THE REGISTRY

The registry stores settings and options for Windows CE. It contains data about hardware, software (including much non-OS software), drivers, users, preferences, etc. It is stored in a binary hive format; its files are stored in the path **\SystemCF\Settings**.

DESCRIPTION OF THE THREE TYPES OF REGISTRIES

- Default Registry. This is the registry that is loaded the first time you boot the unit. It is the registry that will come up until you make modifications to Windows CE. The four default registry hive keys (as listed in Windows CE) are:
    - HKEY_LOCAL_MACHINE: Contains hardware and driver configuration data.
    - HKEY_CURRENT_USER: Contains user configuration data.
    - HKEY_USERS: This hive is reserved for future use.
    - HKEY_CLASSES_ROOT: Contains OLE and file-type matching configuration data.
- Live Registry. This is simply the registry that is running when you start the unit. It is the default registry or a persistent registry, depending on whether you have made any modifications or not.
- Persistent Registry. This is a default registry that has been modified. If you make any changes within the Control Panel, or through applications, this will affect the registry. When saving the registry or by re-starting the unit, the Persistent Registry will now take over. It is important to note that some information will not be written to the registry by initiating a warm boot. In these cases, you must manually save the Persistent Registry. An example of this would be calibrating the display. (See below for more information on how to save the Persistent Registry.

Note: *It is important to note that only one type of registry can be running at any time. Regardless of whether it is the default registry or the persistent registry, it will contain the same hive keys.*

MANUALLY SAVING THE PERSISTENT REGISTRY

The FC-2500 internal memory consists of DRAM and Flash. Typically, any changes made to the FC-2500 including file creation are temporarily stored in the unit's DRAM. You must then copy the files from DRAM to internal flash memory to store the information permanently.

Consequently, if you do not store the information to flash memory and the unit loses power, all information stored in DRAM will be lost. However, whenever you make changes that affect the registry, such as changing settings in the Control Panel or installing software, you can permanently store registry changes without writing to flash memory by using the Persistent Registry.

Note*: The FC-2500 will also store registry information every time you perform a suspend/resume operation.*

To store registry information on the FC-2500 permanently:

1. From the Start menu, select **Programs > Tools> SaveReg**.
2. The FC-2500 will begin saving the registry.

3. After you successfully save the registry, a message box will appear:

4. Tap **OK** to close the message box.

Note: *For more information on how to create and program a Persistent Registry, see the following link:* *http://support.microsoft.com/kb/322269*

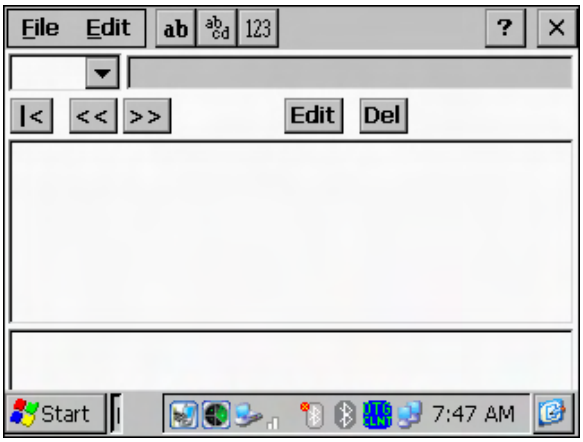
EDITING THE REGISTRY

In some cases, you may need to manually edit particular registry entries. This can be done in two ways: with an application called Regedit.exe, or by editing the registry remotely with the Remote Registry Editor (part of the Visual Studio 2005 suite).

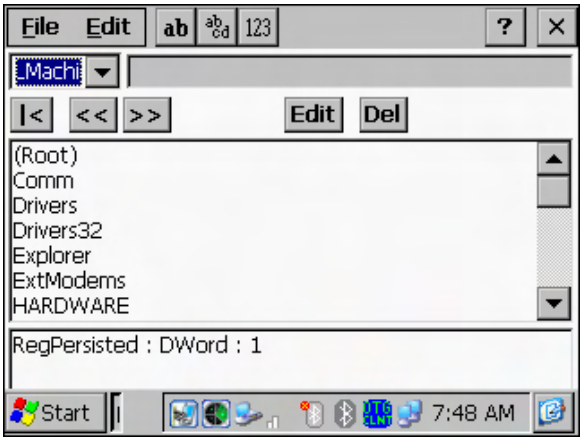
Note: *It is a good idea to backup the registry before making modifications to it. See the section below entitled: "Exporting the Registry" for an explanation on how to do this.*

EDITING WITH REGEDIT.EXE

1. To edit the registry with the Regedit.exe application simply tap **Start>Run** and type **`regedit`.** This will bring up the Registry Editor as shown below.



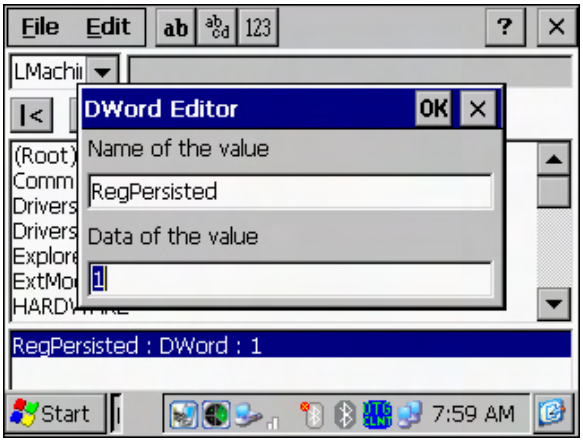
2. To access the appropriate hive key, click on the drop down menu towards the upper left. In the example below, we have selected the HKEY_LOCAL_MACHINE hive.



3. To edit an entry, simply double click it. In this example we will double click the entry at the bottom of the display named "RegPersisted : DWord : 1".

Note: *This is only an example, be very sure you know exactly what you want to modify before making any changes to the Registry. Once again, backup the registry before making any changes!*

4. This will bring up the DWord Editor popup window as shown below.



5. At this point, we can turn this On or Off, by changing the Data of the value to one or zero respectively. The next time you restart the unit, any changes to the registry will take affect. A way to "force" these changes would be to save the Persistent Registry as mentioned previously. Keep in mind that this is only one of several data types that you can modify.

EDITING THE REGISTRY REMOTELY

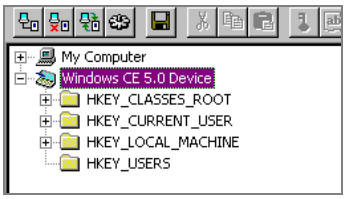
You can also edit the registry of the FC-2500 from a Personal Computer running Microsoft Windows. The benefits of this are the ability to use your PC's keyboard, mouse, and larger display, which collectively allow for faster editing time. However, this requires that Visual Studio 2005 and ActiveSync are installed on the PC and that the unit is connected to the PC.

1. Verify that your FC-2500 is connected to the PC (the best way to do this is via USB), and be sure that the device is recognized by ActiveSync.

2. On the PC, click **Start>All Programs>Microsoft Visual Studio 2005>Visual Studio Remote Tools>Remote Registry Editor**. The Windows CE Remote Registry Editor application will start up and a popup window should appear automatically like the one shown below.

3. Select the Windows CE 5.0 device as shown above and click OK. This should make a connection with the unit. It may take up to a minute.

4. At this point you should see at least two line items: My Computer and the Windows CE 5.0 Device. By expanding the + sign for the Windows CE device (shown below) you will see the different hives, and can edit the registry the way you normally would.



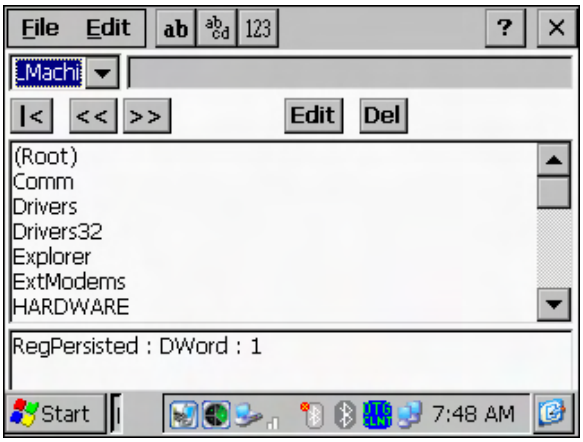
Note: *Make sure you are making edits to the Windows CE device and not My Computer (your local PC). In addition, you will note that the hives are not listed in the same order as they were in the Regedit.exe program on the FC-2500. In the Remote Registry Editor they are listed alphabetically.*

Note: *Another great tool for working with the FC-2500 remotely is a third party application by SOTI, Inc. called Pocket Controller- Pro. For more information, click the following link:*
http://www.soti.net/default.asp?Cmd=Products&SubCmd=PCPro

EXPORTING THE REGISTRY

You can export parts of the registry or the entire registry as you see fit. This is a great way to store a backup of your working registry, and is highly recommended as a safety precaution prior to making edits within the registry. It is important to export this data to a non-volatile area (SystemCF, USB flash drive, or to the network).

1. On the FC-2500 tap **Start>Run>**and type **`Regedit`**. This will bring up the registry editor.
2. Select a hive that you want to backup. In this case we will select the HKEY_LOCAL_MACHINE as shown below.



3. Select the Key, or SubKey that you want to export by tapping on it once. To export the entire HKEY_LOCAL_MACHINE hive you will need to tap **(Root),** which we will use in the example.
4. Then tap **File>Export**. This will bring up the Save As window. Again, make sure to export the data to a non-volatile area of the FC-2500, in this example we are selecting SystemCF as shown below.
5. Name the export file, in this example **`export.reg`** and click OK. This may take up to a minute to save, depending on the key that you selected. It is good practice to name the export file after the key that is being exported, for example, Drivers32.

Note: *The registry files (located at* ***\SystemCF\Settings****) can also be copied to another area of the SystemCF or to a USB flash drive to facilitate a "backup" of the registry.*

IMPORTING THE REGISTRY

To import a portion of the registry, complete the following steps:

1. On the FC-2500 tap **Start>Run>**and type **`Regedit`**. This will bring up the registry editor.
2. Select a location that you want to import to. This needs to be the same key that you selected when you ran the export. Make sure that the particular key is highlighted.
3. Tap **File>Import**.
4. Browse for and select the proper .reg file, then click OK. This will import that portion of the registry, it may take up to a minute pending on the key that you selected for import.

INVALIDATING THE REGISTRY

In some cases you may want to reset the registry back to the Default. This is common if you have a Persistent Registry that has failed (and never exported the registry in the past). To do this, you will need to Invalidate the Registry. Follow the steps below to do this.

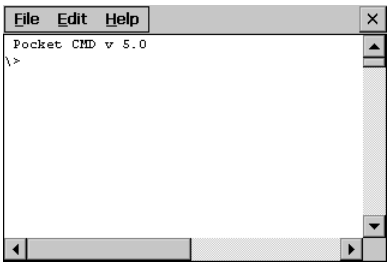
1. Shut down the unit by pressing and holding the power button for 10 seconds. Then release the power button.
2. Briefly press and let go of the power button (**and immediately follow the next step**)
3. Press and hold the SHIFT and W keys simultaneously. This must be done very quickly after the power button has been pressed.
4. There should be a line of text stating that the device is invalidating the persistent registry. If this does not happen, wait for the unit to boot; then repeat the steps until you see the message.

COMMAND PROMPT WINDOWS

The command prompt window allows you to execute limited MS-DOS command line arguments (e.g., `CD SystemCF, Run TEST.BAT`, etc).

