Installation and User's Guide

Digi Connect Wi-Wave™ for Microsoft Windows® CE 5.0 and Microsoft Windows Embedded CE 6.0

90000904_C

Digi International Inc.
11001 Bren Road East
Minnetonka, MN 55343 (USA)
☎ +1 877 912-3444 or +1 952 912-3444
http://www.digiembedded.com

Table of Contents

1.1. Conventions used in this manual

The following is a list of the typographical conventions used in this manual:

Style	Used for file and directory names, programs and command names, command-line options, URL, and new terms.
`Style`	Used in examples to show the contents of files, the output from commands or in the text the C code.
`Style`	Used in examples to show the text that should be typed literally by the user.
#	Used to indicate the listed commands have to be executed as administrator.
$	Used to indicate the listed commands have to be executed as a normal user.
[1]	Used to reference an item of the reference section.

This manual also uses these frames and symbols:

This is a warning. It helps you to solve or to avoid common mistakes or problems

This is a tip. It contains useful information about a topic

```
$   This is a host computer session
$   And this is what you must input (in bold)
```

```
#   This is a target session
#   And this is what you must input (in bold)
```

1.2. Acronyms and abbreviations

AES	Advanced Encryption Standard
API	Application Program Interface
BSP	Board Support Package
CCMP	Counter Mode with Cipher Block Chaining Message Authentication Code Protocol
CPU	Central Processing Unit
DHCP	Dynamic Host Configuration Protocol
DSSS	Direct-Sequence Spread Spectrum

EEPROM	Electronically Erasable Programmable Read Only Memory
EULA	End-User License Agreement
FPGA	Field-Programmable Gate Array
FTP	File Transfer Protocol
IEEE	Institute for Electrical and Electronics Engineers
IOCTL	I/O Control
IP	Internet Protocol
LCD	Liquid Crystal Display
OHCI	Open Host Controller Interface
OS	Operating System
PSK	Pre-Shared Key
QFE	Quick Fix Engineering
SSID	Service set identifier
TKIP	Temporal Key Integrity Protocol
USB	Universal Serial Bus
WEP	Wired Equivalent Privacy
WLAN	Wireless Local Area Network
WPA	Wi-Fi Protected Access
WZCSAPI	Wireless Zero Config Service API
WZCSVC	Wireless Zero Config Service

2. Introduction

2.1. Overview

The Digi Connect Wi-Wave™ is a Wireless LAN USB Adapter. This USB adapter is designed to comply with IEEE 802.11b/g Wireless LAN standard and easy to carry with the PCI Express Minicard. It is suitable for any embedded device running Microsoft Windows® CE 5.0 or Microsoft Windows Embedded CE 6.0.

2.2. Features of the Digi Connect® Wi-Wave

- Complies with the IEEE 802.11b and IEEE 802.11g 2.4Ghz (DSSS) standards
- High data transfer rate – up to 54Mbps.
- Supports 64/128-bit WEP, TKIP and AES encryption.
- Supports open, shared, WPA, WPA-PSK, WPA2 and WPA2-PSK authentication.
- Driver complies with the NDIS 5.0 standard
- Implements Power Save Poll algorithm to save power while connected to an AP.
- Supports 'ndisconfig power' calls to put the driver into different power level modes

The card supports 64/128-bit WEP data encryption, which protects a wireless network from eavesdropping. It also supports the WPA (Wi-Fi Protected Access) feature, which combines IEEE 802.1x, PSK (Pre-Shared Key), and TKIP (Temporal Key Integrity Protocol) technologies. Users are required to authorize before accessing to Access Points or Access Point Routers, and the data transmitted is encrypted/decrypted by a dynamically changed secret key. Furthermore, this adaptor supports WPA2 function, which provides a stronger encryption mechanism through AES (Advanced Encryption Standard), which is a requirement for some corporate and government users.

Encryption *makes data unreadable without a certain deciphering key.*

Authentication *confirms the identity or origin of something or someone.*

3. Requirements

3.1. Host System requirements

The development system must meet the following requirements:

- x86 PC with 500 MHz Pentium III or faster processor; 2 GHz Pentium 4 or equivalent recommended
- Microsoft Windows® 2000 Professional with Service Pack 4 or Windows XP Professional with Service Pack 1.
- CD-ROM or DVD-ROM drive.
- Serial port
- Ethernet network card
- Microsoft Windows® CE
  - Microsoft Windows® CE 5.0 with latest QFE installed. QFE distributed at the end of 2006 and beginning of 2007 added some important wireless support to the OS, so it's important to install them.
  - Or Microsoft Windows® Embedded CE 6.0 with Service Pack 1 installed.

3.2. BSP requirements

- USB host support (full- or high-speed modes)
- Display interface support recommended for development purposes, but not required

3.3. Microsoft Windows® CE OS Configuration

3.3.1. Required components

Go to the Catalog and expand **Core OS > CEBASE**. Then include these elements:

- Communication Services and Networking
  - Networking - General
    - Extensible Authentication Protocol
  - Networking – Local Area Network (LAN)
    - Wireless LAN (802.11) STA - Automatic Configuration 802.1x
- Security
  - Authentication Services
    - Schannel (SSL/TLS)

3.3.2. Recommended catalog components

Other recommended networking utilities and services are:

- Communication Services and Networking
  - Networking - General
    - Network utilities (ipconfig, ping, route)
  - Servers
    - FTP Server
    - Telnet Server
- Shell and User Interface
  - User Interface
    - Network User Interface

If using a WPA Enterprise configuration, the following component may also be needed for certificate management:

- Security
  - Microsoft Certificate Enrollment Tool Sample

4. Installation

This release is installed by executing Setup.exe.The installer wizard will guide you in all required steps.

After the installation has finished, the following components will be on your PC:

- %ProgramFiles%\Digi\AppKits\WiWave:
  - **Doc**: Software and Hardware documentation including this document.
  - **Bin**: Prebuilt binaries for required Microsoft Windows® CE Versions and architectures.
  - **Uninstaller**: Executable to uninstall this release.
  - Release notes and license agreement.

- %_WINCEROOT%\OTHERS\Digi\appkits\WiWave:
  - **src:** Sources of the driver and configuration application.

%ProgramFiles% is an environment variable of your system that provides the path to your Program Files directory (usually C:\Program Files)

%_WINCEROOT% is an environment variable of your system that provides the path to your Microsoft Windows® CE root directory (usually C:\WINCE500 or C:\WINCE600)

5. Integration

5.1. Integration: Overview

The Digi Connect Wi-Wave™ integration described in this chapter assumes that you have already performed the following steps. These steps are fairly general as they are highly dependant on the customer BSP and the Microsoft Windows® CE version used:

Before completing these steps, you will need to have the following applications already installed:

- Platform Builder as described in Chapter 2: Host System Requirements
- A BSP corresponding with the hardware that will be used.

5.1.1. Creating a New Project

If you are creating a new project using the Wizard, follow these steps:

1. Select the BSP that corresponds with the hardware.
2. Choose a Design Template or a Custom Device.
   After the wizard finishes if you are using a Design Template, you will need to perform the following actions need to be done manually:
   - Add missing OS components from the ones listed in Chapter 2: 2.3.Microsoft Windows® CE OS Configuration.
   - Add missing BSP components from the ones listed in Chapter 2: BSP Requirements

   If you are using a Custom Device, you will select all operating systems manually. Those listed in Chapter 2: Microsoft Windows® CE OS Configuration' need to be included. After the wizard finishes, missing BSP components from the ones listed in Chapter 2: BSP Requirements also need to be added manually.
3. Continue with the Connect Wi-Wave integration described in this chapter.

5.1.2. Working with an Existing Project

If you are working with a project that is already built and running for the hardware, follow these steps:

1. Add missing OS components from the ones listed in Chapter 2: 2.3.Microsoft Windows® CE OS Configuration.
2. Add missing BSP components from the ones listed in Chapter 2: BSP Requirements
3. Continue with the Connect Wi-Wave integration described in this chapter.
4. Download the new Microsoft Windows® CE image to the target.

5.2. Integration without sources

5.2.1. Driver and Configuration Application binaries

The Digi Connect Wi-Wave contains pre-built binaries for:

- Following OS Versions:
  - WinCE500
  - WinCE600
- Following Architectures:
  - ARM
  - x86
  - MIPSII

Select your required combination from %ProgramFiles%\Digi\AppKits\WiWave\bin and copy following files to %_WINCEROOT%\platform**YourPlatformName**\files directory:

- Wiwave.dll
- Wiwave.rel
- Wificonf.exe

%ProgramFiles% is an environment variable of your system that provides the path to your Program Files directory (usually C:\Program Files)

%_WINCEROOT% is an environment variable of your system that provides the path to your Microsoft Windows® CE root directory (usually C:\WINCE500 or C:\WINCE600)

***YourPlatformName** has to be replaced by the name of the BSP where the driver wants to be integrated.*

5.2.2. Platform.reg

Add the following entries to your platform.reg file.

You can copy them from file %_WINCEROOT%\OTHERS\Digi\appkits\WiWave\src\driver\Wiwave\Wiwave.reg.

```
;Wiwave Driver
[HKEY_LOCAL_MACHINE\Drivers\USB\ClientDrivers\Wiwave]
   "Dll"="Wiwave.dll"
   "Prefix"="NDS"
   "Miniport"="Wiwave"

;Wiwave Vendor=0x04d0, Product=0x0801
[HKEY_LOCAL_MACHINE\Drivers\USB\LoadClients\1232_2049\Default\Default\Wiwave]
   "DLL"="Wiwave.dll"

[HKEY_LOCAL_MACHINE\Comm\Wiwave]
   "DisplayName"="Wiwave Wifi Interface"
   "Group"="NDIS"
   "ImagePath"="Wiwave.dll"

[HKEY_LOCAL_MACHINE\Comm\Wiwave\Linkage]
```

```
   "Route"=multi_sz:"Wiwave1"

[HKEY_LOCAL_MACHINE\Comm\Wiwave1]
   "DisplayName"="Wiwave Wifi Interface"
   "Group"="NDIS"
   "ImagePath"="Wiwave.dll"

[HKEY_LOCAL_MACHINE\Comm\Wiwave1\Parms]
   "BusNumber"=dword:0
   "BusType"=dword:1
   "Transceiver"=dword:3
   "CardType"=dword:1

;MAC configurable params (Enter values in HEX)
   "tx_power"=dword:0a             ;Default= 10
   "band"=dword:03           ;0= All Bands, 1= Band A, 2= Band B, 3= Bands BG
   "chan_mask"=dword:00003fff      ;Default All 2.4 GHz channels Enabled
   "chan_mask_high"=dword:00000000 ;Default= All Disabled
   "tx_rate"=dword:21c             ;Default= 21c
   "rts_thresh"=dword:92b          ;Default= 92b
   "frag_thresh"=dword:00000600    ;Default= 00000600
   "ibss_master_chan"=dword:03     ;Channel to use for IBSS Master
   "duty_cycle_on"=dword:000000c8  ;Default 200mS, Minimun 100mS
   "duty_cycle_off"=dword:000000c8 ;Default 200mS, Maximum 500mS

;Wireless option parameter values
;
; 0x00000001   Enable antenna diversity
; 0x00000002   Enable short preamble
; 0x00000004   Enable server certificate verification
; 0x00000008   Use only 802.11b rates in 2.4 GHz band
; 0x00000010   Use RTS/CTS protection frames for 802.11g
; 0x00000020   Use fixed transmit rate
; 0x00000040   Enable 802.11 Multi domain capability (802.11d)
; 0x00000080   Antenna Selection 0=Ant1, 1=Ant2
; 0x00000100   Enable 802.11 Power Save Poll
;
   "options"=dword:00000000                ;Default= 00000000

[HKEY_LOCAL_MACHINE\Comm\Wiwave1\Parms\TcpIp]
   "EnableDHCP"=dword:0            ;<======== set to 1 if DHCP enabled.
   "IpAddress"="192.168.42.130"
   "DefaultGateway"="192.168.42.64"
   "UseZeroBroadcast"=dword:0
   "Subnetmask"="255.255.255.0"
;  "DNS"="212.163.200.2"
```

You can customize some of these registry entries, e.g. network address, TCP/IP configuration and the MAC configurable parametres. These values will be read only during the driver startup.

5.2.3. Platform.bib

Add the following entries to your platform.bib file.

