code 93 Lengths parameter. Tap the setting to be changed, then tap an option to change this setting.

#### **SNMP OID**

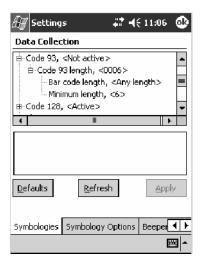
1.3.6.1.4.1.1963.15.3.3.1.1.19.1

# **Options**

Bar code length	0 1	Any length Minimum length (default)
Minimum length	001–254	Minimum length 1-254 (default is 6)



Note: If Bar code length = "1" then Minimum length is entered.



#### **Code 128**

Code 128 is a variable-length, continuous, high-density, alphanumeric symbology that uses multiple element widths and supports the extended ASCII character set.

#### Action

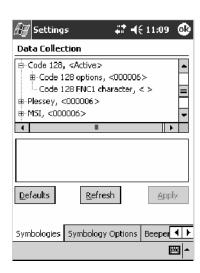
Tap the Code 128 parameter, then select an option to change this parameter setting.

#### **SNMP OID**

1.3.6.1.4.1.1963.15.3.3.1.1.9.1

# **Options**

- 0 Not active (default)
- 1 Active



This illustration is from a 700 Color Computer using a laser scanner.

# **Code 128 Options**

Set the following for the Code 128 parameter. Note that the EAN 128 JC1 and CIP 128 French Pharmaceutical options are not available when you use an imager with your 700 Color Computer.

#### Action

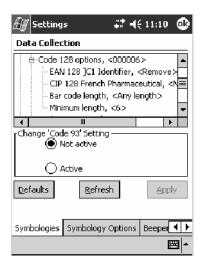
Tap (+) to expand the Code 128 Options parameter, select a setting, then select an option to change this setting.

#### **SNMP OID**

None.

## **Options**

EAN 128 ]C1 Identifier	0	Remove (default)
(Not supported when using an imager)	1	Include
CIP 128 French Pharmaceutical	0	Not active (default)
(Not supported when using an imager)	1	Active
Bar code length	0	Any length (default)
	1	Minimum length
Minimum length	001–254	Minimum length 1–254 (default is 6)



This illustration is from a 700 Color Computer using a laser scanner.

#### Code 128 FNC1 Character

The Code 128 FNC1 character (EAN 128 norms) can be any ASCII character and is used as a separator when multiple identifiers and their fields are concatenated. *Note that this is not available when you use an imager with your 700 Color Computer.* 

Non-printable ASCII characters can be entered using the following syntax where *HH* is the hexadecimal value of the character.

 $\xrack xHH$ 

For example, the GS character, whose hexadecimal value is 1D, would be entered as \x1D. In addition, the following characters have their own identifiers:

- BEL \a
- BS \b
- FF \f
- LF \n
- CR \r
- HT \t
- VT \v

#### **Action**

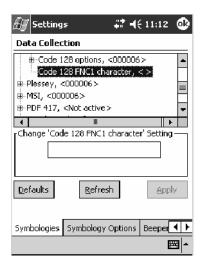
Tap (+) to expand the Code 128 parameter, then type the ASCII characters to be set for the Code 128 FNC1 character parameter.

#### **SNMP OID**

1.3.6.1.4.1.1963.15.3.3.1.1.21.1

## **Options**

Any ASCII character (default is the GS function character — ID hex)



# Plessey

Plessey is a pulse-width modulated symbology like most other bar codes. It includes a start character, data characters, an eight-bit cyclic check digit, and a termination bar. The code is continuous and not self-checking. You need to configure two parameters for Plessey code: Start Code and Check Digit. Note that this is not available when you use an imager with your 700 Color Computer.

#### Action

Tap (+) to expand the **Plessey** parameter, select the setting to be changed, then select an option to change this setting or select an option from the drop-down list.

#### SNMP OID

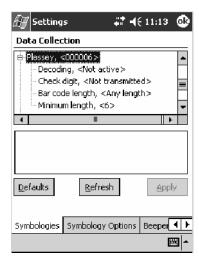
1.3.6.1.4.1.1963.15.3.3.1.1.10.1

#### **Options**

Decoding	0 1	Not active (default) Active
Check digit	0 1	Not transmitted (default) Transmitted
Bar code length	0 1	Any length Minimum length (default)
Minimum length	001–254	Minimum length 1–254 (default is 6)



Note: If Bar code length = "1" then Minimum length is entered.



#### MSI

MSI is a symbology similar to Plessey code (page 312) that includes a start pattern, data characters, one or two check digits, and a stop pattern. *Note that this is not available when you use an imager with your 700 Color Computer.* 

#### Action

Tap (+) to expand the MSI parameter, select the setting to be changed, then select an option to change this setting or select an option from the drop-down list.

#### **SNMP OID**

1.3.6.1.4.1.1963.15.3.3.1.1.15.1

# **Options**

Decoding	0 1	Not active (default) Active
Check digit	0 1 2 3	Mod 10 transmitted (default) Mod 10 not transmitted Double Mod 10 transmitted Double Mod 10 not transmitted
Bar code length	0 1	Any length Minimum length (default)
Minimum length	001–254	Minimum length 1-254 (default is 6)



Note: If Bar code length = "1" then Minimum length is entered.



#### **PDF417**

PDF417 is a stacked two-dimensional symbology that provides the ability to scan across rows of code. Each row consists of start/stop characters, row identifiers, and symbol characters, which consist of four bars and four spaces each and contain the actual data. This symbology uses error correction symbol characters appended at the end to recover loss of data.

Because the virtual wedge translates incoming data into keypad input, the size of the keypad buffer limits the effective length of the label to 128 characters. Longer labels may be truncated. For PDF417 labels of more than 128 characters, you can develop an application that bypasses the keypad buffer.

#### Action

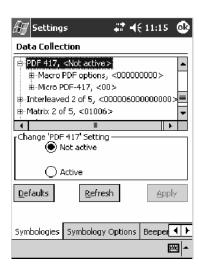
Tap the PDF417 parameter, then select an option to change this parameter setting. Tap (+) to access either the Macro PDF options parameter or the Micro PDF417 parameter.

#### **SNMP OID**

1.3.6.1.4.1.1963.15.3.3.1.1.17.1

# **Options**

- 0 Not active
- 1 Active (default)



This illustration is from a 700 Color Computer using a laser scanner.

# **Macro PDF options**

Macro PDF is used when a long message requires more than one PDF417 label. Note that this is not available when you use an imager with your 700 Color Computer.

• Select **Buffered** to store a multi-label PDF417 message in the Sabre buffer, thus transmitting the entire message when all labels are read.

• Select **Unbuffered** for multi-label PDF417 messages that are too long for the Sabre buffer (memory overflow). Each part of the PDF417 label is transmitted separately, and the host application must then assemble the message using the macro PDF control header transmitted with each label. *Control Header is only present in macro PDF codes and is always transmitted with unbuffered option.* 

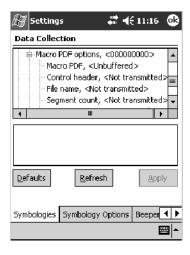
#### Action

Tap (+) to expand the PDF417 parameter, tap (+) to expand the Macro PDF parameter, select a setting to be changed, then select an option to change this setting.

#### **SNMP OID**

1.3.6.1.4.1.1963.15.3.3.1.1.22.1

options		
Macro PDF	0 1	Unbuffered Buffered (default)
Control header	0 1	Not transmitted (default) Transmitted
File name	0 1	Not transmitted (default) Transmitted
Segment count	0 1	Not transmitted (default) Transmitted
Time stamp	0 1	Not transmitted (default) Transmitted
Sender	0 1	Not transmitted (default) Transmitted
Addressee	0 1	Not transmitted (default) Transmitted
File size	0 1	Not transmitted (default) Transmitted
Checksum	0 1	Not transmitted (default) Transmitted



#### Micro PDF417

Micro PDF417 is a multi-row symbology derived from and closely based on PDF417 (page 314). A limited set of symbology sizes is available, together with a fixed level of error correction for each symbology size. Note that this is not available when you use an imager with your 700 Color Computer.

#### **Action**

Tap (+) to expand the PDF417 parameter, tap (+) to expand the Micro PDF417 parameter, select a setting to be changed, then select an option to change this setting.

#### **SNMP OID**

1.3.6.1.4.1.1963.15.3.3.1.1.27.1

Decoding	0 1	Not active (default) Active	
Code 128 Emulation	0 1	Not active (default) Active	



#### Interleaved 2 of 5

Interleaved 2 of 5 (I 2 of 5) is a high-density, self-checking, continuous, numeric symbology used mainly in inventory distribution and the automobile industry.



**Note**: An Interleaved 2 of 5 bar code label must be at least three characters long for the 700 Color Computer to scan and decode correctly.

#### **Action**

Tap (+) to expand the Interleaved 2 of 5 parameter, select the setting to be changed, then tap an option to change this setting or select an option from the drop-down list.

#### SNMP OID

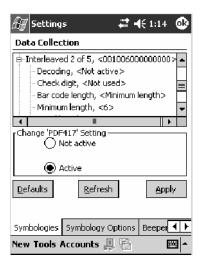
1.3.6.1.4.1.1963.15.3.3.1.1.23.1

#### **Options**

Decoding	0	Not active (default) Active
Check digit	0 1 2 3 4	Not used (default) Mod 10 transmitted Mod 10 not transmitted French CIP transmitted French CIP not transmitted
Bar code length	0 1 2	Any length Minimum length (default) Fixed lengths
Minimum length	003-254	Minimum length 3–254 (default is 6)
Fixed length 1	003-254	Fixed bar code length 3-254 (default is 3)
Fixed length 2	003-254	Fixed bar code length 3-254 (default is 3)
Fixed length 3	003-254	Fixed bar code length 3–254 (default is 3)



Note: If Bar code length = "1" then Minimum length is entered. If Bar code length = "2" then Fixed length 1, Fixed length 2, or Fixed length 3 is entered.



#### Matrix 2 of 5

Matrix 2 of 5 is a numerical symbology. Note that this is not available when you use an imager with your 700 Color Computer.

#### **Action**

Tap (+) to expand the Matrix 2 of 5 parameter, select the setting to be changed, then tap an option to change this setting or select an option from the drop-down list.

#### **SNMP OID**

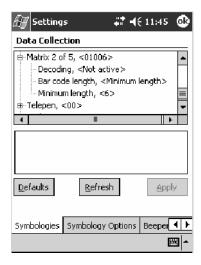
1.3.6.1.4.1.1963.15.3.3.1.1.24.1

# **Options**

Decoding	0 1	Not active (default) Active
Bar code length	0 1	Any length Minimum length (default)
Minimum length	001–254	Minimum length 1-254 (default is 6)



Note: If Bar code length = "1" then Minimum length is entered.



## Telepen

Telepen is an alphanumeric, case-sensitive, full ASCII symbology. *Note that this is not available when you use an imager with your 700 Color Computer.* 

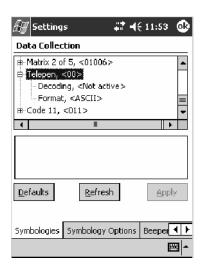
#### **Action**

Tap (+) to expand the **Telepen** parameter, select the setting to be changed, then tap an option to change this setting.

#### **SNMP OID**

1.3.6.1.4.1.1963.15.3.3.1.1.25.1

Decoding	0 1	Not active (default) Active
Format	0 1	ASCII (default) Numeric



#### Code 11

Code 11 is a high density, discrete numeric symbology that is extensively used in labeling telecommunications components and equipment. *Note that this is not available when you use an imager with your 700 Color Computer.* 

#### Action

Tap (+) to expand the Code 11 parameter, select the setting to be changed, then tap an option to change this setting.

#### **SNMP OID**

1.3.6.1.4.1.1963.15.3.3.1.1.26.1

Decoding	0 1	Not active (default) Active
Check digit verification	1 2	1 digit (default) 2 digits
Check digit transmit	0 1	Disable Enable (default)



#### **QR Code**

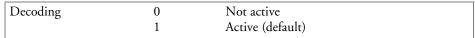
QR Code (Quick Response Code) is a two-dimensional matrix symbology containing dark and light square data modules. It has position detection patterns on three of its four corners and features direct encodation of the Japanese Kana-Kanji character set. It can encode up to 2509 numeric or 1520 alphanumeric characters and offers three levels of error detection. Note that this is not available when you use a laser scanner with your 700 Color Computer or if you are using a 730 Computer.

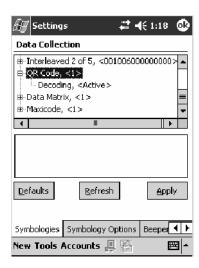
#### **Action**

Tap (+) to expand the **QR Code** parameter, select the setting to be changed, then tap an option to change this setting or select an option from the drop-down list.

#### **SNMP OID**

1.3.6.1.4.1.1963.15.3.3.1.1.35.1





#### **Data Matrix**

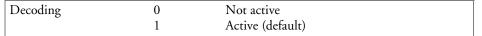
A two-dimensional matrix symbology, which is made of square modules arranged within a perimeter finder pattern. The symbology utilizes Error Checking and Correcting (ECC) algorithm with selectable levels for data error recovery and Cyclic Redundancy Check algorithm to validate the data. The character set includes either 128 characters conforming to ISO 646 (ANSI X3.4 - 1986) or 256 extended character set. Maximum capacity of a symbol is 2335 alphanumeric characters, 1556 8-bit byte characters or 3116 numeric digits. Note that this is not available when you use a laser scanner with your 700 Color Computer or if you are using a 730 Computer.

#### Action

Tap (+) to expand the **Data Matrix** parameter, select the setting to be changed, then tap an option to change this setting.

#### **SNMP OID**

1.3.6.1.4.1.1963.15.3.3.1.1.34.1





#### MaxiCode

MaxiCode is a fixed-size 2-D matrix symbology which is made up of offset rows of hexagonal elements arranged around a unique circular finder pattern. ASCII data is encoded in six-bit symbol characters. The symbol contains 33 rows which are alternately 30 and 29 elements wide. There are five different code sets. A single MaxiCode symbol can encode up to 93 characters of data. Note that this is not available when you use a laser scanner with your 700 Color Computer or if you are using a 730 Computer.

#### Action

Tap (+) to expand the MaxiCode parameter, select the setting to be changed, then tap an option to change this setting.

#### SNMP OID

1.3.6.1.4.1.1963.15.3.3.1.1.33.1

Decoding	0	Not active	
	1	Active (default)	



# **Symbology Options**



To access the settings from the 700 Color Computer, tap **Start** > **Settings** > the **System** tab > the **Data Collection** icon to access its control panel applet.

Use the right and left arrows to scroll to the **Symbology Options** tab, then tap this tab to access its parameters. The following are parameters for bar code symbology options. *Note that these are listed in the order of their appearance within the Symbology Options tab.* 

# Symbology ID

Identifies the bar code symbology in which data is encoded by prepending a user-specified symbology identifier to the data. You can prepend one of these types of character strings to identify the symbology:

- User-defined ASCII Character (Option 1):
  A user-defined symbology identifier is a single ASCII character. You can assign a custom identifier character to each bar code symbology. Note that this is not available when you use an imager with your 700 Color Computer.
- AIM ISO/IEC Standard (Option 2 Required to define symbology IDs): The AIM Standard has a three-character structure which indicates the symbology and optional features. See the AIM ISO/IEC Standard for information.

#### Action

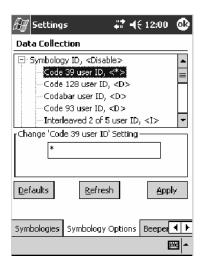
Select **Symbology ID**, then select an option to change this parameter setting. Tap (+) to expand the **Symbology ID** parameter, then select any of the user ID parameters listed. See the top of the next page for a sample screen of the Code 39 user ID.

#### SNMP OID

1.3.6.1.4.1.1963.15.3.3.4.1.22.1

- 0 Disable (default)
- 1 User defined (disabled when using an imager)
- 2 ISO/IEC Standard





#### Code 39 User ID

If "1" was selected in the **Symbology ID** parameter, you can set your own ASCII character to identify Code 39 bar code data. *Note that this is not available when you use an imager with your 700 Color Computer.* 

Action: Tap (+) to expand the Symbology ID parameter, select the Code 39 user

ID parameter, then enter a user ID value to change this parameter setting.

**SNMP OID:** 1.3.6.1.4.1.1963.15.3.3.4.1.3.1

**Options:** x where x is a single ASCII character. Default is asterisk (\*).

#### Code 128 User ID

If "1" was selected in the **Symbology ID** parameter, you can set your own ASCII character to identify Code 128 bar code data. *Note that this is not available when you use an imager with your 700 Color Computer.* 

Action: Tap (+) to expand the Symbology ID parameter, select the Code 128 user

ID parameter, then enter a user ID value to change this parameter setting.

**SNMP OID:** 1.3.6.1.4.1.1963.15.3.3.4.1.5.1

**Options:** x where x is a single ASCII character. Default is asterisk (\*).

#### Codabar User ID

If "1" was selected in the **Symbology ID** parameter, you can set your own ASCII character to identify Codabar bar code data. *Note that this is not available when you use an imager with your 700 Color Computer.* 

Action: Tap (+) to expand the Symbology ID parameter, select the Codabar user

ID parameter, then enter a user ID value to change this parameter setting.

**SNMP OID:** 1.3.6.1.4.1.1963.15.3.3.4.1.2.1

**Options:** x where x is a single ASCII character. Default is D.

#### Code 93 User ID

If "1" was selected in the **Symbology ID** parameter, you can set your own ASCII character to identify Code 93 bar code data. *Note that this is not available when you use an imager with your 700 Color Computer.* 

Action: Tap (+) to expand the Symbology ID parameter, select the Code 93 user

ID parameter, then enter a user ID value to change this parameter setting.

**SNMP OID:** 1.3.6.1.4.1.1963.15.3.3.4.1.4.1

**Options:** x where x is a single ASCII character. Default is asterisk (\*).

#### Interleaved 2 of 5 User ID

If "1" was selected in the **Symbology ID** parameter, you can set your own ASCII character to identify Interleaved 2 of 5 bar code data. *Note that this is not available when you use an imager with your 700 Color Computer.* 

Action: Tap (+) to expand the Symbology ID parameter, select the Interleaved 2 of 5 user ID parameter, then enter a user ID value to change this parameter.

ter setting.

**SNMP OID:** 1.3.6.1.4.1.1963.15.3.3.4.1.10.1

**Options:** x where x is a single ASCII character. Default is I (not lowercase L).

#### PDF417 User ID

If "1" was selected in the **Symbology ID** parameter, you can set your own ASCII character to identify PDF417 bar code data. *Note that this is not available when you use an imager with your 700 Color Computer.* 

Action: Tap (+) to expand the Symbology ID parameter, select the PDF417 user

ID parameter, then enter a user ID value to change this parameter setting.

SNMP OID: 1.3.6.1.4.1.1963.15.3.3.4.1.12.1

**Options:** x where x is a single ASCII character. Default is an asterisk (\*).

#### MSI User ID

If "1" was selected in the **Symbology ID** parameter, you can set your own ASCII character to identify MSI bar code data. *Note that this is not available when you use an imager with your 700 Color Computer.* 

Action: Tap (+) to expand the Symbology ID parameter, select the MSI user ID

parameter, then enter a user ID value to change this parameter setting.