To open the Command Prompt window:

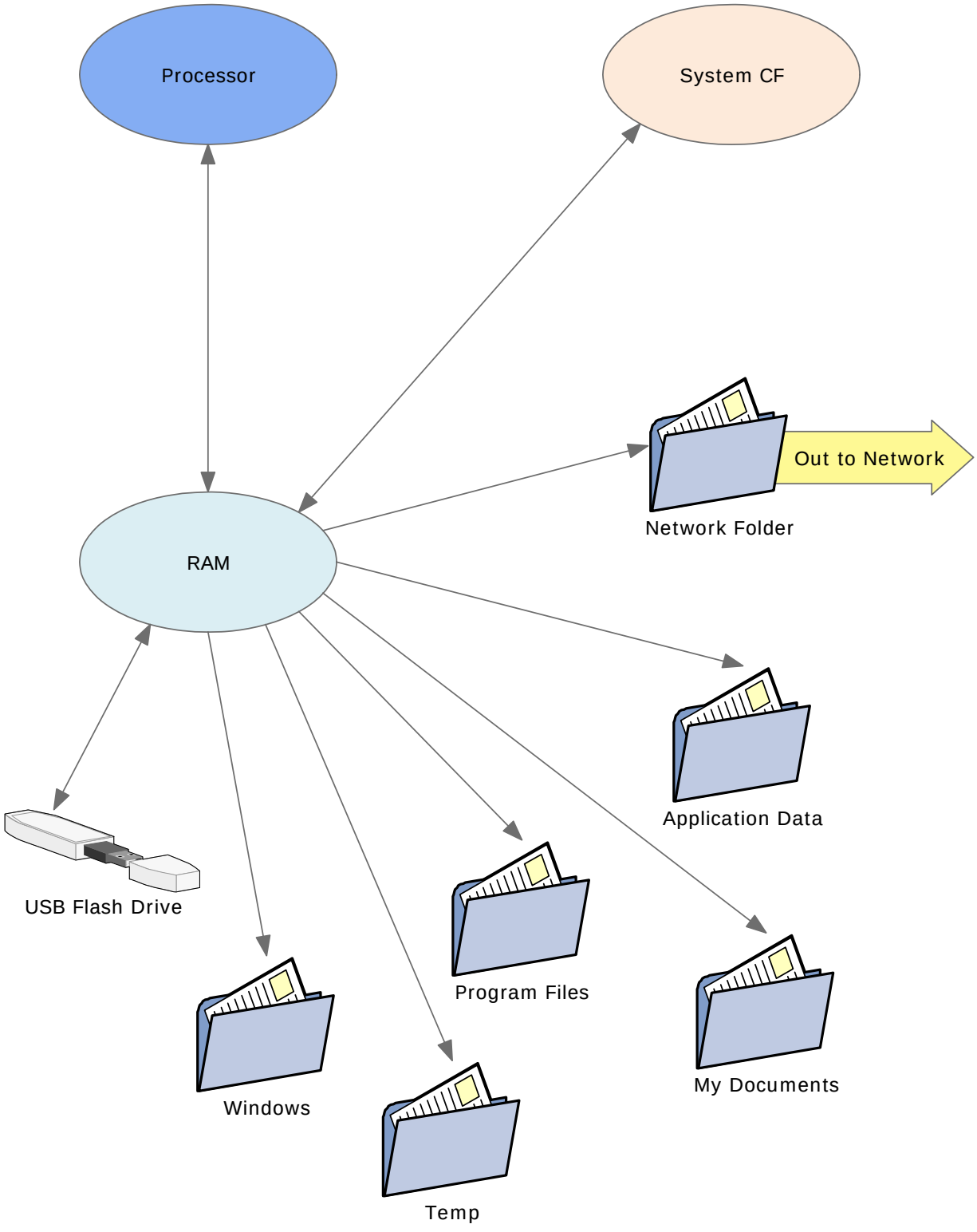
1. From the Start menu, select **Programs > Tools> Command Prompt**. The Command Prompt window opens:



2. You can then type in your commands. For a list of supported commands, type **Help** and press **Enter**.
3. To end the session, type **Exit** and press **Enter**.

PROCESSING, MEMORY SYSTEMS & STORAGE

The Processor, RAM and Non-Volatile (NV) storage all must interact in an organized, quick, and consistent way to offer the user an efficient session. In this section we'll diagram these different components, which will follow with the basic descriptions of the processor used and the memory systems/storage systems.



PROCESSOR

The FC-2500 utilizes the Marvel® PXA270 processor; an integrated system-on-a-chip microprocessor for high performance, dynamic, low-power portable handheld and hand-set devices as well as embedded platforms. It incorporates the XScale® technology which complies with the ARM* version 5TE instruction set (excluding floating-point instructions) and follows the ARM* programmer's model. The PXA270 processor also provides Intel® Wireless MMX™ media enhancement technology, which supports integer instructions to accelerate audio and video processing. In addition, it incorporates Wireless Intel Speedstep® Technology, which provides sophisticated power management capabilities enabling excellent MIPS/mW performance.

The processor stores all information temporarily in RAM. From there the information is either kept in RAM (within one of several volatile folders, as depicted above), or it is sent to SystemCF, USB flash drive, or to a network share.

When calculating (or benchmarking) the amount of data a processor can calculate the term MFLOPS is often used meaning millions of floating point operations per second. In a standard LinPack test the FC-2500 measures 1.7 to 1.8 MFLOPS.*

RAM

The FC-2500 comes with 256 MB of SDRAM. This is considered volatile storage in part because if you cold or warm boot the unit, the memory will be erased. The beauty of using SDRAM is that it synchronizes to the system bus speed (and ultimately the processor), allowing for a more complex pattern of operation than other memory options. Write times will be the fastest when dealing with SDRAM. In a standard Page Fill Read test with caching turned on, the FC-2500 consistently measures at 260-265 MB/sec.*

VOLATILE FOLDERS IN RAM

The SDRAM contains the operating system and any folders created during system initialization. These folders include \ or the root known as My Device, Application Data, My Documents, Program Files, Temp, and Windows. More folders can be added programmatically by the user and copies into RAM every time the system boots through the use of tools such as FileCopy.F2c, or TTStartup. In a standard Read Test of information from the root, the FC-2500 consistently measures at 126-127 MB/sec.*

SYSTEMCF FOLDER

The only folder on the FC-2500 that provides non-volatile (permanent) storage is the SystemCF folder. Information stored in other folders will be lost when you remove power from the FC-2500. You can however, have the FC-2500 automatically copy files from the SystemCF to other folders when booting up. This can be done with filecopy.f2c, FC-Creator, or the ColdBootCopy utility within TTStartup. The SystemCF folder resides on a compact flash card which is separate from RAM, though it shows up directly within the root of My Device. The size of the SystemCF compact flash card is 2GB. Write times to SystemCF are much slower than RAM as depicted above and slightly slower than writing to the root directory. In a standard Read Test of information in the SystemCF folder, the FC-2500 consistently measures at 38-39 MB/sec.*

USB FLASH DRIVES

These flash drives (also known as ThumbDrives) are slower than conventional SDRAM, but are portable in nature, allowing the user to move their data from the FC-2500 to other sources, and perhaps increase their storage space if needed. The FC-2500 accepts OTG compliant USB 2.0 Full speed devices. It is important to note that you will need a mini-A to type A female USB cable or dongle. Write times to USB flash drives will be considerably slower than that of RAM and less than SystemCF as well. The standard transfer rate of a USB 2.0 Full Speed device is 1.5 MB/sec.

NETWORK STORAGE

Network storage is accessed through the path **My Device>Network**. From there you can view computers, and/or view your local name redirects to remote shares. Network storage could be on a PC, a server, or even another FC-2500 if configured correctly. For more information on network storage and how to make network connections, see the section entitled: "Network Connections". Write times to network shares will always be slower than writing to RAM (or any other option on the FC-2500, barring external USB devices), since you are connecting to a remote computer. Write times are also subjective in the fact that they are based on: type of network, network throughput and network usage.

***Note:** The benchmarking information used in this section was obtained through the use of a program called Q-Bench Pro by Qualnetics Corporation. More information on this can be found at the following link: http://www.qualnetics.com/qbench.php*

CHAPTER 4

CONFIGURATION

THE CONTROL PANEL

The table below lists the available control panel functions on the FC-2500.

Table 4-1: Control Panel Functions

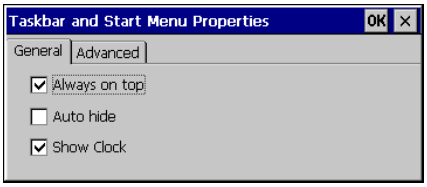
Icon	Function	Description
	**Aux CF Card**	This function enables you to enable/disable power to CF cards installed in the internal CF Card slot, which not user accessible.
	**Aux Switch**	For units with a second COM that supplies 5VDC output, use this function to set the default power state (On or Off), and test the connected devices.
	**Backlight**	Use this function to adjust the backlight setting for the following conditions: Line Active, Line Active Inactive, Battery Active and Battery Inactive. (For ways to increase battery life, see Power Management)
	**Beep Select**	Use this function to change the frequency, volume and duration properties of the beep.
	**Certificates**	Use this function to import, view or remove certificates, which protect your personal information on the Internet, and protect your computer from unsafe software.
	**CPU Speed**	Use this function to determine the current CPU and cold boot-up speed. Allowable selections are 312 MHz and 624 MHz). (For ways to increase battery life, see Power Management)
	**Date/Time**	Use this function to adjust the date, time and time zone. The Date/Time is backed up by a special Real Time Clock Lithium-ion battery. (For more information on this battery, see The Rechargeable Lithium Backup Battery)
	**Dialing**	Use this function to adjust the dialing location settings and dialing patterns when using a modem.
	**Display**	Use this function to adjust the backlight timeout, change the background image or change the desktop color scheme.
	**Display Rotation**	Use this function to rotate the screen 180 degrees (upside down).
	**Hot Keys**	Use this function to assign functionality to the unit's eight programmable keys.
	**Internet Options**	Use this function to set up connections, security settings and internet related functions.
	**Keyboard**	Use this function to change the repeat delay and repeat rate.
	**Network and Dial-up Connections**	Use this function to change network adapter settings and/or set up identification for remote networks.
	**Owner**	Use this function to enter the owner name, address, phone numbers, notes and network ID.
	**Password**	Use this function to enable password protection and set a password.

Icon	Function	Description
	**PC Connection**	Use this function to enable direct connection to a desktop computer
	**Power**	Use this function to: ▪ Check battery power ▪ Set device to turn off when idle ▪ Set up power schemes ▪ Check the power levels of your system devices
	**Regional Settings**	Use this function to change the appearance of region specific information, such as date, time and currency.
	**Remove Programs**	This function enables you to remove programs installed in RAM.
	**Startup**	Use this function in conjunction with the TTStartup program to enable File Copy (from ColdBootCopy) and/or enable Session backup/restore. It also allows you to select how many back up sessions you can have before the oldest session backup is overwritten. (For more information on how to use this see the section entitled: "TTStartup")
	**Stylus**	Use this function to recalibrate the touch screen and adjust the stylus double-tap rate.
	**System**	Use this function to view system information, change the RAM (Program/Storage memory) division, change the device name and change the device description.
	**Version**	Use this function to find out the operating system and NK version of the unit. Also identify the type of processor used and the configuration of the device.
	**Wi-Fi**	This function opens the Summit Client Utility allowing you to create and configure WLAN connections. (For more information, see the section entitled: "Summit WLAN Communications")

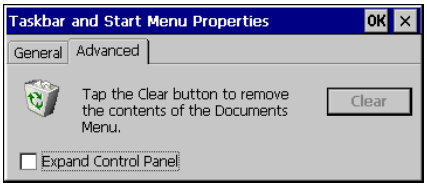
TASKBAR AND START MENU SETTINGS

To change the Taskbar and Start Menu settings:

4. Select **Start > Settings> Taskbar & Start Menu**. The Taskbar and Start Menu Properties dialog box opens:
5. Select the **General** tab:



6. Check **Always on Top** to ensure that the taskbar is always visible, even when a program appears in a full window (maximized).
7. Check **Auto hide** to display the taskbar just when you point to the taskbar area.
8. Check **Show Clock** to display the time of day in the taskbar.
9. Select the **Advanced** tab:



10. Tap the **Clear** button to remove the contents of the documents menu.
11. Check the **Expand Control Panel** box to display the contents of the Control Panel as items on the Settings | Control Panel menu.
12. Tap **OK** to save the settings and exit the menu.

CHAPTER 5

CONNECTIONS

SUMMIT WLAN COMMUNICATIONS

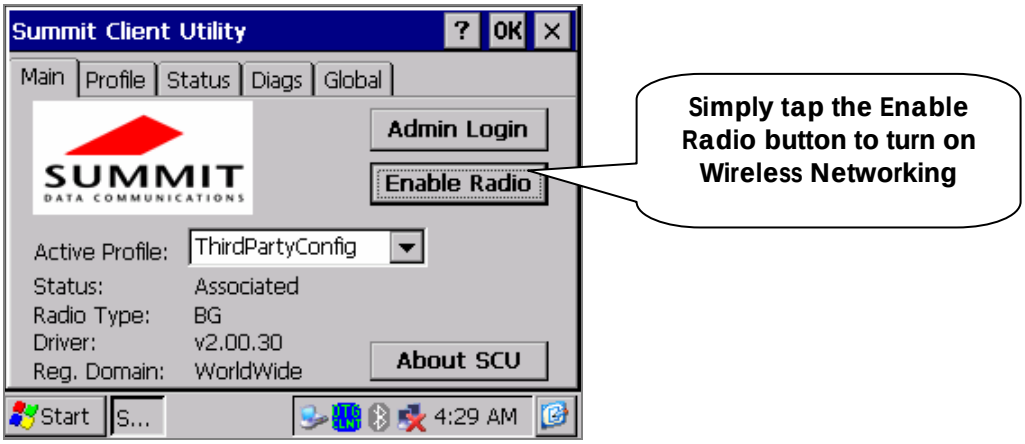
The FC-2500 comes standard with a Summit WLAN 802.11b/g Mini Module. Maximum data rates supported are 54 Mbps (in G mode) and 11 Mbps (in B mode). Security standards include WEP, WPA, and WPA2, with configurable TKIP and AES encryption. It is compliant with Cisco's proprietary wireless protocols via CCX version 4. This WLAN module can be configured, diagnosed, and troubleshot within the Summit Client Utility. As of April 2008, the FC-2500 uses Version 2 of the Summit Client Utility.

ACCESSING THE SUMMIT CLIENT UTILITY

By default, the FC-2500 should search for wireless networks automatically utilizing the Microsoft® Zero Configuration utility. If it does search for wireless networks automatically, skip to the next section entitled "**Connecting to a WLAN**". However, to do this, the WLAN radio feature must be enabled in the Summit Client Utility (SCU). The following shows how to access the SCU and turn the WLAN radio on.