Under Windows CE 5.00:

```
Wiwave.dll       $(_FLATRELEASEDIR)\Wiwave.dll    NK   SH
wzctool.exe      $(_FLATRELEASEDIR)\wzctool.exe   NK
wificonf.exe     $(_FLATRELEASEDIR)\wificonf.exe  NK
```

Under Windows CE 6.00:

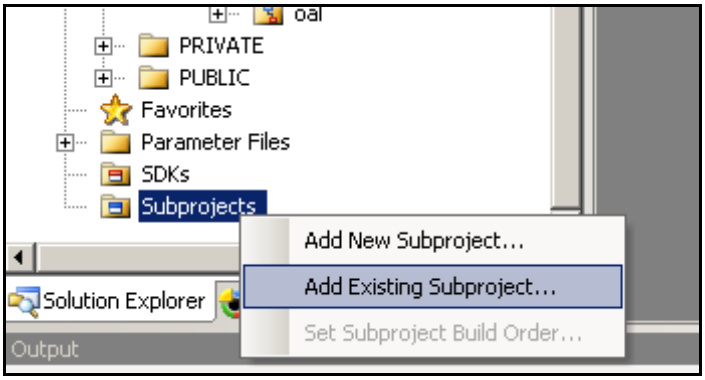
```
Wiwave.dll       $(_FLATRELEASEDIR)\Wiwave.dll    NK   SHK
wzctool.exe      $(_FLATRELEASEDIR)\wzctool.exe   NK
wificonf.exe     $(_FLATRELEASEDIR)\wificonf.exe  NK
```

5.3. Installation with sources

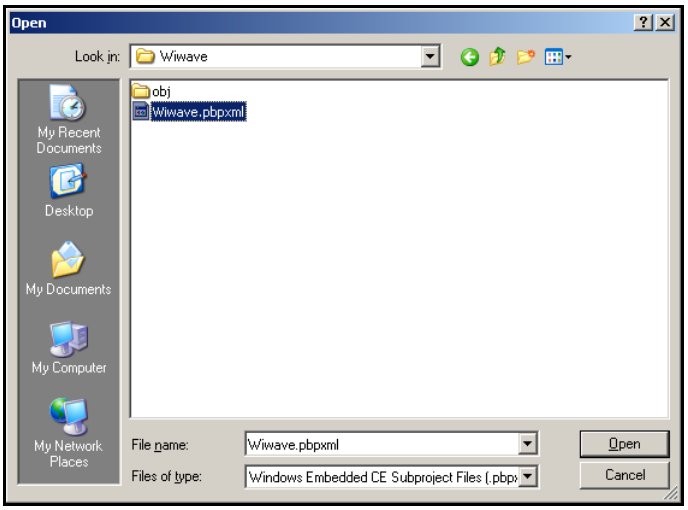
5.3.1. Driver

After you have completely built your project, it's time to Add a new Subproject for the Wi-Wave Module:

1. Open the Solution Explorer and right-click over the Subprojects. Choose 'Add Existing Subproject':

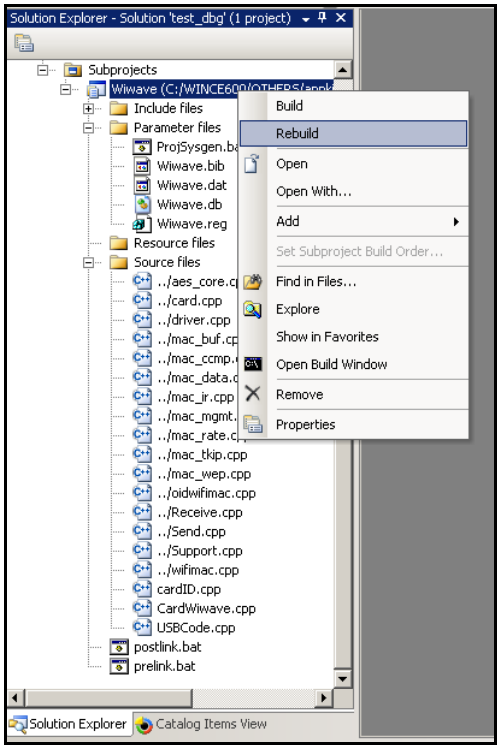


2. Navigate to %_WINCEROOT%\OTHERS\Digi\appkits\WiWave\src\driver\Wiwave directory and select Wiwave.pbpxml:



3. Now you can see the parameter files and the source files.
4. Right click over WiWave and select Rebuild and then Makeimage.

The driver build should be rebuilt and included in the final nk.bin image together with the necessary registry entries.

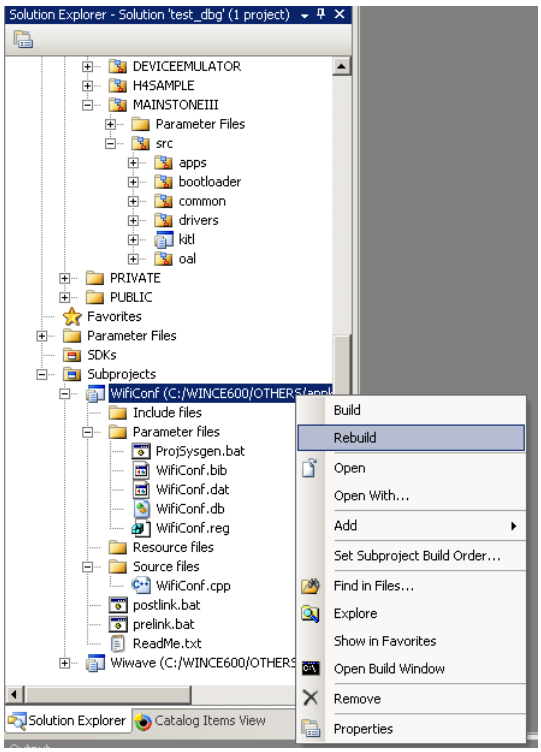


5.3.2. Application

Use the following path to select WifiConf.pbpxml

%_WINCEROOT%\OTHERS\ Digi\appkits\WiWave\src\WifiConf ,

Use the same steps as you did for the Driver.



6. SYSGEN the project

Once you have configured the OS as described in chapter 3.3 and installed the driver as described in chapter 4, you have completed the initial setup. Press the SYSGEN button so both the operating system and the BSP will be build.

Please verify that the wiwave.dll is inside your new image and that the correct registry entries are placed in the final registry file (reginit.ini).

7. Driver Start

Once your project has been built, download it to the target device and start it up..

If you have the Digi Connect Wi-Wave™ connected to the USB host connector of the target, it should be automatically detected, and you should see a console message similar to the following:

```
[Wi-Wave]: Loading Wireless Driver Version 1.8 ... OK.
```

More information may be shown depending on the debug level established in the driver.

Before Driver version 1.5, MAC address was stored and read in the eeprom using little endian format. From 1.5 on, it's done in big endian.

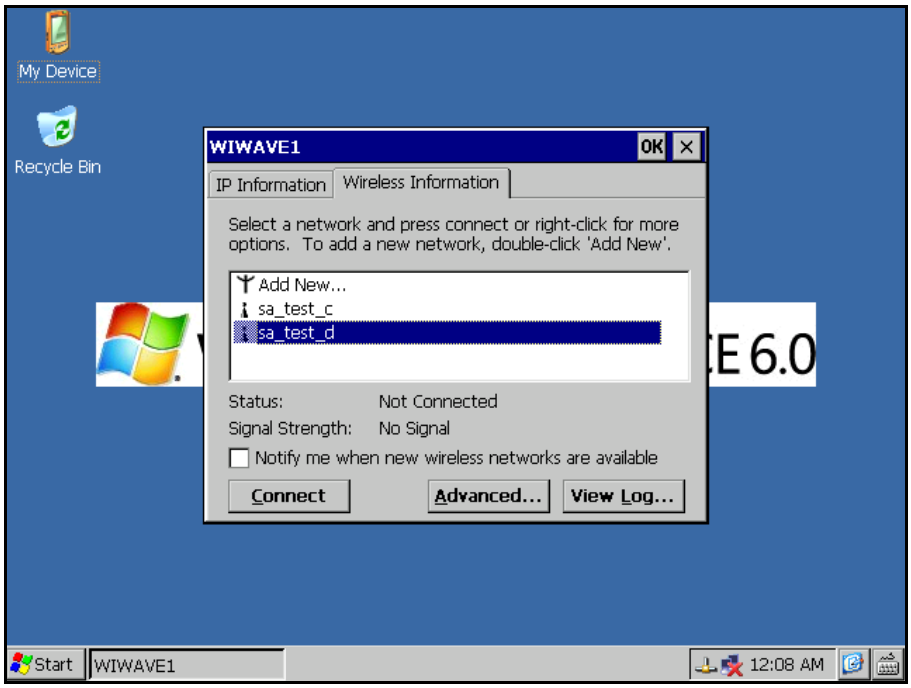
Conversion of modules running previous versions of the driver are done automatically first time driver version 1.5 starts.

Following message will be seen:

[Wi-Wave]: MacUsbReadEEPROM recovering from Little Endian.

Make sure your device reports the same address printed on the module label. Should follow this pattern: 00:40:9d:xx:xx:xx

If a display is available, the first time a window opens, it will show the Access Points in range. From there, you can select an Access Point to connect to. Also, a small connection icon with a red X appears in the taskbar indicating that the Connect Wi-Wave is not yet connected to any Access Point.



To prevent this dialog from showing, uncheck ***Notify me when new wireless networks are available****. This setting is stored in the Registry in RAM. To save the Registry permanently into NVRAM revise your BSP documentation.*

When this dialog is opened, Microsoft Windows® CE asks the wireless driver to scan for surrounding Access Points every 30 seconds.

If you are already connected to an Access Point, scanning could cause you to lose the connection.

If a display is not available, the driver load can be verified by entering the following console commands:

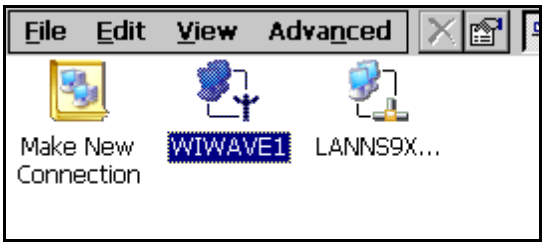
```
\> wzctool -e
wifi-card [0] = WIWAVE1
```

If any preferred network is saved in the Registry, the target automatically connects to it.

8. WLAN network settings

The network settings for the WLAN adapter are initially taken from platform.reg file.

To modify the WLAN network settings in Windows Embedded CE, go to the Control Panel in the target device: **Start > Settings > Control Panel > Network > Dial-up Connections**.



Network settings modified in the Microsoft Windows® Embedded CE 6.0 are stored in RAM memory and remain valid until the target is reset. To save these settings permanently in NVRAM, revise your BSP documentation.

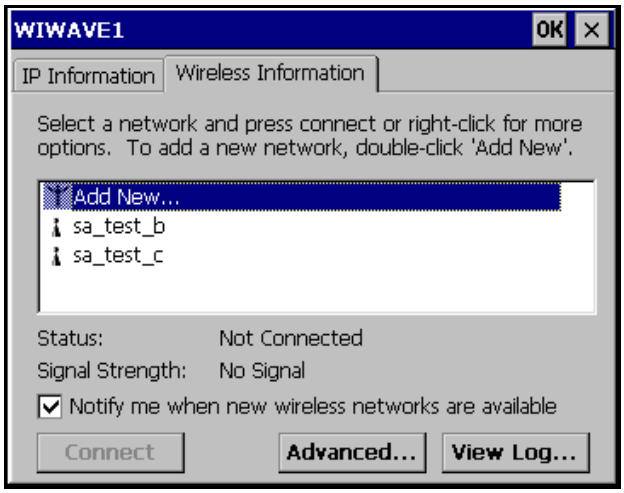
9. Connect to an Access Point (infrastructure mode)

The Digi Connect Wi-Wave™ wireless interface can be connected to an Access Point in several ways. All of them go through the WZCSAPI (see the official Microsoft Windows ® CE online help documentation for more information about the Access Point).

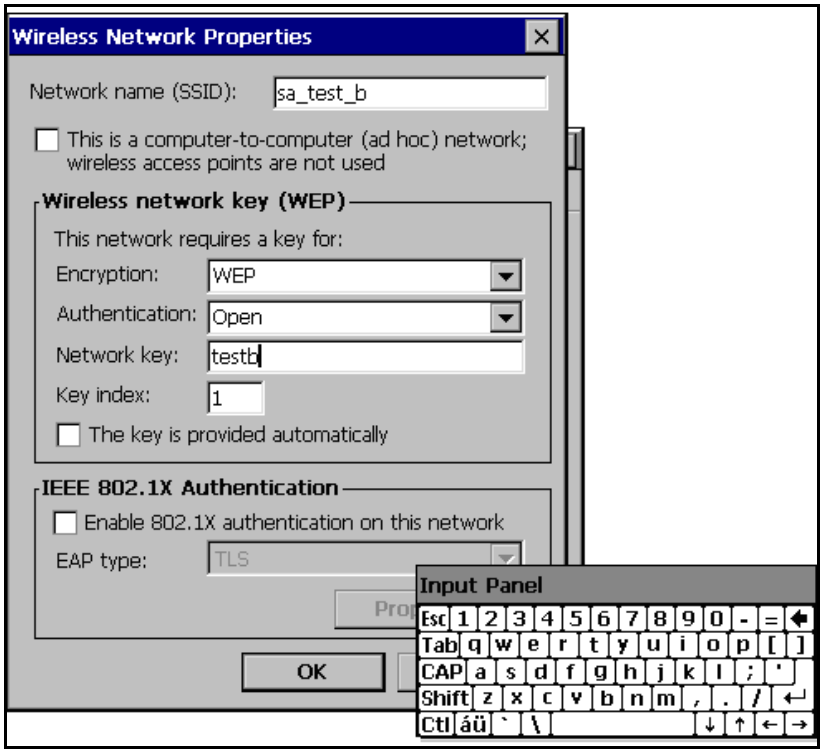
9.1. Graphic mode

1. Double-clicking the wireless icon on the taskbar shows the window with a list of the wireless Access Points in range.

If the wireless icon is selected shortly after the driver was loaded, the Access Point list might be empty because the Connect Wi-Wave wireless interface is still scanning for Access Points.