**SNMP OID:** 1.3.6.1.4.1.1963.15.3.3.4.1.11.1

**Options:** x where x is a single ASCII character. Default is D.

# **Plessey User ID**

If "1" was selected in the **Symbology ID** parameter, you can set your own ASCII character to identify Plessey bar code data. *Note that this is not available when you use an imager with your 700 Color Computer.* 

Action: Tap (+) to expand the Symbology ID parameter, select the Plessey user

ID parameter, then enter a user ID value to change this parameter setting.

SNMP OID: 1.3.6.1.4.1.1963.15.3.3.4.1.13.1

**Options:** x where x is a single ASCII character. Default is D.

#### Standard 2 of 5 User ID

If "1" was selected in the **Symbology ID** parameter, you can set your own ASCII character to identify Standard 2 of 5 bar code data. *Note that this is not available when you use an imager with your 700 Color Computer.* 

Action: Tap (+) to expand the Symbology ID parameter, select the Standard 2 of

5 user ID parameter, then enter a user ID value to change this parameter

setting.

**SNMP OID:** 1.3.6.1.4.1.1963.15.3.3.4.1.23.1

**Options:** x where x is a single ASCII character. Default is D.

#### **UPC A User ID**

If "1" was selected in the **Symbology ID** parameter, you can set your own ASCII character to identify UPC-A (Universal Product Code) bar code data. *Note that this is not available when you use an imager with your 700 Color Computer.* 

Action: Tap (+) to expand the Symbology ID parameter, select the UPC A user

ID parameter, then enter a user ID value to change this parameter setting.

SNMP OID: 1.3.6.1.4.1.1963.15.3.3.4.1.6.1

**Options:** x where x is a single ASCII character. Default is A.

## **UPC E User ID**

If "1" was selected in the **Symbology ID** parameter, you can set your own ASCII character to identify UPC-E bar code data. *Note that this is not available when you use an imager with your 700 Color Computer.* 

Action: Tap (+) to expand the Symbology ID parameter, select the UPC E user

ID parameter, then enter a user ID value to change this parameter setting.

**SNMP OID:** 1.3.6.1.4.1.1963.15.3.3.4.1.7.1

**Options:** x where x is a single ASCII character. Default is E.

#### **EAN 8 User ID**

If "1" was selected in the **Symbology ID** parameter, you can set your own ASCII character to identify EAN-8 bar code data. *Note that this is not available when you use an imager with your 700 Color Computer.* 

Action: Tap (+) to expand the Symbology ID parameter, select the EAN 8 user

ID parameter, then enter a user ID value to change this parameter setting.

**SNMP OID:** 1.3.6.1.4.1.1963.15.3.3.4.1.8.1

**Options:** x where x is a single ASCII character. Default is xFF.

#### EAN 13 User ID

If "1" was selected in the **Symbology ID** parameter, you can set your own ASCII character to identify EAN-13 (European Article Numbering) bar code data. *Note that this is not available when you use an imager with your 700 Color Computer.* 

Action: Tap (+) to expand the Symbology ID parameter, select the EAN 13 user

ID parameter, then enter a user ID value to change this parameter setting.

**SNMP OID:** 1.3.6.1.4.1.1963.15.3.3.4.1.9.1

**Options:** x where x is a single ASCII character. Default is F.

## Matrix 2 of 5 User ID

If "1" was selected in the **Symbology ID** parameter, you can set your own ASCII character to identify Matrix 2 of 5 bar code data. *Note that this is not available when you use an imager with your 700 Color Computer.* 

Action: Tap (+) to expand the Symbology ID parameter, select the Matrix 2 of 5

user ID parameter, then enter a user ID value to change this parameter

setting.

**SNMP OID:** 1.3.6.1.4.1.1963.15.3.3.4.1.24.1

**Options:** x where x is a single ASCII character. *Default is D.* 

#### Telepen User ID

If "1" was selected in the **Symbology ID** parameter, you can set your own ASCII character to identify Telepen bar code data. *Note that this is not available when you use an imager with your 700 Color Computer.* 

Action: Tap (+) to expand the Symbology ID parameter, select the Telepen user

ID parameter, then enter a user ID value to change this parameter setting.

**SNMP OID:** 1.3.6.1.4.1.1963.15.3.3.4.1.25.1

**Options:** x where x is a single ASCII character. Default is an asterisk (\*).

#### Code 11 User ID

If "1" was selected in the **Symbology ID** parameter, you can set your own ASCII character to identify Code 11 bar code data. *Note that this is not available when you use an imager with your 700 Color Computer.* 

Action: Tap (+) to expand the Symbology ID parameter, select the Code 11 user

ID parameter, then enter a user ID value to change this parameter setting.

**SNMP OID:** 1.3.6.1.4.1.1963.15.3.3.4.1.16.1

**Options:** x where x is a single ASCII character. Default is asterisk (\*).

#### **Prefix**

Prepends a string of up to 20 ASCII characters to all scanned data.

#### Action

Tap the **Prefix** parameter, then enter a prefix value to change this parameter setting.

#### **SNMP OID**

1.3.6.1.4.1.1963.15.3.3.4.1.29.1

# **Options**

Acceptable values are up to 20 ASCII characters. Embedded null (<NUL >) characters are not allowed. Default is no characters (disabled).



### **Suffix**

Appends a string of up to 20 ASCII characters to all scanned data.

### Action

Tap the **Suffix** parameter, then enter a suffix value to change this parameter setting.

### **SNMP OID**

1.3.6.1.4.1.1963.15.3.3.4.1.30.1

## **Options**

Acceptable values are up to 20 ASCII characters. Embedded null (<NUL >) characters are not allowed. Default is no characters (disabled).



## **Beeper/LED**



To access the settings from the 700 Color Computer, tap **Start** > **Settings** > the **System** tab > the **Data Collection** icon to access its control panel applet.

Use the right and left arrows to scroll to the Beeper/LED tab, then tap this tab to access its parameters.

Most of these functions are not available when using an imager. The following table shows which functions are supported either by an imager or by a laser scanner.

Beeper Function	lmager	Laser Scanner
Beeper	X	
Beeper Volume		X
Beeper Frequency		X
Good Read Beeps		X
Good Read Beep Duration		X

The following are parameters for features on the 700 Color Computer. *Note that these are listed in the order of their appearance.* 

### **Beeper**

Sets the volume for the good read beep. Note that this is not available when you use a laser scanner with your 700 Color Computer.

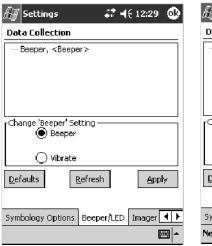
### **Action**

Tap the **Beeper** parameter, then select an option to change this parameter setting.

### **SNMP OID**

1.3.6.1.4.1.1963.15.3.1.4.1.6.1

- 1 Beeper (default)
- 4 Vibrate (not supported on 730 Computers)





700 Color with Imager Screen

730 Screen

## **Beeper Volume**

Sets the volume for the good read beep. Note that this is not available when you use an imager with your 700 Color Computer.

### Action

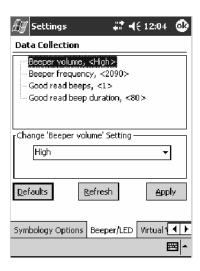
Tap the **Beeper volume** parameter, then select an option to change this parameter setting.

### **SNMP OID**

1.3.6.1.4.1.1963.15.3.1.4.1.6.1

## **Options**

- 0 Low
- 1 High (default)
- 2 Medium
- 3 Off
- 4 Vibrate



## **Disabling the Volume**



To disable the beeper, tap Start > Settings > the Personal tab > Sounds & Notifications > the Volume tab, drag the System volume slider bar to the left "Silent" position, then tap ok to exit this applet. See Chapter 1, "Introduction" for more information.

## **Beeper Frequency**

Sets the frequency for the good read beep. Note that this is not available when you use an imager with your 700 Color Computer.

### **Action**

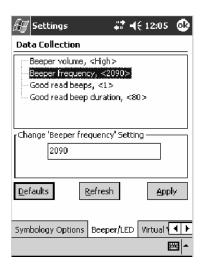
Tap the Beeper frequency parameter, then enter a frequency value to change this parameter setting.

### **SNMP OID**

1.3.6.1.4.1.1963.15.3.1.4.1.7.1

## **Options**

1000-4095 (default is 2090)



## **Good Read Beeps**

Sets the number of good read beeps. Note that this is not available when you use an imager with your 700 Color Computer.

### Action

Tap the Good read beeps parameter, then select an option to change this parameter setting.

### **SNMP OID**

1.3.6.1.4.1.1963.15.3.1.4.1.8.1

- 0 No beeps
- 1 One beep (default)
- 2 Two beeps



## **Good Read Beep Duration**

Sets the duration of the good read beep. Note that this is not available when you use an imager with your 700 Color Computer.

### Action

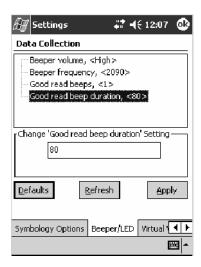
Tap the Good read beep duration parameter, then enter a duration value to change this parameter setting.

### **SNMP OID**

1.3.6.1.4.1.1963.15.3.1.4.1.9.1

## **Options**

0–2550 Beep duration in milliseconds. (default is 80)



## **Imager**



**Note:** These instructions do *not* apply to the 730 Computer.



To access the settings from the 700 Color Computer, tap **Start** > **Settings** > the **System** tab > **Data Collection** to access its control panel applet.

Use the right and left arrows to scroll to the **Imager** tab, then tap this tab to access its parameters.

The following are parameters for the imager. Note that these are listed in the order of their appearance within the Imager tab.

### **Aimer LED Duration**

The Aimer LED Duration controls the time the Aimer LED is turned on when the scan button is pressed. After this time, images are captured for decoding. The purpose is to position the Aimer LED on the bar code symbol before attempting to decode the bar code. *Note that this is not available when you use a laser scanner with your 700 Color Computer.* 

### Action

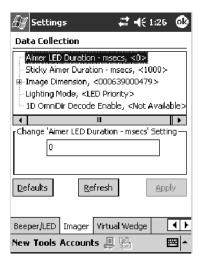
Tap the Aimer LED Duration parameter, then enter a value to change this setting. Note that values must be in 50 ms increments, such as 500, 650, or 32500. Values not entered in 50 ms increments are rounded down. For example, 2489 ms is rounded down to 2450 ms, 149 ms is rounded down to 100 ms, etc..

### **SNMP OID**

1.3.6.1.4.1.1963.15.3.3.3.1.1.21.1

## **Options**

0-65500 ms (Default is 0)



## **Sticky Aimer Duration**

The Sticky Aimer Duration controls the time the Aimer LED stays on after the a bar code read completes or after the trigger button is released. *Note that this is not available when you use a laser scanner with your 700 Color Computer.* 

### Action

Tap the Sticky Aimer Duration parameter, then enter a value to change this setting. Note that values must be in 50 ms increments, such as 500, 650, or 32500. Values not entered in 50 ms increments are rounded down. For example, 2489 ms is rounded down to 2450 ms, 149 ms is rounded down to 100 ms, etc..

### **SNMP OID**

1.3.6.1.4.1.1963.15.3.3.3.1.1.24.1

## **Options**

0-65535 ms (Default is 1000)



## **Image Dimension**

The image dimensions control the vertical size of the image for decoding. This can restrict the image to one bar code when otherwise, there might be more than one bar code in the image to be decoded. *Note that this is not available when you use a laser scanner with your 700 Color Computer.* 

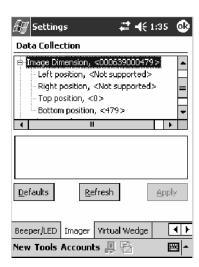
### Action

Tap the **Image dimension** parameter, select the position to be changed, then tap an option or enter a value to change this position.

### SNMP OID

1.3.6.1.4.1.1963.15.3.3.3.1.1.22.1

Left position	0	Not supported
Right position	0	Not supported
Top position	0-478	Position in pixels (Default is 0)
Bottom position	0-479	Position in pixels (Default is 479)



## **Lighting Mode**

The Lighting Mode sets the lighting mode of the imager. When set to "LED Priority," the imager depends more on ambient lighting to illuminate the bar code for reading. When set to "Aperture Priority," the imager uses its built-in LED to illuminate the bar code for reading. Note that this is not available when you use a laser scanner with your 700 Color Computer.

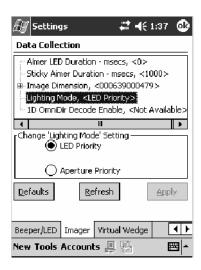
### **Action**

Tap the **Lighting Mode** parameter, then select an option to change this parameter setting.

### **SNMP OID**

1.3.6.1.4.1.1963.15.3.3.3.1.1.23.1

- 0 LED Priority (default)
- 1 Aperture Priority



### 1D OmniDir Decode Enable

The 1D OmniDir Decode Enable affects the scanning abilities of the IT4000 Imager. With 1D omni directional enabled, the imager is able to decode images and bar code labels regardless of the orientation of the label. With 1D omni directional disabled, the imager only decodes labels in the direction of the aimer LED. *Note that this is not available when you use a laser scanner with your 700 Color Computer.* 

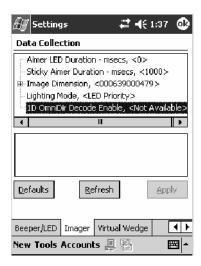
### Action

Tap the 1D OmniDir Decode Enable parameter, then select an option to change this parameter setting.

### **SNMP OID**

1.3.6.1.4.1.1963.15.3.3.3.1.1.25.1

- 0 Disabled
- 1 Enabled (default)



## **Virtual Wedge**



To access the settings from the 700 Color Computer, tap **Start** > **Settings** > the **System** tab > **Data Collection** to access its control panel applet.

Use the right and left arrows to scroll to the Virtual Wedge tab, then tap this tab to access its parameters.

The following are parameters for the virtual wedge scanner. Note that these are listed in the order of their appearance within the Virtual Wedge tab.

## **Virtual Wedge**

Enables or disables the virtual wedge for the internal scanner. The virtual wedge retrieves scanned Automatic Data Collection (ADC) data and sends it to the keypad driver so that the 700 Color Computer can receive and interpret the data as keypad input.

Because the virtual wedge translates incoming data into keypad input, the size of the keypad buffer limits the effective length of the label to 128 characters. Longer labels may be truncated. For labels of more than 128 characters, you need to develop an application that bypasses the keypad buffer.

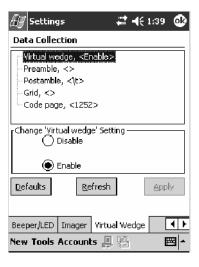
### **Action**

Tap the Virtual Wedge parameter, then tap an option to change this parameter setting.

### SNMP OID

1.3.6.1.4.1.1963.15.3.2.1.1.2.1

- 0 Disable
- 1 Enable (default)



### **Preamble**

Sets the preamble that precedes any data you scan with the 700 Color Computer. Common preambles include a data location number or an operator number.

### **Action**

Tap the **Preamble** parameter, then enter a preamble value to change this parameter setting.

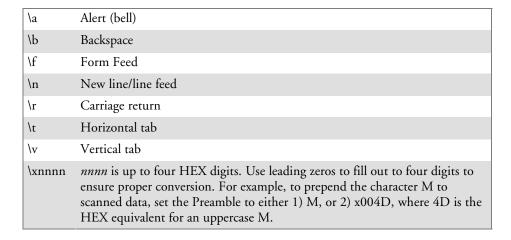
### **SNMP OID**

1.3.6.1.4.1.1963.15.3.2.1.1.3.1

### **Syntax**

**ADdata** 

where *data* is any acceptable values up to 31 ASCII characters. Embedded null (<NUL >) characters are not allowed. Below are the non-printing characters you can use for Virtual Wedge Preambles. *Default is blank*.







Note: When you enter the AD command without data, the preamble is disabled. If you want to use quotation marks or the following combinations of characters as part of the appended data, separate those characters from the AD command with quotes. If you do not use quotes as described here, the 700 Color Computer interprets the characters as another configuration command:

AD

AE

AF

KC

BV

EX

DF

### **Example**

To use the two-character string BV as a preamble, scan this command (as a Code 39 label) or send this command through the network: \$+AD"BV"

### **Postamble**

Sets the postamble that is appended to any data you scan with the 700 Color Computer. Common postambles include cursor controls, such as tabs or carriage return line feeds.

### **Action**

Tap the **Postamble** parameter, then enter a postamble value to change this parameter setting.

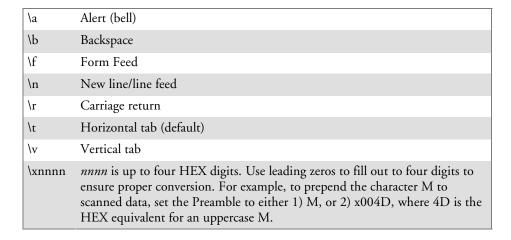
### **SNMP OID**

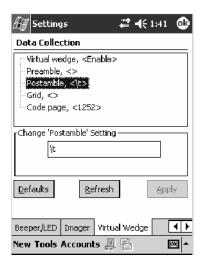
1.3.6.1.4.1.1963.15.3.2.1.1.4.1

### **Syntax**

**AEdata** 

where *data* is any acceptable values up to 31 ASCII characters. Embedded null (<NUL >) characters are not allowed. Below are the non-printing characters you can use for Virtual Wedge Postambles:







Note: When you enter the AE command without data, the postamble is disabled. If you want to use quotation marks or the following combinations of characters as part of the appended data, separate those characters from the AE command with quotes. If you do not use quotes as described here, the 700 Color Computer interprets the characters as another configuration command.

AD

AE

AF

KC

BV

EX

DF

### **Example**

To use the two-character string BV as a postamble, scan this command (as a Code 39 label) or send this command through the network: \$+AE\*BV"

### Grid

Sets the virtual wedge grid, which filters the data coming from this 700 Color Computer. The data server supports data filtering, which allows you to selectively send scanned data. The virtual wedge grid is similar to the "format" argument of the C Runtime Library scan function.

### Action

Tap the **Grid** parameter, then enter a grid value to change this parameter setting.

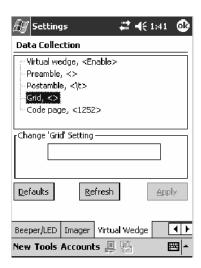
### **SNMP OID**

1.3.6.1.4.1.1963.15.3.2.1.1.5.1

### **Syntax**

AF<symID> filter-expression= > editing-expression where:

- <symID> The AIM symbology ID (optional).
- filter-expression
   Any character string that includes valid filter expression values. See the user manual provided with your resouce kit via the IDL for a list of valid filter expression values.
- editing-expression
   Any character string that includes valid editing expression values. See the user manual provided with your resouce kit via the IDL for a list of valid editing expression values.



## **Code Page**

Sets the virtual wedge code page. The code page controls the translation from the character set of the raw collected data to Unicode, which is the character set expected by Windows CE applications. The default code page is 1252, which is the Windows Latin 1 (ANSI) character set.

### Action

Tap the Code Page parameter, then select an option to change this parameter setting.

### **SNMP OID**

1.3.6.1.4.1.1963.15.3.2.1.1.6.1

## **Options**

The only acceptable value for the code page parameter is "1252," which is the default.