1. Open the Summit Client Utility. There should be a shortcut on the desktop called "Summit Client Utility"; simply double-tap it. If the shortcut is not there, the path to the application is **Start>Programs>Summit>scu**

2. Once you have opened the application, you will see an "Enable Radio" button towards the top right of the display. Tap it once to enable the WLAN radio as shown below.



3. Once you enable the WLAN radio, the Zero Configuration utility should start searching for networks automatically. If it does not, warm boot the FC-2500 by tapping: **Start>Programs>Tools>Warm Boot.**

CONNECTING TO A WLAN

If the FC-2500 is not associated with a wireless access point then the Zero Configuration utility should start automatically, every time you boot the FC-2500. The utility will subsequently popup on the screen at a set interval. This will continue until the unit is associated with an access point. If the Zero Configuration utility is not displayed, you can access it by double tapping the Zero Configuration icon on the task bar and selecting the "Wireless Information" tab. When you do so, you will see the screen depicted below. Now the FC-2500 is ready to be associated with an access point.

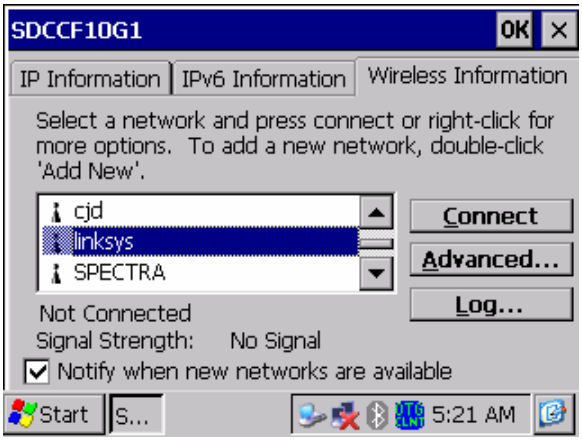
ASSOCIATING WITH AN ACCESS POINT

There are two ways to associate with an access point; through the use of the Zero Configuration utility or the Summit Client Utility. First, we will show how to associate with Zero Configuration.

Note: *Zero Configuration may be limited in authentication techniques. If you need a higher level of authentication and encryption use the SCU.*

Connecting with Zero Configuration

1. First, select the access point that you will use for wireless communications. In this example we are using an access point called "Linksys®". Scroll down until you see the correct access point (if there is more than one) and tap it once to select it.



2. Then tap Connect to begin the association process. You will then be prompted to enter information as shown in the figure below.

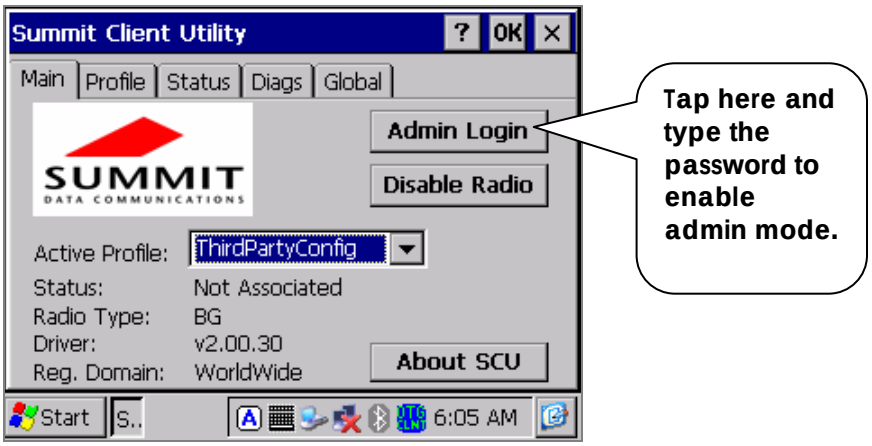
If there is encryption and/or a special authentication mode that the access point requires, you would configure that in this screen. Also, if the access point requests a special network key (otherwise known as a pass phrase) then you would need to disable the "The key is provided automatically" checkbox and type that pass phrase in the field labeled "Network Key". If there are no encryption or authentication options to be configured then skip to step 3.

3. When you are finished setting the options, tap the OK button at the top right of the screen.

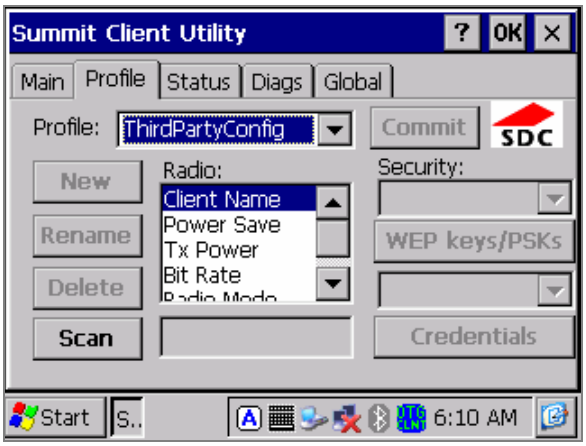
Connecting with the Summit Client Utility (SCU)

To connect to a wireless network with the SCU, you will have to use the built-in "ThirdPartyConfig" profile, or create a new profile within the SCU application. Creating new profiles (and scanning for networks) requires being in "Admin Mode". To enable Admin Mode, simply tap the Admin Login button on the main screen of the SCU and type in the password (if there is one) when prompted. Then continue with the steps below to make the wireless connection.

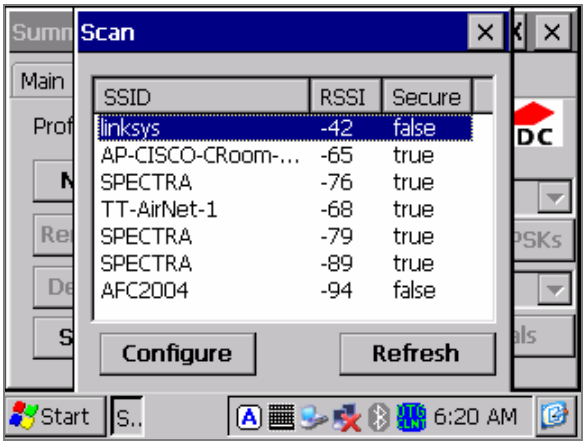
Note: *When typing more complex passwords (case sensitive, alpha-numeric, etc…)try using CeKeys. To access this utility tap Start>Programs>Tools>CeKeys*



1. Tap the "Profile" tab. You should see the screen depicted below



2. Now, tap the Scan button towards the bottom right of the screen and select the appropriate wireless access point.

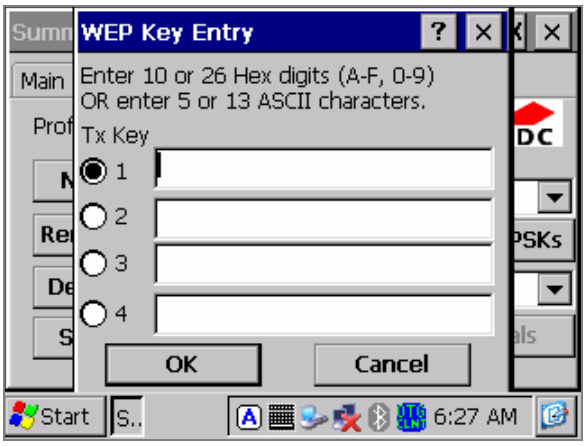


3. Tap "Configure" to configure the connection. In this example, we are selecting the Linksys® access point.

Note: *If the "Configure" button is grayed out, then you need to enable Admin Mode.*

4. On the next screen, SCU will confirm that you want to create a new profile named after the access point. Tap yes to continue.

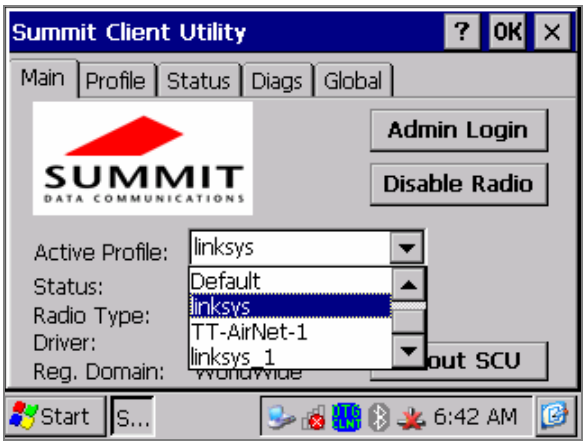
5. If your access point requires it, you will then be prompted to type in appropriate network key in the field as shown below. If no key is required, the profile (and subsequent connection) will be created automatically.



6. Tap the Commit button to commit these changes. (If you tap the OK button, the program will also ask you if you want to commit the changes.)

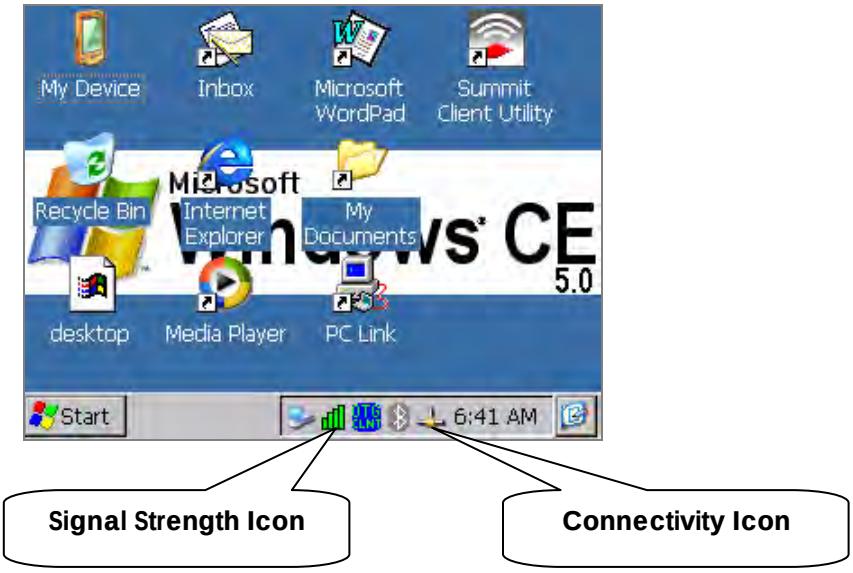
7. To make this the default profile (and use this access point by default every time the FC-2500 boots up) tap the Main tab.

8. Then select the profile from the drop down list as shown below. Keep in mind that you will need to be in Admin Mode to accomplish this.



9. This will require a power cycle. Tap OK for any open windows and warm boot the FC-2500.

10. When you reboot, the FC-2500 should connect to the default wireless access point. You will then see the connectivity and signal strength icons in their active state, located in the task bar as shown below.



At this point you can also access the SCU by double tapping the Signal Strength icon.

CONFIGURING A WLAN CONNECTION WITH THE SUMMIT UTILITY

In this section we'll concentrate on a few common changes you might make to the configuration.

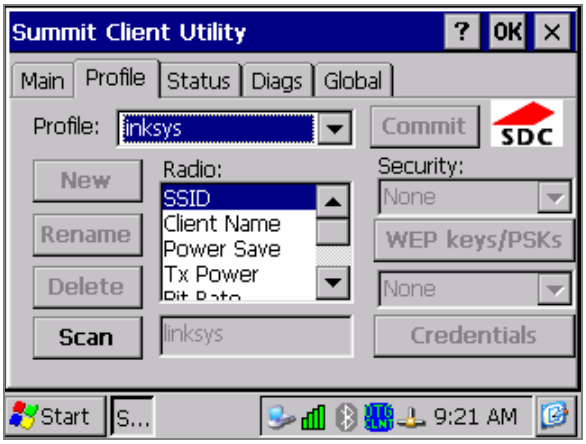
Radio Mode

The radio can operate at B, G, or a combination BG. It is configured by default to BG mode. However, in some cases you may have to set the radio to one specific mode. The following shows how to do this step by step.

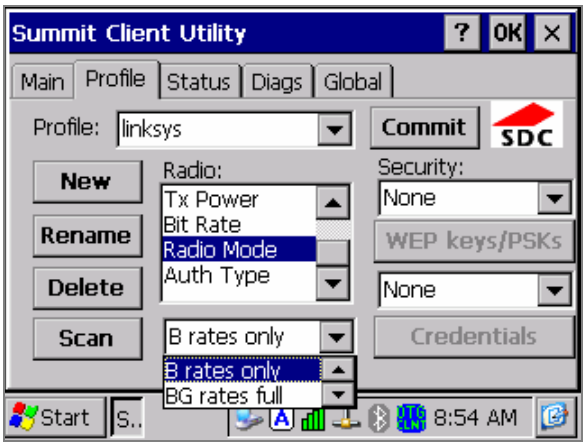
1. Open the SCU

2. Enable Admin Mode.

3. Tap the Profile tab and select the appropriate profile that you want to change. In this case we are using the "Linksys®" profile.