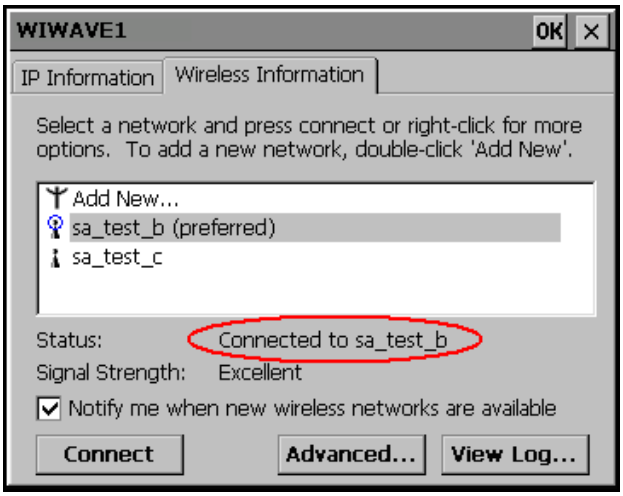
2. Click the desired Access Point, and click **Connect**. The **Wireless Network Properties** window opens. Depending on the Access Point authentication and encryption configuration, different information may need to be entered.



The virtual keyboard can be used for inputing characters.

3. Click **OK**. The Properties window closes and the previous window is displayed again.

The status text beside the selected Access Point passes through several states (depending on the authentication): **Scanning**, **Associating**, **Associated**, **Authenticating**, **Authenticated...**and finishes with **Connected**. Then, the WiFi Taskbar icon changes to blue, and the red X disappears.



Now the target can be accessed from any device on the network segment. For example, you can ping the Digi Connect Wi-Wave™ from a wireless PC.

If a simple ping does not work, it is probable that the Access Point and the IP of the wireless interface are not within the same network segment. Check the network settings of the WLAN, as seen in topic 8.

9.2. Command line mode

If a graphic display is not available, there is a command line application named **wzctool** for connecting to an Access Point. Execute the **wzctool** application with **/help** option to learn its syntax:

```
\> wzctool /help
wzctool usage:
options:
 -e              Enumerate wireless cards.
 -q <Card Name>  Query wireless card.
 -c <Card Name> -ssid AP-SSID -auth open -encr wep -key 1/0x1234567890
     connect to AP-SSID with given parameters.     Use -c -? for detail.
 -reset          Reset WZC configuration data. Wireless card will disconnect
    if it was connected.
 -set <Card Name> <parameter> Set WZC variables.
     Use -set -? for detail.
 -refresh        Refresh entries.
 -registry       configure as registry.
     Use -registry -? for detail.
 -enablewzcsvc   enable WZC service.
 -disablewzcsvc  disable WZC service.
 -?  shows help message
if no arg is given, wzctool will reads and set as settings in the registry.
Use '-registry -?' for detail
if no <Card Name> is given, wzctool will find the first WiFi card and use
this card.
\>
```

9.2.1. wzctool syntax

To get information about the available wireless interfaces, execute **wzctool –q**. This example shows that two networks are available and one of them is a preferred network.

```
\> wzctool -q
wireless card found: WIWAVE1
WZCQueryInterfaceEx() for WIWAVE1
In flags used       = [0x7FFFFFF]
Returned out flags  = [0x07EFFFF]
wzcGuid             = [WIWAVE1]
wzcDescr            = [WIWAVE1]
BSSID = 00:17:94:FD:99:C0 (this wifi card is associated state)
Media Type          = [0]
Configuration Mode  = [0000A002]
   zero conf enabled for this interface
   802.11 OIDs are supported by the driver/firmware
Infrastructure Mode = [1]  Infrastructure net (connected to an Access Point)
Authentication Mode = [4]  Ndis802_11AuthModeWPAPSK
rdNicCapabilities   = 96 bytes
   dwNumOfPMKIDs            : [3]
   dwNumOfAuthEncryptPairs  : [11]
   Pair[1]
      AuthmodeSupported          [Ndis802_11AuthModeOpen]
      EncryptStatusSupported     [Ndis802_11WEPDisabled]
   Pair[2]
```

```
      AuthmodeSupported          [Ndis802_11AuthModeOpen]
      EncryptStatusSupported     [Ndis802_11WEPEnabled]
   Pair[3]
      AuthmodeSupported          [Ndis802_11AuthModeShared]
      EncryptStatusSupported     [Ndis802_11WEPEnabled]
   Pair[4]
      AuthmodeSupported          [Ndis802_11AuthModeWPA]
      EncryptStatusSupported     [Ndis802_11Encryption2Enabled]
   Pair[5]
      AuthmodeSupported          [Ndis802_11AuthModeWPA]
      EncryptStatusSupported     [Ndis802_11Encryption3Enabled]
   Pair[6]
      AuthmodeSupported          [Ndis802_11AuthModeWPAPSK]
      EncryptStatusSupported     [Ndis802_11Encryption2Enabled]
   Pair[7]
      AuthmodeSupported          [Ndis802_11AuthModeWPAPSK]
      EncryptStatusSupported     [Ndis802_11Encryption3Enabled]
   Pair[8]
      AuthmodeSupported          [Ndis802_11AuthModeWPA2]
      EncryptStatusSupported     [Ndis802_11Encryption2Enabled]
   Pair[9]
      AuthmodeSupported          [Ndis802_11AuthModeWPA2]
      EncryptStatusSupported     [Ndis802_11Encryption3Enabled]
   Pair[10]
      AuthmodeSupported          [Ndis802_11AuthModeWPA2PSK]
      EncryptStatusSupported     [Ndis802_11Encryption2Enabled]
   Pair[11]
      AuthmodeSupported          [Ndis802_11AuthModeWPA2PSK]
      EncryptStatusSupported     [Ndis802_11Encryption3Enabled]
rdPMKCache   = 0 bytes
WEP Status          = [4]  <unknown value>
SSID = sa_test_c
Capabilities =
     WPA/TKIP capable
     WPA2/AES capable

[Available Networks] SSID List [2] entries.

******** List Entry Number [0] ********
   Length                  = 196 bytes.
   dwCtlFlags              = 0x00000010
   MacAddress              = 00:17:94:FD:99:C0
   SSID                    = sa_test_c
   Privacy                 = 4  Privacy enabled (encrypted with
                             [Ndis802_11Encyption2Enabled])
   RSSI                    = -37 dBm (0=excellent, -100=weak signal)
   NetworkTypeInUse        = NDIS802_11FH
   Configuration:
      Struct Length    = 32
      BeaconPeriod     = 90 kusec
      ATIMWindow       = 0 kusec
      DSConfig         = 2437000 kHz (ch-6)
      FHConfig:
         Struct Length = 0
         HopPattern    = 0
         HopSet        = 0
         DwellTime     = 0
   Infrastructure          = Ndis802_11Infrastructure
   SupportedRates          = 1.0,2.0,5.5,11.0,6.0,12.0,18.0,24.0, (Mbit/s)
   KeyIndex                = <not available> (beaconing packets don't have
                             this info)
   KeyLength               = <not available> (beaconing packets don't have
                             this info)
   KeyMaterial             = <not available> (beaconing packets don't have
                             this info)
   Authentication          = 4  Ndis802_11AuthModeWPAPSK
   rdUserData length       = 0 bytes.
```

```
******** List Entry Number [1] ********
   Length                  = 196 bytes.
   dwCtlFlags              = 0x00000010
   MacAddress              = 00:13:46:9B:A8:55
   SSID                    = sa_test_d
   Privacy                 = 6  Privacy enabled (encrypted with
                             [Ndis802_11Encyption3Enabled])
   RSSI                    = -46 dBm (0=excellent, -100=weak signal)
   NetworkTypeInUse        = NDIS802_11FH
   Configuration:
      Struct Length    = 32
      BeaconPeriod     = 100 kusec
      ATIMWindow       = 0 kusec
      DSConfig         = 2472000 kHz (ch-13)
      FHConfig:
         Struct Length = 0
         HopPattern    = 0
         HopSet        = 0
         DwellTime     = 0
   Infrastructure          = Ndis802_11Infrastructure
   SupportedRates          = 1.0,2.0,5.5,11.0,6.0,9.0,12.0,18.0, (Mbit/s)
   KeyIndex                = <not available> (beaconing packets don't have
                             this info)
   KeyLength               = <not available> (beaconing packets don't have
                             this info)
   KeyMaterial             = <not available> (beaconing packets don't have
                             this info)
   Authentication          = 7  Ndis802_11AuthModeWPA2PSK
   rdUserData length       = 0 bytes.

[Preferred Networks] SSID List [1] entries.

******** List Entry Number [0] ********
   Length                  = 196 bytes.
   dwCtlFlags              = 0x00000013
   MacAddress              = 00:17:94:FD:99:C0
   SSID                    = sa_test_c
   Privacy                 = 4  Privacy enabled (encrypted with
                             [Ndis802_11Encyption2Enabled])
   RSSI                    = -37 dBm (0=excellent, -100=weak signal)
   NetworkTypeInUse        = NDIS802_11FH
   Configuration:
      Struct Length    = 32
      BeaconPeriod     = 90 kusec
      ATIMWindow       = 0 kusec
      DSConfig         = 2437000 kHz (ch-6)
      FHConfig:
         Struct Length = 0
         HopPattern    = 0
         HopSet        = 0
         DwellTime     = 0
   Infrastructure          = Ndis802_11Infrastructure
   SupportedRates          = 1.0,2.0,5.5,11.0,6.0,12.0,18.0,24.0, (Mbit/s)
   KeyIndex                = <not available> (beaconing packets don't have
                             this info)
   KeyLength               = <not available> (beaconing packets don't have
                             this info)
   KeyMaterial             = <not available> (beaconing packets don't have
                             this info)
   Authentication          = 4  Ndis802_11AuthModeWPAPSK
   rdUserData length       = 0 bytes.

rdCtrlData length   = 0 bytes

parameter setting in Zero Config
tmTr = 3000 mili-seconds (Scan time out)
tmTp = 2000 mili-seconds (Association time out)
```

```
tmTc = 60000 mili-seconds (Periodic scan when connected)
tmTf = 60000 mili-seconds (Periodic scan when disconnected)
\>
```

9.2.2. Connect to an Access Point

To connect to a specific Access Point, use the **–c** option. For example, to connect to an Access Point with SSID **myAPname** with WPA-PSK authentication with TKIP encryption and password **fY5jHot6**, execute:

```
\> wzctool -c WIWAVE1 -ssid myAPname -auth wpa-psk -encr tkip -key fY5jHot6
```

If connecting to an Access Point with SSID **myAPname** open authentication and no encryption, the command is:

```
\> wzctool -c WIWAVE1 -ssid myAPname -auth open -encr disabled
```

9.2.3. Source code

The source code of the **wzctool** utility is available in the directory **%_WINCEROOT%\PUBLIC\COMMON\OAK\DRIVERS\NETSAMP\WZCTOOL**. It can be used as an example for controlling and configuring the Connect Wi-Wave wireless interface.

10. Connect to a computer (ad hoc mode)

10.1. Join to a peer

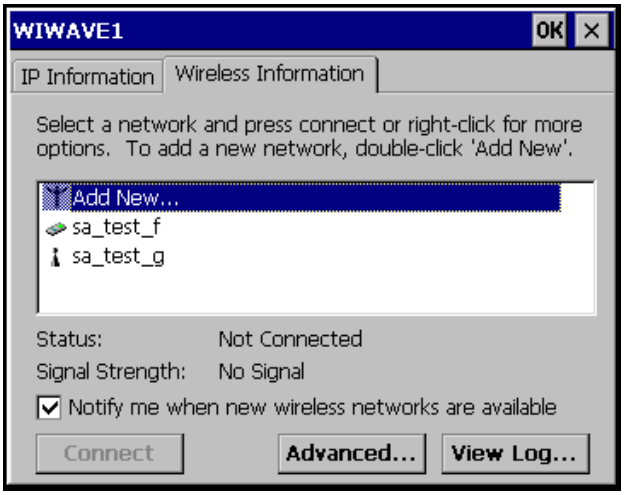
You can connect to a peer computer in ad hoc mode by using the same steps shown in the previous chapter: Connect to an Access Point (Infrastructure Mode).

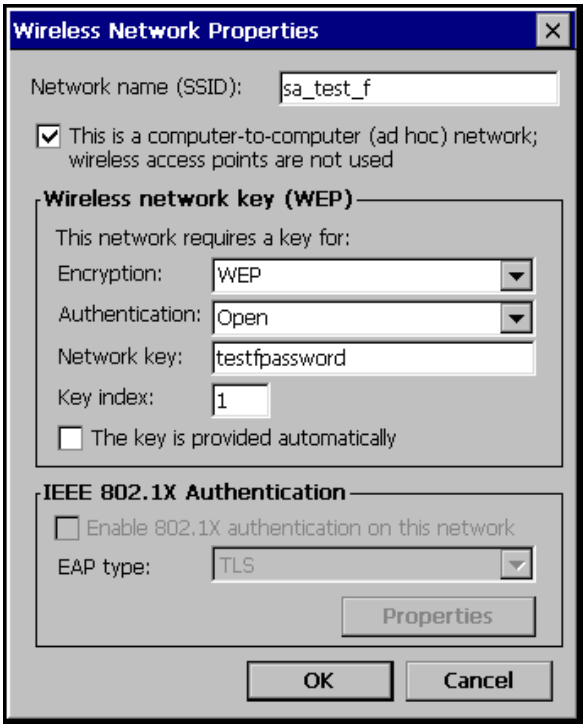
When queried with wzctool –q, a device configured as ad hoc reports something like this:.

```
Length                  = 196 bytes.
   dwCtlFlags              = 0x00000000
   MacAddress              = 66:3E:C2:49:AF:60
   SSID                    = sa_test_f
   Privacy                 = 0  Privacy enabled (encrypted with
                             [Ndis802_11WEPEnabled])
   RSSI                    = -55 dBm (0=excellent, -100=weak signal)
   NetworkTypeInUse        = NDIS802_11FH
   Configuration:
      Struct Length    = 32
      BeaconPeriod     = 90 kusec
      ATIMWindow       = 0 kusec
      DSConfig         = 2442000 kHz (ch-7)
      FHConfig:
         Struct Length = 0
         HopPattern    = 0
         HopSet        = 0
         DwellTime     = 0
   Infrastructure          = NDIS802_11IBSS
   SupportedRates          = 1.0,2.0,5.5,11.0,6.0,9.0,12.0,18.0, (Mbit/s)
   KeyIndex                = <not available> (beaconing packets don't have
                             this info)
   KeyLength               = <not available> (beaconing packets don't have
                             this info)
   KeyMaterial             = <not available> (beaconing packets don't have
                             this info)
   Authentication          = 0  Ndis802_11AuthModeOpen
   rdUserData length       = 0 bytes.
```

10.1.1. Graphic mode

The only difference from connecting in infrastructure mode is that ad hoc devices have a different icon than Access Points. In the Properties window, the check box This is a computer-to-computer (ad hoc) network is automatically selected, and the Encryption and Authentication combo boxes are preconfigured, depending on the ad hoc device configuration.