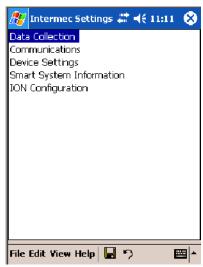
# **Intermec Settings Control Panel Applet**

You may have the Intermec Settings control panel applet. Information about the settings you can configure with this applet is described in the *Intermec Computer Command Reference Manual*. The online manual is available from the Intermec web site at www.intermec.com.

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



To access the settings from the 700 Color Computer, tap **Start** > **Settings** > the **System** tab > **Intermec Settings** to access its control panel applet.



# **SNMP Control Panel Applet**



**Note:** This applet is not available in units with PSM Build 3.00 or newer. To determine your PSM build version, tap **Start** > **Programs** > **File Explorer** > the **PSMinfo** text file.



If your unit has PSM Build 3.00 or newer, then you may have the Intermec Settings control panel applet in place of the SNMP applet. Information about the settings you can configure with the Intermec Settings applet is described in the Intermec Computer Command Reference Manual. The online manual is available from the Intermec web site at www.intermec.com.

Simple Network Management Protocol (SNMP) parameters include identification information, security encryption, security community strings, and traps.



To access the settings from the 700 Color Computer, tap **Start** > **Settings** > the **System** tab > **SNMP** to access its control panel applet.

Tap a tab to access its menus. These tabs represent three groups of settings or parameters:

- Security (starting on the next page)
- Traps (starting on page 357)
- Identification (starting on page 359)

## **Security**



To access the settings from the 700 Color Computer, tap **Start** > **Settings** > the **System** tab > **SNMP** > the **Security** tab to access its parameters.

The following are parameters that affect encryption and community strings. Note that these are listed in the order of their appearance within the Security tab.

## **Read Only Community**

Sets the read-only community string for this 700 Color Computer, which is required for processing of SNMP get and get next requests.

### **Action**

Tap the Read Only Community parameter, then enter a community string to change this parameter setting.

### **SNMP OID**

1.3.6.1.4.1.1963.10.5.1.2.0

### **Options**

The read-only community string can be up to 128 ASCII characters. Default is Public.



## **Read/Write Community**

Sets the read/write community string, which is required for processing of SNMP set requests by this 700 Color Computer. An SNMP packet with this name as the community string will also process SNMP get and next requests.

### Action

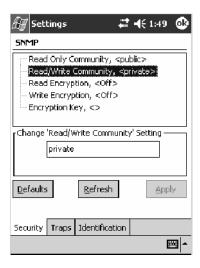
Tap the Read/Write Community parameter, then enter a community string to change this parameter setting.

### **SNMP OID**

1.3.6.1.4.1.1963.10.5.1.3.0

### **Options**

The read/write community string can be up to 128 ASCII characters. Default is Private.



## **Read Encryption**

Sets the packet-level mode of security for SNMP read-only requests. If you enable read encryption, all received SNMP get and get next packets have to be encrypted or the packet will not be authorized. If encryption is enabled, you can only use software provided by Intermec Technologies.



Note: To enable security encryption, you also need to set the Security Encryption Key (page 356).

### Action

Tap the **Read Encryption** parameter, then select an option to change this parameter setting.

### **SNMP OID**

1.3.6.1.4.1.1963.10.5.1.4.0

1	On	SNMP get and get next packets must be encrypted	
2	Off	SNMP packets do not have to be encrypted (default)	



## **Write Encryption**

Sets the packet-level mode of security for SNMP read/write requests. If you enable write encryption, all SNMP packets that are received with the read/write community string have to be encrypted or the packet will not be authorized. You need to use software from Intermec Technologies that supports encryption.



**Note**: To enable security encryption, you also need to set the Security Encryption Key (page 356).

### Action

Tap the **Write Encryption** parameter, then select an option to change this parameter setting.

## **SNMP OID**

1.3.6.1.4.1.1963.10.5.1.5.0

1	On	SNMP packets must be encrypted	
2	Off	SNMP packets do not have to be encrypted (default)	



## **Encryption Key**

Identifies the key that this 700 Color Computer uses to encrypt or decipher SNMP packets. Encryption is used only by software provided by Intermec Technologies. If encryption is enabled, SNMP management platforms will not be able to communicate with the 700 Color Computer. The encryption key is returned encrypted.

### **Action**

Tap the Encryption Key parameter, then enter a security encryption key value to change this parameter setting.



Note: You also need to set either Read Encryption (page 354) or Write Encryption (page 355) or both.

### SNMP OID

1.3.6.1.4.1.1963.10.5.1.6.0

## **Options**

The encryption key can be from 4 to 20 ASCII characters. Default is NULL.



## **Traps**



To access the settings from the 700 Color Computer, tap **Start** > **Settings** > the **System** tab > **SNMP** > the **Traps** tab to access its parameters.

The following are authentication and threshold parameters for traps. *Note that these are listed in the order of their appearance within the Traps tab.* 

### **Authentication**

Determines whether to send authentication traps. When trap authentication is enabled, an authentication trap is sent if an SNMP packet is received by the master agent with an invalid community string.

### **Action**

Tap the **Authentication** parameter, then select an option to change this parameter setting.

### **SNMP OID**

1.3.6.1.4.1.1963.10.5.2.2.0

- 1 On (default)
- 2 Off



### **Threshold**

Determines the maximum number of traps per second that the master agent generates. If the threshold is reached, the trap will not be sent.

### **Action**

Tap the **Threshold** parameter, then enter a threshold value to change this parameter setting.

### **SNMP OID**

1.3.6.1.4.1.1963.10.5.2.3.0

### **Options**

Any positive integer value. Default is 10.



## **Identification**



To access the settings from the 700 Color Computer, tap **Start** > **Settings** > the **System** tab > **SNMP** > the **Identification** tab to access its parameters.

The following are parameters for contact, location, and name information for support purposes. *Note that these are listed in the order of their appearance within the Identification tab.* 

### **Contact**

Sets the contact information for the person responsible for this 700 Color Computer.

### Action

Tap the **Contact** parameter, then enter the name of your contact representative to change this parameter setting.

### SNMP OID

1.3.6.1.2.1.1.4.0

### **Options**

The identification contact may be up to 255 ASCII characters. Default is no characters or blank.



### Name

Sets the assigned name for this 700 Color Computer.

### Action

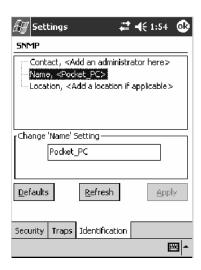
Tap the Name parameter, then enter the name of your 700 Color Computer to change this parameter setting.

### **SNMP OID**

1.3.6.1.2.1.1.5.0

## **Options**

The identification name may be up to 255 ASCII characters. Default is no characters or blank.



### Location

Sets the identification location for this 700 Color Computer, such as "Shipping."

### Action

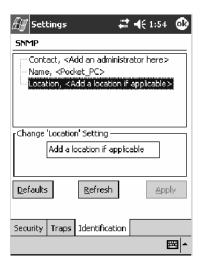
Tap the **Location** parameter, then enter the location of where your 700 Color Computer to change this parameter setting.

### **SNMP OID**

1.3.6.1.2.1.1.6.0

### **Options**

The identification location may be up to 255 ASCII characters. Default is no characters or blank.



# **Unit Information Control Panel Applet**



**Note:** This applet is not available in units with PSM Build 3.00 or newer. To determine your PSM build version, tap **Start** > **Programs** > **File Explorer** > the **PSMinfo** text file.



If your unit has PSM Build 3.00 or newer, then you may have the Intermec Settings control panel applet in place of the Unit Information applet. Information about the settings you can configure with the Intermec Settings applet is described in the *Intermec Computer Command Reference Manual*. The online manual is available from the Intermec web site at www.intermec.com.

Unit Information is a read-only control panel applet that provides information about your 700 Color Computer, such as software version builds, available CAB files, and the internal battery status.



To access the settings from the 700 Color Computer, tap **Start** > **Settings** > the **System** tab > **Unit Information** to access its control panel applet.

Tap a tab to access its menus. These tabs represent three groups of settings or parameters:

- Versions (next paragraph)
- Battery Status (starting on page 363)
- CAB Files (starting on page 364)

## **Versions**

You can view the latest software build version on your 700 Color Computer by accessing the **Unit Information** control panel applet.



To access the settings from the 700 Color Computer, tap **Start** > **Settings** > the **System** tab > **Unit Information** > the **Versions** tab to view the latest software build version. Tap **ok** to exit this information.







730 Screen

## **Battery Status**

You can view the battery status for your 700 Color Computer by accessing the **Unit Information** control panel applet.



To access the settings from the 700 Color Computer, tap **Start** > **Settings** > the **System** tab > **Unit Information** > the **Battery Status** tab to view the current status. Tap **ok** to exit this information.



## **CAB Files**

You can view the latest developer or released version of each CAB file from Intermec Technologies Corporation that are installed in your 700 Color Computer via the **Unit Information** control panel applet. *Custom CAB files are not displayed in this applet.* See the *Software Tools User's Manual* for more information about these files.



To access the information from the 700 Color Computer, tap **Start** > **Settings** > the **System** tab > **Unit Information** > the **CAB Files** tab to view the current CAB file versions. Tap **ok** to exit this information.



When a CAB file is built, a registry entry is created with a build number for that file. This CAB Files control panel applet looks for a registry key for each CAB file installed. When the registry entry is found, the CAB file name and version number information are displayed. If a CAB file has not been installed, then its information is not displayed.

Below is a list of CAB files from Intermec Technologies that are available for your 700 Color Computer with their latest developer or released version of the software build. Should you need to add any of these to your 700 Color Computer, contact an Intermec representative.

#### • BtMainStack:

Installation of the Main Bluetooth Stack is handled automatically as part of the operating system boot-up procedure. See Chapter 4, "Network Support," for more information about Bluetooth wireless printing.

### • Comm Port Wedge:

The software build for the Comm Port Wedge. Note that the Comm Port Wedge CAB file is available.

### • NPCPTest:

This installs a Norand<sup>®</sup> Portable Communications Protocol (NPCP) Printing test application which will print to an Intermec<sup>®</sup> 4815, 4820, or 6820 Printer. See Chapter 5, "Printer Support," for more information.

### • S9C Upgrade:

Installs the files needed to upgrade the S9C scanner firmware. See the 700 Color Management Tools for information on upgrading the firmware.

### WinCfg:

Configures the NRINET.INI file, launches the NRINet client, and loads and unloads the LAN and WLAN device drivers.

### • Wireless Printing Sample:

Installs a sample application that developers can use for reference when they are developing their own Wireless Printing applications. The source code for this application is included as part of the Bluetooth Resource Kit in the IDL, which is available as a download from the Intermec web site at www.intermec.com. Contact your Intermec representative for more information.

### • ActiveX Control Tools:

This lists some of the CAB files that may be available with which to install ActiveX Control Tools. *See the online help for more information.* 

### • AXCommunication:

Communication controls that transmit or receive messages from input connections.

#### • AXFileTransfer:

File transfer controls that transmit and receive files using the Trivial File Transfer Protocol (TFTP).

### • AXReaderCommand:

Reader command functions that modify and retrieve configuration information from your 700 Color Computer.

### • AXVWedge:

The virtual wedge control that retrieves scanned ADC data and sends it to the keyboard driver to interpret data as keyboard input.

# **Utilities Control Panel Applet**

The Utilities control panel applet examines and modifies settings and operational modes of specific hardware and software on the 700 Color Computer, including the dock switch, registry storage, wakeup mask, and application launch keys.



To access the settings from the 700 Color Computer, tap **Start** > **Settings** > the **System** tab > **Utilities** to access its control panel applet.

Use the left and right arrows to scroll through the tabs along the bottom of the control panel applet, then tap a tab to access its menus. These tabs represent the following groups of settings or parameters:

- Dock Switch (next paragraph)
- Registry Save (page 367)
- Wakeup Mask (page 368)
- App Launch (page 369)

### **Dock Switch**

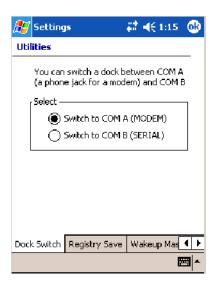


From the 700 Color Computer, tap **Start** > **Settings** > the **System** tab > **Utilities** > the **Dock Switch** tab to access the Dock Switch control panel applet.

Use this applet to control the position of the dock switch. This can be set either to a COM A (phone jack for a modem) position or to a COM B (serial) position.

If switched to COM B and suspended, the terminal has the following behavior:

- If the 700 Color Computer is on charge, the dock switch remains switched to COM B.
- If the 700 Color Computer is off charge, the dock switch switches to COM A and remain in this position until the 700 Color Computer resumes charge.



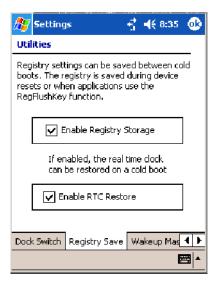
### **Registry Save**



From the 700 Color Computer, tap **Start** > **Settings** > the **System** tab > **Utilities** > the **Registry Save** tab to access the Registry Save control panel applet.

For Windows Mobile 2003, the only medium available for saving the registry is the Flash File System (PSM). Registry data is stored in the "\Flash\_File\_Store\Registry" path. Check **Enable Registry Storage** to enable this function.

To ensure that the 700 Color Computer restores the real-time clock after a cold-boot, check the **Enable RTC Restore** option. *Note that this does not apply to the 730 Computer.* 





700 Color Screen

730 Screen

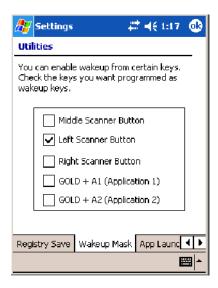
### **Wakeup Mask**



From the 700 Color Computer, tap Start > Settings > the System tab > Utilities > the Wakeup Mask tab to access the Wakeup Mask control panel applet.

This applet programs three scanner buttons and the A1 and A2 application keys to be "wakeup" or resume keys. That is, to prompt the 700 Color Computer to "wake up" or resume activity after going to "sleep" as a result of being inactive after a length of time. This information will remain between warm and cold boots.

Check the appropriate box, then tap **ok** to apply your settings.



Based on your setting, do the following to "wake up" the 700 Color Computer.

If you select:	Then do this on Numeric Keyboard	Then do this on Alphanumeric Keyboard
Middle Scanner Button	Squeeze the button on the Scan Handle	Squeeze the button on the Scan Handle
Left Scanner Button	Squeeze the left scanner button	Squeeze the left scanner button
Right Scanner Button	Squeeze the right scanner button	Squeeze the right scanner button
GOLD + A1 (Application 1)	Press [Gold] © (A)	Press [Gold/White] (AAT)
GOLD + A2 (Application 2)	Press [Gold] $\bigcirc$ $\bigcirc$ $\bigcirc$ $\bigcirc$	Press [Gold/White]   BA2

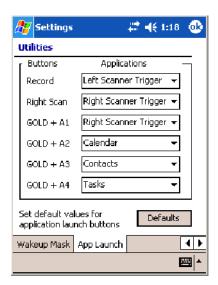
### **App Launch**



From the 700 Color Computer, tap **Start** > **Settings** > the **System** tab > **Utilities**, then scroll to the right to tap the **App Launch** tab to access the Application Launch control panel applet.

This applet programs or maps two scanner buttons and four application keys to start up to six applications. *Note that the left scanner button also acts as the record button.* 

For 700 Color Computers with either a laser scanner or an imager, default mappings are shown in the following illustration.



For 700 Color Computers without either a laser scanner or an imager, the default maps the Record, Calendar, Contacts, and Tasks applications the top four and the A3 and A4 buttons are "unassigned" or available for two more applications.



**Note**: Record, Calendar, Contacts, and Tasks are Pocket PC applications. See Chapter 2, "Windows Mobile 2003," for more information about these applications.

- To assign an application to a button, select an application from the applicable drop-down list box.
- To assign a new application, select the "Add new application" option, which brings up an Open File dialog and browse SD or CF storage cards for new applications.
- To disable or unmap a currently mapped application from a corresponding button, select "unassigned" from the applicable drop-down list.
- To restore these buttons to their defaults, tap **Defaults** in the lower right corner.

### Appendix A — Configurable Settings



**Note**; You cannot map an application to more than one button. Should you assign the same application to two buttons, a verification prompt will appear after the second button to confirm whether you want to remap the application. If you tap **Yes**, the applet changes the first button to "unassigned" and map the application to the second button.



Note: All changes are activated immediately upon selection.

# **Wireless Network Control Panel Applet**



**Note**: See Chapter 4, "*Network Support*," for information about the 802.11b or 802.11b/g radio module.

### **About the Wireless Network**

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

- Infrastructure networks get you onto your corporate network and the internet. Your 700 Color Computer establishes a wireless connection to an access point, which links you to the rest of the network. When you connect to a network via an access point, you are using the 802.11b or 802.11b/g infrastructure mode.
- Ad-hoc networks are private networks shared between two or more clients, even with no access point.

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

Intermec Technologies recommends using security measures with wireless networks to prevent unauthorized access to your network and to ensure your privacy of transmitted data. The following are required elements for secure networks:

- Authentication by both the network and the user
- Authentication is cryptographically protected
- Transmitted data

There are many schemes available for implementing these features.

# **Terminology**

Below are terms you may encounter as you configure your wireless network:

- CKIP (Cisco Key Integrity Protocol)
  This is Cisco's version of the TKIP protocol, compatible with Cisco
  Airnet products.
- EAP (Extensible Authentication Protocol)
   802.11b or 802.11b/g uses this protocol to perform authentication.
   This is not necessarily an authentication mechanism, but is a common framework for transporting actual authentication protocols. Intermec Technologies provides a number of EAP protocols for you to choose the best for your network.

- TKIP (Temporal Key Integrity Protocol)
  This protocol is part of the IEEE 802.11i encryption standard for wireless LANs., which provides per-packet key mixing, a message integrity check and a re-keying mechanism, thus overcoming most of the weak points of WEP. This encryption is more difficult to crack than the standard WEP. Weak points of WEP include:
  - No Installation Vector (IV) reuse protection
  - Weak keys
  - No protection against message replay
  - No detection of message tampering
  - No key updates
- WEP (Wired Equivalent Privacy) encryption
  With preconfigured WEP, both the client 700 Color Computer and
  access point are assigned the same key, which can encrypt all data be-

access point are assigned the same key, which can encrypt all data between the two devices. WEP keys also authenticate the 700 Color Computer to the access point — unless the 700 Color Computer can prove it knows the WEP key, it is not allowed onto the network.

WEP keys are only needed if they are expected by your clients. There are two types available: 64-bit (5-character strings, 12345) (default) and 128-bit (13-character strings, 1234567890123). Enter these as either ASCII (12345) or Hex (0x3132333435).

• WPA (Wi-Fi Protected Access)

This is an enhanced version of WEP that does not rely on a static, shared key. It encompasses a number of security enhancements over WEP, including improved data encryption via TKIP and 802.11b or 802.11b/g authentication with EAP.

WiFi Alliance security standard is designed to work with existing 802.11 products and to offer forward compatibility with 802.11i.

### **Configuring Your Wireless Network**



To start 802.11b or 802.11b/g communications on the 700 Color Computer, tap **Start** > **Settings** > the **System** tab > **Wireless Network** to access the Profile Wizard for the 802.11b or 802.11b/g radio module.