4. In the "Radio" field, scroll down to the Radio Mode option.

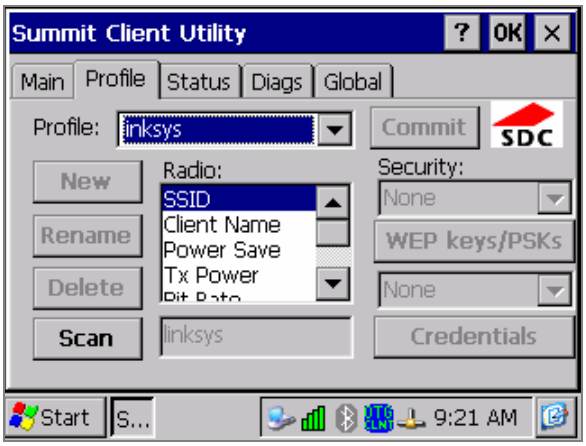


5. Click on the drop down menu below the "Radio" field. At this point you would select the mode that will allow your FC-2500 to best communicate with the wireless access point. For example, some access points only work in B mode. Others work in G only. When you are successfully associated with an access point, the default option for this is called "BG rates full", and will work in most environments.

TX Power

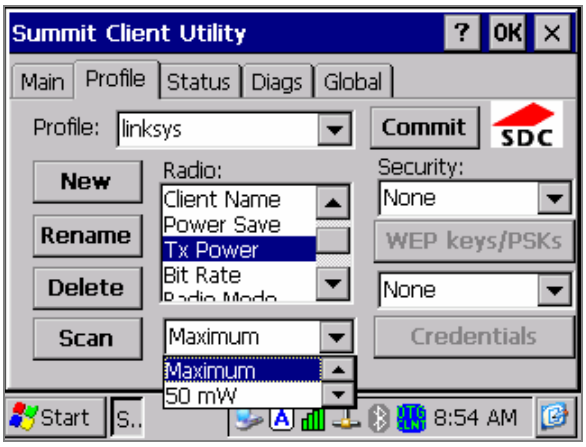
The transmit power of the mini module is normally set to maximum. However, it is possible to modify this so that the module transmits at a lower rating and conserves power. The following shows how this can be done step by step.

1. Open the SCU
2. Enable Admin Mode
3. Tap the Profile tab and select the appropriate profile that you want to change. In this case we are using the "Linksys®" profile.



4. Select TX Power from the "Radio" Field.

5. Then click on the drop down menu underneath the "Radio" field. From here you can select the exact power rating in mW as shown below.



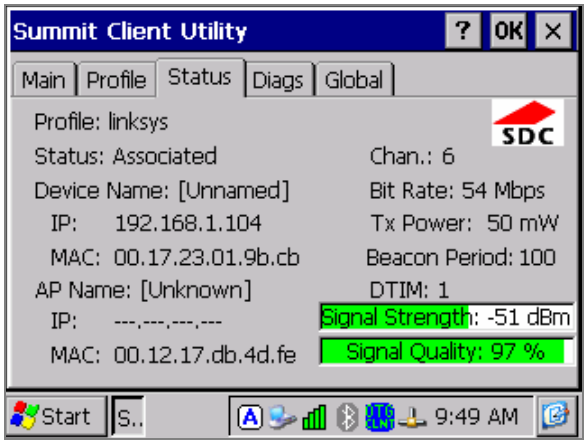
Note: *If you are limited to 50 mW, you may have to change your Regulatory Domain. This can be done in the Summit Manufacturing Utility (SMU). For example, the WorldWide setting will limit you to an output of 50 mW, but the FCC setting will allow you to go beyond this to 80 mW.*

DIAGNOSING A WLAN CONNECTION WITH THE SUMMIT UTILITY

The Summit Client Utility offers several ways to analyze, diagnose, and troubleshoot your wireless connection. In this section we will speak to a few of those. Remember to enable Admin mode whenever making any modifications in the SCU.

Status

The Status tab gives a variety of information about the wireless connection as shown below.



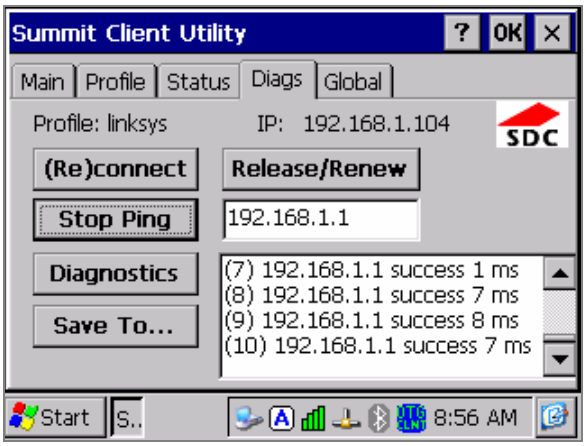
In the screen above you can see the profile that we are associated with as well as the FC-2500's device name (if it has one). You can also see:

- the IP address assigned to the FC-2500 by the wireless access point
- the MAC address that the FC-2500 is using
- the channel we are using for wireless communications
- the current Bit Rate which will actively change on this screen as you increase or decrease the distance between the FC-2500 and the access point
- the current transmit power
- the beacon period
- and the signal strength/signal quality meters

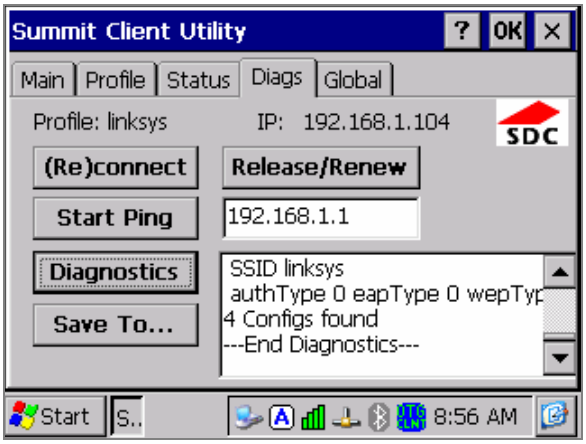
Diagnostics

The "Diags" tab offers several networking diagnostics.

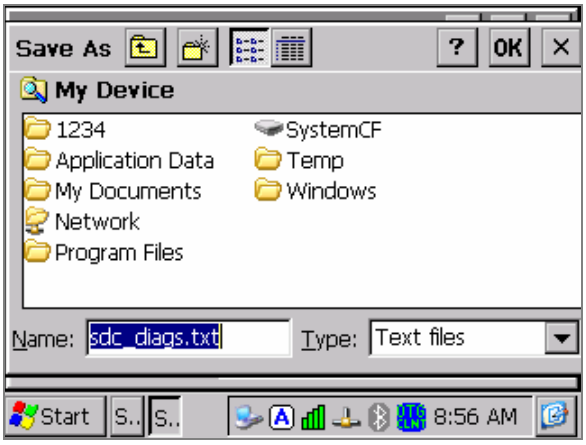
- (Re) connect: If the FC-2500 is removed from the range of a wireless access point it will lose its connection. When returning to the access point's range, press this button to reconnect.
- Release/Renew: This option allows the FC-2500 to release it's IP address and attempt to gain a new one. This is useful when an access point is reconfigured, power cycled, or if a new access point is introduced to the network.
- Start/Stop Ping: This utility tests network connectivity from the FC-2500 to the access point. In the example below, the SCU diagnostics is "pinging" the access point: 192.168.1.1. This will continue until the "Stop Ping" button is pressed. This utility can be helpful when trying to attain a networking baseline, or for determining how fast the connection is in general. For example, anything less than 10 ms is a decent return time for a ping on a wireless network.



- Diagnostics: By pressing the Diagnostics button, you will get a host of information including the SSID you are connected to, the configurations available, the networking and diagnostic settings, etc…



This information can also be outputted as a text file for archival. This is done with the default name sdc_diags.txt as shown below.



Note: *Remember to save .txt files (and any other files) to non-volatile areas of the system, including the SystemCF or USB flash drive.*

Note: *For more information on the Summit Client Utility, see the Documentation section of Summit Data Communications' website at: http://www.summitdatacom.com/documentation.htm and download the User Guide for Version 2.*

CREATING AN ACTIVESYNC CONNECTION

ActiveSync is a desktop utility program (available as a free download from Microsoft) that allows you to synchronize certain types of information between a PC and the FC-2500. You can also use ActiveSync to transfer files and install programs on the FC-2500.

ActiveSync 4.1 supports PC sync via USB, JETT•connect cable or a Bluetooth connection. It does not support Remote PC Sync (via WLAN or LAN). For information on how to make an ActiveSync connection using Bluetooth, refer to the JETT•XL/eye Bluetooth User's Guide manual.

Note: *We recommend you use ActiveSync version 4.5 or greater.*

USB CONNECTION

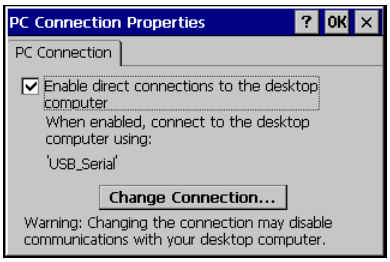
By default, the FC-2500 is setup to make an automatic ActiveSync connection and prompt you to create a partnership when you connect a USB cable between the FC-2500 and you PC. Refer to the Creating a Partnership section for further instructions.

SERIAL CONNECTION

If you are using a DE-9 cable, you will need to perform the following procedures to make an ActiveSync connection:

CONFIGURING THE FC-2500

6. On the FC-2500, tap **Settings> Control Panel > PC Connection**. The PC Connection Properties Dialog box appears.

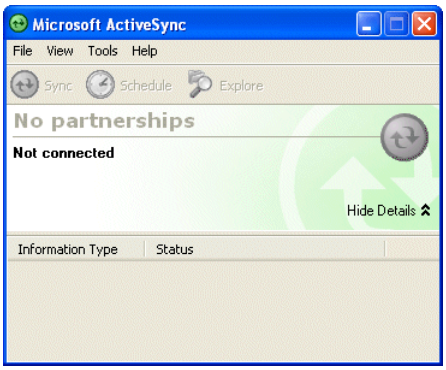


7. Tap **Change Connection**. The Change Connection dialog box appears.

8. Select **PC-115** and tap **OK**. The PC Connection Properties Dialog box reappears.
9. Tap **OK** to exit.
10. Close the Control Panel and return to the desktop.

CONFIGURING THE PC

1. On the PC, start **ActiveSync**.



2. On the ActiveSync menu bar, select **File** and then **Connection Settings**. The Connection Settings dialog box opens.

3. Check the **Allow connection to one of the following box**, and assign the number of the available COM port (typically COM1).

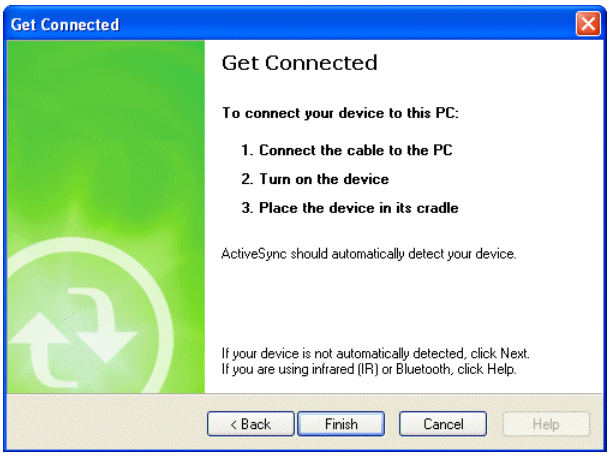
***Note**: If you have to share a COM port with a device that has a different connectivity program, you will need to free the COM port from ActiveSync before using the COM port for the other device. The next time you want to connect a device to ActiveSync, you will need to assign the COM port back to ActiveSync.*

4. If needed, change how the PC should connect to when passing through ActiveSync (i.e., the FC-2500 can use the computer's network connection as if it were its own to download files or browse the Internet). The default setting is **Automatic**.

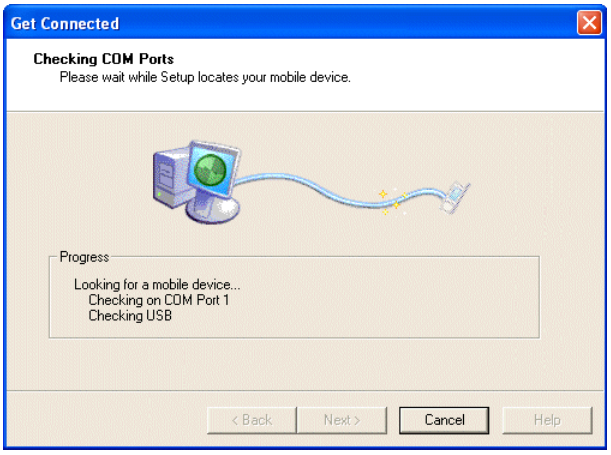
***Note**: For additional security, disable network bridging on the PC (specifically, bridging to a Remote NDIS adapter) before connecting to the PC to pass though to the Internet or a network. Proceed to the next section. Do not close this dialog box.*

CREATING AN INITIAL CONNECTION

1. Plug one end of the interface cable into your computer and the other end into the FC-2500.
2. In the Connection Settings dialog box, click **Connect**.
3. When using a serial cable, the Get Connected wizard will appear.

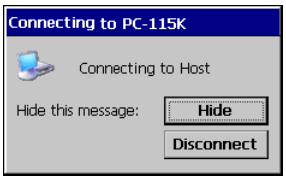


4. Click **Next**. ActiveSync will then attempt to establish a connection with the FC-2500 (this process may take several seconds).