10.1.2. Command line mode

If a graphic display is not available, use the **wzctool** application to connect to an ad hoc device For example, to connect to an ad hoc device configured with **SSID=sa_test_f** with open authentication and wep128 encryption with index 1 and password **pass7ujH,** the command is:

```
\> wzctool -c WIWAVE1 -adhoc -ssid sa_test_f -auth open -encr wep -key 1/pass7ujH
```

10.2. Initiating the communication (Master Mode)

You can create a new ad hoc network so other devices can connect to your device in ad hoc mode.

The wireless channel used must be configured previously. The driver pulls it from the following registry entry:

```
[HKEY_LOCAL_MACHINE\Comm\Wiwave1\Parms]
     "ibss_master_chan"=dword:03      ;Channel to use for IBSS Master
```

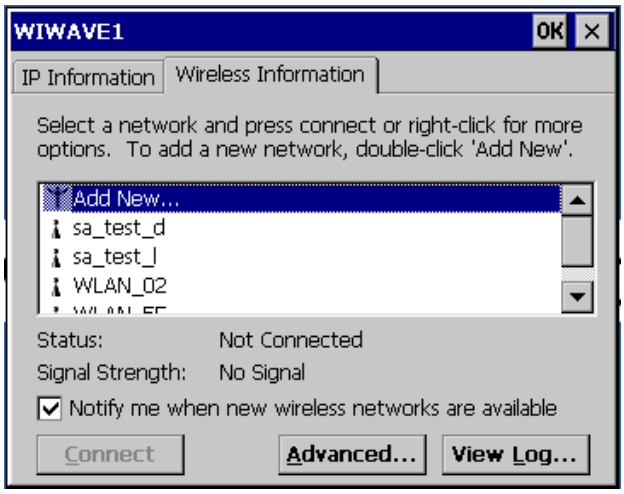
It is possible to change it after the driver has started through following the Wificonf application:

```
\> wificonf -ibss_master_chan 1
WIWAVE1 Adapter Detected: Digi Wi-Wave Wireless LAN Adapter.
OID_WIFIMAC_GET_IBSS_MASTER_CHAN=0x3
OID_WIFIMAC_SET_IBSS_MASTER_CHAN set ibss_master_chan=0x1
```

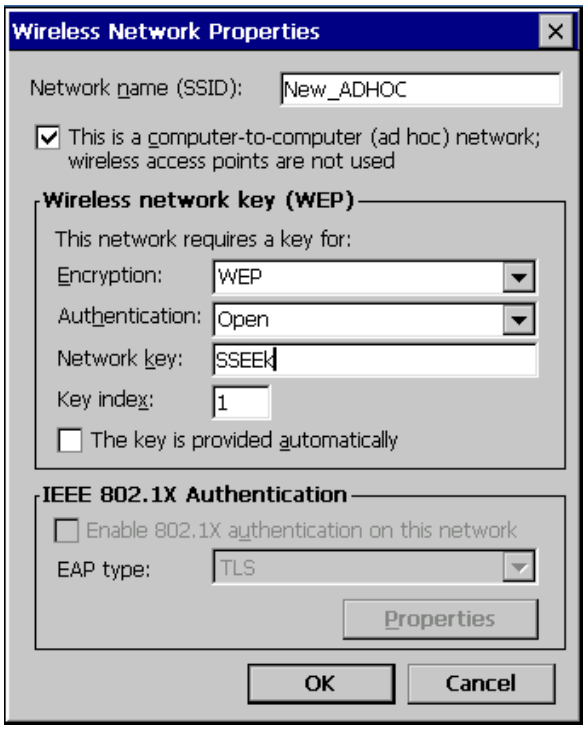
10.2.1. Graphic mode

As a new network, the SSID will not be in the list of available networks.

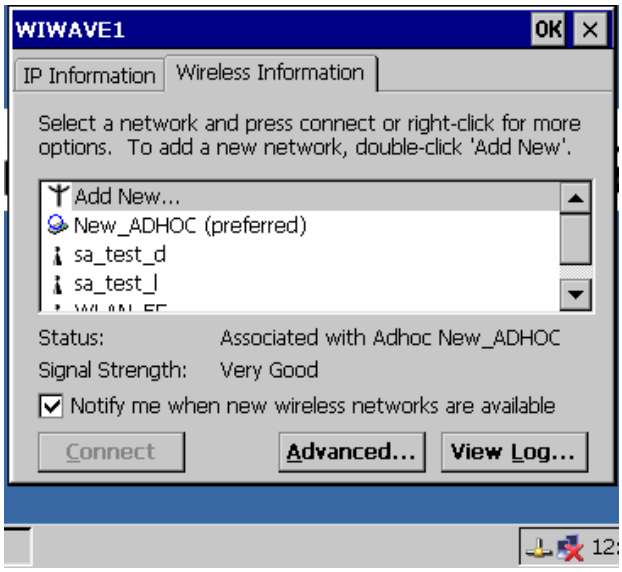
1. To ad a new network, double-click on ‘Add New.’ The Properties screen appear with blank fields.



2. Type in the new network name, and check the ‘This is a computer-to-computer ad hoc network’ checkbox, Select open authentication and the desired encryption (only Open and WEP are supported). Then press OK.

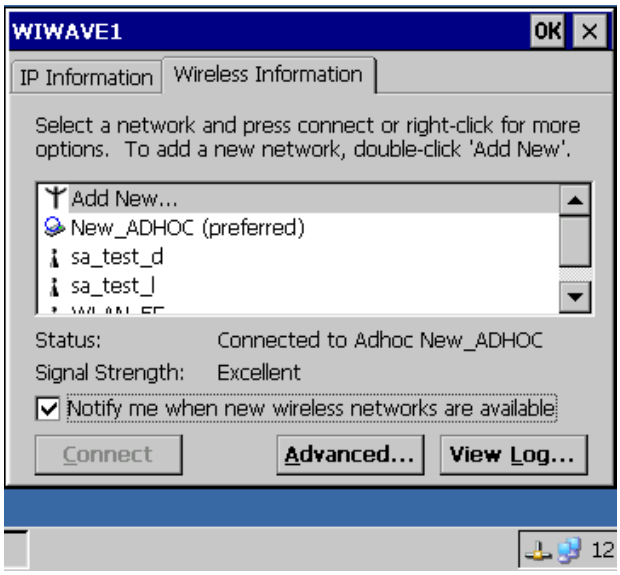


The status text beside the selected Access Point passes through several states and finishes with **Associated**. The WiFi Taskbar icon still appears with a red X. .



Our device has now stated to distribute the SSID. Now other devices can see it.

When other device connect with it, the status text changes into **Connected**, the WiFi Taskbar icon changes to blue and the red X disappears.



10.2.2. Command line mode

If a graphic display is not available, use the **wzctool** application to create a new ad hoc network. To create the same network as in the graphic example, the command is:

```
\> wzctool -c WIWAVE1 -adhoc -ssid New_ADHOC -auth open -encr wep -key
1/SSEEk
```

Our device has now stated to distribute the ssid. Now other devices can see it. The internal driver status at this point will be 5 (Started ad hoc IBSS).

```
\> wificonf -status
WIWAVE1 Adapter Detected: Digi Wi-Wave Wireless LAN Adapter.
OID_WIFIMAC_STATION_STATE= 5 (Started ad hoc IBSS)
```

When other device connect with us, the internal driver status changes to 4 (Joined ad hoc IBSS)

```
\> wificonf -status
WIWAVE1 Adapter Detected: Digi Wi-Wave Wireless LAN Adapter.
OID_WIFIMAC_STATION_STATE= 4 (Joined ad hoc IBSS)
```

11. Automatic connection to Preferred Networks after reset or Power-on

Microsoft Windows® CE will automatically try to connect to previous Preferred Networks if the Registry has been saved into NVRAM before turning off the board. To save the Registry permanently into NVRAM revise your BSP documentation.

If this feature is not working and your platform doesn't keep the filesystem after reset, please, verify that your current registry contain following entries:

```
[HKEY_LOCAL_MACHINE\Init\BootVars]
                "MasterKeysInRegistry"=dword:1
```

12. Authentication and encryption

As mentioned previously, authentication is the process of confirming the identity, and encryption is the process that makes information unreadable for unauthorized users. the Connect Wi-Wave WLAN interface Microsoft Windows® CE 6.0 driver supports the following methods:

12.1. Supported methods

Authentication	Infrastructure mode	Ad hoc mode
open	X	X
shared	X	X
WPA-PSK (WPA Personal)	X	
WPA2-PSK (WPA2 Personal)	X	
WPA (WPA Enterprise)	X	
WPA2 (WPA2 Enterprise)	X	

Encryption	Infrastructure mode	Ad hoc mode
no encryption	X	X
WEP 64/128 bits	X	X
TKIP	X	
AES-CCMP	X	

12.2. Authentication and encryption combinations

There are several combinations of authentication and encryption methods. When queried with the **wzctool**, the driver reports them as follows:

```
\> wzctool -q
wireless card found: WIWAVE1
   dwNumOfAuthEncryptPairs  : [11]
   Pair[1]
      AuthmodeSupported          [Ndis802_11AuthModeOpen]
      EncryptStatusSupported     [Ndis802_11WEPDisabled]
   Pair[2]
      AuthmodeSupported          [Ndis802_11AuthModeOpen]
      EncryptStatusSupported     [Ndis802_11WEPEnabled]
   Pair[3]
      AuthmodeSupported          [Ndis802_11AuthModeShared]
      EncryptStatusSupported     [Ndis802_11WEPEnabled]
   Pair[4]
      AuthmodeSupported          [Ndis802_11AuthModeWPA]
      EncryptStatusSupported     [Ndis802_11Encryption2Enabled]
   Pair[5]
      AuthmodeSupported          [Ndis802_11AuthModeWPA]
      EncryptStatusSupported     [Ndis802_11Encryption3Enabled]
   Pair[6]
      AuthmodeSupported          [Ndis802_11AuthModeWPAPSK]
      EncryptStatusSupported     [Ndis802_11Encryption2Enabled]
   Pair[7]
      AuthmodeSupported          [Ndis802_11AuthModeWPAPSK]
      EncryptStatusSupported     [Ndis802_11Encryption3Enabled]
   Pair[8]
      AuthmodeSupported          [Ndis802_11AuthModeWPA2]
      EncryptStatusSupported     [Ndis802_11Encryption2Enabled]
   Pair[9]
      AuthmodeSupported          [Ndis802_11AuthModeWPA2]
      EncryptStatusSupported     [Ndis802_11Encryption3Enabled]
   Pair[10]
      AuthmodeSupported          [Ndis802_11AuthModeWPA2PSK]
      EncryptStatusSupported     [Ndis802_11Encryption2Enabled]
   Pair[11]
      AuthmodeSupported          [Ndis802_11AuthModeWPA2PSK]
      EncryptStatusSupported     [Ndis802_11Encryption3Enabled]
```

Encryption2 is TKIP and Encryption3 is AES-CCMP.

These combinations can be grouped as follows:

- Open authentication and encryption.
- Open authentication with WEP encryption.
- WPA-PSK or WPA2-PSK authentication with TKIP or AES-CCMP encryption.
- WPA and WPA2 Enterprise authentication.