A profile contains all the information necessary to authenticate you to the network, such as login name, password or certificate, and protocols by which you are authenticated.

You can have up to four profiles for different networks. For example, you may have different login names or passwords on different networks, or you may use a password on one network, and a certificate on another.

Use the Profiles page to select and configure between the networking environments assigned to this 802.11b or 802.11b/g radio.



### Profile:

Tap the drop-down list to choose between four different profiles assigned to this unit, then tap **Edit Select Profile**, make the changes needed for this profile (*starting on the next page*), then tap **OK** to return to the Profiles page.

• Enable Microsoft's Wireless Zero Config

Check this box to enable Microsoft's Wireless Zero Config application. This effectively disables the Intermec software solution for 802.11b or 802.11b/g, including configuration via the CORE application and the Wireless Network control panel applet.

### Basic

Use the Basic page to set the network type, name, and manage battery power for this profile. Tap **ok** or **OK** to return to the Profiles page.

### • Profile Label:

Enter a unique name for your profile.

### • Network type:

Tap the drop-down list to select either "Infrastructure" if your network uses access points to provide connectivity to the corporate network or internet; or "Ad-Hoc" to set up a private network with one or more participants.

### • Channel:

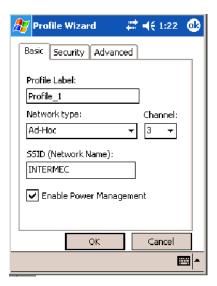
If you selected "Ad-Hoc" for the network type, select the channel on which you are communicating with others in your network. There are up to 11 channels available.

### • SSID (Network Name):

This assumes the profile name *unless another name is entered in this field*. If you want to connect to the next available network or are not familiar with the network name, enter "ANY" in this field. Consult your LAN administrator for network names.

### • Enable Power Management:

Check this box to conserve battery power (default), or clear this box to disable this feature.



### **Security**

The following are available from the 8021x Security drop-down list. Note that the last four methods are available if you have purchased the security package. Contact your Intermec representative for information.

- None (below)
- PEAP (page 378)
- TLS (page 382)
- TTLS (page 385)
- LEAP (page 389)

### None

Use "None" to disable 802.11b or 802.11b/g Security and enable either WEP or WPA-PSK encryption.

### To Disable 802.1x Security

- 1 Set 8021x Security as "None."
- 2 Set Association to "Open."
- **3** Set Encryption to "None."



### **To Enable WEP Encryption**

- 1 Set 8021x Security as "None."
- **2** Set **Association** to either "Open" if WEP keys are not required; or "Shared" when WEP keys are required for association.
- **3** Set Encryption to "WEP." See page 372 for information about WEP encryption.
- **4** If you had set **Association** to "Shared," then select a data transmission key from the **Data TX Key** drop-down list near the bottom of this screen, then enter the encryption key for that data transmission in the appropriate **Key** # field.



### To Enable WPA Encryption Using a Preshared Key

- 1 Set 8021x Security as "None."
- **2** Set **Association** to "WPA." See page 372 for information about WPA encryption.
- **3** Skip Encryption as it is automatically set to "TKIP." See page 372 for more information about TKIP.
- **4** Enter the temporal key as ASCII (12345) in the **Pre-Shared Key** field.



### **PEAP** (Protected EAP)

This protocol is suitable for performing secure authentication against Windows domains and directory services. It is comparable to EAP-TTLS (see page 385), both in its method of operation and its security, though not as flexible. This does not support the range of inside-the-tunnel authentication methods supported by EAP-TTLS. Microsoft and Cisco both support this protocol.

Use "PEAP" to configure the use of PEAP as an authentication protocol and to select "Open," "WPA," or "Network EAP" as an association mode.

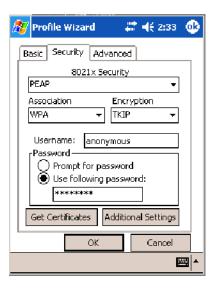
### To Enable PEAP with an Open Association

- 1 Set 8021x Security as "PEAP."
- **2** Set Association to "Open."
- **3** Skip Encryption as it is automatically set to "WEP." See page 372 for information about WEP encryption.
- 4 Enter your unique user name and password to use this protocol. Select **Prompt for password** to have the user enter this password each time to access the protocol; or leave **Use following password** as selected to automatically use the protocol without entering a password.
- **5** Tap **Get Certificates** to obtain or import server certificates. See page 388 for more information.
- **6** Tap **Additional Settings** to assign an inner PEAP authentication and set options for server certificate validation and trust. See page 381 for more information.



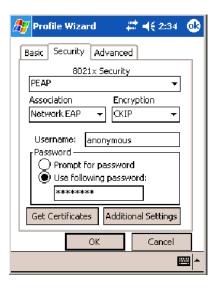
### To Enable PEAP with WPA Encryption

- 1 Set 8021x Security as "PEAP."
- **2** Set **Association** to "WPA." See page 372 for information about WPA encryption.
- **3** Skip Encryption as it is automatically set to "TKIP." See page 372 for more information about TKIP.
- 4 Enter your unique user name and password to use this protocol. Select **Prompt for password** to have the user enter this password each time to access the protocol, or leave **Use following password** as selected to automatically use the protocol without entering a password.
- **5** Tap Get Certificates to obtain or import server certificates. See page 388 for more information.
- **6** Tap Additional Settings to assign an inner PEAP authentication and set options for server certificate validation and trust. See page 381 for more information.



### To Enable PEAP with Network EAP

- 1 Set 8021x Security as "PEAP."
- **2** Set **Association** to "Network EAP." See page 371 for information about EAP.
- **3** Set Encryption to either "WEP" or "CKIP." See page 371 for information about CKIP and page 372 for information about WEP encryption.
- 4 Enter your unique user name and password to use this protocol. Select **Prompt for password** to have the user enter this password each time to access the protocol, or leave **Use following password** as selected to automatically use the protocol without entering a password.
- **5** Tap Get Certificates to obtain or import server certificates. See page 388 for more information.
- **6** Tap Additional Settings to assign an inner PEAP authentication and set options for server certificate validation and trust. See page 381 for more information.

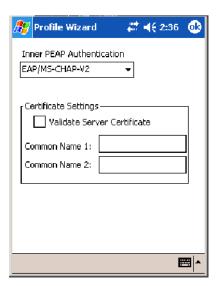


### **Additional Settings**

1 Select an authentication method from the Inner PEAP Authentication drop-down list.

EAP/MS-CHAP-V2	Authenticates against a Windows Domain Controller and other non-Windows user databases. This is Microsoft's implementation of PEAP.
EAP/Token Card	Use with token cards. The password value entered is never cached. This is Cisco's implementation of PEAP.
EAP/MD5-Challenge	Message Digest 5. A secure hashing authentication algorithm.

- **2** Check **Validate Server Certificate** to verify the identity of the authentication server based on its certificate when using TTLS or PEAP.
- **3** Enter the Common Names of trusted servers. Note that if these fields are left blank, the server certificate trust validation is not performed or required.
- **4** Click **ok** to return to the Security page.



### TLS (EAP-TLS)

EAP-TLS is a protocol that is based on the TLS (Transport Layer Security) protocol widely used to secure web sites. This requires both the user and authentication server have certificates for mutual authentication. While cryptically strong, this requires corporations that deploy this to maintain a certificate infrastructure for all their users.

Use "TLS" to configure the use of EAP-TLS as an authentication protocol, and select either "Open" or "WPA" as an association mode.

### To Enable TLS with an Open Association

- 1 Set 8021x Security as "TLS."
- **2** Set **Association** to "Open."
- **3** Skip Encryption as it is automatically set to "WEP." See page 372 for information about WEP encryption.
- **4** Enter your unique **Subject Name** and **User Name** to use this protocol.
- **5** Tap **Get Certificates** to obtain or import server certificates. See page 388 for more information.
- **6** Tap **Additional Settings** to set options for server certificate validation and trust. See page 384 for more information.



### To Enable TLS with WPA Encryption

- 1 Set 8021x Security as "TLS."
- **2** Set **Association** to "WPA." See page 372 for information about WPA encryption.
- **3** Skip Encryption as it is automatically set to "TKIP." See page 372 for more information about TKIP.
- **4** Enter your unique **Subject Name** and **User Name** as credentials for this profile.
- **5** Tap **Get Certificates** to obtain or import server certificates. See page 388 for more information.
- **6** Tap **Additional Settings** to set options for server certificate validation and trust. See page 384 for more information.



### **Additional Settings**

- 1 Check Validate Server Certificate to verify the identity of the authentication server based on its certificate when using PEAP or TLS.
- **2** Enter the Common Names of trusted servers. Note that if these fields are left blank, the server certificate trust validation is not performed or required.
- **3** Click **ok** to return to the Security page.



### TTLS (EAP-Tunneled TLS)

This protocol provides authentication like EAP-TLS (see page 382) but does not require certificates for every user. Instead, authentication servers are issued certificates. User authentication is done using a password or other credentials that are transported in a securely encrypted "tunnel" established using server certificates.

EAP-TTLS works by creating a secure, encrypted tunnel through which you present your credentials to the authentication server. Thus, inside EAP-TTLS there is another *inner authentication protocol* that you must configure via Additional Settings.

Use "TTLS" to configure the use of EAP-TTLS as an authentication protocol, and select either "Open" or "WPA" as an association mode.

### To Enable TTLS with an Open Association (default configuration)

- 1 Set 8021x Security as "TTLS."
- **2** Set Association to "Open."
- **3** Skip Encryption as it is automatically set to "WEP." See page 372 for information about WEP encryption.
- 4 Enter your unique user name and password to use this protocol. Select **Prompt for password** to have the user enter this password each time to access the protocol, or leave **Use following password** as selected to automatically use the protocol without entering a password.
- **5** Tap Get Certificates to obtain or import server certificates. See page 388 for more information.
- **6** Tap **Additional Settings** to assign an inner TTLS authentication and an inner EAP, and set options for server certificate validation and trust. See page 387 for more information.



### To Enable TTLS with WPA Encryption

- 1 Set 8021x Security as "TTLS."
- **2** Set **Association** to "WPA." See page 372 for information about WPA encryption.
- **3** Skip Encryption as it is automatically set to "TKIP." See page 372 for more information about TKIP.
- 4 Enter your unique user name and password to use this protocol. Select Prompt for password to have the user enter this password each time to access the protocol, or leave Use following password as selected to automatically use the protocol without entering a password.
- **5** Tap Get Certificates to obtain or import server certificates. See page 388 for more information.
- **6** Tap **Additional Settings** to assign an inner TTLS authentication and an inner EAP, and set options for server certificate validation and trust. See page 387 for more information.



### **Additional Settings**

1 Select an authentication protocol from the **Inner TTLS Authentication** drop-down list:

PAP	Password Authentication Protocol. A simple authentication protocol that sends security information in the clear.	
СНАР	Challenge Handshake Authentication Protocol. Use of Radius to authenticate a terminal without sending security data in the clear. Authenticates against non-Windows user databases. You cannot use this if authenticating against a Windows NT Domain or Active Directory.	
MS-CHAP; MS-CHAP-V2	Authenticates against a Windows Domain Controller and other non-Windows user databases.	
PAP/Token Card	Use with token cards. The password value entered is never cached.	
EAP	Extensible Authentication Protocol. See page 371 for information about EAP.	

- **2** If you select "EAP" for the inner authentication protocol, then select an inner EAP protocol from the **Inner EAP** drop-down list.
- **3** Enter the Common Names of trusted servers. Note that if these fields are left blank, the server certificate trust validation is not performed or required.
- **4** Check **Validate Server Certificate** to verify the identity of the authentication server based on its certificate when using TTLS, PEAP, and TLS.
- **5** Enter the **Anonymous EAP-TTLS Name** as assigned for public usage. Use of this outer identity protects your login name or identity.
- **6** Click **ok** to return to the Security page.



### **To Get Certificates**

Certificates are pieces of cryptographic data that guarantee a public key is associated with a private key. They contain a public key and the entity name that owns the key. Each certificate is issued by a certificate authority.

Use this page to import a certificate onto the 700 Color Computer.

### **Root Certificates**

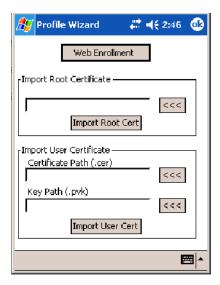
- **1** Tap the <<< button next to the **Import Root Certificate** field to select the root certificate (DER-encoded .CER file) to import.
- **2** Click **Import Root Cert** to install the selected certificate.

### **User Certificate**

- 1 Tap the <<< button next to the Certificate Path field to select the user certificate (DER-encoded .CER file without the private key) to import.
- 2 Tap the <<< button next to the **Key Path** field to select the private key (.PVK file) which corresponds to the user certificate chosen in step 1.
- **3** Tap **Import User Cert** to install the selected certificate.

### **Web Enrollment**

Tap **Web Enrollment** to obtain a user certificate over the network from an IAS Server. Tap **ok** to return to the Security page.



### **LEAP** (Cisco Lightweight EAP)

LEAP is the Cisco Lightweight version of EAP. See page 371 for information about EAP.

Use "LEAP" to configure the use of LEAP as an authentication protocol, select "Open," "WPA," or "Network EAP" as an association mode, or assign Network EAP. *Note that this defaults to the Network EAP*.

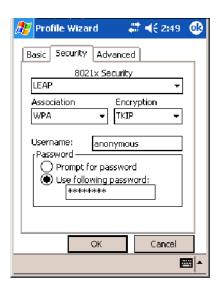
### To Enable LEAP with an Open Association

- 1 Set 8021x Security as "LEAP."
- **2** Set Association to "Open."
- **3** Skip Encryption as it is automatically set to "WEP." See page 372 for information about WEP encryption.
- **4** Enter your unique User Name to use this protocol.
- 5 Select Prompt for password to have the user enter this password each time to access the protocol, or leave Use following password as selected to automatically use the protocol without entering a password.



### To Enable LEAP with WPA Encryption

- 1 Set 8021x Security as "LEAP."
- **2** Set **Association** to "WPA." See page 372 for information about WPA encryption.
- **3** Skip Encryption as it is automatically set to "TKIP." See page 372 for more information about TKIP.
- **4** Enter your unique User Name to use this protocol.
- 5 Select Prompt for password to have the user enter this password each time to access the protocol, or leave Use following password as selected to automatically use the protocol without entering a password.



### To Enable LEAP with Network EAP

- 1 Set 8021x Security as "LEAP."
- **2** Set **Association** to "Network EAP," an EAP protocol for the network. See page 371 for information about EAP.
- **3** Set Encryption to either "WEP" or "CKIP." See page 371 for information about CKIP and page 372 for information about WEP encryption.
- **4** Enter your unique User Name to use this protocol.
- **5** Select **Prompt for password** to have the user enter this password each time to access the protocol, or leave **Use following password** as selected to automatically use the protocol without entering a password.



### **Advanced**

Use this page to configure additional settings for this profile. Tap **ok** or **OK** to return to the Profiles page.

### • Detect Rogue APs:

Wireless NICs and APs associate based on the SSID configured for the NIC. Given an SSID, the BSSID with the strongest signal is often chosen for association. After association, 802.1x authentication may occur and during authentication credentials to uniquely identify a user — these are passed between the NIC and the AP.

The base 802.1x technology does not protect the network from "rogue APs." These can mimic a legitimate AP to authentication protocols and user credentials. This provides illegal users ways to mimic legitimate users and steal network resources and compromise security.

Check this box to detect and report client behavior suspected of being rogue APs. Once a rouge AP is detected, your 700 Series Computer no longer associates with that AP until you perform a warm boot.

Clear this box to solve AP connection problems that result when an AP gets put on the rogue AP list due to inadvertant failed authentications and *not* because it is a *real* rouge.

### • Enable mixed cell:

Mixed cell is a profile-dependent setting. If enabled, you can connect to mixed cell without using WEP, then you can query the cell to determine whether you can use encryption.

### Enable Logging:

Check this box to log what activity incurs for this profile.

# **Other Configurable Parameters**

The following parameters can be configured by sending reader commands through the network or from an application. See "*Using Reader Commands*" on page 394 for more information.

Reader Command	Description	Options
Audio Volume	Changes the volume of all audio signals.	<ul> <li>Off</li> <li>Very quiet</li> <li>Quiet</li> <li>Normal (default)</li> <li>Loud</li> <li>Very loud</li> </ul>
Automatic Shutoff	Sets the length of time the 700 Color Computer remains on when there is no activity. When you turn on the 700 Color Computer, it either resumes exactly where it was when you turned it off or boots and restarts your application.	<ul> <li>1 minute</li> <li>2 minutes</li> <li>3 minutes (default)</li> <li>4 minutes</li> <li>5 minutes</li> </ul>
Backlight Timeout	Sets the length of time that the display backlight remains on. If you set a longer timeout value, you use the battery power at a faster rate.	10 10 seconds 30 30 seconds 60 1 minute (default) 120 2 minutes 180 3 minutes 240 4 minutes 300 5 minutes
Date/Time	Sets the current date and time.	Date Year 0000–9999 (1999)  Month 1–12 (6)  Day 1–31 (1)  Time Hour 0–23 (0)  Minute 0–59 (00)  Second 0–59 (00)
Key Clicks	Enables or disables the keypad clicks. The 700 Color Computer emits a click each time you press a key or decode a row of a two-dimensional symbology.	<ul><li>0 Disable clicks</li><li>1 Enable soft key clicks</li><li>2 Enable loud key clicks (default)</li></ul>

# **Using Reader Commands**

After the 700 Color Computer is connected to your network, you can send the 700 Color Computer a reader command from an application to perform a task, such as changing the time and date. Some reader commands temporarily override the configuration settings and some change the configuration settings.

### **Change Configuration**

The Change Configuration command must precede any configuration command. If you enter a valid string, the 700 Color Computer configuration is modified and the computer emits a high beep. To send the Change Configuration command through the network, use the \$+ [command] syntax where command is the two-letter command syntax for the configuration command followed by the value to be set for that command.

You can also make changes to several different commands by using the \$+ [command]...[command n] syntax. There are seven configuration command settings that can be changed in this way. See each command for information on respective acceptable "data" values.

Command	Syntax
Audio Volume	BVdata
Automatic Shutoff	EZdata
Backlight Timeout	DFdata
Key Clicks	KCdata
Virtual Wedge Grid	AFdata
Virtual Wedge Postamble	AEdata
Virtual Wedge Preamble	ADdata



Note: See pages 344 and 346 for more information about the Virtual Wedge Postamble and Virtual Wedge Preamble commands.

### Example 1

To change the Beep Volume to Off, you can send this string to the 700 Color Computer through the network: \$+BV0 where:

\$+ Indicates Change Configuration.
 BV Specifies the Audio Volume parameter.
 0 Specifies a value of Off.

### Example 2

To change the Beep Volume to Very Quiet and the Virtual Wedge Grid to 123: \$+BV1AF123

where:

\$+ Indicates Change Configuration
 BV1 Specifies Audio Volume, set to Very Quiet (1)
 AF123 Specifies Virtual Wedge Grid, set to a value of 123.

### **Set Time and Date**

This command sets the date and time on the 700 Color Computer. The default date and time is *June 1, 1999 at 12:00 AM*.