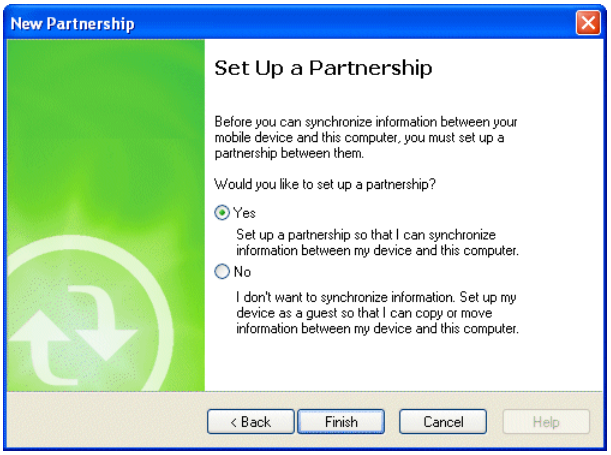


5. On the FC-2500 desktop, tap **PC Link**.

6. The following message box appears:



7. When ActiveSync successfully establishes communications with the FC-2500, the New Partnership dialog box will appear on the PC.



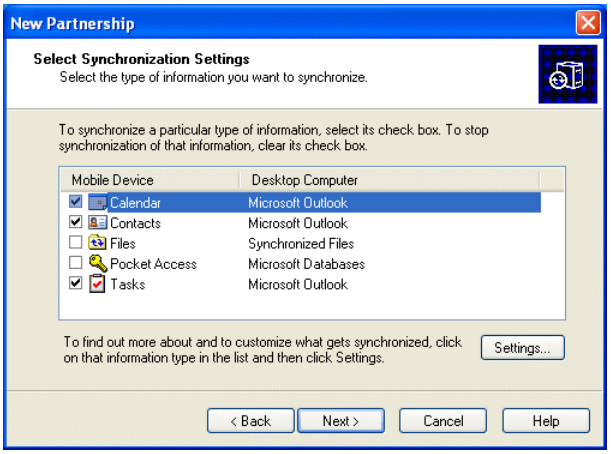
8. You can now choose to setup a partnership to synchronize files or connect as a "guest." Refer to the following sections to further instructions.

CREATING A PARTNERSHIP

When starting ActiveSync for the first time, you have the option to create a partnership that allows you to synchronize selected items between the FC-2500 and the PC.

To create a partnership during your initial setup:

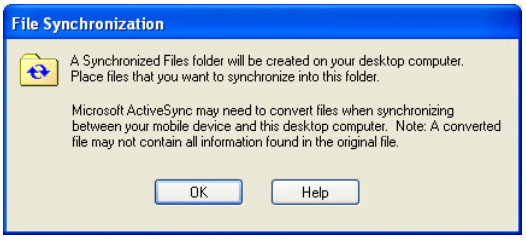
9. Select **Yes** in the New Partnership wizard and then click **Next**. ActiveSync will then prompt you to select the type of information you want to synchronize.



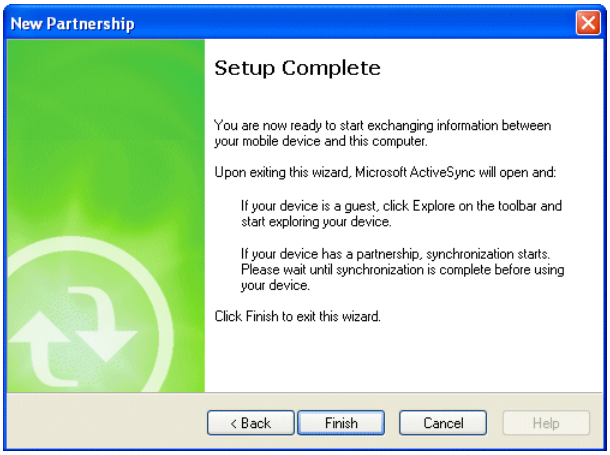
10. Check or clear the boxes that pertain to the type of information you want to synchronize.

***Note**: the FC-2500 does not support Pocket Access or the synchronization of Pocket Access files*

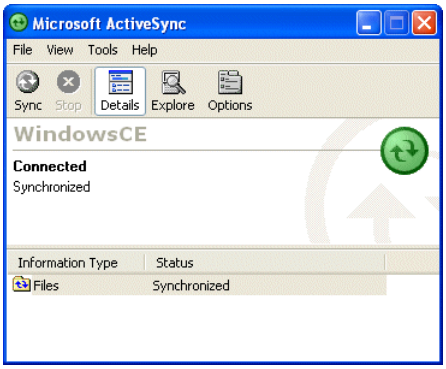
11. If you choose to synchronize files, the following message will appear:



12. Click **OK** to continue.



13. Click **Finish** to complete the setup. ActiveSync will then create a folder on your desktop and attempt to synchronize the item types you selected with similar items on the FC-2500 (e.g., if you selected **Files** and placed a file in the My Documents folder on the FC-2500, ActiveSync will transfer it to the FC-2500's folder on the desktop.



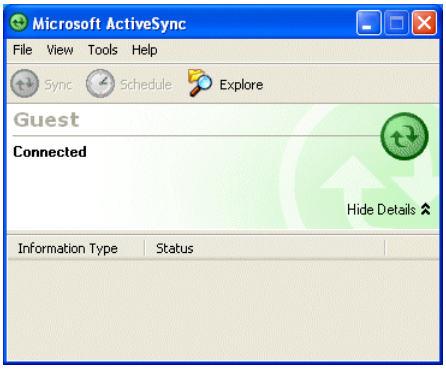
On the FC-2500, an icon indicating an ActiveSync connection will appear in the system tray.

CREATING A GUEST CONNECTION

If you do not require synchronization or plan to connect to the PC on a regular basis, you can set up a "guest" connection to perform tasks such as coping files or installing programs.

To create a guest connection during your initial setup:

1. Select **No** in the New Partnership wizard and then click **Next**. The ActiveSync main menu will reappear. A status of "Connected" should appear in the ActiveSync window.

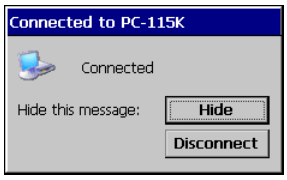


On the FC-2500, an icon indicating an ActiveSync connection will appear in the system tray.

DISCONNECTING ACTIVESYNC

To terminate the ActiveSync connection:

1. On the FC-2500, double-tap the connection icon in the System Tray. The PC Status dialog box appears.



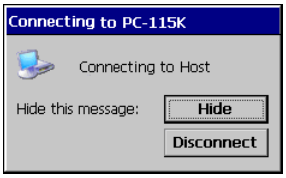
2. Tap **Disconnect**.

SUBSEQUENT SERIAL CONNECTIONS

To perform subsequent ActiveSync connections when using a DE-9 cable:

1. If needed, connect your interface cable between your desktop computer and the FC-2500.
2. On the FC-2500 desktop, tap **PC Link**.

3. On the FC-2500, the following message box appears:



4. If you have created partnership, the ActiveSync will automatically reconnect and attempt to synchronize any files. If you have not created a partnership, ActiveSync will display the New Partnership wizard. In which case, you can create a partnership or a guest connection.

NETWORK CONNECTIONS

The FC-2500 can connect to network resources (shared folders, e-mail, etc…) as long as it has a WLAN, Bluetooth®, or a wired network connection. There are a couple ways to do this: through the browser, the command line, or making persistent network connections.

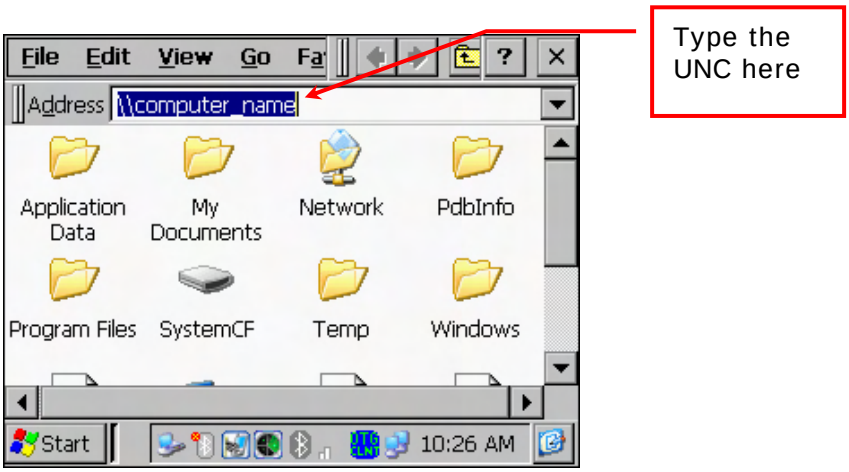
Note: *For more information on how to connect to wireless networks, see "Summit WLAN Connections".*

CONNECTING TO NETWORK RESOURCES VIA BROWSING

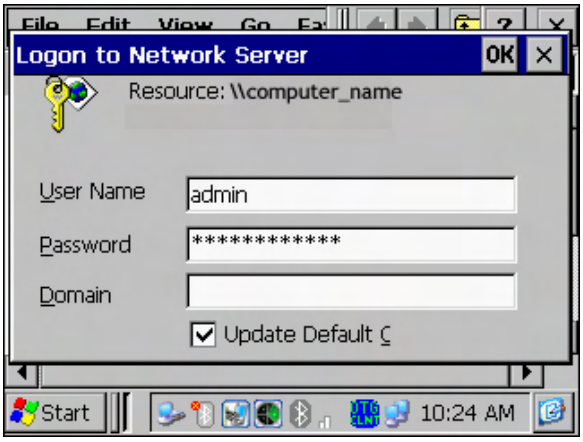
Browsing is probably the easiest way to connect to network resources. Browsing will allow you to access other computers and their shared folders. To browse to a computer you can simply access My Device and double tap the Network icon. However, if you are not a member of a network, you may have difficulty seeing (or accessing) individual computers. The better way is to connect directly to the shared resource. Follow the steps below to do this:

1. Double tap the My Device icon on the Desktop.
2. In the Address field, type the name of the computer and share you would like to access (this is also known as a UNC or Universal Naming Convention address). An example is shown below.

   a) The proper syntax for this would be **`\\computer_name\share_name`** where "computer_name" is the computer you are attempting to connect to, and "share_name" is the share that is located on that computer. You do not need to type the entire local path to that share, only the computer name and the share name are necessary. In some cases you only need to type the computer name to gain access to some resources.



3. The first time you attempt to connect to this resource, a window will appear asking you for the proper credentials which will authenticate you to the remote system. These credentials include a username, a password, and (optionally) a Domain name, if the remote computer is part of one. This is shown in the next figure.



Subsequent connections to this resource do not require you to type the credentials unless you restart (warm or cold boot) the unit.

4. At this point, the remote computer (or remote computer's particular share) should show up on the display as depicted below.

Any information inside that share will be accessible unless advanced permissions are in effect.

Of course, typing in these credentials over and over again can be taxing. To permanently add one set of credentials, see the section entitled: "Setting up Persistent Identification for Remote Networks" below.

CONNECTING TO NETWORK RESOURCES VIA THE COMMAND LINE INTERFACE

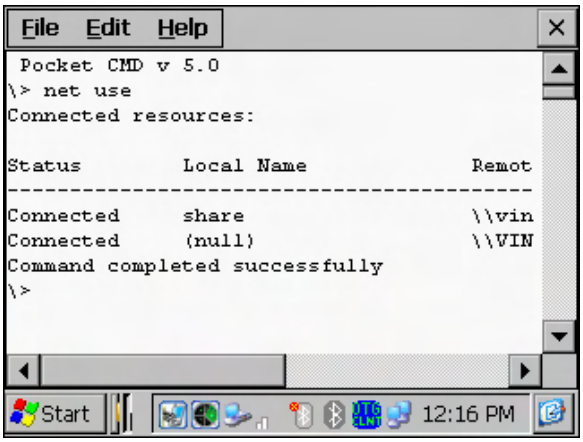
For people who wish to use the command line interface (CLI) instead of a GUI, you can connect to network resources as well. This is done primarily with the **`Net Use`** command.

Note: *For all intensive purposes, the terms command line interface, CLI, and command prompt can be used interchangeably in this section.*

The Net Use command can be used to view your current network connections, and to make new network connections.

To view your current network connections, do the following:

1. Open the Command Prompt. Tap Start>Programs>Tools>Command Prompt.
2. Type the command **`Net Use`**. An example of this is below.



In this example, we have a share already connected. If no network connections were made up to this point, there would be no information in the table. However, the "Command Completed successfully" message would show up at the bottom indicating that the command ran properly.

After a short timeout, the status of the connection will change from "Connected" to "Disconnected" for security reasons.

Connecting to network resources is slightly different than on a PC. Instead of selecting a drive letter for mapping, we select a Local Name (since there are no drive letters in Windows CE!).