12.3. Open authentication without encryption

When queried with **wzctool –q**, an Access Point configured as previously described displays this information:

```
Length                = 196 bytes.
   dwCtlFlags         = 0x00000000
   MacAddress         = 00:13:46:9B:A8:53
   SSID               = sa_test_a
   Privacy            = 1  Privacy disabled (wireless data is not encrypted)
   RSSI               = -54 dBm (0=excellent, -100=weak signal)
   NetworkTypeInUse   = NDIS802_11FH
   Configuration:
      Struct Length    = 32
      BeaconPeriod     = 100 kusec
      ATIMWindow       = 0 kusec
      DSConfig         = 2472000 kHz (ch-13)
      FHConfig:
         Struct Length = 0
         HopPattern    = 0
         HopSet        = 0
         DwellTime     = 0
   Infrastructure      = Ndis802_11Infrastructure
   SupportedRates      = 1.0,2.0,5.5,11.0,6.0,9.0,12.0,18.0, (Mbit/s)
   KeyIndex            = <not available> (beaconing packets don't have this
                          info)
   KeyLength           = <not available> (beaconing packets don't have this
                          info)
   KeyMaterial         = <not available> (beaconing packets don't have this
                          info)
   Authentication      = 0  Ndis802_11AuthModeOpen
   rdUserData length   = 0 bytes.
```

12.3.1. Connect in graphic mode

To connect to an Access Point in graphic mode, select the Access Point and click **Connect**. Because the Access Point configuration is automatically recognized, these fields are already filled in:

- **Encryption: disabled**
- **Authentication: Open**

12.3.2. Connect in command line mode

To connect in command line mode, use the wzctool command, specifying only the SSID of the Access Point, which in the example, is **sa_test_a**:

```
\> wzctool -c WIWAVE1 -ssid sa_test_a -auth open -encr disabled
```

12.4. Open authentication with WEP encryption

When queried with **wzctool –q**, an Access Point configured with open authentication with WEP encryption with this configuration displays this information:

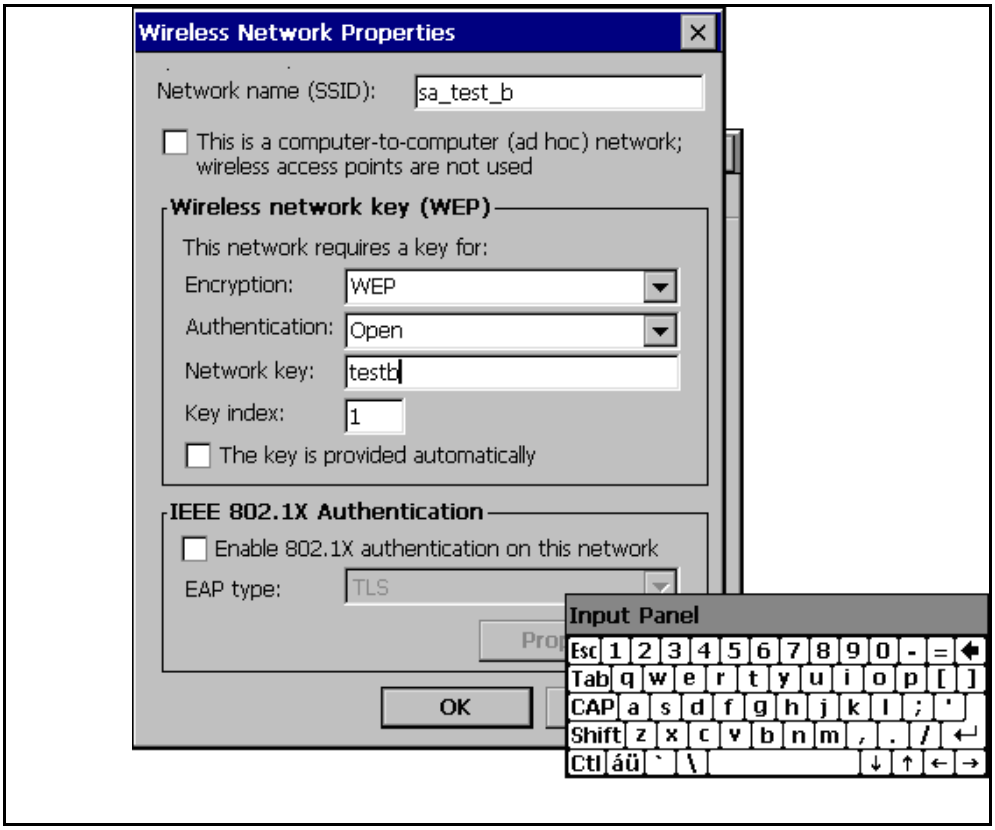
```
Length                  = 196 bytes.
   dwCtlFlags           = 0x00000000
   MacAddress           = 00:13:46:9B:A8:54
   SSID                 = sa_test_b
   Privacy                        =  0    Privacy  enabled  (encrypted  with
[Ndis802_11WEPEnabled])
   RSSI                 = -59 dBm (0=excellent, -100=weak signal)
   NetworkTypeInUse     = NDIS802_11FH
   Configuration:
      Struct Length     = 32
      BeaconPeriod      = 100 kusec
      ATIMWindow        = 0 kusec
      DSConfig          = 2472000 kHz (ch-13)
      FHConfig:
         Struct Length = 0
         HopPattern     = 0
         HopSet         = 0
         DwellTime      = 0
   Infrastructure       = Ndis802_11Infrastructure
   SupportedRates       = 1.0,2.0,5.5,11.0,6.0,9.0,12.0,18.0, (Mbit/s)
   KeyIndex             = <not available> (beaconing packets don't have this
                           info)
   KeyLength            = <not available> (beaconing packets don't have this
                           info)
   KeyMaterial          = <not available> (beaconing packets don't have this
                           info)
   Authentication       = 0  Ndis802_11AuthModeOpen
   rdUserData length    = 0 bytes.
```

12.4.1. Connect in graphic mode

To connect in graphic mode, select the Access Point and click **Connect**. Because the Access Point configuration is automatically recognized, these fields are already filled:

- **Encryption: WEP**
- **Authentication: Open**

The check box **The key is provided automatically** is selected by default. Deselect it and enter **Network Key** and the **Key index** values as needed for the Access Point. Use the virtual keyboard to input the password.



12.4.2. Connect in command line mode

To connect in command line mode, provide the SSID of the Access Point (in the example, **sa_test_a**) and the WEP password and password index (in the example, **index=1** and **password=testb**:

```
\> wzctool -c WIWAVE1 -ssid sa_test_b -auth open -encr wep -key 1/testb
```

The password can also be specified in hexadecimal format (**testb = 0x7465737462**):

```
\> wzctool -c WIWAVE1 -ssid sa_test_b -auth open -encr wep -key
1/0x7465737462
```

12.5. WPA-PSK authentication with TKIP encryption

This topic shows connecting to an Access Point configured with WPA-PSK authentication with TKIP encryption. The connection method for other combinations of authentication (WPA-PSK or WPA2-PSK)and encryption (TKIP or AES-CCMP) vary only in the parameters specified.

When queried with **wzctool –q,** an Access Point configured with WPA-PSK + TKIP displays this information:

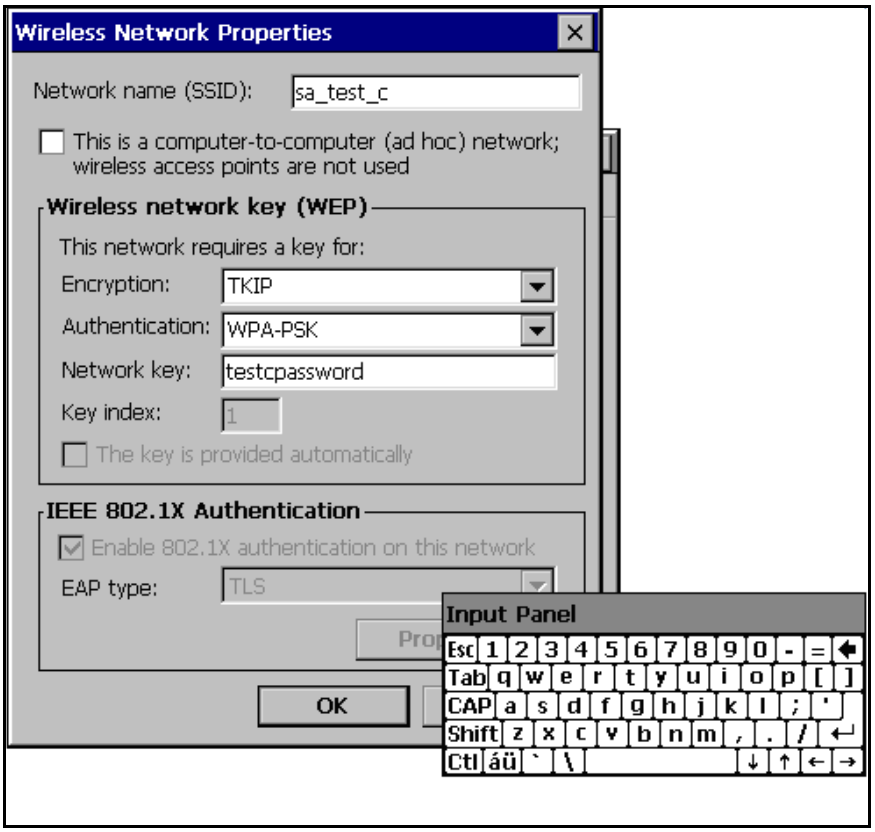
```
Length                  = 196 bytes.
   dwCtlFlags           = 0x00000010
   MacAddress           = 00:17:94:FD:99:C0
   SSID                 = sa_test_c
   Privacy                        =  4    Privacy  enabled  (encrypted  with
[Ndis802_11Encryption2Enabled])
   RSSI                 = -44 dBm (0=excellent, -100=weak signal)
   NetworkTypeInUse     = NDIS802_11FH
   Configuration:
      Struct Length     = 32
      BeaconPeriod      = 90 kusec
      ATIMWindow        = 0 kusec
      DSConfig          = 2437000 kHz (ch-6)
      FHConfig:
         Struct Length  = 0
         HopPattern     = 0
         HopSet         = 0
         DwellTime      = 0
   Infrastructure       = Ndis802_11Infrastructure
   SupportedRates       = 1.0,2.0,5.5,11.0,6.0,12.0,18.0,24.0, (Mbit/s)
   KeyIndex             = <not available> (beaconing packets don't have this
                          info)
   KeyLength            = <not available> (beaconing packets don't have this
                          info)
   KeyMaterial          = <not available> (beaconing packets don't have this
                          info)
   Authentication       = 4  Ndis802_11AuthModeWPAPSK
   rdUserData length    = 0 bytes.
```

12.5.1. Connect in graphic mode

To connect to the Access Point in graphic mode, select the Access Point and click **Connect**. Because the Access Point configuration is automatically recognized, these fields are already filled:

- **Encryption: TKIP**
- **Authentication: WPA-PSK**

Only the **Network key** field must be entered. Enter the pre-shared key that is configured in the Access Point. Use the virtual keyboard to input the password.



12.5.2. Connect in command line mode

To connect to the Access Point in command line mode, enter the **wzctool** command. Specify the SSID of the Access Point (in the example, **sa_test_c**) and the WPA-PSK password (in the example, **testcpassword**):

```
\> wzctool -c WIWAVE1 -ssid sa_test_c -auth wpa-psk -encr tkip -key
testcpassword
```

12.6. WPA2-PSK authentication with AES-CCMP encryption

WPA2-PSK authentication with AES-CCMP encryption is a different combination of the same method seen in topic 12.5.

When queried with **wzctool –q**, an Access Point configured with WPA2-PSK + AES-CCMP displays this information:

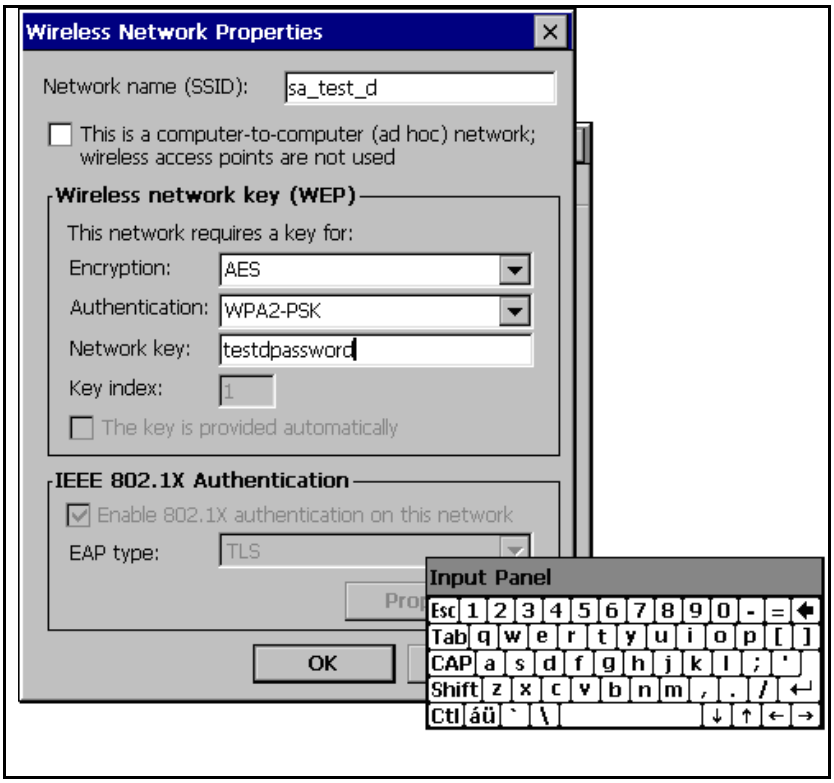
```
Length                  = 196 bytes.
   dwCtlFlags           = 0x00000010
   MacAddress           = 00:13:46:9B:A8:55
   SSID                 = sa_test_d
   Privacy              = 6  Privacy enabled (encrypted with [Ndis802_11Encr
yption3Enabled])
   RSSI                 = -37 dBm (0=excellent, -100=weak signal)
   NetworkTypeInUse     = NDIS802_11FH
   Configuration:
      Struct Length     = 32
      BeaconPeriod      = 100 kusec
      ATIMWindow        = 0 kusec
      DSConfig          = 2472000 kHz (ch-13)
      FHConfig:
         Struct Length  = 0
         HopPattern     = 0
         HopSet         = 0
         DwellTime      = 0
   Infrastructure       = Ndis802_11Infrastructure
   SupportedRates       = 1.0,2.0,5.5,11.0,6.0,9.0,12.0,18.0, (Mbit/s)
   KeyIndex             = <not available> (beaconing packets don't have this
                           info)
   KeyLength            = <not available> (beaconing packets don't have this
                           info)
   KeyMaterial          = <not available> (beaconing packets don't have this
                           info)
   Authentication       = 7  Ndis802_11AuthModeWPA2PSK
   rdUserData length    = 0 bytes.
```

12.6.1. Connect in graphic mode

To connect to an Access Point in graphic mode, select the Access Point and click **Connect**. The Access Point configuration is automatically recognized, so these fields are already filled:

- **Encryption: AES**
- **Authentication: WPA2-PSK**

Only the **Network key** field must entered. Enter the pre-shared key configured in the Access Point. Use the virtual keyboard to input the password.