From the network, send the following:

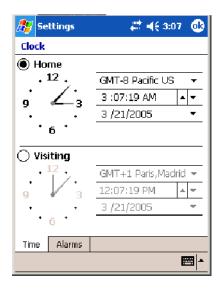
### /+ yyyymmddhhmmss

where acceptable values for the date are:

УУУУ	0000–9999	Year
mm	01–12	Month of the year
dd	01-31	Day of the month
hh	00-23	Hour
mm	00-59	Minutes
SS	00-59	Seconds



You can also set the time and date by using Configuration Management in Unit Manager, or by using the Clock control panel applet in the Settings menu. To access this control panel applet, tap **Start** > **Settings** > the **System** tab > the **Clock** icon to access its control panel applet.



# **Configuration Bar Codes**

You can change some settings on your 700 Color Computer by scanning the following Code 39 bar code labels.

• You can use the Data Collection control panel to set the three Virtual Wedge parameters (*starting on page 343*).



**Note**: When you use a bar code creation utility to make a scannable bar code label, the utility probably adds opening and closing asterisks automatically. Asterisks are included here for translation purposes.

### **Audio Volume**



Note: The Audio Volume parameter information is on page 393.

Turn Audio Off



\*\$+BV0\*

Set Audio Volume to very quiet



\*\$+VB1\*

Set Audio Volume to quiet



\*\$+BV2\*

Set Audio Volume to normal (default)



\*\$+BV3\*

Set Audio Volume to loud



\*\$+BV4\*

Set Audio Volume to very loud



\*\$+BV5\*

### **Automatic Shutoff**



Note: The Automatic Shutoff parameter information is on page 393.

Set Automatic Shutoff to 1 minute



\*\$+EZ1\*

Set Automatic Shutoff to 2 minutes



\*\$+EZ2\*

Set Automatic Shutoff to 3 minutes (default)



\*\$+EZ3\*

Set Automatic Shutoff to 4 minutes



\*\$+EZ4\*

Set Automatic Shutoff to 5 minutes



\*\$+EZ5\*

### **Backlight Timeout**



Note: The Backlight Timeout parameter information is on page 393.

Backlight Timeout 10 seconds



\*\$+DF10\*

Backlight Timeout 30 seconds



\*\$+DF30\*

Backlight Timeout 1 minute (default)



\*\$+DF60\*

### Appendix A — Configurable Settings

Backlight Timeout 2 minutes



\*\$+DF120\*

**Backlight Timeout 3 minutes** 



\*\$+DF180\*

**Backlight Timeout 4 minutes** 



\*\$+DF240\*

**Backlight Timeout 5 minutes** 



\*\$+DF300\*

# **Key Clicks**



Note: The Key Clicks parameter information is on page 393.

Disable key clicks



\*\$+KC0\*

Enable soft key clicks



\*\$+KC1\*

Enable loud key clicks (default)



\*\$+KC2\*

## Virtual Wedge Grid, Preamble, Postamble

The following parameters are user-configurable strings. Refer to a full ASCII chart for more information.

### Grid

For Virtual Wedge Grid, the first part of the bar code would be the following, which can include a string of up to 240 characters. *Parameter information starts on page 348*.



\*\$+AF

### **Preamble**

For Virtual Wedge Preamble, the first part of the bar code would be below, followed by a string of up to 31 characters (no < NUL >) and an asterisk. Default is no characters. Parameter information is on page 344.



\*\$+AD

### **Postamble**

For Virtual Wedge Postamble, the first part of the bar code would be below, followed by a string of up to 31 characters (no <NUL>) and an asterisk. Default is no characters. Parameter information is on page 346.



\*\$+AE

### Appendix A — Configurable Settings

# **B** Troubleshooting

This appendix contains a brief explanation of what you can do to trouble-shoot your 700 Color Mobile Computer.

### The 700 Color Computer does not respond when connected to a power supply.

Make sure your ac adapter or dock is plugged in and is making good contact with your mobile computer.

### The 700 Color Computer will not turn on when the I/O key is pressed.

The battery may be low and need recharging.

### The Power status LED starts blinking.

The Power status LED provides the status of your battery pack when it is connected to a charging device. See Chapter 1, "Introduction," for LED descriptions.

### The 700 Color Computer will not turn on and the screen is blank.

The battery may be *critically low*. Put your mobile computer in a charger for at least five minutes, remove it from the dock, then press the I/O button. The screen should turn on.

Put your mobile computer back in the charger and continue to charge your mobile computer for at least four hours to make sure the battery is fully charged.

### The 700 Color Computer will not turn on when placed in a dock.

Make sure the dock is plugged in and your mobile computer is securely seated in the dock.

### The 700 Color Computer shuts down during operation.

You may have a very low battery. Put the battery in a charger.

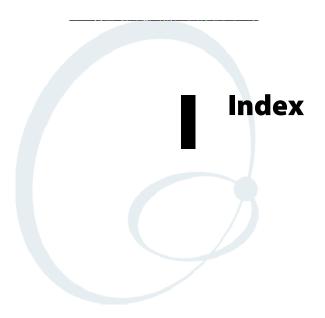
### The 700 Color Computer does not turn off.

The 700 Color Computer does not turn off while it is processing data. If this situation continues for a long period of time, it does run down the battery unless you have it connected to an external power, or in a charging dock.

Should the 700 Color Computer lock up, perform a warm-boot. If the mobile computer stays locked up, then perform a cold-boot. See Chapter 1, "Introduction," for information on performing warm-boots and cold-boots.

# The 700 Color Computer takes a long time to boot up after a warm- or cold-boot is performed.

Normal recovery time is 30 to 45 seconds. If the recovery takes longer, then contact support personnel.



The Classes and Functions Index covers classes and functions for the 700 Series Color Mobile Computer.

The General Index covers all topics. Those in italics are figures, those in bold are tables.

The Files Index is to assist you in locating descriptions for device drivers, applications, utilities, batch files, or other files within this publication.

Classes and Functions	Cold boot, IOCTL_HAL_COLDBOOT, 260
	ConfigureProfile(), 283 Copyfiles, [DefaultInstall], 228
A	[CopyFiles], file_list_section
	destination_filename, 231
add_registry_section, [AddReg]	flags, 231
flags, 232	source_filename, 231
registry_root_string, 232	CreateEvent(), 294
value_name, 232 AddReg, [DefaultInstall], 228	CreateFile()
[AddReg], add_registry_section	DTR printing, 204, 205
flags, 232	IrDA printing, 198
registry_root_string, 232	NPCP printing, 199, 200
value_name, 232	
AddWep(), 278	D
ANT_DIVERSITY, GetDiversity(), 271	[DefaultInstall]
ANT_PRIMARY, GetDiversity(), 271	AddReg, 228
ANT_SECONDARY, GetDiversity(), 271	CESelfRegister, 228
AppName, [CEStrings], 225	CESetupDLL, 228
Asset management, DeviceURL parameter, 243	CEShortcuts, 228
	Copyfiles, 228
В	Deprecated functions, 288
Basic connect/disconnect functions, 268	DeregisterDevice(), 199
BlockSize, FTP Server, 243	DTR printing, 204
BuildMax, [CEDevice], 226	[DestinationDirs], file_list_section, 230
BuildMin, [CEDevice], 226	DeviceIOControl(), 266
C	DTR printing, 204
[CEDevice]	NPCP printing, 199
BuildMax, 226	DeviceIoControl(), NPCP printing, 200, 201
BuildMin, 226	Device UPL FTP Server, 243
ProcessorType, 226	DeviceURL, FTP Server, 243
UnsupportedPlatforms, 226	disk_ordinal, [SourceDiskNames], 228
VersionMax, 226	DllRegisterServer, 228 DllUnregisterServer, 228
VersionMin, 226	Diffolitegisterserver, 220
CESelfRegister, [DefaultInstall], 228	E
CESetupDLL, [DefaultInstall], 228	EnableSuppLogging(), 287
CEShortcuts, [DefaultInstall], 228	EnableWep(), 278
[CEShortcuts], shortcut_list_section	EnableZeroConfig(), 284
shortcut_filename, 233	EncryptionStatus(), 279
shortcut_type_flag, 233	EncryptWepKeyForRegistry(Deprecated), 288
target_file/path, 233	ERROR_INSUFFICIENT_BUFFER
target_file_path, 233	IOCTL_HAL_ITC_READ_PARM, 251
CESignature	IOCTL_HAL_ITC_WRITE_SYSPARM, 255
[SourceDiskNames], 228	ERROR_INVALID_PARAMETER
[Version], 224	IOCTL_HAL_ITC_READ_PARM, 251
[CEStrings]	IOCTL_HAL_ITC_WRITE_SYSPARM, 255
AppName, 225	F
InstallDir, 225 ClassID field values	File Transfer Protocol. See FTP
VN_CLASS_ASIC, 252	file_list_section
VN_CLASS_BOOTSTRAP, 252	[CopyFiles]
VN_CLASS_KBD, 252	destination_filename, 231
CloseHandle()	flags, 231
DTR printing, 204, 205	source_filename, 231
IrDA printing, 198	[DestinationDirs], 230
NPCP printing, 199, 200	filename, [SourceDiskFiles], 229
r r r	, [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

FTP	GetCCXStatus(), 277
client, 245	GetCurrentDriverName(), 287
configurable parameters, 243	GetDiversity(), 271
BlockSize, 243	GetLinkSpeed(), 271
DeviceName, 243	GetMac(), 272
DeviceURL, 243	GetMedia(Deprecated), 288
PauseAtStartup, 244	GetMedium(Deprecated), 288
FTPDCMDS subdirectory, 247	GetNetworkMode(), 272
RTC 959, 247	GetNetworkType(), 273
server, 245	GetNicStats(Deprecated), 288
server requests	GetPowerMode(), 274
CDUP, 245	GetRadioIpAddress(), 277
CWD, 245	GetRSSI(), 274
DELE, 245	GetRTSThreshold(Deprecated), 288
HELP, 245	GetSSID(), 273
LIST, 245	GetTXPower(), 275
MKD, 245	GetWepStatus(), 276
MODE, 245	н
NLST, 245	
NOOP, 245	HAL, verion of Pocket PC
PASS, 245	IOCTL_HAL_GET_BOOTLOADER_VERINFO,
PWD, 245	259
QUIT, 245	IOCTL_HAL_GET_OAL_VERINFO, 258
RETR, 245	Helper functions, 283
RMD, 245	1
RNFR, 245	•
RNTO, 245	ID field values
SITE, 246	IOCTL_HAL_ITC_READ_PARM
SITE ATTRIB, 246	ITC_NVPARM_80211_INSTALLED, 253
SITE BOOT, 246	ITC_NVPARM_80211_RADIOTYPE, 253
	ITC_NVPARM_ANTENNA_DIVERSITY, 252
SITE COPY, 246	ITC_NVPARM_BLUETOOTH_INSTALLED,
SITE EKEY, 247	254
SITE EVAL, 247	ITC_NVPARM_CONTRAST, 252
SITE EXIT, 246	ITC_NVPARM_DISPLAY_TYPE, 252
SITE GVAL, 247	ITC_NVPARM_ECN, 252
SITE HELP, 246	ITC_NVPARM_EDBG_SUBNET, 252
SITE KILL, 246	ITC_NVPARM_EDG_IP, 252
SITE LOG, 246	ITC_NVPARM_ETHERNET_ID, 251
SITE PLIST, 246	ITC_NVPARM_INTERMEC_DATACOLLEC-
SITE PVAL, 247	TION_HW, 253
SITE RUN, 246	ITC_NVPARM_INTERMEC_DATACOLLEC-
SITE STATUS, 247	TION_SW, 253
SITE TIMEOUT, 247	ITC_NVPARM_INTERMEC_SOFT-
STOR, 245	WARE_CONTENT, 252
SYST, 245	ITC_NVPARM_LAN9000_INSTALLED, 254
TYPE, 245	ITC_NVPARM_MANF_DATE, 251
USER, 245	
XCUP, 245	ITC_NVPARM_MCODE, 252
XCWD, 245	ITC_NVPARM_RTC_RESTORE, 253
XMKD, 245	ITC_NVPARM_SERIAL_NUM, 251
XPWD, 245	ITC_NVPARM_SERIAL2_INSTALLED, 254
XRMD, 245	ITC_NVPARM_SERVICE_DATE, 251
stopping server from application, 248	ITC_NVPARM_SIM_PRO-
support, 245	TECT_HW_INSTALLED, 254
web browsers, 247	ITC_NVPARM_SIM_PRO-
neo otomocio, a 1/	TECT_SW_INSTALLED, 254
G	ITC_NVPARM_VERSION_NUMBER, 252
G	ITC_NVPARM_VIBRATE_INSTALLED, 254
GetAssociationStatus(), 269	ITC_NVPARM_WAN_FREQUENCY, 253
GetAuthenticationMode(), 270	ITC_NVPARM_WAN_INSTALLED, 253
GetBSSID(), 270	ITC_NVPARM_WAN_RADIOTYPE, 253

ITC NVPARM WAN RI, 252	ITC_DEVID_WANRADIO_SIEMENS_MC46, 253
IOCTL_HAL_ITC_WRITE_SYSPARM	ITC_DEVID_WANRADIO_SIERRA_SB555, 253
ITC_DOCK_SWITCH, 256	ITC_DEVID_WANRADIO_XIRCOM_GEM3503, 253
ITC_ WAKEUP_MASK, 256	ITC_IFTP_STOP, 248
ITC_AMBIENT_FRONTLIGHT, 256	ITC_NVPARM_80211_INSTALLED, 253
ITC_AMBIENT_KEYBOARD, 256	ITC_NVPARM_80211_RADIOTYPE, 253
ITC_REGISTRY_SAVE_ENABLE, 256	ITC_NVPARM_ANTENNA_DIVERSITY, 252
IDNA	ITC_NVPARM_BLUETOOTH_INSTALLED, 254
DeviceName, 243	ITC_NVPARM_CONTRAST, 252
DeviceURL, 243	ITC_NVPARM_DISPLAY_TYPE, 252
IDNATarget, FTP Server, 244	ITC_NVPARM_ECN, 252
InstallDir, [CEStrings], 225	ITC_NVPARM_EDBG_SUBNET, 252
Intermec Device Network Announcement. See IDNA	ITC_NVPARM_EDG_SUBNET, 2)2 ITC_NVPARM_EDG_IP, 252
IOCTL_GET_CPU_ID, 265	ITC_NVPARM_EDG_IF, 232 ITC_NVPARM_ETHERNET_ID, 251
IOCTL_HAL_COLDBOOT, 260, 291	ITC_NVPARM_INTERMEC_DATACOLLEC-
IOCTL_HAL_GET_BOOT_DEVICE, 262	TION_HW, 253
IOCTL_HAL_GET_BOOTLOADER_VERINFO, 259	ITC_NVPARM_INTERMEC_DATACOLLEC-
IOCTL_HAL_GET_DEVICE_INFO, 250	TION_SW, 253
IOCTL_HAL_GET_DEVICEID, 257	ITC_NVPARM_INTERMEC_SOFTWARE_CON-
IOCTL_HAL_GET_OAL_VERINFO, 258	TENT, 252
IOCTL_HAL_GET_RESET_INFO, 261	ITC_NVPARM_LAN9000_INSTALLED, 254
IOCTL_HAL_ITC_READ_PARM, 251	ITC_NVPARM_MANF_DATE, 251
IOCTL_HAL_ITC_WRITE_SYSPARM, 255	ITC_NVPARM_MCODE, 252
IOCTL_HAL_REBOOT, 263, 291	ITC_NVPARM_RTC_RESTORE, 253
IOCTL_HAL_WARMBOOT, 260, 291	ITC_NVPARM_SERIAL_NUM, 251
IOCTL_LOAD_NDIS_MINIPORT, 266	ITC_NVPARM_SERIAL2_INSTALLED, 254
IOCTL_PROCESSOR_INFORMATION, 264	ITC_NVPARM_SERVICE_DATE, 251
IOCTL_UNLOAD_NDIS_MINIPORT, 266	ITC_NVPARM_SIM_PROTECT_HW_INSTALLED,
isDHCPEnabled(), 286	254
isOrinoco(), 284	ITC_NVPARM_SIM_PROTECT_SW_INSTALLED,
isSupplicantRunning(), 285	254
isZeroConfigEnabled(), 284	ITC_NVPARM_VERSION_NUMBER, 252
ITC_ DOCK_SWITCH, 256	ITC_NVPARM_VIBRATE_INSTALLED, 254
ITC_ WAKEUP_MASK, 256	ITC_NVPARM_WAN_FREQUENCY, 253
ITC_AMBIENT_FRONTLIGHT, 256	ITC_NVPARM_WAN_INSTALLED, 253
ITC_AMBIENT_KEYBOARD, 256	ITC_NVPARM_WAN_RADIOTYPE, 253
ITC_DEVID_80211RADIO_INTEL_2011B, 253	ITC_NVPARM_WAN_RI, 252
ITC_DEVID_80211RADIO_MAX values	ITC_REGISTRY_SAVE_ENABLE, 256
ITC_DEVID_80211RADIO_INTEL_2011B, 253	
ITC_DEVID_80211RADIO_NONE, 253	K
ITC_DEVID_80211RADIO_NONE, 253	KernelIoControl
ITC_DEVID_INTERMEC_EVIO, 253	IOCTL_GET_CPU_ID, 265
ITC_DEVID_INTERMEC2D_IMAGER, 253	IOCTL HAL COLDBOOT, 260, 291
ITC DEVID OEM2D IMAGER, 253	IOCTL_HAL_GET_BOOT_DEVICE, 262
ITC_DEVID_SCANHW_MAX values	IOCTL HAL GET BOOTLOADER VERINFO,
ITC_DEVID_INTERMEC_EVIO, 253	259
ITC_DEVID_INTERMEC2D_IMAGER, 253	IOCTL_HAL_GET_DEVICE_INFO, 250
ITC_DEVID_OEM2D_IMAGER, 253	IOCTL_HAL_GET_DEVICEID, 257
ITC_DEVID_SCANHW_NONE, 253	IOCTL HAL GET OAL VERINFO, 258
ITC_DEVID_SE900_LASER, 253	IOCTL_HAL_GET_RESET_INFO, 261
ITC_DEVID_SE900HS_LASER, 253	IOCTL_HAL_ITC_READ_PARM, 251
ITC_DEVID_SCANHW_NONE, 253	IOCTL_HAL_ITC_WRITE_SYSPARM, 255
ITC DEVID SE900 LASER, 253	IOCTL HAL REBOOT, 263, 291
ITC_DEVID_SE900HS_LASER, 253	IOCTL_HAL_WARMBOOT, 260, 291
ITC_DEVID_WANRADIO_NONE, 253	IOCTL_PROCESSOR_INFORMATION, 264
ITC_DEVID_WANRADIO_SIEMENS_MC45, 253	KernelloControl(), 250
	ixincho control(), 270