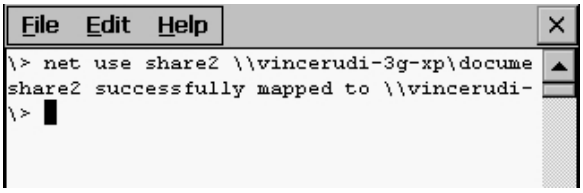
To create a new network connection, do the following:

1. Open the Command Prompt. Tap Start>Programs>Tools>Command Prompt.
2. Find out the name of the computer and the share that you want to connect to.
3. Make the connection by typing the proper syntax, example syntax follows:

   `Net use [local name] [\\computer_name\share_name]`

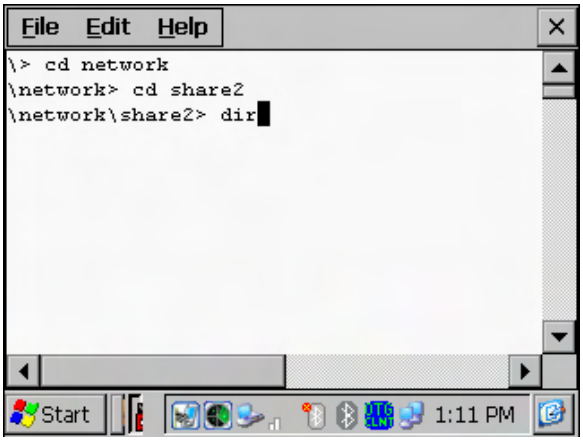
Note: *Brackets indicate that the syntax within them is a variable. Do not type the brackets!*

So for example, we might type: **`Net use share2 \\testcomputer\testshare`** where "share2" is out local name, "testcomputer" is our computer_name and "testshare" is our share_name. An actual share completing successfully is shown below:



4. As long as the message says that is successfully mapped, then you should be able to access the share within the CLI or in the GUI.

Keep in mind that we are not using drive letters; instead we are using local names, and that the local names are simply redirecting to resources on the network. Therefore, to access the share in question, we would navigate to the Network folder (if working in the GUI, this is directly inside of My Device, if in the CLI, it is directly within the root directory), and then the name of the Share folder. Navigating to this directory in the CLI is depicted in the figure below:



In the above example we started in the root directory and typed **`cd network`** to navigate to the network folder, then **`cd share2`** to navigate to the Local Name known as share2. Typing **`Dir`** allows you to view the contents of that remote resource.

The GUI version of this would look like something like the figure listed below:

Note: *You can also create shares on the FC-2500, effectively making it a file server. This is done by modifying the registry. More information can be found on this at the following link: http://msdn.microsoft.com/en-us/library/aa459150.aspx (be sure to scroll down about half way to see the section on making the file server).*

SPECIFYING AUTHENTICATION PARAMETERS WHEN CONNECTING TO NETWORK RESOURCES FOR THE FIRST TIME IN THE CLI

One thing to keep in mind is that the remote system will ask for authentication the first time you connect to one of its resources. If you are working in the command line, you will have to append the previously covered **`net use`** command with the proper authentication. Two examples follow:

To connect to a computer that is not a member of a domain:

```
Net use [local share]  [\\computer_name\share_name] [/user:%username%]
[/password:%password%]
```

Note: *Do not type the brackets, they indicate variable information. This also goes for the percentage % signs. They specify that the term inside them is also a variable. So instead of %username%, you would type whatever the username is (e.g. cfrost) that is allowed access to the remote resource.*

To connect to a computer that *is* a member of a domain:

```
Net use [local share]  [\\computer_name\share_name]
[/user:%domain_name%\%username%] [/password:%password%]
```

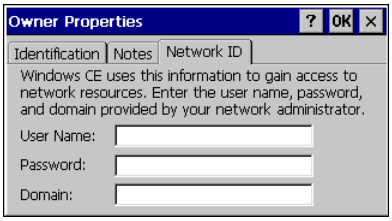
Remember to include the extension of your domain (.com, .net, .local, etc…) in the %domain_name% variable.

SETTING UP PERSISTENT IDENTIFICATION FOR REMOTE NETWORKS

When connecting to a network, you may need to enter a user name, password, and domain name for gain access. This can become tedious if you are connecting to the same resources over and over, and if you are restarting the FC-2500 often (which is very likely). To set up authentication permanently, proceed through the following steps:

To set up identification for remote networks:

1. Select Start > Settings > Control Panel.
2. Double-tap **Owner**. The Owner Properties dialog box appears.
3. Select the **Network ID** tab. Enter the **User Name**, **Password**, and (optionally) the **Domain** used to gain access to the network.



4. Tap **OK** to exit.

CONNECTING TO A MAIL SERVER

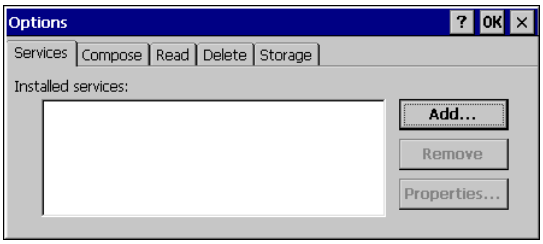
You can send and receive e-mail by connecting to a POP3 or IMAP4 server. Inbox contains an e-mail service for each method you use. For either service, you must establish a connection to your Internet service provider (ISP) or to the appropriate mail server in your local area network. In addition to creating this connection, you must also create the e-mail service.

Prior to setting up a service, you should obtain the following information from your ISP or network administrator: POP3 or IMAP4 server name, SMTP host name, user name, password and domain name (for network connections only).

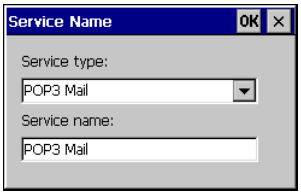
***Notes**: Windows CE 5.0 does not support other mail protocols such as AOL or services that use special authentication, such as MSN. However, you can gain access to the Internet through these services. If you use the same service to connect to different mailboxes, set up and name a different service for each connection. For additional information about the inbox, refer to Windows CE 5.0 online help.*

To connect to your POP3 or IMAP4 mail server:

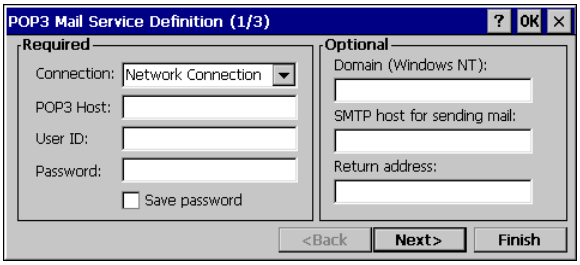
1. Select Start > Programs > Inbox > Services > Options. The Options dialog box opens.



2. Select the **Services** tab and tap **Add**. The Service Name dialog box opens.



3. From the **Service type** list, select **POP3 Mail** or **IMAP4 Mail**, enter a unique name for the service (you cannot change this name once entered) and tap **OK**. The Mail Service Setup wizard appears.



4. In the Required panel:

   - Select the name of the connection you created to connect to the mail server. If you are receiving e-mail through a network (Ethernet) connection, select Network Connection.

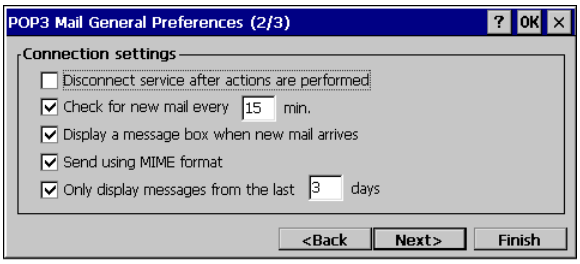
     If you want Inbox to use your current connection, select **(none)**.

     If you have not created a connection, select **Create new connection**, double-tap the Make New Connection icon, and follow the instructions in the wizard. When finished, select Inbox in the Taskbar and continue setting up Inbox.
   - Enter the **POP3 Host** or **Server** (IMAP4) name of the mail server you use to receive and send messages.
   - Enter the **User ID** (user name or mailbox ID) assigned to you.
   - Enter the **Password** you will use to access this mail account. If you do not want a prompt to enter the password each time you connect, select **Save password**.

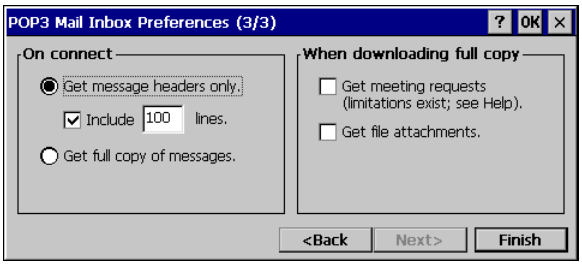
5. In the Optional panel:

   - If connecting to a network that uses Windows NT domain security, enter the **Windows NT domain name**.
   - If your mail service uses a separate server for SMTP, enter the **SMTP Host name**. For POP3 Mail service with an ISP, the ISP must use an SMTP mail gateway.
   - Enter your **return e-mail address**.

6. Tap **Next**. The General Preferences dialog box opens.



7. Choose any of the settings, all of which are optional, then click Next**.** The Inbox Preferences dialog box opens.



8. Choose any of the settings as needed, then click **Finish**. The Mail Service Setup wizard closes and the Options dialog box reappears.

***Note**: Receiving entire messages consumes storage memory.*

9. Close the Options dialog box to return to the Inbox. .

CHAPTER 6

USING VISUAL STUDIO® 2005

INTRODUCTION

Visual Studio 2005 is an integrated development environment that allows you to build applications for the FC-2500. You can use Visual C# or Visual Basic to write managed applications that run on the .NET Compact Framework, or you can write native applications using Visual C++. The section will help you become familiar with FC-2500 application development using Visual Studio.

USING THE .NET COMPACT FRAMEWORK

In Visual Studio 2005 managed projects, all platforms target version 2.0 of the .NET Compact Framework unless otherwise noted.

The number and members of classes differ from what is available for developing desktop applications. In managed projects using the .NET Compact Framework, fewer classes are available for devices, and the complement of classes typically differs among platforms.

The same is true for native projects, where only a subset of Windows APIs, MFC classes, or ATL components are available. You can determine which classes are available by viewing the documentation, by using IntelliSense, or by using the Visual Studio Object Browser while your project is active.

As with desktop applications, you can access native code by using platform invoke. The .NET Compact Framework provides limited support for COM interop. It does not support creating COM objects in managed code or interoperating with ActiveX controls.

Some language items can differ; for example, not all Visual Basic keywords used for desktop development are supported.

Some code snippets provided in Visual Studio documentation for desktop projects may generate build errors in device projects.

There are design considerations, such as the form factor of the device, power usage, memory constraints, and other details, that are not factors for desktop development.

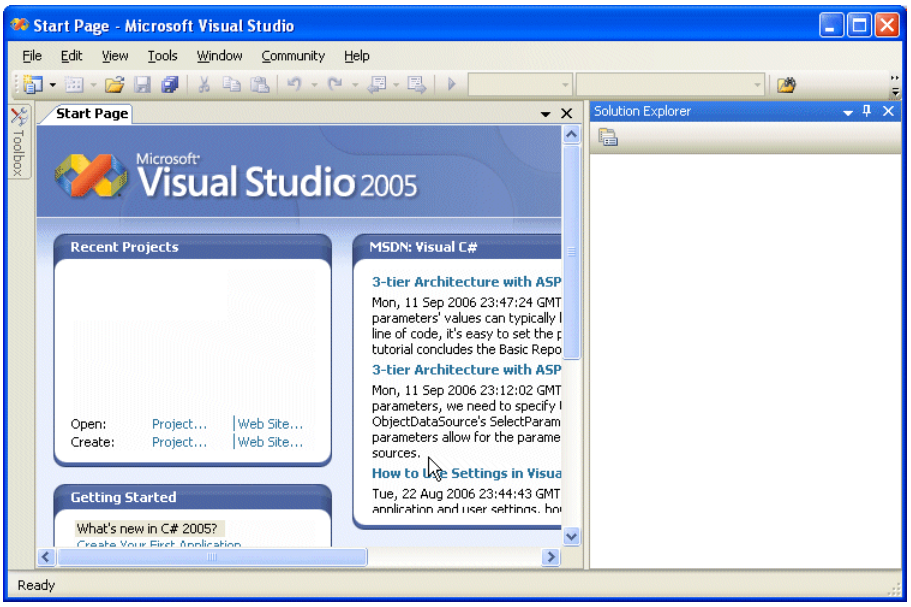
BUILDING A "HELLO WORLD" APPLICATION

The following procedures describe how to create a simple "Hello World" application and deploy it to the FC-2500.