12.6.2. Connect in command line mode

To connect to the Access Point in command line mode, enter the **wzctool** command. Specify the SSID of the Access Point (in the example, **sa_test_d**) and the WPA2-PSK password (in the example, **testdpassword**):

```
\> wzctool -c WIWAVE1 -ssid sa_test_d -auth wpa2-psk -encr aes -key
testdpassword
```

12.7. WPA Enterprise authentication

When queried with **wzctool –q,** an Access Point configured with WPA or WPA2 Enterprise displays this information:

```
Length                  = 196 bytes.
   dwCtlFlags           = 0x00000010
   MacAddress           = 00:13:46:9B:A8:55
   SSID                 = sa_test_d
   Privacy              = 6  Privacy enabled (encrypted with [Ndis802_11Encr
yption3Enabled])
   RSSI                 = -37 dBm (0=excellent, -100=weak signal)
   NetworkTypeInUse     = NDIS802_11FH
   Configuration:
      Struct Length     = 32
      BeaconPeriod      = 100 kusec
      ATIMWindow        = 0 kusec
      DSConfig          = 2472000 kHz (ch-13)
      FHConfig:
         Struct Length = 0
         HopPattern     = 0
         HopSet         = 0
         DwellTime      = 0
   Infrastructure       = Ndis802_11Infrastructure
   SupportedRates       = 1.0,2.0,5.5,11.0,6.0,9.0,12.0,18.0, (Mbit/s)
   KeyIndex             = <not available> (beaconing packets don't have this
                           info)
   KeyLength            = <not available> (beaconing packets don't have this
                           info)
   KeyMaterial          = <not available> (beaconing packets don't have this
                           info)
   Authentication       = 6  Ndis802_11AuthModeWPA
   rdUserData length    = 0 bytes.
```

12.7.1. Connect in graphic mode

To connect to the Access Point in graphic mode, select the Access Point and click **Connect**. The Access Point configuration is automatically recognized, so these fields are already filled in:

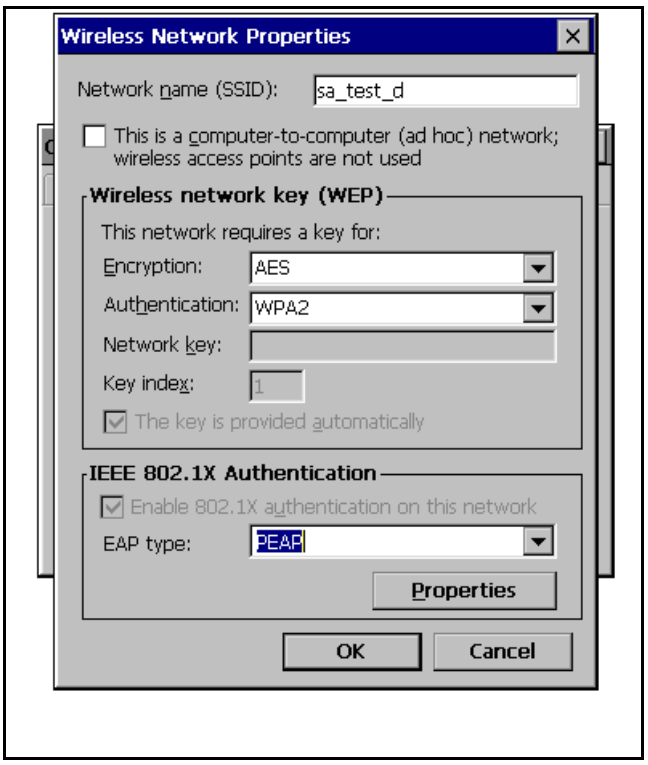
Encryption: AES

Authentication: WPA2

Then select the desired EAP type to use. The settings and requested information displayed depend on the EAP type.

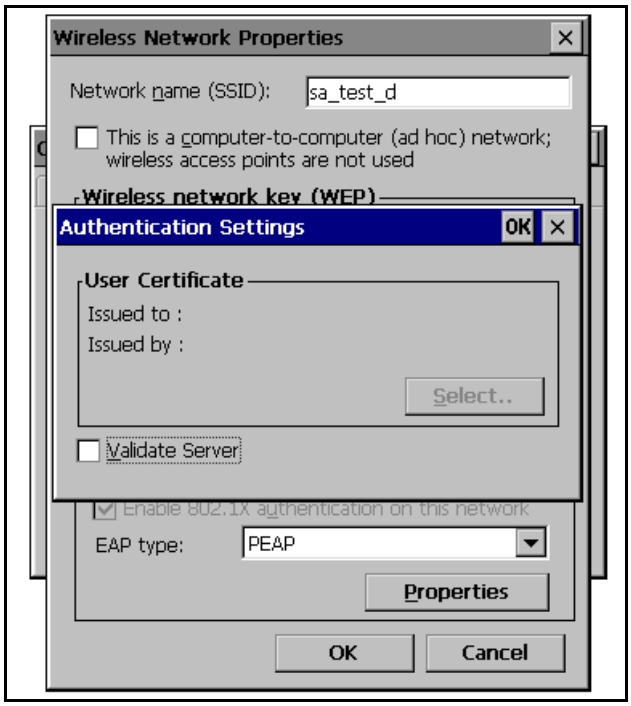
For PEAP:

From the **EAP type** combo box, select **PEAP**.

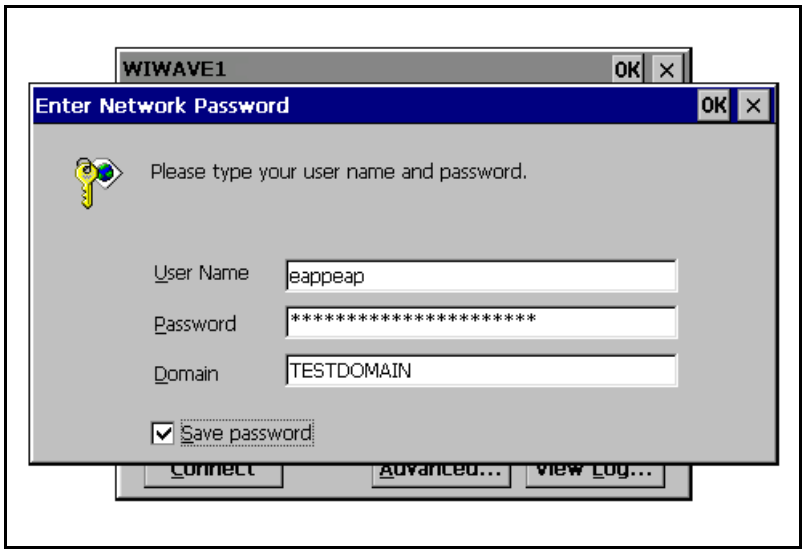


Click the **Properties** button. The **Authentication Settings** dialog is displayed. For the **Validate Server** checkbox, if the correct server certificates are installed, leave it selected. If Server Certificates are not installed or the server does not support this feature, deselect it.

The date of the system must be correctly set and match the server certificate valid period

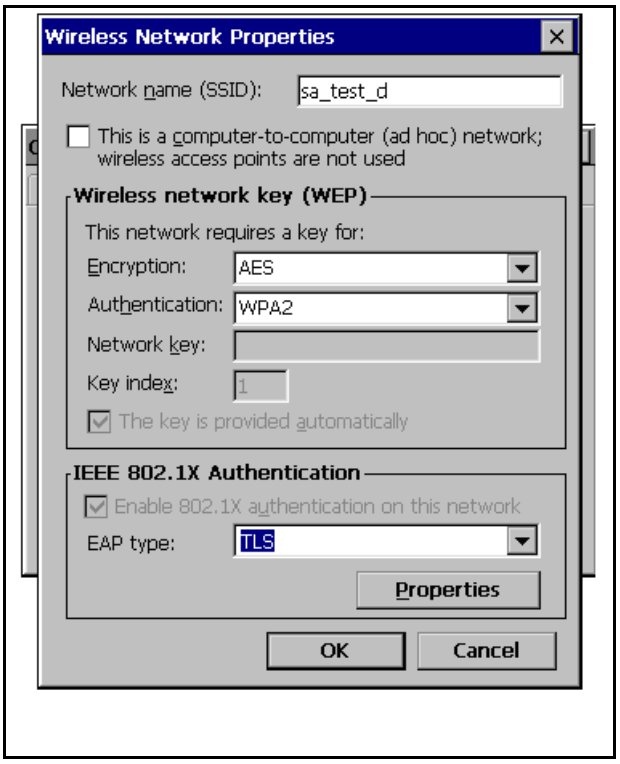


Click **OK** buttons twice. If the network has never been reached using this method, the system will prompt for **User Name**, **Password**, and **Domain**:



For TLS:

From the **EAP type** combo box, select **TLS**.

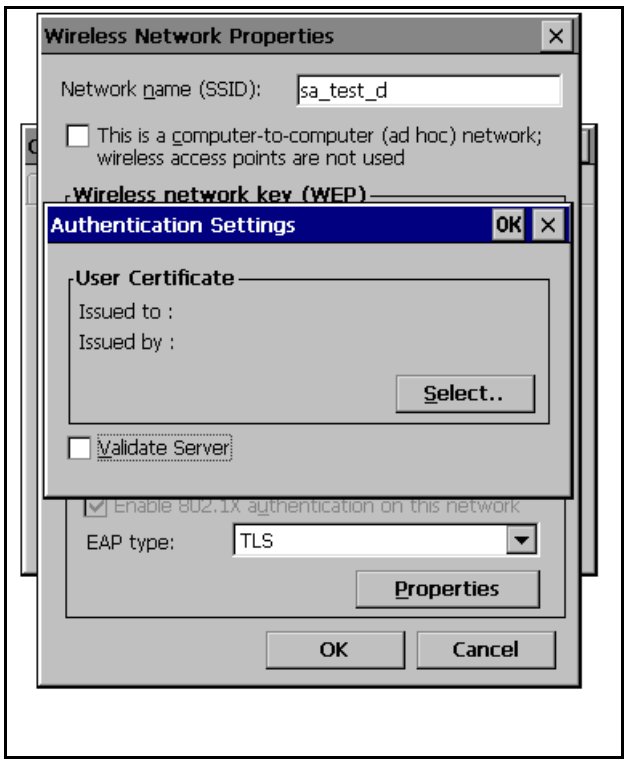


Click the **Properties** button.

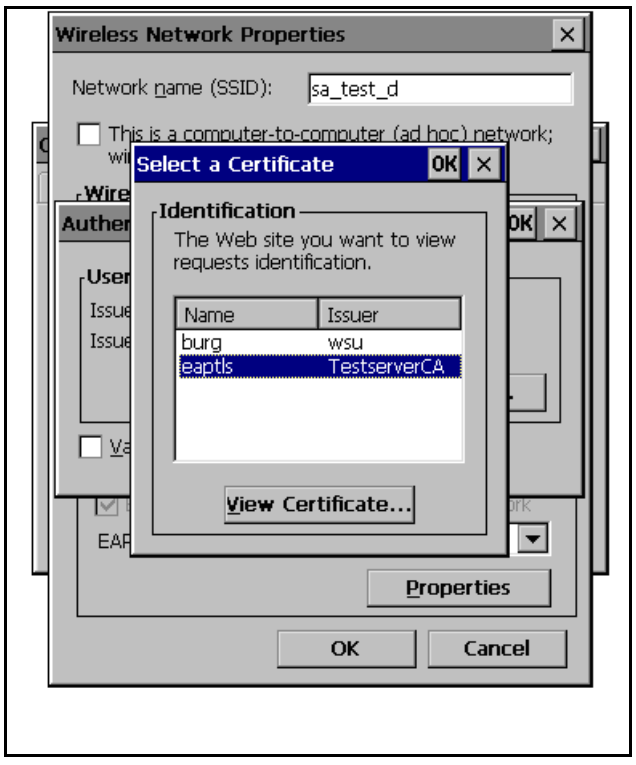
The **Authentication Settings** dialog is displayed. For the **Validate Server** checkbox, if the correct server certificates are installed, leave this setting selected. If Server Certificates are not installed or the server doesn't support this feature, deselect it.

The date of the system must be correctly set and match the server certificate valid period.

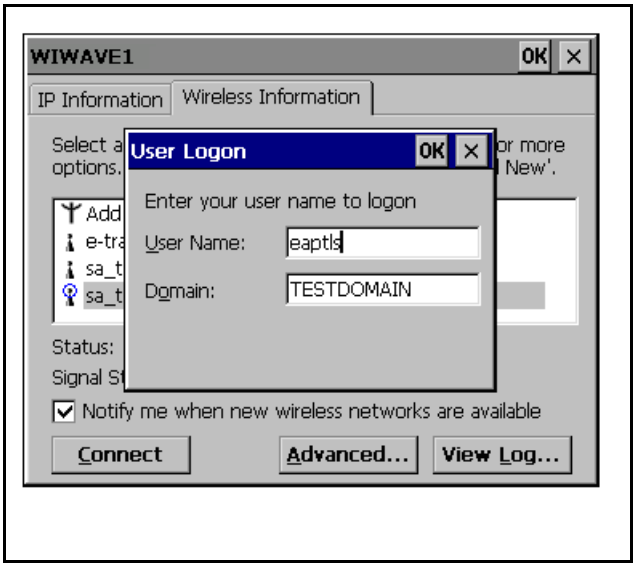
Click the **Select** button.