L	NDIS_ENCRYPTION_1_ENABLED
lpBytesReturned	EncryptionStatus(), 279
ÍOCTL_GET_CPU_ID, 265	GetWepStatus(), 276
IOCTL_HAL_GET_BOOT_DEVICE, 262	NDIS_ENCRYPTION_1_KEY_ABSENT
IOCTL_HAL_GET_BOOTLOADER_VERINFO,	EncryptionStatus(), 279
259	GetWepStatus(), 276
IOCTL_HAL_GET_DEVICE_INFO, 250	NDIS_ENCRYPTION_2_ENABLED
IOCTL_HAL_GET_DEVICEID, 257	EncryptionStatus(), 279
IOCTL_HAL_GET_OAL_VERINFO, 258	GetWepStatus(), 276
IOCTL_HAL_GET_RESET_INFO, 261	NDIS_ENCRYPTION_2_KEY_ABSENT
IOCTL_HAL_ITC_READ_PARM, 251	EncryptionStatus(), 279
IOCTL_HAL_ITC_WRITE_SYSPARM, 255	GetWepStatus(), 276
IOCTL_PROCESSOR_INFORMATION, 264	NDIS_ENCRYPTION_3_ENABLED
lpInBuf	EncryptionStatus(), 279
IOCTL_GET_CPU_ID, 265	GetWepStatus(), 276
IOCTL_HAL_COLDBOOT, 260	NDIS_ENCRYPTION_3_KEY_ABSENT
IOCTL_HAL_GET_BOOT_DEVICE, 262	EncryptionStatus(), 279
IOCTL_HAL_GET_BOOTLOADER_VERINFO,	GetWepStatus(), 276
259	NDIS_ENCRYPTION_DISABLED
IOCTL_HAL_GET_DEVICE_INFO, 250	EncryptionStatus(), 279
IOCTL_HAL_GET_DEVICEID, 257	GetWepStatus(), 276
IOCTL_HAL_GET_OAL_VERINFO, 258	NDIS_ENCRYPTION_NOT_SUPPORTED
IOCTL_HAL_GET_RESET_INFO, 261	EncryptionStatus(), 279
IOCTL_HAL_ITC_READ_PARM, 251	GetWepStatus(), 276
IOCTL_HAL_ITC_KEAD_TAKNI, 251 IOCTL_HAL_ITC_WRITE_SYSPARM, 255	NDIS_MIXED_CELL_OFF, SetMixedCellMode(), 282
IOCTL_HAL_REBOOT, 263	NDIS_MIXED_CELL_ON, SetMixedCellMode(), 282
	NDIS_NET_AUTO_UNKNOWN
IOCTL_HAL_WARMBOOT, 260	GetNetworkMode(), 272
IOCTL_PROCESSOR_INFORMATION, 264	SetNetworkMode(), 281
lpInBufSize	NDIS_NET_MODE_ESS
IOCTL_GET_CPU_ID, 265	GetNetworkMode(), 272
IOCTL_HAL_COLDBOOT, 260	SetNetworkMode(), 281
IOCTL_HAL_GET_BOOT_DEVICE, 262	NDIS_NET_MODE_IBSS
IOCTL_HAL_GET_DEVICE_INFO, 250	GetNetworkMode(), 272
IOCTL_HAL_GET_DEVICEID, 257	SetNetworkMode(), 281
IOCTL_HAL_GET_OAL_VERINFO, 258	NDIS_NET_MODE_UNKNOWN
IOCTL_HAL_GET_RESET_INFO, 261	GetNetworkMode(), 272
IOCTL_HAL_REBOOT, 263	SetNetworkMode(), 281
IOCTL_HAL_WARMBOOT, 260	NDIS_NET_TYPE_DS, GetNetworkType(), 273
lpOutBuf	
IOCTL_GET_CPU_ID, 265	NDIS_NET_TYPE_FH, GetNetworkType(), 273
IOCTL_HAL_COLDBOOT, 260	NDIS_NET_TYPE_OFDM_2_4G
IOCTL_HAL_GET_BOOT_DEVICE, 262	GetNetworkMode(), 272
IOCTL_HAL_GET_BOOTLOADER_VERINFO,	SetNetworkMode(), 281
259	NDIS_NET_TYPE_OFDM_5G
IOCTL_HAL_GET_DEVICE_INFO, 250	GetNetworkMode(), 272
IOCTL_HAL_GET_DEVICEID, 257	SetNetworkMode(), 281
IOCTL_HAL_GET_OAL_VERINFO, 258	NDIS_NET_TYPE_UNDEFINED, GetNetworkType(),
IOCTL_HAL_GET_RESET_INFO, 261	273
IOCTL_HAL_ITC_READ_PARM, 251	NDIS_NETWORK_EAP_MODE_OFF
IOCTL_HAL_ITC_WRITE_SYSPARM, 255	GetCCXStatus(), 277
IOCTL_HAL_REBOOT, 263	SetCCXStatus(), 282
IOCTL_HAL_WARMBOOT, 260	NDIS_NETWORK_EAP_MODE_ON
IOCTL_PROCESSOR_INFORMATION, 264	GetCCXStatus(), 277
84	SetCCXStatus(), 282
M	NDIS_POWER_LEVEL_1, GetTXPower(), 275
ManifestName, FTP Server, 244	NDIS_POWER_LEVEL_15, GetTXPower(), 275
N	NDIS_POWER_LEVEL_30, GetTXPower(), 275
N	NDIS_POWER_LEVEL_5, GerTXPower(), 275
nDeviceId, NLEDGetDeviceInfo, 290	NDIS_POWER_LEVEL_63, GetTXPower(), 275

NDIS_POWER_LEVEL_UNKNOWN, GetTXPower(),	IOCTL_HAL_GET_BOOTLOADER_VERINFO,
275	259
NDIS_RADIO_ASSOCIATED, GetAssocationStatus(),	IOCTL_HAL_GET_DEVICE_INFO, 250
269	IOCTL_HAL_GET_DEVICEID, 257
NDIS_RADIO_AUTH_MODE_AUTO	IOCTL_HAL_GET_OAL_VERINFO, 258
GetAuthenticationMode(), 270	IOCTL_HAL_GET_RESET_INFO, 261
SetAuthenticationMode(), 280	IOCTL_HAL_ITC_READ_PARM, 251
NDIS_RADIO_AUTH_MODE_ERROR	IOCTL_HAL_ITC_WRITE_SYSPARM, 255
GetAuthenticationMode(), 270	IOCTL_HAL_REBOOT, 263
SetAuthenticationMode(), 280	IOCTL_HAL_WARMBOOT, 260
NDIS_RADIO_AUTH_MODE_OPEN	IOCTL_PROCESSOR_INFORMATION, 264
GetAuthenticationMode(), 270	0
SetAuthenticationMode(), 280	Object store
NDIS_RADIO_AUTH_MODE_SHARED	IOCTL_HAL_COLDBOOT, 260
GetAuthenticationMode(), 270	IOCTL_HAL_REBOOT, 263
SetAuthenticationMode(), 280	IOCTL_HAL_WARMBOOT, 260
NDIS_RADIO_AUTH_MODE_WPA	Oldstyle device ID, 257
GetAuthenticationMode(), 270	OSVERSIONINFO.dwBuildNumber, 226
SetAuthenticationMode(), 280	OSVERSIONINFO.dwVersionMajor, 226
NDIS_RADIO_AUTH_MODE_WPA_NONE	OSVERSIONINFO.dwVersionMinor, 226
GetAuthenticationMode(), 270	
SetAuthenticationMode(), 280	P
NDIS_RADIO_AUTH_MODE_WPA_PSK	PauseAtStartup, FTP Server, 244
GetAuthenticationMode(), 270	pInput, NLEDSetDevice, 290
SetAuthenticationMode(), 280	Pocket PC
NDIS_RADIO_POWER_AUTO	IOCTL_HAL_GET_BOOTLOADER_VERINFO,
GetPowerMode(), 274	259
SetPowerMode(), 281	IOCTL_HAL_GET_OAL_VERINFO, 258
NDIS_RADIO_POWER_MODE_CAM	pOutput, NLEDGetDeviceInfo, 290
GetPowerMode(), 274	Processor information, IOCTL_PROCESSOR_IN-
SetPowerMode(), 281	FORMATION, 264
NDIS_RADIO_POWER_MODE_FAST_PSP	ProcessorType, [CEDevice], 226
GetPowerMode(), 274	Provider, [Version], 224
SetPowerMode(), 281	Q
NDIS_RADIO_POWER_MODE_PSP GetPowerMode(), 274	Query Information functions, 269
SetPowerMode(), 2/4 SetPowerMode(), 281	R
NDIS_RADIO_POWER_UNKNOWN	RadioConnect(), 268
GetPowerMode(), 274	RadioDisassociate(), 269
SetPowerMode(), 2/4	RadioDisconnect(), 268
NDIS_RADIO_SCANNING, GetAssociationStatus(), 269	ReadFile(), NPCP printing, 199
nInBufSize	Reboot methods
IOCTL_HAL_GET_BOOTLOADER_VERINFO,	IOCTL_HAL_COLDBOOT, 291
259	IOCTL_HAL_REBOOT, 291
IOCTL_HAL_ITC_READ_PARM, 251	IOCTL_HAL_WARMBOOT, 291
IOCTL_HAL_ITC_WRITE_SYSPARM, 255	RegFlushKey(), 129, 241
IOCTL_PROCESSOR_INFORMATION, 264	RegisterDevice(), 199
nInfold, NLEDGetDeviceInfo, 290	DTR printing, 204
NLED_COUNT_INFO, NLEDGetDeviceInfo, 290	Registry
NLED_SETTINGS_INFO_ID, NLEDGetDeviceInfo,	FTP Server parameters, 243
290	save location, IOCTL_HAL_ITC_WRITE_SYSPARM
NLED_SUPPORTS_INFO_ID, NLEDGetDeviceInfo,	255
290	RegOpenKeyEx(), 293
NLEDGetDeviceInfo, 290	RegQueryValueEx(), 293
NLEDSetDevice, 290	RegSetValueEx(), 293
nOutBufSize	RemoveWep(), 283
IOCTL_GET_CPU_ID, 265	RenewDHCP(), 286
IOCTL_HAL_COLDBOOT, 260	ResetRadioToSystemSave(), 287
IOCTL_HAL_GET_BOOT_DEVICE, 262	RFC 959, 247

Root, FTP Server, 244	SwitchPacketDriver(), 288
S	SYSTEMINFO.dwProcessorType, 226
Set information functions, 278 SetAuthenticationMode(), 280 SetCCXStatus(), 282 SetChannel(), 280 SetDiversity(Deprecated), 288 SetMixedCellMode(), 282 SetNetworkMode(), 281 SetPowerMode(), 281 SetPowerMode(), 281 SetRTSThreshold(Deprecated), 288 SetSID(), 282 SetTXRate(Deprecated), 288 SHFullScreen(), 241 shortcut_list_section, [CEShortcuts] shortcut_filename, 233 shortcut_type_flag, 233	U UDP, FTPDCE, 245 UDP broadcasts, IDNATarget parameter, 244 UnsupportedPlatforms, [CEDevice], 226 UUID, 257  V [Version]
target_file/path, 233 target_file_path, 233 Signature, [Version], 224 SIM cards protection hardware, 254 protection software, 254 software installed, 254 [SourceDiskFiles], filename, 229 [SourceDiskNames] CESignature, 228 disk_ordinal, 228 SourceDisksNames.MIPS, 229 SourceDisksNames.SH3, 229 StartScanList(), 285 StartSupplicant(), 285 StopSupplicant(), 286	W  WAN radio IDs  ITC_DEVID_WANRADIO_NONE, 253  ITC_DEVID_WANRADIO_SIEMENS_MC45, 253  ITC_DEVID_WANRADIO_SIEMENS_MC46, 253  ITC_DEVID_WANRADIO_SIERRA_SB555, 253  Warm boot  IOCTL_HAL_REBOOT, 263  IOCTL_HAL_WARMBOOT, 260  Wireless TCP/IP installations, BlockSize parameter, 243  WriteFile()  DTR printing, 204, 205  IrDA printing, 198  NPCP printing, 199, 200
string_key, [Strings], 225 [Strings], string_key, 225	Xscale processor ID, IOCTL_GET_CPU_ID, 265

General Index	profile security information, WEP encryption, 376 profiles, 373 advanced settings, 392
N	basic information, 374
Numbers	security information, 375
1470 Imager. See Imager	SSID (network name), 374
1551/1553 Tethered Scanner	TTLS, WPA encryption, 386
See also Tethered scanner	WPA authentication, Zero Configuration, 111
configuring, 218	WPA authentication with pre-shared key, Zero Config-
reset to factory defaults, 221	uration, 112
troubleshooting, 221	WPA encryption, 377
1D laser scanner, about, 207	zero configuration, WEP encryption, 110
1D OmniDir Decode Enable, configuration parameter,	802.11 CF radio CORE module, 138
342	installing available modules, 137
2D Imager, about, 207	loading a module, 137
4820 printer, NPCP driver, 199	Α
6804DM printer	
DTR driver, 204	Abstract Syntax Notation.1. See ASN.1
IrDA driver, 198	Accessory list, 24
6804T printer	Accounts, via Inbox, 79
DTR driver, 204	Activation wizard
IrDA driver, 198	Phone application, CDMA radios, 150
6805A printer	phone application, CDMA radios, 144, 146
DTR driver, 204 IrDA driver, 198	ActiveSync
	ActiveSync Help, 49
6806 printer DTR driver, 204	adding programs to Start many 47
IrDA driver, 198	adding programs to Start menu, 47 Folder behavior connected to e-mail server, 78
6808 printer DTR driver, 204	installing applications, 121 Microsoft Reader, 93
IrDA driver, 198	Pocket Internet Explorer
printer support, 197	favorite links, 96
681T printer, DTR driver, 204	mobile favorites, 97
6820 printer	Mobile Favorites folder, 96
IrDA driver, 198	replicating registry settings, 124
NPCP driver, 199	Start menu icon, 31
printer support, 197	URL, 48
6920 Communications Server, ManifestName parameter,	Windows Mobile, 48
244	ActiveX control tools, unit information control panel, CAB
740 Color Computer, 292	files, 365
781 printers, DTR driver, 204	AD command, with/without data, 345
782T printer, printer support, 197	Adding bookmarks, Microsoft Reader, 95
802.11	Adding drawings to text, Microsoft Reader, 95
antenna color code, 133	Adding programs
API, 267	ActiveSync, 46
channel, 374	Pocket Internet Explorer, 46
communications setup, 135, 373	to the Start menu, 47
configuration profiles, 267	via ActiveSync, 47
CORE module, 138	via File Explorer, 47
LEAP	Windows Mobile, 45
network EAP, 391	Adjusting settings, Windows Mobile, 45
WPA encryption, 390	Adobe Acrobat Reader, URL, 184
network type, 374	AE command, with/without data, 347
PEAP	Aimer LED Duration, configuration parameter, 338
network EAP, 380	All-Day events, Calendar, 52
WPA encryption, 379	creating, 56
profile label, 374	Alpha plane on keypad, 293

Alphanumeric keypad	AXReaderCommand, 365
alpha (blue) key sequences, 16	AXVWedge, 365
[gold/white] key sequences, 14	
registry settings	В
alpha plane, 293	Backlight control panel applet
gold plane, 293	ambient light sensor, 2
unshifted plane, 293	keypad, 12
scan codes, 296	Backlight timeout
Ambient lighting, 2	bar code configuration, 393, 397
Annotations index, Microsoft Reader, 95	configuration parameter, 393
Antenna, radio type, 133	Bar code configuration
APIs	audio volume, 393
802.11, 267	automatic shutoff, 393
AT command interface, 184	backlight timeout, 393
IrSock, 198	key clicks, 393
App launch, control panel applet, 369	Bar codes
Application keys	configuration
app launch control panel applet, 369	audio volume, 396
wakeup mask control panel applet, 368	automatic shutoff, 397
Appointments	backlight timeout, 397
Calendar	Code 39, 396
adding a note, 58	key clicks, 398
assigning to a category, 60	internal scanner supported symbologies, 212
changing, 55	scanning labels, 396
creating, 55	supported symbologies, 303
deleting, 62	tethered scanner supported symbologies, 222
finding, 62	Battery
making recurring, 59	ambient lighting, 2
setting a reminder, 57	low battery conditions, 6
viewing, 53	RAM maintenance, 6
via Calendar, 50	specifications, 26
APS linear imager, about, 207	status, 5
ASCII	Battery status, unit information control panel applet, 363
printing, 198	Beeper
printing to a port, port print method, 198	configuration parameter
raw text to printer, 198	frequency, 335
ASN.1, 195	volume, 333, 334
AT command interface, 184	disabling the volume, 334
testing, 185	selecting a volume, 9
Attaching notes to text, Microsoft Reader, 95	silencing the volume, 11
Audio, phone application, 160	supported functions, 332
Audio control panel applet, input mixing, 8	volume, turning it on, 7
Audio files, Windows Media Player, 92	when not available
Audio system	beeper frequency, 335
external headset jack, 4	good read beep duration, 337
microphone, 4	good read beeps, 336
speaker, 3	Bell Mobility activation process, 183
AutoCab, command line syntax, 130	Block recognizer, Windows Mobile input panel, 36
AutoFTP, 248	Bluealps CORE module
AutoIP, 194	installing available modules, 137
Automatic Private IP. See AutoIP	loading a module, 137
Automatic shutoff	Bluetooth
bar code configuration, 393, 397	accessing, 189
configuration parameter, 393	activating, 189
Autostart FTP, 248	unit information control panel, main stack CAB file,
AvantGo channels, Pocket Internet Explorer, 98	364
AXCommunication, 365	WPport, 191
AXFileTransfer, 365	Bluetooth compatibility, network support, 189

Books, Microsoft Reader	Carrier, location of ESN, 165
adding bookmarks, 95	Categories
adding drawings, 95	calendar, 51
annotations index, 95	contacts, assigning to, 69
attaching notes, 95	CDMA/1xRTT, 140
copying, 95	activation with SB555 Watcher, 165
downloading, 93	Bell Mobility, 183
highlighting, 95	Sprint, 172
reading, 94	Telus, 183
removing, 95	Verizon, 168
searching, 95	antenna color code, 133
Browsing the Internet, Pocket Internet Explorer, 99	AT command set, 184
Build information, software, 20	copying files from web site, 162
build information, software, 20	via Microsoft ActiveSync, 163
C	
CAB files	via storage cards, 163
	CORE module, 140
after the extraction, 236	location of ESC, 165
creating, 224	phone activation, 144, 146
INF files, 224	phone application, 144
with CAB Wizard, 239	setting up, 162
installation functions, SETUP.DLL, 236	terminology, 162
placing files onto storage card, 128	CEImager
unit information control panel applet, 364	location of the executable file, 128
Cabinet Wizard	migrating AUTORUN.DAT files, 128
creating CAB files, 239	Channel, 802.11 radio module, 374
troubleshooting, 240	Cisco compatible extension specifications, 24
using the application, 224	Clock
Cabling, scanner, 216	restore real-time after cold-boot, 367
Calendar	setting date and time, 395
all day events, 52	Windows Mobile settings, 45
creating, 56	Closing drivers, NPCP, 200
appointments	Codabar, configuration parameter, 306
adding a note, 58	user ID, 325
assigning to a category, 60	Code 11, configuration parameter, 320
changing, 55	user ID, 329
creating, 55	Code 128, configuration parameter, 309
deleting, 62	FNC1 character, 311
finding, 62	user ID, 325
making recurring, 59	Code 39, configuration parameter, 304
	user ID, 325
setting a reminder, 57	Code 93, configuration parameter, 308
viewing, 53	1
categories, 51	length, 308
meetings, sending a request, 61	user ID, 326
options, changing, 63	Code Division Multiple Access. See CDMA/1xRTT
Pocket Outlook, 50	Cold boot, performing, 19
recurrence pattern, 53	COM A, modem position, 366
Start menu icon, 31	COM B, serial position, 366
synchronizing, 51	COM port
Call Guard alert, enable while roaming, Phone application,	configuration, 214
CDMA radios, 153	wedge settings, 214
Call history, Phone application	COM1, NPCP parameter, 199
CDMA radios, 149	COM1 port, 198
GSM radios, 155	Comm port wedge
Capacitor, internal super, 6	disabling, 214
Capturing thoughts and ideas, via Notes, 75	enabling, 213
Card support	error messages, 214
CompactFlash cards, 21	limitations, 216
MultiMediaCards, 21	settings, 214
radios, 23	unit information control panel, 364
Secure Digital cards, 21	Command line syntax, AutoCab, 130
O ,	, , , , , , , , , , , , , , , , , , , ,