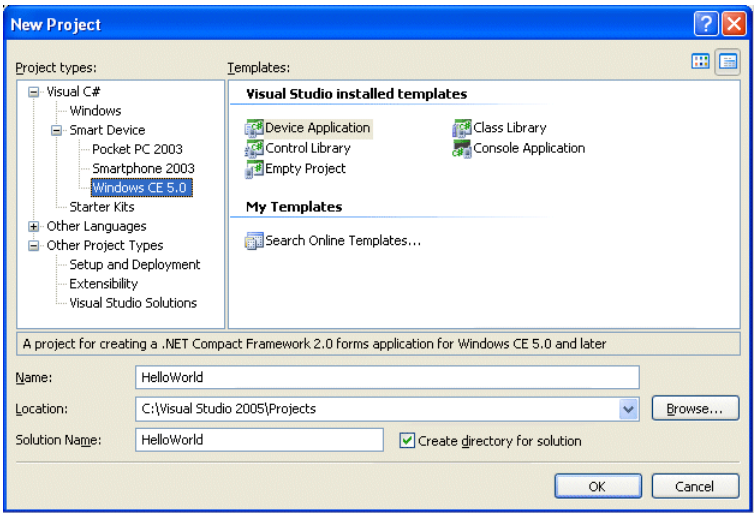
CREATING THE PROJECT

To create C# smart device project for the FC-2500

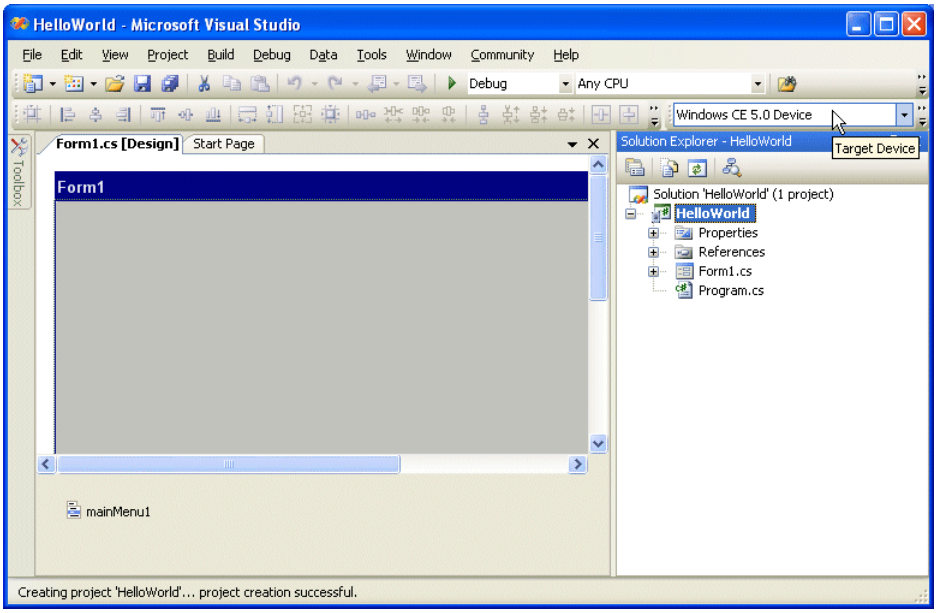
10. Create an ActiveSync connection.
11. Start **Visual Studio 2005**.



12. From the File menu, select **New**, and then click **Project**. The New Project dialog box appears.



13. Under Project Types, select **Visual C#**, **Smart Device** and then **Windows CE 5.0**.
14. Under Templates, select **Device Application**.
15. Type the **Name** (such as HelloWorld) and then select the **Location** of the project (such as C:\Visual Studio\Projects).
16. Click **OK** to create the project. The Visual Studio Form Designer opens.

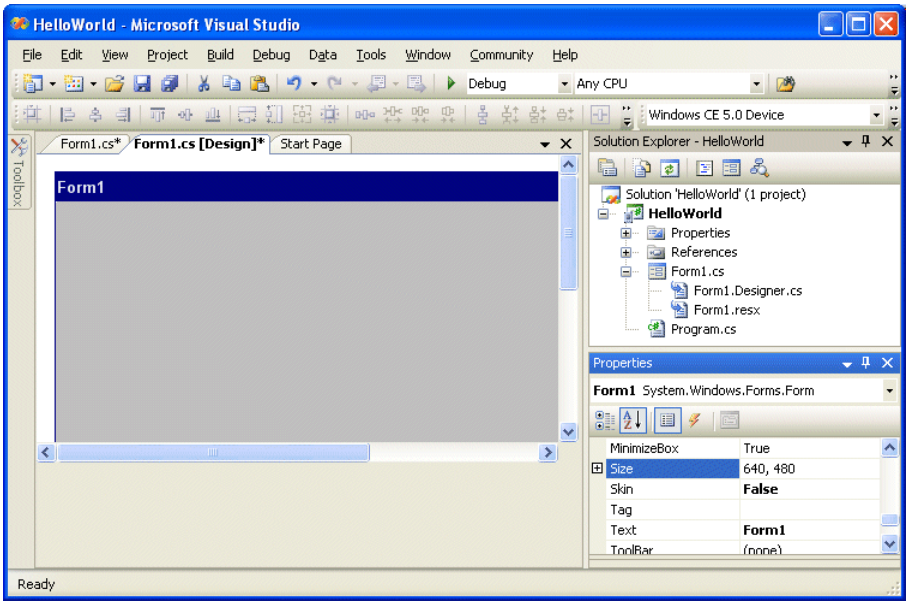


17. Select **Windows CE 5.0 Device** as the Target Device. You are now ready to begin application development.

EDITING THE FORM

To resize the form to fit on the screen and add a button to the Hello World application:

18. Delete **mainMenu1** as it is unnecessary for this example and then right click on **Form1** to open its properties.

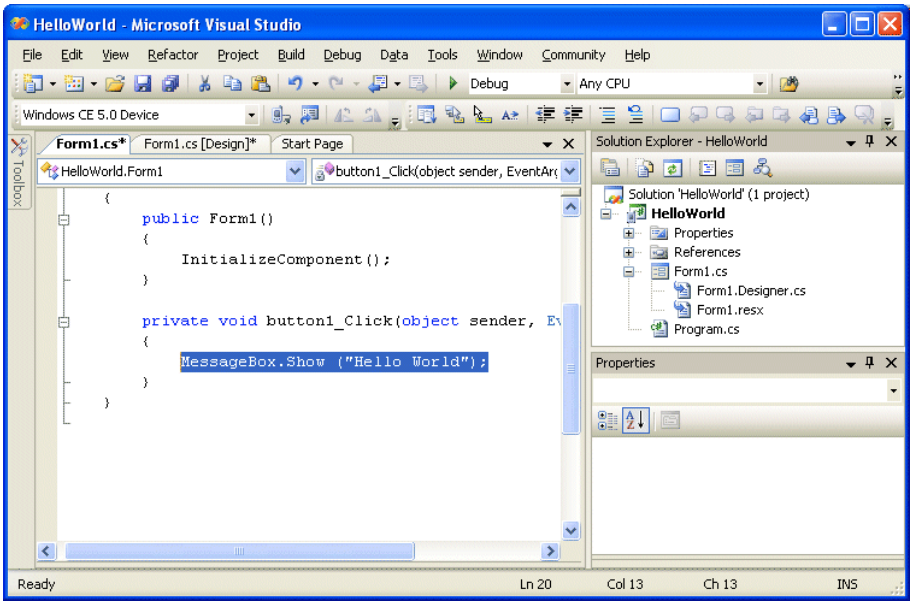


19. In the Properties Window, change the **Size** property to **200,150** and the **Text** property to "**Hello World**." The result should look like:

20. From the **Toolbox**, drag a **Button** control onto **Form1**. Right click on the button to open its properties and then change the **Text** property to "**Click Me!**" The result should look like:

21. Double-click the button to open the Code Editor and then enter the following code within the button's Click event handler:

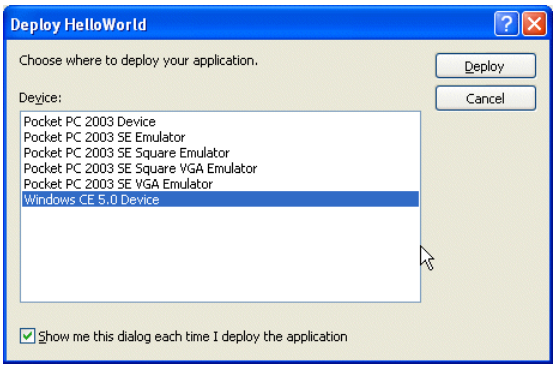
```
MessageBox.Show ("Hello World");
```



BUILDING AND DEPLOYING THE APPLICATION

To build and deploy the Hello World application to the FC-2500:

22. On the Standard toolbar, choose **Release** from the Solution Configurations list box.
23. From the Build menu, select **Deploy HelloWorld**. Visual Studio will then build the application and display the Deploy Selection dialog box.



24. Choose **Windows CE 5.0 Device** as the target smart device and then click **Deploy**. Visual Studio will transfer and install the Hello World application to the "Program Files" folder on the FC-2500.

RUNNING THE APPLICATION ON THE FC-2500

To run the HelloWorld application on the FC-2500:

25. On the FC-2500, open **My Device**, navigate to the "Program files\Hello World" folder .
26. Double tap the HelloWorld application to execute the program.

27. Tap **Click Me!** to display "Hello Word."

28. Tap **OK** to close the form.

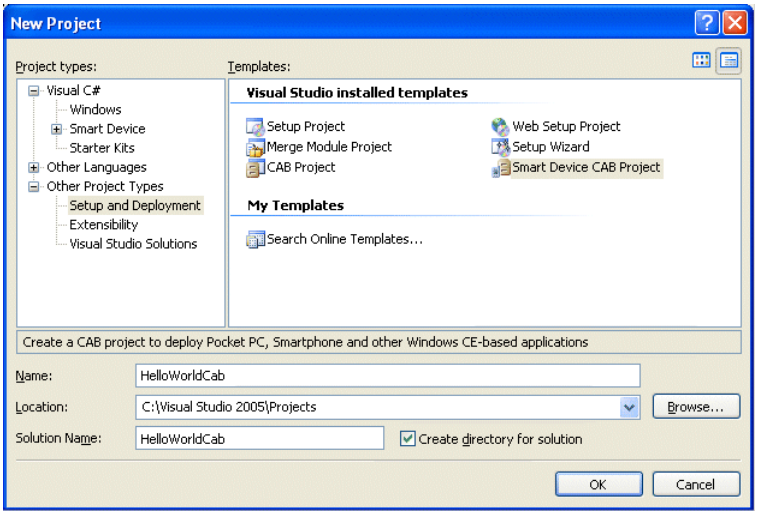
CREATING A REDISTRIBUTABLE CAB FILE

The following procedure will create a redistributable CAB file that contains the “Hello World” created in the previous section and how to deploy it on the FC-2500.

To create a redistributable CAB file project:

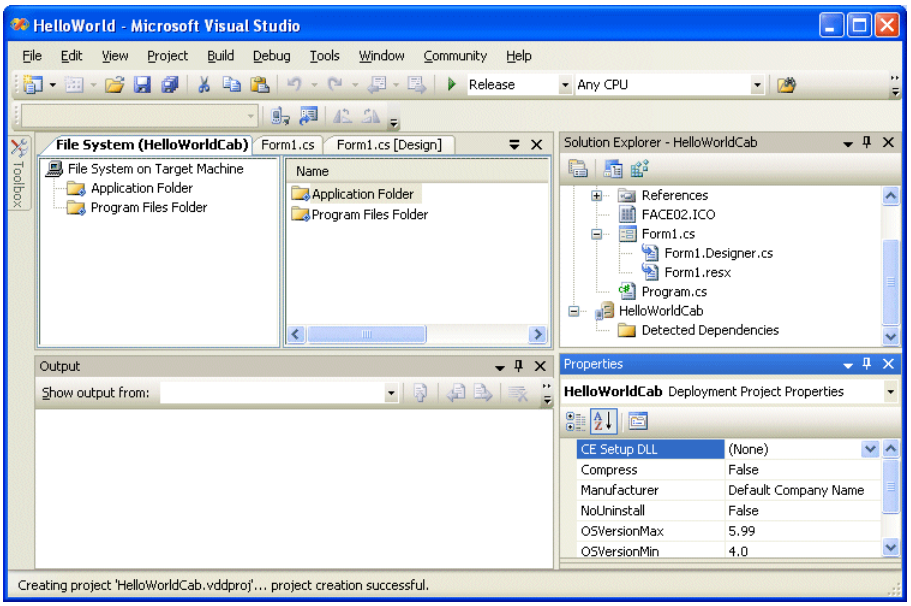
29. Open the Hello World Project that you created in previous section.

30. From the File menu, select **Add**, and then **New Project**. The New Project dialog box appears.



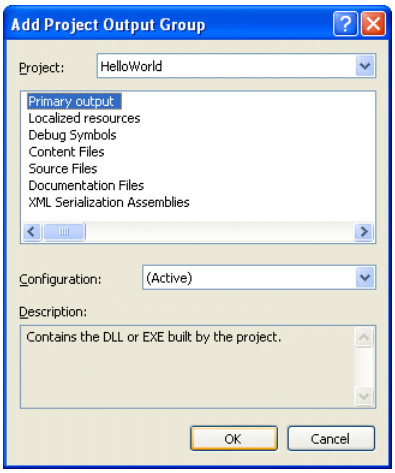
31. Under Project Types, select **Other Project Types** and then **Setup and Deployment**. Under Templates, select **Smart Cab Project**.

32. Type the **Name** (such as HelloWorldCab), select the **Location** of the project (such as C:\Visual Studio\Projects) and then click **OK** to create the project. The Visual Studio Cab Project Designer opens.



33. In the Properties window, change the **Manufacturer** property to "**My Company**" and the **ProductName** property to "Hello World.”

34. In the File System Editor, right-click the **Application Folder,** select **Add** and then **Project Output**. The Add Project Output dialog box opens.

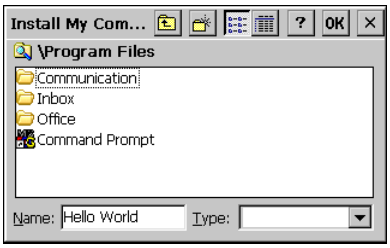


35. Select **Primary Output** and click **OK**.

36. From the **Build** menu, select Build **HelloWorld**. Visual Studio will then construct the cab file and place it in a subfolder of the location you selected when you created the project (such as C:\Visual Studio\Projects\HelloWorld\HelloWorldCab\Release).