Select the Client Certificate to use.



Click **OK** buttons three times. If the network has never been reached using this method, the system will prompt for **User Name** and **Domain**:



Generating and installing certificates is done using a standard Microsoft Windows® CE process. Description of the process is out of the scope of this document.

13. Access Points supporting several authentication and encryption methods

Modern Access Points can be configured to support several authentication and encryption methods at the same time. For example, an Access Point can be configured to support, , any combination of WPA-PSK or WPA2-PSK authentication and TKIP or AES-CCMP encryption.

Such configuration offers the possibility to connect to the Access Point with any of the four resultant combinations at the same time.

14. Band and Channel Mask configuration

14.1. Overview

There are several bands a wireless adapter can support (a, b, g). Each band implies certain channels (frequencies), modulations, and baud rates. Currently, the Connect Wi-Wave supports:

	a	b	bg
Connect Wi-Wave		X	X

14.2. Configuration

The bands that are supported by the device are read by the driver from the system registry:

```
[HKEY_LOCAL_MACHINE\Drivers\USB\ClientDrivers\Wiwave]

  "band"=dword:03  ;0= All Bands, 1= Band A, 2= Band B, 3= Bands BG
```

They can also be read and changed at run-time using the wificonf application:

```
\> Wificonf -band
OID_WIFIMAC_GET_BAND= 3 (Bands BG)
\>
\> Wificonf -band 2
OID_WIFIMAC_GET_BAND= 3 (Bands BG)
OID_WIFIMAC_SET_BAND= 2 (Band B)
\>
```

The driver reads the Channel Masks from the registry:

```
[HKEY_LOCAL_MACHINE\Drivers\USB\ClientDrivers\Wiwave]

  "chan_mask"=dword:00003fff  ;Default All 2.4 GHz channels Enabled
  "chan_mask_high"=dword:00000000 ;Default= All Disabled
```

Channel Masks can be configured at run-time with the wificonf application:

```
\> Wificonf -chan_mask
OID_WIFIMAC_GET_CHAN_MASK=0x3fff
\>
\> Wificonf -chan_mask 0x30f0
OID_WIFIMAC_SET_CHAN_MASK=0x30f0
\>
\> Wificonf -chan_mask_high
OID_WIFIMAC_GET_CHAN_MASK_HIGH=0x0
```

The Channel Mask is composed of two 32bit registers: “chan_mask” and “chan_mask_high” (chan_mask_high is always 0 for the Connect Wi-Wave). Each bit represents a channel thus the mask is capable of specifying 64 channels (some channels do not exist).

Bit 0 of 'chan_mask' is channel 1, bit 31 of 'chan_mask' is channel 32.

Bit 0 of 'chan_mask_high' is channel 33 ...

A 1 in the mask means that the channel is enabled.

By default all supported bands and channels on the Connect Wi-Wave module are enabled.

When a band configuration changed all valid channels for that band are automatically enabled. To disable a channel just set the bit for that corresponding channel to ‘0’.

This table shows default channel mask after a band change:

Mask \ Band	0	1	2	3
	all	a	b	Bg
chan_mask	0xFFFF3FFF	0xFFFF0000	0x00003FFFF	0x00003FFFF
chan_mask_high	0x0007FFFF	0x0007FFFF	0x00000000	0x00000000

15. Ndisconfig Power Levels for Digi Wireless Adapters

15.1. Overview

Microsoft Windows CE 6.00 can now put the ndis driver into different power level state via a new 'power' option in the 'ndisconfig.exe' tool. The Connect Wi-Wave modules implement two levels. The following table shows which power state the Connect Wi-Wave will be in for each ndis power level.

dX	Connect Wi-Wave State
d0	ON
d1	ON
d2	ON
d3	OFF
d4	OFF

When the connect Wi-Wave is in the OFF state it will not receive or transmit any data.

The minimum FPGA revision on the Connect Wi-Wave that supports low power modes is:
HW Rev=0x030d
FW Rev=0x030d

15.2. Usage

Use 'ndisconfig power' command to query current state.

Use 'ndisconfig power set wiwave1 d3' to put the Connect Wi-Wave module in low power.

Use 'ndisconfig power set wiwave1 d0' to put the Connect Wi-Wave module in normal power state.

```
\> ndisconfig power
     Adapter   Power State
     -------   -----------
   WIWAVE1     D0
\>
\> ndisconfig power set wiwave1 d3
     Adapter   Power State
     -------   -----------
   WIWAVE1     D3
\>
\> ndisconfig power set wiwave1 d0
     Adapter   Power State
     -------   -----------
   WIWAVE1     D0
```

15.3. Power Measurements

Check on the Hardware Reference Manual for current and power comsumptioms at ON and OFF states.

15.4. WindowsCE 5.0 usage

In Microsoft Windows CE 5.0 the ndisconfig.exe 'power' option is not available. To remedy this the Wificonf.exe tool has been expanded to allow the user to put the Connect Wi-Wave into the supported power levels.

When using wificonf.exe to change driver power level, the operating system and NDIS layers are not aware of the new driver status. Unexpected functionality may arise.

'Ndisconfig power', if available, is preferred.

```
\> wificonf -power
OID_PNP_QUERY_POWER= d0
\>
\> wificonf -power d3
OID_PNP_QUERY_POWER= d0
OID_PNP_SET_POWER d3 Success
```

16. Power Safe Poll implementation for Digi Wireless Adapters

16.1. Overview

This mode allows the Connect Wi-Wave to go into a low power mode for configurable amounts of time and still allow it to send and receive data.

When entering the Power Safe Poll mode the Connect Wi-Wave will send a notification to the AP that it is entering this mode. The AP will then buffer the Connect Wi-Wave's packets until the Connect Wi-wave returns from low power mode to receive the data.

The Connect Wi-Wave will stay in low power mode for duty_cycle_off milliseconds. It will stay on for duty_cycle_on milliseconds.

Note: Power Save Poll is only active when the device is connected (associated or authenticated to an AP. Power Save Poll is not active while scanning channels or in Ad-Hoc mode.

The minimum FPGA revision on the Connect Wi-Wave that supports low power modes is:

HW Rev=0x030d
FW Rev=0x030d

Some Access Points doesn't support this functionality.

When using one of these Access Points, don't enable Power Save Poll on the wireless driver.

16.2. Configuration

The time the device is on and off is read by the driver from the registry:

```
 [HKEY_LOCAL_MACHINE\Comm\Wiwave1\Parms]

;MAC configurable params (Enter values in HEX)
   "duty_cycle_on"=dword:000000c8  ;Default 200mS, Minimun 100mS
   "duty_cycle_off"=dword:000000c8 ;Default 200mS, Maximum 500mS
```

It can be read and changed at run time using the wificonf application:

```
\> Wificonf -duty_cycle_on
OID_WIFIMAC_GET_DUTY_CYCLE_ON=200
\>
\> Wificonf -duty_cycle_on 500
OID_WIFIMAC_GET_DUTY_CYCLE_ON=200
OID_WIFIMAC_SET_DUTY_CYCLE_ON set duty_cycle_on=500
\>
\> Wificonf -duty_cycle_off
OID_WIFIMAC_GET_DUTY_CYCLE_OFF=200
\>
```

```
\> Wificonf -duty_cycle_off 500
OID_WIFIMAC_GET_DUTY_CYCLE_OFF=200
OID_WIFIMAC_SET_DUTY_CYCLE_OFF set duty_cycle_off=500
```

There are some restrictions on these times:

Time (mS)	Max	Min	Default
On	na	100	200
Off	500	na	200

Bit 8 of the 'options' parameter in the registry toggles this feature on and off, by default it is disabled:

```
 [HKEY_LOCAL_MACHINE\Comm\Wiwave1\Parms]

;Wireless option parameter values
;
; 0x00000001   Enable antenna diversity
; …
; 0x00000100   Enable 802.11 Power Save Poll
;
   "options"=dword:00000000                ;Default= 00000000
```

It can be read and changed at run time using the wificonf application:

```
\> Wificonf -options
OID_WIFIMAC_GET_OPTIONS=0x0
\>
\> Wificonf -options 0x100
OID_WIFIMAC_GET_OPTIONS=0x0
OID_WIFIMAC_SET_OPTIONS set options=0x100
```

Remember to use 'wificonf -save_params' command and save the registry in order to keep changes done with the wificonf.exe application persistent.

16.3. Performance

Due to the nature of Power Safe Poll mode networking responsiveness and speed will be sacrificed for better power consumption. This is obviously due to the fact that the device will not be receiving or sending data during the duty_cycle_off periods.

To get a general idea of the kinds of effects this will have on network traffic a simple test was performed using the FTP server.

First, a control run was done with the feature disabled. A file was transferred to and from the device to record a benchmark.

Then Power Save Poll mode was switched on with different settings and the test was applied again to see the different speeds.

Duty Cycle (%) On (mS) Off (mS)	100 (No PS) na na	33 200 400	50 200 200	50 400 400
Upload Percentage	na	25%	39%	43%
Download Percentage	na	29%	42%	47%

16.4. Power Measurements

Current measurements at on and off states are exactly the same as the ones measured in previous chapter 15.3 'Ndisconfig Power levels'. Check them on the Hardware Reference Manual.

17. Driver Unload

17.1. Using 'ndisconfig adapter' commands

The Digi Connect Wi-Wave driver can be unloaded at run time using following console command:

```
Ndisconfig adapter del wiwave1
```

You should see a console message similar to the following:

```
[Wi-Wave]: UnLoading Wireless Driver… OK.
```

More information may be shown depending on the debug level established in the driver.

If a display is available, the small connection icon in the taskbar should disappear.

If a display is not available, the driver unload can be verified by entering the following console commands:

```
\> wzctool -e
System has no wireless card.
```

The Digi Connect Wi-Wave card will be left in low power mode like in the off state measured in previous chapter 15.3 'Ndisconfig Power levels'.

The Digi Connect Wi-Wave driver can be loaded again at run time using following console command:

```
Ndisconfig adapter add wiwave wiwave1
```

* Note that first wiwave entrie doesn't have the '1' index.

17.2. Removing usb cable or power supply.

If the cable connecting the Digi Connect Wi-Wave card with the usb host connector of the main board is unplugged, or the power supply of the Digi Connect Wi-Wave card is removed, the usb host driver will notify the driver that will automatically unloaded.

You should see a console message similar to the following:

```
[Wi-Wave]: UnLoading Wireless Driver… OK.
```

More information may be shown depending on the debug level established in the driver.

If a display is available, the small connection icon in the taskbar should disappear.

If a display is not available, the driver unload can be verified by entering the following console commands:

```
\> wzctool -e
System has no wireless card.
```

Obviously, the Digi Connect Wi-Wave driver can be loaded again at run time by plugging the cable or powering the Digi Connect Wi-Wave card.

18. Wireless configuration tool

WifiConf.exe is a custom application provided to give access to non-standard functionality/features of the driver that cannot be accessed through standard Microsoft Windows® CE tools like graphic **NetUI** or command-line **wzctool**.

This application is in the Windows folder of the target

To execute the application and see its syntax, type **wificonf** in a console or telnet session:

```
\> wificonf
Application to configure and save the wireless settings. Revision 1.4
Copyright(c) 2008 by Digi International Inc.

Usage: WifiConf [Adapter] <options>

Where Adapter:
If only one Digi Wireless Adapter is used, can be omitted.
If more than one Digi Wireless Adapter is used, use one of Supported
adapters:
WIWAVE1, CCW9CWIFI1 or CCW9MWIFI1.

Where options are:
  -status
        Displays driver internal current status.

  -stats clear|read
        Clears or reads statistics.

  -tx_power [HexValue]
        Reads/Sets tx_power.

  -band [DecValue]
       Reads/Sets band: 0=all bands, 1=band A, 2= band B, 3=bands BG.
  -chan_mask [HexValue]
        Reads/Sets chan_mask.
  -chan_mask_high [HexValue]
        Reads/Sets chan_mask_high.
  -ibss_master_chan [DecValue]
        Reads/Sets ibss_master_chan.
  -tx_rate [DecValue]
        Reads/Sets tx_rate.
  -rts_thresh [DecValue]
        Reads/Sets rts_thresh.
  -frag_thresh [DecValue]
        Reads/Sets frag_thresh.
  -duty_cycle_on [DecValue]
       Reads/Sets duty_cycle_on.
  -duty_cycle_off [DecValue]
       Reads/Sets duty_cycle_off.
  -options [HexValue]
        Reads/Sets options.

  -save_params
        Save wireless settings in system registry.

  -fpga_rev
        Displays FPGA Revision.

  -mac [xx:xx:xx:xx:xx:xx]
        Reads/Sets mac address stored in eeprom (only for Connect Wi-Wave
module).

Examples:
WifiConf -status
```

```
        Reads and display driver status
WifiConf -stats read
        Reads and display driver statistics
WifiConf -chan_mask
        Reads current value of chan_mask. Ex: 0x1002=Only ch13 & ch2 enabled
WifiConf -chan_mask 0x1003
        Sets value of chan_mask to 0x1003: ch13, ch2 and ch1 enabled
```

18.1. Digi Wireless Adapters

There will normally only be one Digi wireless Adapter available on a module. Then [Adapter] can be omitted from the command line and the available one will be used automatically:

```
\> wificonf –options
WIWAVE1 Adapter Detected: Digi Wi-Wave Wireless LAN Adapter.
WIWAVE1 Adapter Used:      Digi Wi-Wave Wireless LAN Adapter.
OID_WIFIMAC_GET_OPTIONS=0x0
\>
```