Common Object Resource Environment. See CORE	PDF417, 314
Communications	user ID, 326
DTR, 205	plessey, 312
NPCP, 202	user ID, 327
CompactFlash cards	prefix, 330
card support, 21	QR code, 321
installing applications, 122	security
migrating applications, 128	encryption key, 356
packaging an application, 120	read encryption, 354
specifications, 25	read-only community string, 352
Computer shutdown, 6	read/write community string, 353
Configuration parameters	write encryption, 355
1D OmniDir decode enable, 342	SNMP, security subnet mask, 324
aimer LED duration, 338	standard 2 of 5, 305
automatic shutoff, 393	user ID, 327
backlight timeout, 393	sticky aimer duration, 339
beeper, 333	suffix, 331
frequency, 335	telepen, 319
volume, 334	user ID, 328
codabar, 306	trap
user ID, 325	authentication, 357
code 11, 320	threshold, 358
user ID, 329	UPC
code 128, 309	A user ID, 327
FNC1 character, 311	E user ID, 327
user ID, 325	UPC/EAN, 307
code 39, 304	virtual wedge, 343
user ID, 325	code page, 349
code 93, 308	grid, 348
length, 308	postamble, 346
user ID, 326	preamble, 344
datamatrix, 322	volume, 393
date/time, 393	Configuring service settings, Phone application, GSM ra-
EAN	dios, 157
13 user ID, 328	Connecting to
8 user ID, 328	an ISP, 100
good read	e-mail server, 116
beep duration, 337	work, 104
beeps, 336	Connecting to a mail server, via Inbox, 79
identification	Connections
contact, 359	See also Getting connected
location, 361	directly to e-mail server, 116
name, 360	ending, 116
image dimension, 340	setting up an e-mail account, 116
interleaved 2 of 5, 317	to an ISP, 100
user ID, 326	via modem, 100
key clicks, 393	to work, 104
lighting mode, 341	via modem, 105
macro PDF, 314	via VPN server, 113
matrix 2 of 5, 318	via modem
user ID, 328	to an ISP, 100
maxicode, 323	to work, 105
micro PDF417, 316	via VPN server, to work, 113
MSI, 313	via wireless network, 108
user ID, 326	Conserving battery power, 2

Contacts	Copying, contacts, 69
adding a note, 68	Copying text, Microsoft Reader, 95
adding a telephone number	CORE, 137
CDMA radios, 149	802.11 CF module, 138
GSM radios, 155	802.11 radio module
adding to speed dial, 72	details, 139
assigning to a category, 69	general, 138
changing, 67	accessing from
changing options, 72	Programs panel, 137
copying, 69	Today screen, 138
creating, 65, 67	activating, 137
deleting, 71	installing available modules, 137
finding, 71	loading a module, 137
MSN Messenger	module for 802.11 NIC, 267
managing, 90	WAN monitor, GSM/GPRS, 142
sending messages, 91	WAN radio module
working with, 89	CDMA/1xRTT, 140
Pocket Outlook, 64	general, 142
sending a message, 70	Creating
Start menu icon, 31	a modem connection
synchronizing, 65	to an ISP, 100
viewing, 66	to work, 105
Control panel applets	a VPN server connection, to work, 113
Audio, 8	a wireless network connection, 108
backlight, 2, 12	CAB files, 224
clock, 395	with CAB Wizard, 239
data collection, 302	contacts via Contacts, 65
beeper volume, 9	document via Pocket Word, 82
beeper/LED, 332	drawing via Notes, 41
imager, 338	INF files, 224
symbologies, 303	note via Notes, 75
symbology options, 324	task via Tasks, 74
vibrator, 22	workbook via Pocket Excel, 86
	WOIRDOOK VIA I OCKET LACEI, OO
virtual wedge, 343	D
intemec settings	
beeper volume, 10	Data, Phone application, CDMA radios, 152
vibrator, 23	Data collection
intermec settings, 302, 350	configuration parameters
phone settings	1D OmniDir decode enable, 342
CDMA radios, 151	aimer LED duration, 338
GSM radios, 156	beeper, 333
	beeper frequency, 335
power	
battery status, 5	beeper volume, 334
RAM maintenance, 6	codabar, 306
SNMP, 351	codabar user ID, 325
identification, 359	code 11, 320
security, 352	code 11 user ID, 329
traps, 357	code 128, 309
system, wireless network, 135, 373	code 128 FNC1 character, 311
unit information, 362	code 128 user ID, 325
	code 39, 304
battery status, 5, 363	
CAB files, 364	code 39 user ID, 325
versions, 20, 362	code 93, 308
utilities, 366	code 93 length, 308
app launch, 369	code 93 user ID, 326
dock switch, 366	datamatrix, 322
registry save, 129, 367	EAN-13 user ID, 328
wakeup mask, 368	EAN-8 user ID, 328
wireless network, 371	good read beep duration, 337
Converting writing to text, 39	good read beeps, 336

image dimension, 340	opening, 200
interleaved 2 of 5, 317	reading from, 200
interleaved 2 of 5 user ID, 326	removing, 199
lighting mode, 341	writing to, 200
macro PDF, 314	O'Neil. <i>See</i> DTR printing
matrix 2 of 5, 318	DTR printing, 204
matrix 2 of 5 user ID, 328	closing driver, 205
maxicode, 323	communications, 205
micro PDF417, 316	opening driver, 205
MSI, 313	removing driver, 204
MSI user ID, 326	writing to driver, 205
PDF417, 314	-
PDF417 user ID, 326	E
plessey, 312	E-mail account, setting up an account, 116
plessey user ID, 327	E-mail server, getting connected, 116
prefix, 330	EAN, configuration parameter, 307
QR code, 321	13 user ID, 328
standard 2 of 5, 305	8 user ID, 328
standard 2 of 5 user ID, 327	Editing a profile, 373
sticky aimer duration, 339	Edition information, 28
suffix, 331	Emails, SMS messages via Phone application
telepen, 319	CDMA radios, 150
telepen user ID, 328	GSM radios, 156
UPC-E user ID, 327	Ending a connection, 116
UPC-A user ID, 327	Environmental specifications, 25
UPC/EAN, 307	Epson Escape Sequences, 198
virtual wedge, 343	Error messages
virtual wedge code page, 349	comm port wedge, 214
virtual wedge grid, 348	tethered scanner, 214
virtual wedge postamble, 346	ESN, location on computer, 165
virtual wedge preamble, 344	Ethernet, communications setup, 134
vibrator, 22	ETSI GSM 07.05 interface specifications, 184
Data Matrix, configuration parameter, 322	ETSI GSM 07.07 interface specifications, 184
Date, setting, 395	Expansion slot specifications, 25
Date/Time, configuration parameter, 393 DHCP, 194	F
	Favorite links, Pocket Internet Explorer, 96
replicating registry settings, 123	File Explorer
Display full screen, 241 Display specifications, 24	adding programs to Start menu, 47
Dock switch, control panel applet, 366	removing programs, 47
Docks, modem support, 17	Windows Mobile, 44
DRAM	Find feature, Windows Mobile, 44
low battery shutdown, 6	Flash File Store
maintenance, 6	migrating applications, 128
Drawing mode, Pocket Word, 85	packaging an application, 120
Drawing on the screen	Flash file system, control panel applet, 367
See also Notes	Folder behavior connected to e-mail server
Pocket Word, 85	ActiveSync, 78
Drivers	IMAP4, 78
DTR	POP3, 78
communications, 205	SMS, 78
installing, 204	FRAME_NOT_ACKED, 201
opening, 205	FTP
removing, 204	configurable parameters
writing to, 205	IDNATarget, 244
NPCP	ManifestName, 244
closing, 200	Root, 244
communications, 202	heartbeat, 245
I/O controls, 201	server, installing applications, 122
installing, 199	FTPDCMDS subdirectory, FTP support, 247
C	//

Full screen display, 241	lighting mode, 341
G	maxicode, 323 QR code, 321
GDI approach, 198	sticky aimer duration, 339
General Packet Radio Service. See GSM/GPRS	settings, 215
Getting connected	supported
directly to an e-mail server, 116	beeper functions, 332
ISP, 100	functions, 338
setting up an e-mail account, 116	symbologies, 303
to an ISP, 100	symbologies not available
creating a modem connection, 100	CIP 128 French Pharmaceutical, 310
to work, 104	Code 11, 320
creating a modem connection, 105	Code 128 FNC1 character, 311
creating a VPN server connection, 113	EAN 128 ]C1, 310
creating a wireless network connection, 108	Macro PDF, 314
Windows Mobile, 100	Matrix 2 of 5, 318
	micro PDF417, 316
Gold plane on keypad, 292	and the second s
Good read, configuration parameter	Telepen, 319
beep duration, 337	symbology user IDs not available
beeps, 336	Codabar, 325
Grid data, configuration parameter, 348	Code 11, 329
GSM/GPRS, 142	Code 128, 325
antenna color code, 133	Code 39, 325
AT command set	Code 93, 326
MC45, 184	EAN 13, 328
MC46, 184	EAN 8, 328
CORE module, 142	Interleaved 2 of 5, 326
phone application, 154, 159	Matrix 2 of 5, 328
и	MSI, 326
Н	PDF417, 326
Handset	Plessey, 327
phone application, 159	Standard 2 of 5, 327
volume, 161	Telepen, 328
Headset jack, external, 4	UPC A, 327
Hiding your location from everyone except 911, CDMA	UPC E, 327
radios, 151	vibrator, enabling, 22
Highlighting text, Microsoft Reader, 95	IMAP4, Folder behavior connected to e-mail server, 78
	Inbox
I	accounts, 79
I/O controls, NPCP driver, 201	composing/sending messages, 81
Identification, configuration parameter	connecting to a mail server, 79
contact, 359	downloading messages from server, 80
location, 361	getting connected, 100
name, 360	managing e-mail messages and folders, 78
IDNA	Pocket Outlook, 77
IDNATarget, 244	Start menu icon, 31
ManifestName, 244	synchronizing e-mail messages, 77
Image dimension, configuration parameter, 340	using My Text, 43
Imager	INF files, creating, 224
beeper functions not available	Input Mixing, Audio control panel applet, 8
beeper frequency, 335	Input panel
good read beep duration, 337	block recognizer, 36
good read beeps, 336	keyboard, 35
beeper/LED parameters, beeper, 333	letter recognizer, 37
	Pocket Word, 83
control panel appet, data collection, 338	
data collection parameters	selecting typed text, 37
1D OmniDir decode enable, 342	transcriber, 37 Windows Mobile, 32
aimer LED duration, 338	
datamatrix, 322 image dimension, 340	word suggestions, 35 Installation functions, SETUP.DLL, 236
mage dimension, 740	mstanation functions, SET UF.DLL, 230

Installing applications using a storage card, 122	alphanumeric alpha (blue) key sequences, 16
using CompactFlash cards, 122	[gold/white] key sequences, 14
using Secure Digital cards, 123	scan codes, 296
with ActiveSync, 121	backlight control panel applet, 12
with FTP Server, 122	change notification, 294
Installing drivers	driver registry settings, 294
DTR, 204	numeric
NPCP, 199	alpha (blue) key sequences, 15
Instant messaging, 87	[gold] key sequences, 13
Integrated scanners. See Internal scanners	scan codes, 295
Interface specifications, ETSI GSM 07.0x, 184	planes, 292
Interleaved 2 of 5, configuration parameter, 317	remapping, 292
user ID, 326	sample registry keys, 298
Intermec part numbers, 24	specifications, 25
Intermec settings, 302, 350	L
beeper volume, 10	
vibrator, 23	Laser scanner
INTERMEC_PACKET_DRIVER, SwitchPacketDriver(),	configuration parameters, 300
288	data collection parameters
Internal scanners	beeper frequency, 335 beeper volume, 334
configuring, 210	codabar, 306
specifications, 25	codabar user ID, 325
supported symbologies, 212	code 11, 320
Internet explorer	code 11 user ID, 329
software build version, 20	code 128, 309
Windows Mobile 2003 edition, 28	code 128 FNC1 character, 311
Internet Service Provider. See ISP	code 128 user ID, 325
IOCTL_NPCP_BIND, 201	code 39, 304
IOCTL_NPCP_CANCEL, 201	code 39 user ID, 325
IOCTL_NPCP_CLOSE, 201 IOCTL_NPCP_ERROR, 201	code 93, 308
IOCTL_NPCP_FLUSH, 201	code 93 length, 308
IP address, replicating registry settings, 123	code 93 user ID, 326
IrDA printing, 198	EAN-13 user ID, 328
ISP	EAN-8 user ID, 328
connecting to via Windows Mobile, 100	good read beep duration, 337
creating, a modem connection, 100	good read beeps, 336
Pocket Internet Explorer, 96	interleaved 2 of 5, 317
Windows Mobile, 100	interleaved 2 of 5 user ID, 326
ITC_KEYBOARD_CHANGE, CreateEvent(), 294	macro PDF, 314
ITU-T interface specifications, 184	matrix 2 of 5, 318
1	matrix 2 of 5 user ID, 328
K	micro PDF417, 316
W 1 P T 1 72	MSI, 313
Keeping a to-do list, via Tasks, 73	MSI user ID, 326
Key clicks	PDF417, 314
bar code configuration, 393, 398	PDF417 user ID, 326
configuration parameter, 393	plessey, 312
Key sequences	plessey user ID, 327
alpha (blue) keys alphanumeric, 16	prefix, 330 standard 2 of 5, 305
numeric, 15	standard 2 of 5, 505 standard 2 of 5 user ID, 327
[gold] keys, numeric, 13	suffix, 331
[gold/white] keys, alphanumeric, 14	telepen, 319
Keyboard	telepen user ID, 328
See also Keypad	UPC-E user ID, 327
Windows Mobile input panel, 35	UPC-A user ID, 327
Keypad	UPC/EAN, 307
advanced remapping, 294	virtual wedge, 343
	2

virtual wedge code page, 349	OIDs, 196		
virtual wedge grid, 348	Micro PDF417, configuration parameter, 316		
virtual wedge postamble, 346	Microphone, 4		
virtual wedge preamble, 344	phone application, 159		
SNMP configuration parameters	Microprocessor, specifications, 25		
identification contact, 359	Microsoft Developer Network Library. See MSDN library		
identification location, 361	Microsoft Exchange e-mail account, 87		
identification name, 360	Microsoft Passport account, 87		
security encryption key, 356	Microsoft Reader		
security read encryption, 354	books		
security read-only community string, 352	downloading, 93		
security read/write community string, 353	reading, 94		
security subnet mask, 324	removing, 95		
security write encryption, 355	features, 95		
trap authentication, 357	adding bookmarks, 95		
trap threshold, 358	adding drawings, 95		
supported	annotations index, 95		
beeper functions, 332	attaching notes, 95		
symbologies, 303	copying text, 95		
symbologies not available	highlighting text, 95		
Datamatrix, 321	searching for text, 95		
datamatrix, 322	using the library, 93		
maxicode, 323	Windows Mobile, 93		
LEAP	Microsoft's Wireless Zero Config, 373		
802.11 radio module	Migrating applications		
network EAP, 391	Flash File Store, 128		
WPA encryption, 390	CompactFlash storage cards, 128		
profile security information, 389	Secure Digital storage cards, 128		
WEP encryption, 389	Migrating to a 700 Color Computer, 130		
LED status, 17	Mobile Favorites, Pocket Internet Explorer, 97		
battery, 17	Mobile Favorites folder, Pocket Internet Explorer, 96		
scanning keypad/shift and notification, 17	Modem position, COM A, 366		
Letter recognizer, Windows Mobile input panel, 37	Modems		
Letting your location be visible, CDMA radios, 151	creating a connection		
Library, Microsoft Reader, 93	to an ISP, 100		
Lighting Mode, configuration parameter, 341	to work, 105		
Line printing, 198	specifications, 25		
Location, Phone application, CDMA radios, 151	MP3 files, Windows Media Player, 92		
LPT9 printer device, 199	MSDN library, 248		
	MSDN Windows CE documentation, 194		
M	MSI, configuration parameter, 313		
Macro PDF, configuration parameter, 314	user ID, 326		
Managing e-mail messages and folders, via Inbox, 78	MSN Messenger		
	about, 87		
Matrix 2 of 5, configuration parameter, 318 user ID, 328	accounts		
	Microsoft Exchange e-mail, 87		
MaxiCode, configuration parameter, 323	Microsoft Passport, 87		
Meetings Calendar conding a request 61	contacts		
Calendar, sending a request, 61	managing, 90		
via Calendar, 50	sending messages, 91		
Memory and storage, specifications, 25	working with, 89		
Menus, Windows Mobile settings, 45	setting up an account, 88		
Messages	using My Text, 43		
sending to, contacts, 70	MultiMediaCards, card support, 21		
via Inbox	TI,		
composing/sending, 81	N		
downloading from server, 80			
MIBs	NDIS_SUPP_LOGGING_OFF, EnableSuppLogging(),		
ASN.1, 195	287		
files, 195	NDIS_SUPP_LOGGING_ON, EnableSuppLogging(),		
object identifier, 196	287		

NDISUIO_PACKET_DRIVER, SwitchPacketDriver(), 288	Object Store, packaging an application, 120 Opening drivers
Network adapters	DTR, 205
antenna color code, 133	NPCP, 200
Ethernet communications, 134	Operating system, specifications, 25
no networking, 135	Owner information, Windows Mobile settings, 45
wireless 802.11, 134	6-, -,
wireless printing, 189	Р
Network EAP	
LEAP security method, 391	Packaging an application
PEAP security method, 380	CompactFlash storage cards, 120
Network settings, Phone application, GSM radios, 158	Flash File Store, 120
Network type, 802.11 radio module, 374	Object Store, 120
NLED driver, vibrator, 289	Persistent Storage Manager, 120
	Secure Digital storage cards, 120
NLED_SETTINGS_INFO_ID, NLEDSetDevice, 290	Page format printing, 198
Notes	Password
adding to	Pocket Excel, 87
appointments, 58	Windows Mobile settings, 45
contacts, 68	Patent information, xxiii
creating a note, 75	PB20 printers, printer support, 197
drawing on the screen, 41	PDF417
creating a drawing, 41	about the laser scanner, 207
selecting a drawing, 41	configuration parameter, 314
Pocket Outlook, 75	user ID, 326
recording a message, 42	PEAP
Start menu icon, 31	802.11 radio module
synchronizing notes, 76	network EAP, 380
writing on the screen, 38	
alternate writing, 39	WPA encryption, 379
converting writing to text, 39	profile security information, 378
selecting the writing, 38	WEP encryption, 378
tips for good recognition, 40	Performing a cold boot, 19
NPCP printing, 199	Performing a warm boot, 19
about, 199	Persistent Storage Manager. See PSM
closing driver, 200	Phone application
COM1 parameters, 199	CDMA radios, 144
communications, 202	activation wizard, 144, 146, 150
driver I/O controls, 201	adding contact to speed dial, 72, 149
installation, 199	call history, 149
LPT9, 199	customizing phone settings, 151
opening driver, 200	enable Call Guard alert while roaming, 153
reading from driver, 200	hiding your location except from 911, 151
removal, 199	letting your location be visible for everyone, 151
sample code, 202	reset connection settings for PCS Vision, 152
unit information control panel, NPCPTEST CAB file,	sending SMS messages, 150
	toggle between automatic or Sprint roaming, 153
364	update your PCS Vision profile, 152
writing to driver, 200	view current phone settings, 153
Numeric keypad	voice mail, 150
alpha (blue) key sequences, 15	GSM radios, 154
[gold] key sequences, 13	adding contact to speed dial, 155
registry settings	call history, 155
alpha plane, 293	customizing phone settings, 156
gold plane, 293	finding, setting, selecting networks, 158
unshifted plane, 293	sending SMS messages, 156
scan codes, 295	service settings, 157
	view current phone settings, 158
0	Phone Info
O'Neil printing	Phone application, CDMA radios, 153
See also DTR printer	WAN info, GSM radios, 158
installing driver, 204	Phone jack position, control panel applet, 366