In case a WIWAVE Adapter is plugged to the usb connector of a Digi ConnectCore module with wireless support, if [Adapter] is omitted the Wificonf application will request a specific Adapter:

```
\> wificonf –options
WIWAVE1 Adapter Detected: Digi Wi-Wave Wireless LAN Adapter.
CCW9MWIFI1 Adapter Detected: Digi ccw9mWifi Wireless LAN Adapter.
More than one Adapter Detected. Specify desired one as first argument.
\>
\> wificonf wiwave1 -status
WIWAVE1 Adapter Used: Digi Wi-Wave Wireless LAN Adapter.
OID_WIFIMAC_GET_OPTIONS=0x0
\>
```

18.2. Display wireless status information

To display status information, type in this command:

```
\> wificonf –status
WIWAVE1 Adapter Detected: Digi Wi-Wave Wireless LAN Adapter.
OID_WIFIMAC_STATION_STATE=2 (Associated with ESS)
OID_WIFIMAC_GET_CURRENT_TX_RATE= 540 → 54000 kbps
OID_802_11_RSSI= -59 dBm (Very Good)
OID_802_11_BSSID=00:17:94:fd:99:c0
\>
```

18.3. Display transmission driver statistics

This command reads, or clears, reception and transmission driver statistics, which are more detailed than standard NDIS statistics:

```
\> wificonf -stats clear
WIWAVE1 Adapter Detected: Digi Wi-Wave Wireless LAN Adapter.
OID_WIFIMAC_RESET_STATS Statistics Reseted
\> wificonf -stats read
WIWAVE1 Adapter Detected: Digi Wi-Wave Wireless LAN Adapter.
OID_WIFIMAC_GET_STATS:
txBytes       = 105872866
txFrames      = 80694
txBCFrames    = 802
rxBytes       = 7253725
rxFrames      = 46558
```

```
rxBCFrames     = 46236
txRTS          = 0
txRetries      = 8375
txDropRetry    = 0
txDropBC       = 0
txDropAssoc    = 18
rxRTS          = 0
rxRetries      = 448
rxDropSize     = 0
rxDropBuffer   = 0
rxDropInvalid = 86
rxDropDup      = 319
rxDropAge      = 0
rxDropDecrypt = 53
rxDropOverrun = 0
rxDropReplay   = 0
rxDropMICFail = 53
\>
```

18.4. Display FPGA Revision

To display the revision of the FPGA, type in this command:

```
\> wificonf –fpga_rev
WIWAVE1 Adapter Detected: Digi Wi-Wave Wireless LAN Adapter.
OID_WIFIMAC_GET_FPGA_REV. HW Rev=0x030d
OID_WIFIMAC_GET_FPGA_REV. FW Rev=0x030d
\>
```

18.5. Display and Modify permanent MAC Address stored on Wi-Wave eeprom

To read the current MAC address, type this command without any argument:

```
\> wificonf –mac
WIWAVE1 Adapter Detected: Digi Wi-Wave Wireless LAN Adapter.
OID_WIFIWAVE_GET_MAC= 00:40:9d:69:00:03
\>
```

To modify the current parameter, add a value to the command line:

```
\> wificonf –mac 00:40:9d:69:00:05
WIWAVE1 Adapter Detected: Digi Wi-Wave Wireless LAN Adapter.
OID_WIFIWAVE_GET_MAC= 00:40:9d:69:00:03
OID_WIFIWAVE_SET_MAC Success
\>
```

Before Driver version 1.5, MAC address was stored and read in the eeprom using little endian format. From 1.5 on, it's done in big endian.

Conversion of modules running previous versions of the driver are done automatically first time driver version 1.5 starts.

Following message will be seen:

```
[Wi-Wave]: MacUsbReadEEPROM recovering from Little Endian.
```

Make sure your device reports the same address printed on the module label. Should follow this pattern: 00:40:9d:xx:xx:xx

18.6. Commands for configuring driver parameters

To display or modify driver parameters, use these commands:

- WifiConf tx_power [HexValue]
- WifiConf band [DecValue]
- WifiConf chan_mask [HexValue]
- WifiConf chan_mask_high [HexValue]
- WifiConf ibss_master_chan_ [DecValue]
- WifiConf tx_rate [DecValue]
- WifiConf rts_thresh [DecValue]
- WifiConf frag_thresh [DecValue]
- WifiConf duty_cycle_on [DecValue]
- WifiConf duty_cycle_off [DecValue]
- WifiConf options [HexValue]

Some commands take decimal values, while others take hexadecimal values.

To read the current parameter value, type this command without any argument:

```
\> WifiConf -chan_mask
WIWAVE1 Adapter Detected: Digi Wi-Wave Wireless LAN Adapter.
OID_WIFIMAC_GET_CHAN_MASK=0x1001
\>
```

To modify the current parameter, add a value to the command line:

```
\> WifiConf -chan_mask 1003
WIWAVE1 Adapter Detected: Digi Wi-Wave Wireless LAN Adapter.
OID_WIFIMAC_GET_CHAN_MASK=0x1001
OID_WIFIMAC_SET_CHAN_MASK set chan_mask=0x1003
\>
```

18.7. Store parameters to Registry

If any parameter of the WLAN interface is modified with the WifiConf tool, the new configuration can be stored into the Registry key [**HKEY_LOCAL_MACHINE\Comm\wiwave1\Parms**] with this command:

```
\> WifiConf -save_params
WIWAVE1 Adapter Detected: Digi Wi-Wave Wireless LAN Adapter.
Configuration saved in registry. Remember to save registry in persistent
storage!!!
\>
```

With this command, the new settings are stored in the Registry in RAM. To save this Registry permanently into NVRAM revise your BSP documentation.

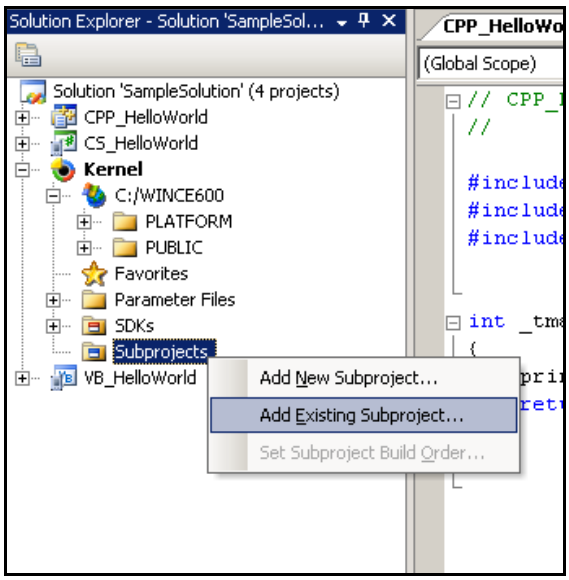
18.8. Source code for WifiConf

The **WifiConf** source code is at %_WINCEROOT%\OTHERS\Digi\appkits\WiWave\src\WifiConf folder. Either modify it or use it as example to create other applications that access the WLAN driver.

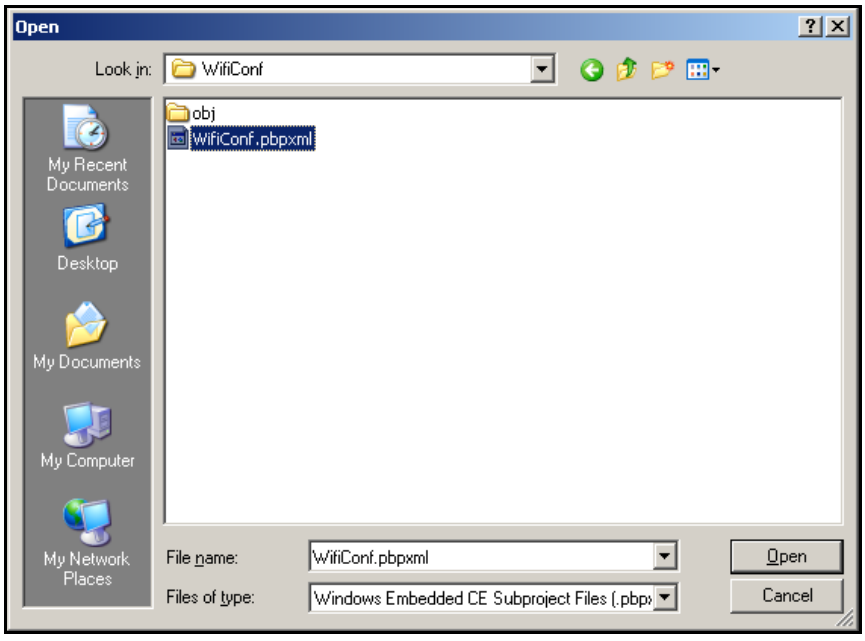
18.9. Build the WifiConf tool

WiFiConf can be built as a Microsoft Windows® CE native code C++ application. To add the **WifiConf** application to an OS design:

1. Select the OS design in the Solution explorer (in this case, **Kernel**) and select the **Subprojects** element in it. Right-click and select **Add Existing Subproject.**



2. Open the folder containing the **WifiConf** source code, select file **WifiConf.pbpxml** and click **Open**.



The **WifiConf** tool is added to the OS design.

18.9.1. About the code

The application gets access to the standard NDIS Wi-Wave driver through the NDISUIO interface. After the interface is opened, both standard and not standard OIDs can be executed. The standard OIDs are those described in the standard NDIS specification, such as OID_802_11_SSID, OID_802_11_INFRASTRUCTURE_MODE, etc.

The driver's non standard OIDs are described in the file

%_WINCEROOT%\OTHERS\Digi\appkits\WiWave\src\driver\oidwifimac.h. Description of the OIDs follow.

Name	Description	Related Registry entry	Range
OID_WIFIMAC_GET_TX_POWER OID_WIFIMAC_SET_TX_POWER	Transmit Power	"tx_power"	0 to 15
OID_WIFIMAC_SET_BAND OID_WIFIMAC_GET_BAND	Wireless Band	"band"	0=All 1=A 2=B 3=BG
OID_WIFIMAC_SET_CHAN_MASK OID_WIFIMAC_GET_CHAN_MASK	Bitmap of allowed channels. Low Part: Ch1 to 32	"chan_mask"	Bit 0 is channel 1, and so on.
OID_WIFIMAC_SET_CHAN_MASK_HIGH OID_WIFIMAC_GET_CHAN_MASK_HIGH	Bitmap of allowed channels. High Part: Ch33 to 64	"chan_mask_high"	Bit 0 is channel 33, and so on.
OID_WIFIMAC_SET_IBSS_MASTER_CHAN OID_WIFIMAC_GET_IBSS_MASTER_CHAN	Channel to use when creating new AdHoc connections	"ibss_master_chan"	1 to 13
OID_WIFIMAC_GET_TX_RATE OID_WIFIMAC_SET_TX_RATE	Maximum transmit rate (in units of 100 kbps), so 540 == 54 mbps	"tx_rate"	Max 540
OID_WIFIMAC_GET_RTS_THRESH OID_WIFIMAC_SET_RTS_THRESH	RTS threshold, 0 to use default.	"rts_thresh"	0 to 2347
OID_WIFIMAC_GET_FRAG_THRESH OID_WIFIMAC_SET_FRAG_THRESH	Fragmentation threshold, 0 to use default.	"frag_thresh"	0 to 2346
OID_WIFIMAC_GET_DUTY_CYCLE_ON OID_WIFIMAC_SET_DUTY_CYCLE_ON	If Power Save Poll is supported and enabled (options bit 0x100): Milliseconds at On state. Default 200 mS	"duty_cycle_on"	Min 100
OID_WIFIMAC_GET_DUTY_CYCLE_OFF OID_WIFIMAC_SET_DUTY_CYCLE_OFF	If Power Save Poll is supported and enabled (options bit 0x100): Milliseconds at Off state. Default 200 mS	"duty_cycle_off"	Max 500
OID_WIFIMAC_GET_OPTIONS OID_WIFIMAC_SET_OPTIONS	Bitmap of options: 0x0001 Enable antenna diversity	"options"	N/A

	0x0002 Enable short preamble 0x0004 Enable server certificate verification 0x0008 Use only 802.11b rates in 2.4 GHz band 0x0010 Use RTS/CTS protection frames for 802.11g 0x0020 Use fixed transmit rate 0x0040 Enable 802.11 Multi domain capability(802.11d) 0x0080 Antenna Selection 0x0100 Enable 802.11 Power Save Poll		
OID_WIFIMAC_GET_STATS OID_WIFIMAC_RESET_STATS	Get/Reset Wi-Wave internal statistics	N/A	N/A
OID_WIFIMAC_STATION_STATE	Gets internal driver state: WLN_ST_STOPPED, WLN_ST_SCANNING, WLN_ST_ASSOC_ESS, WLN_ST_AUTH_ESS, WLN_ST_JOIN_IBSS, WLN_ST_START_IBSS	N/A	0 to 5
OID_WIFIMAC_GET_CURRENT_TX_RATE	Get current TX rate	N/A	Max 540
OID_WIFIMAC_GET_FPGA_REV	Get FPGA revision	N/A	N/A
OID_WIFIWAVE_GET_MAC OID_WIFIWAVE_SET_MAC	Get/Set Wi-Wave MAC address stored in eeprom	N/A	N/A
OID_WIFIMAC_ROAMING_OTHER_BSS	Forces Driver to Roam	N/A	1 to 2

Index