Phone settings	with/without data, 345		
control panel applet	Prefix, configuration parameter, user ID, 330		
CDMA radios, 151	Printer support, 198		
GSM radios, 156	IrDA printer driver, 198		
customizing via Phone application	NPCP printer driver, 199		
CDMA radios, 151	O'Neil printer driver, 204		
GSM radios, 156	Profile label, 802.11 radio module, 374		
network settings, GSM radios, 158	Profiles		
view current settings via Phone application, CDMA	802.11 radio module, 373		
radios, 153	advanced settings, 392		
view current settings via WAN info, GSM radios, 158	basic information, 374		
PhoneUtility, 159	security information, 375		
ring, 160	editing, 373		
vibrate, 160	Programs, adding or removing, Windows Mobile, 45		
Physical dimensions, specifications, 26	PSM		
Planes, keypad, 292	determining build version, 18		
Plessey, configuration parameter, 312	packaging an application, 120		
user ID, 327	PSM build, 302		
Pocket Excel			
about, 86	Q		
creating a workbook, 86	QR code, configuration parameter, 321		
Pocket Internet Explorer	R		
about, 96	7-		
adding programs, 46	Radios		
AvantGo channels, 98	See also Network adapters		
browsing the Internet, 99	card support, 23		
favorite links, 96	Reader commands, 394		
getting connected, 100	configuration change, 394		
mobile favorites, 97	date and time settings, 395		
Mobile Favorites folder, 96	Reading from drivers, NPCP, 200		
software build, 20	Real-Time Clock, restore after cold-boot, 367		
Start menu icon, 31	Record button, recording a message, 42		
viewing mobile favorites and channels, 99	Recording, via Notes, 42		
Pocket Outlook, 50	Recording a message, Pocket Word, 84		
Calendar, 50	Recording mode, Pocket Word, 84		
Pocket Word	Recovery CD		
about, 82	AutoCab method, 130		
creating a document, 82	AUTOUSER.DAT file, 129		
drawing mode, 85	RegFlushKey() API, 241		
recording mode, 84	S9C upgrade, 365		
synchronizing, 85	updating the system software, 127		
tips, 87	Recurrence pattern, Calendar, 53		
typing mode, 83	RegFlush utility, 129		
writing mode, 84	Registry		
POP3, Folder behavior connected to e-mail server, 78	confirm the new regisry file, 126		
Postamble	copy the REGFLUSH.CAB file, 124		
configuration parameter, 346	delete the old registry save, 124		
with/without data, 347	keypad remapping, 294		
Power	load the application, 125		
	replicating settings, 123		
control panel	sample view of key mapping, 298		
battery status, 5 RAM maintenance, 6	update other computers, 126		
specifications, 26	writing to a storage card, 129		
	Registry Save, control panel applet, 367		
Windows Mobile settings, 45 Preamble			
configuration parameter, 344			

Registry settings	backlight timeout, 393		
AutoCfg, 194	date/time, 393		
AutoFTP, 249	key clicks, 393		
AutoInterval, 194	volume, 393		
AutoIP/DHCP, 194	utilities configuration, button wakeup mask, 368		
DhcpMaxRetry, 194	Scanner cabling, 216		
DhcpRetryDialogue, 194	Scheduling appointments and meetings, via Calendar, 50		
EnableDHCP, 194			
	SDMMC Disk, 128		
keypad driver, 294	Searching for text, Microsoft Reader, 95		
keypad planes	Secure Digital cards		
alpha, 293	card support, 21		
gold, 293	installing applications, 122, 123		
unshifted, 293	migrating applications, 128		
Regulatory approvals, specifications, 26	packaging an application, 120		
Removing drivers	specifications, 25		
DTR, 204	Security, configuration parameter		
NPCP, 199	encryption key, 356		
Removing programs, Windows Mobile, 45, 47	read encryption, 354		
Replicating registry settings, 123	read-only community string, 352		
Reset button, 19	read/write community string, 353		
Reset connection settings for PCS Vision, Phone applica-	subnet mask, 324		
tion, CDMA radios, 152	write encryption, 355		
Roaming, toggle between automatic or Sprint, Phone ap-	Selecting, drawing via Notes, 41		
plication, CDMA radios, 153	Sending and receiving messages, via Inbox, 77		
RTC. See Real-Time Clock	Serial port, modem support, 17		
	Serial position, COM B, 366		
\$			
S9C, unit information control panel, upgrade files, 365	Services, Phone application, GSM radios, 157		
Sabre 1551E or 1553	Setting date and time, 395		
See also Tethered scanner	Setting up an e-mail account, 116		
cabling, 216	SETUP.DLL, installation functions, 236		
settings, 215	SIM cards		
Sample code, NPCP printing, 202	IMSI assigned		
SB555 Watcher	CDMA/1xRTT, 141		
activation, 165	GSM/GPRS, 143		
Bell Mobility, 183	installation status, GSM/GPRS, 143		
Sprint, 172	phone number assigned, GSM/GPRS, 142		
Telus, 183	Simple Network Management Protocol. See SNMP		
Verizon, 168	SMS, Folder behavior connected to e-mail server, 78		
copying files to computer, 162	SMS messages, Phone application		
via Microsoft ActiveSync, 163	CDMA radios, 150		
via storage cards, 163	GSM radios, 156		
location of ESN, 165	Snap-on modems, 17		
Scan codes	SNMP, 195		
	configuration parameters		
alphanumeric keypad, 296	identification contact, 359		
numeric keypad, 295	identification location, 361		
SCAN Mute, Audio control panel applet, 8	identification name, 360		
Scanner	security encryption key, 356		
beeper volume	security read encryption, 354		
selecting, 9	security read-only community string, 352		
turning it off, 11			
turning it on, 7	security read/write community string, 353		
mute feature, turning it off, 8	security subnet mask, 324		
specifications, 25	security write encryption, 355		
unit configuration parameters	trap authentication, 357		
automatic shutoff, 393	trap threshold, 358		

SNMP OIDs	telepen, 319		
1D OmniDir decode enable, 342	user ID, 328		
aimer LED duration, 338	trap		
beeper, 333	authentication, 357		
frequency, 335	threshold, 358		
volume, 334	UPC		
codabar, 306	A user ID, 327		
user ID, 325	E user ID, 327		
code 11, 320	UPC/EAN, 307		
user ID, 329	virtual wedge, 343		
code 128, 309	code page, 349		
FNC1 character, 311	grid, 348		
user ID, 325	postamble, 346		
code 39, 304	preamble, 344		
user ID, 325	Software versions, 20, 362		
code 93, 308	700 Series Computer, 20		
length, 308	unit information control panel applet, 364		
user ID, 326	Speaker, 3		
datamatrix, 322	Speakerphone		
EAN	phone application, 159		
13 user ID, 328	volume, 161		
8 user ID, 328	Specifications, 24		
good read	Cisco compatible extensions, 24		
beep duration, 337	display, 24		
beeps, 336	environmental, 25		
identification	expansion slots, 25		
contact, 359	integrated scanner options, 25		
location, 361	integrated wireless, 25		
name, 360	keypad options, 25		
image dimension, 340	memory and storage, 25		
interleaved 2 of 5, 317	microprocessor, 25		
user ID, 326	modems, 25		
lighting mode, 341	operating system, 25		
macro PDF, 314	physical dimensions, 26		
matrix 2 of 5, 318	power, 26		
user ID, 328	regulatory approvals, 26		
maxicode, 323	standard communications, 26		
micro PDF417, 316	Speed dial, Phone application		
MSI, 313	CDMA radios, 149		
user ID, 326	GSM radios, 155		
PDF417, 314			
user ID, 326	Sprint activation process, 172 SSID (network name), 802.11 radio module, 374		
plessey, 312	Standard 2 of 5, configuration parameter, 305		
user ID, 327	user ID, 327		
prefix, 330	Standard communications, specifications, <b>26</b>		
QR code, 321	Standard communications, specifications, 20 Start Menu, adding programs, 47		
security	via ActiveSync, 47		
encryption key, 356	via File Explorer, 47		
read encryption, 354	Static IP, replicating registry settings, 123		
read-only community string, 352	Status icons, Windows Mobile, 30		
read/write community string, 353	Sticky Aimer Duration, configuration parameter, 339		
write encryption, 355	Storage media, 21		
security subnet mask, 324	specifications, 25		
standard 2 of 5, 305	Stream device driver		
user ID, 327	NPCPPORT.DLL, 199		
sticky aimer duration, 339	ONEIL.DLL, 204		
suffix, 331	Suffix, configuration parameter, 331		

Symbologies	TLS		
internal scanner supported symbologies, 212	802.11 profile		
scanning labels, 396	certificates, 388		
tethered scanner supported symbologies, 222	WPA encryption, 383		
user IDs	profile security information		
Codabar, 325	WEP encryption, 382		
Code 11, 329	WPA encryption, 383		
Code 128, 325	Today, Windows Mobile settings, 45		
Code 39, 325	Today screen, Windows Mobile, 30		
Code 93, 326	Tools CD		
EAN 13, 328	CAB files, 122, 364		
EAN 8, 328	CE Imager, 128		
Interleaved 2 of 5, 326	Comm Port Wedge CAB file, 364		
Matrix 2 of 5, 328	management tools installed on desktop, 121		
MSI, 326	MIB files, 195		
PDF417, 326	sample NPCP code, 202		
Plessey, 327	wireless printing sample, 365		
Standard 2 of 5, 327	Tracking people, via Contacts, 64		
Telepen, 328	Transcriber, Windows Mobile input panel, 37		
UPC A, 327	Trap configuration parameters		
UPC E, 327	authentication, 357		
when not available	threshold, 358		
imager, 312, 313, 314, 316, 318, 319, 320	Traps, control panel appet, SNMP, 357		
laser scanner, 321, 322, 323	Troubleshooting		
Synchronizing	1551/1553 Tethered Scanners, 221		
AvantGo channels, 98	CAB Wizard, 240		
Calendar, 51	does not turn off, 402		
contacts, 65	I/O key, 402		
e-mail messages, 77	power status LED, 402		
favorite links, 96	power supply, 402		
mobile favorites, 97	shuts down during operation, 402		
notes, 76	slow recovery after a boot, 402		
Pocket Word, 85	unit does not turn on, 402		
Tasks, 74	unit does not turn on in dock, 402		
System, Phone application, CDMA radios, 153	TTLS		
	802.11 radio module, WPA encryption, 386		
T	profile security information, WEP encryption, 385		
Tasks	Typing mode, Pocket Word, 83		
creating a task, 74	Typing on the screen, Pocket Word, 83		
Pocket Outlook, 73	U		
Start menu icon, 31	Unit, configuration parameters		
synchronizing, 74	automatic shutoff, 393		
TCP/IP client, DHCP server, 194	backlight timeout, 393		
Telepen, configuration parameter, 319	date/time, 393		
user ID, 328	key clicks, 393		
Telus activation process, 183	volume, 393		
Testing AT commands, 185	Unit information		
Tethered scanner	battery status, 363		
capabilities, 216	CAB files, 364		
disabling, 214	ActiveX control tools, 365		
enabling, 214	Bluetooth stack, 364		
error messages, 214	Comm Port Wedge, 364		
limitations, 216	NPCP printer, 364		
settings, 214	S9C Upgrade, 365		
supported symbologies, 222	Windows configuration, 365		
Text messages, Windows Mobile, 43	wireless printing sample, 365		
Time, setting, 395	versions, 20, 362		
Tips for working, Pocket Excel, 87	Unit Manager, date/time, 393		

Unshifted plane on keypad, regular keypad, 292	GSM/GPRS, 142		
UPC, configuration parameter, 307	installing available modules, 137		
A user ID, 327	loading a module, 137		
E user ID, 327	WAN rado CORE module		
Update your PCS Vision profile, Phone application,	installing available modules, 137		
CDMA radios, 152	loading a module, 137		
Updating, bootloader, 121	WAP pages, 96		
URLs	connecting to an ISP, 100		
ActiveSync, 48	Warm boot, performing, 19		
Adobe Acrobat Reader, 184	Watcher applications		
AT command interface	activating, Sprint, 173		
CDMA/1xRTT SB555, 184	downloading, Sprint, 173		
GPRS/GSM MC45, 184	using		
GPRS/GSM MC46, 184	Sprint, 176		
full screen display, 241	Verizon, 168		
MIBs, 195	Web browsers, FTP support, 247		
Microsoft Exchange e-mail account, 88	Web pages, 96		
Microsoft Passport account, 88	connecting to an ISP, 100		
Microsoft support, 29	Welch Allyn 1470 Imager		
MSDN Windows CE downwards at 104	cabling, 216		
MSDN Windows CE documentation, 194	settings, 215		
Windows Mobile, 29	WEP encryption		
Windows Mobile support, 29	LEAP security method, 389		
Utilities control panel applet	PEAP security method, 378		
app launch, 369 dock switch, 366	profile security information, 375, 376		
registry save, 367	TLS security method, 382 TTLS security method, 385		
wakeup mask, 368	zero configuration, 110		
wakeup mask, 500	Windows CE documentation (MSDN), 194		
V	Windows configuration, unit information control panel,		
Verizon activation process, 168	WinCfg CAB file, 365		
Vibrator	Windows Media files, Windows Media Player, 92		
enabling, 22	Windows Media Player		
phone application, 160	Start menu icon, 31		
programming, 289	Windows Mobile, 92		
Video files, Windows Media Player, 92	Windows Mobile		
Viewing mobile favorites and channels, Pocket Internet	ActiveSync, 48		
Explorer, 99	basic skills, 30		
Virtual wedge	Calendar, 50		
bar code configuration	command bar, 32		
grid, 399	Contacts, 64		
postamble, 399	edition information, 28		
preamble, 399	getting connected, 100		
configuration parameter, 343	Inbox, 77		
code page, 349	MSN Messenger, 87		
grid, 348	navigation bar, 32		
postamble, 346	Notes, 75		
preamble, 344	notifications, 33		
Voice mail, Phone application, CDMA radios, 150	Pocket Excel, 86		
Volume	Pocket Word, 82		
bar code configuration, 393, 396	pop-up menus, 33		
configuration parameter, 393	programs, 31		
phone application, 159, 161	status icons, 30		
VPN server, creating a connection, to work, 113	support URLs, 29		
W	Tasks, 73		
	Today screen, 30		
Wakeup mask, control panel applet, 368	where to find information, 29		
WAN monitor CORE module CDMA/1xRTT, 140	Windows Media Player, 92		
	writing on the screen, 38		

Wireless network, 135, 373 creating a connection, 108 specifications, 25	getting connected, 104 WPA authentication 802.11 radio module, Zero Configuration, 111
Wireless printing	with pre-shared key, Zero Configuration, 112
Bluetooth compatible module, 189	WPA encryption
unit information control panel, WP_SAMPLE.CAB file,	802.11 radio module, 377
365	LEAP security method, 390
Wireless WAN	PEAP security method, 379
AT command interface	TLS security method, 383
CDMA/1xRTT SB555, 184	TTLS security method, 386
GPRS/GSM MC45, 184	WPport, 191
GPRS/GSM MC46, 184	Writing mode, Pocket Word, 84
CDMA/1xRTT, 140	Writing on the screen
GSM/GPRS, 142	See also Notes
testing AT commands, 185	Pocket Word, 84
Work	Writing to drivers
creating	DTR, 205
a modem connection, 105	NPCP, 200
a VPN server connection, 113	

Files Index	NLEDSetDevice, 290 NPCPPORT.DLL, 199 NRINET.INI, 365
Numbers	•
80211API.DLL, 267	0
80211CONF.EXE, 267	OEMIOCTL.H
80211SCAN.EXE, 267	IOCTL_GET_CPU_ID, 265 IOCTL_HAL_COLDBOOT, 260
802PM.DLL, 267	IOCTL_HAL_COLDBOOT, 200 IOCTL HAL GET BOOT DEVICE, 262
A	IOCTL HAL GET BOOTLOADER VERINFO,
AUTOUSER.DAT, 122, 123	259
	IOCTL_HAL_GET_OAL_VERINFO, 258
C	IOCTL_HAL_GET_RESET_INFO, 261
CABWIZ.DDF, 239	IOCTL_HAL_ITC_READ_PARM, 251
CABWIZ.EXE, 224, 239 CEIMAGER.EXE, 128	IOCTL_HAL_ITC_WRITE_SYSPARM, 255
COREDLL.DLL, 289	IOCTL_HAL_REBOOT, 263 IOCTL HAL WARMBOOT, 260
CPL802.CPL, 267	ONEIL.DLL, 204
D	P
DEVICEID.H, 257	PKFUNCS.H
E	IOCTL_HAL_GET_DEVICEID, 257
EXITME.BIN, 247	IOCTL_PROCESSOR_INFORMATION, 264 PRISMNDS.DLL, 267
F	
-	R
FTPDCE.EXE, 245, 247 AutoFTP, 249	REBOOTME.BIN, 247
FTP Server, 242	REGFLUSH.CAB, 123
FTPDCE.TXT, 247	RESETMEPLEASETXT, 236
	RPM.EXE, 229 RPMCE212.INI, 229
I THE PART OF THE PART OF	
INTERMEC.MIB, 195	S
ITCADC.MIB, 195	SETUP.DLL, 228, 236
ITCSNMP.MIB, 195 ITCTERMINAL.MIB, 195	DllMain, 236
· · · · · · · · · · · · · · · · · · ·	Sprint_Watcher_PPC_2002-03xxx.CAB, 174
M	Т
MAKECAB.EXE, 239	TAHOMA.TTF, 229
MOD80211.DLL, 267	U
N	URODDSVC.EXE, 267
NETWLAN.DLL, 267	UKODDS V C.EAE, 20/
NLED.H, 290	W
NI EDC D 1-f- 200	WICECTART INL 220

WCESTART.INI, 229

NLEDGetDeviceInfo, 290



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

**tel** 425.348.2600 **fax** 425.355.9551 www.intermec.com

700 Series Color Mobile Computer User's Manual - July 2005



P/N 961-054-031 REV H