37. Via Activesync, drag and drop the cab file to the **SystemCF** folder on the FC-2500.

38. On the FC-2500, open **My Device**, navigate to the **SystemCF** folder and double-tap the cab file to install it.



39. By default, the CAB file installation program will attempt to install the HelloWorld application to the **Program Files** folder (which is part of temporary virtual memory). To store the application permanently, select the **SystemCF** folder instead.

40. To run the HelloWorld application, open the Hello World folder and double-tap **HelloWorld**.

CHAPTER 7

INCORPORATING THE JETTCE.DLL

INTRODUCTION

JETTce.DLL provides developers a method to incorporate some or all of the following functions into their eMbedded C++ 4.0, Visual Studio .NET 2003 and Visual Studio 2005 applications. The JETTce.DLL is included with the FC-2500 operating system and is located in the Windows folder.

- **Keypad Backlight**—on units with LED backlight keypads, this function enables you to control the keypad backlight
- **Auxiliary Power**—on units with a COM port that supplies power, these function enable you to control the power for connected devices, such as RFID modules and bar code readers.
- **Soft Keyboard**—these functions enable you to control the FC-2500's soft keyboard program (CeKeys)
- **Screen Brightness**—these functions enable you to increase/decrease the screen brightness in 20 percent increments
- **Suspend/Resume**—these functions enable you to control suspend and resume tasks
- **Display Version Number**—this function returns the version number of the operating system
- **MAC Address**—this function returns the MAC address of the FC-2500
- **LEDs**—this function enable you to control the LEDs on the FC-2500

KEYPAD BACKLIGHT FUNCTIONS

RUNWAYLEDS

For FC-2500s with LED backlight keypads, this function enables you to turn off or on the keypad backlight

SYNTAX

```
DWORD RunwayLEDs( DWORD dwFunction );
```

PARAMETERS

Arguments	Values
dwFunction	To turn the LED backlight for the keypad off, set the value to 0 To turn the LED backlight for the keypad on, set the value to1 Any other value will just return the status

RETURN VALUES

Values	Description
0	Indicates that the LED backlight for the keypad is off
1	Indicates that the LED backlight for the keypad on

AUXILIARY POWER FUNCTIONS

TURNAUXSWITCHON

This function turns on the power for auxiliary devices.

SYNTAX

```
void TurnAuxSwitchOn( void );
```

PARAMETERS

None

RETURN VALUES

None

TURNAUXSWITCHOFF

This function turns off the power for auxiliary devices.

SYNTAX

```
void TurnAuxSwitchOff( void );
```

PARAMETERS

None

RETURN VALUES

None

AUXSWITCHISON

This function returns the power status (On/Off) for auxilary devices.

SYNTAX

```
BOOL AuxSwitchIsOn( void );
```

PARAMETERS

None

RETURN VALUES

Values	Description
TRUE	Indicates auxiliary power is on.
FALSE	Indicates auxiliary power is off.

SOFT KEYBOARD FUNCTIONS

ISCEKEYSRUNNING

This function returns the status of the CeKeys data entry program, indicating if the program is currently active. If the program is running, it may be minimized.

SYNTAX

```
BOOL IsCeKeysRunning( void );
```

PARAMETERS

None

RETURN VALUES

Values	Description
TRUE	Indicates that CeKeys is active.
FALSE	Indicates that CeKeys is inactive.

ISCEKEYSDISPLAYED

This function returns the display status of the CeKeys data entry program, indicating whether the program is minimized or open. If the CeKeys program is not currently running, the return value is FALSE.

SYNTAX

```
BOOL IsCeKeysDisplayed( void );
```

PARAMETERS

None

RETURN VALUES

Values	Description
TRUE	Indicates that CeKeys is open on the desktop.
FALSE	Indicates that CeKeys is minimized in the system tray or inactive.

RUNCEKEYS

This function allows the CeKeys data entry program to be started programmatically.

SYNTAX

```
BOOL RunCeKeys( void );
```

PARAMETERS

None

RETURN VALUES

Values	Description
TRUE	Indicates that CeKeys successfully executed.
FALSE	Indicates that CeKeys failed to execute.

DISPLAYCEKEYS

This function opens the CeKeys data entry program (if running) on the desktop.

SYNTAX

```
BOOL DisplayCeKeys( void );
```

PARAMETERS

None

RETURN VALUES

Values	Description
TRUE	Displays CeKeys (if running) on the desktop.
FALSE	Indicates that it could not display CeKeys.

HIDECEKEYS

This function minimizes the CeKeys data entry program (if running) in the system tray.

SYNTAX

```
BOOL HideCeKeys( void );
```

PARAMETERS

None

RETURN VALUES

Values	Description
TRUE	Minimizes CeKeys (if running) in the system tray.
FALSE	Indicates that it could not hide CeKeys.

CENTERCEKEYS

This function places the CeKeys data entry program (if running) in the center of the desktop.

SYNTAX

```
BOOL CenterCeKeys( void );
```

PARAMETERS

None

RETURN VALUES

Values	Description
TRUE	Places CeKeys (if running) in the center of the desktop.
FALSE	Indicates that it could not center CeKeys.

SHUTDOWNCEKEYS

This function terminates the CeKeys data entry program.

SYNTAX

```
BOOL ShutDownCeKeys ( void );
```

PARAMETERS

None

RETURN VALUES

Values	Description
TRUE	Indicates that it successfully closed CeKeys (if running).
FALSE	Indicates that it could not close CeKeys.

SCREEN BRIGHTNESS FUNCTIONS

INCBRIGHTNESS

This function enables you to increase the screen brightness in 20 percent increments.

SYNTAX

```
void IncBrightness( void );
```

PARAMETERS

None

RETURN VALUES

None

DECBRIGHTNESS

This function enables you to decrease the screen brightness in 20 percent increments. This function may aid in maximizing battery life.

SYNTAX

```
void DecBrightness( void );
```

PARAMETERS

None

RETURN VALUES

None

SUSPEND/RESUME FUNCTIONS

SUSPENDDEVICE

This function allows you to suspend the FC-2500 programmatically. This function may aid in maximizing battery life.

SYNTAX

```
void SuspendDevice( void );
```

PARAMETERS

None

RETURN VALUES

None

SUSPEND_KEY_LOCKOUT_OFF

This function enables Suspend/Resume using the FC-2500 keypad.

SYNTAX

```
BOOL Suspend_Key_Lockout_Off( void );
```

PARAMETERS

None

RETURN VALUES

Values	Description
TRUE	Indicates that the keypad Suspend/Resume key is active.
FALSE	Indicates that the function failed.

SUSPEND_KEY_LOCKOUT_ON

This function disables the Suspend/Resume using the FC-2500 keypad.

SYNTAX

```
BOOL Suspend_Key_Lockout_On( void );
```

PARAMETERS

None

RETURN VALUES

Values	Description
TRUE	Indicates that the keypad Suspend/Resume key is not active.
FALSE	Indicates that the function failed.

SUSPEND_KEY_LOCKOUT

This function returns the status of the current suspend/resume lockout state.

SYNTAX

```
BOOL Suspend_Key_Lockout( PBOOL pbLockoutState );
```

PARAMETERS

Values	Description
pbLockoutState	[Out] Pointer to BOOL that will receive the status of the suspend lockout state.

RETURN VALUES

Values	Description
TRUE	Indicates success.
FALSE	Indicates failure.

SUSPEND_KEY_LOCKOUT_STATE

This function allows you enable, disable and return the status of the Suspend/Resume key.

SYNTAX

```
BOOL Suspend_Key_Lockout_State( PDWORD pdwLockoutState );
```

PARAMETERS

Values	Description
pdwLockoutState	[In] Pointer to a DWORD that holds one of the following enumerated values: • LOCKOUT_OFF = 0 • LOCKOUT_ON = 1 • LOCKOUT_READ = 2 [Out] If set to LOCKOUT_READ, upon return, the location pointed to will hold the current lockout state: TRUE = 1 if lockout is on, FALSE = 0 if lockout is off

RETURN VALUES

Values	Description
TRUE	Indicates success.
FALSE	Indicates failure.

MACADDRESS FUNCTIONS

The FC-2500's MAC address is a 48-bit globally unique value that is composed of the 12-bit Two Technologies, Inc. organizationally unique identifier (OUI) that is registered with the IEEE Registration Authority and a 12-bit Two Technologies, Inc. assigned serial number. The 48-bit GUI is unique among all Ethernet and Bluetooth devices worldwide. The 48-bit MAC address is typically viewed as six pairs of hexadecimal digits, e.g. 00:03:CF:12:34:56.

GETMACADDRESS

This function returns the 48-bit MAC address of the FC-2500 as two 32-bit DWORDs. The lowest 32-bits of the MAC address are written into the location pointed to by pLowMac. The highest 16-bits of the MAC address are written into the low bit positions of location pointed to by pHighMac and the remaining bit positions are set to zero.

For example:

High Value: 0x00000030, Low Value: 0xCF001453

Since the high 24-bits of the MAC address are fixed for all Two Technologies, Inc. devices, the value in pHighMac will always be the same. You can use the pLowMac value to uniquely identify each FC-2500.

SYNTAX

```
BOOL GetMacAddress( PDWORD pLowMac, PDWORD pHighMac );
```

PARAMETERS

Values	Description
pHighMac	[Out] Points to an integer variable that will receive the highest 16-bits of the MAC address
pLowMac	[Out] Points to an integer variable that will receive the lowest 32-bits of the MAC address.

RETURN VALUES

Values	Description
TRUE	Indicates success.
FALSE	Indicates failure.

DISPLAY VERSION FUNCTIONS

GETNKBINVERSION

This function returns a 32-bit value that contains the OS version number and the build image version number.

SYNTAX

```
DWORD GetNkBinVersion( void );
```

PARAMETERS

None

RETURN VALUES

Values	Description
MSB	Most Significant Byte = Windows CE major version number
NSB	Next Most Significant Byte = Windows CE minor version number
LSS	Least significant Short (two bytes) = build image (NK.GZ) version number

For example a return value of 05 00 0038 would indicate the following:

Windows CE Major Version = 5

Windows CE Minor Version = 0

Build Image (NK.GZ) Version = 26

You can also use a byte packed structure to access each of the fields, for example:

```
#include <pshpack1.h>                  // pack on byte boundaries
typedef struct
{
      USHORT usNkBinVersion;           // 16 bit field for the individual build number
      UCHAR  ucCeMinorVersion;         // 8 bit field for the CE minor version
      UCHAR  ucCeMajorVersion;         // 8 bit field for the CE major version
}  VERSION_INFO, *PVERSION_INFO;
#include <poppack.h>                   // restore packing
```

LED FUNCTIONS

LEDUPDATE

This function enables you to turn on and turn off the LEDs on the FC-2500.

SYNTAX

```
BOOL LedUpdate( LEDVALUE ledVal, BOOL bValue );
```

PARAMETERS

Values	Description
ledVal	[In] One of the following enumerated values specifying the LED to control: ▪ LED_2ND = 0 ▪ LED_SHIFT = 1 ▪ LED_CTRL = 2 ▪ LED_ALT = 3 ▪ LED_CAPS = 4
bValue	[In] TRUE turns on specified LED. FALSE turns off specified LED.

RETURN VALUES

Values	Description
TRUE	Indicates success.
FALSE	Indicates failure.

CHAPTER 8

GENERATING AUDIO TONES

INTRODUCTION

The FC-2500 has a beep driver that you can incorporate into an application for various purposes, such as notifying an operator that a malfunction occurred or that a process has finished. You can access the beep driver directly or via Beep.DLL. The Beep.DLL is included with the FC-2500 operating system and is located in the Windows folder.

ACCESSING THE BEEP DRIVER DIRECTLY

The following code snippets (written in eMbedded Visual C++ 4.0) shows you how to incorporate beep driver calls directly into an application

BEEP DRIVER FILE STRUCTURE

The following code shows the file structure used by the beep driver. It also lists the minimum and maximum allowable values for each parameter.

```
#ifndef __BEEP_H__
#define __BEEP_H__
typedef struct
{
        DWORD dwFrequency;              // Frequency
        DWORD dwVolume;                 // Volume
        DWORD dwDurationMs;             // Beep duration in ms
}  BEEP_USER,*PBEEP_USER;

#define FREQUENCY_MIN 56                // Frequency Min
#define FREQUENCY_MAX 20000             // Frequency Max
#define VOLUME_MIN 1                    // Normalized Volume Min
#define VOLUME_MAX 100                  // Normalized Volume Max
#define DURATIONMS_MIN 0x0              // DurationMs Min
#define DURATIONMS_MAX 10000            // DurationMs Max - 10 secs
BEEP_USER bsBeep;
HANDLE hBEPDevice = NULL;
DWORD dwNumberofBytesRead = 0;
#endif
```

CREATING AND WRITING TO THE BEEP DRIVER FILE

The following code shows how to specify the beep parameters, create a file to store the parameters, validate that the parameters are within range and then write the values to the file.

If you are generating a series of beeps, you should put a delay of at least one millisecond between each beep to avoid sound overlap.

```
bsBeep.dwFrequency = 500;
bsBeep.dwVolume = 20;
bsBeep.dwDurationMs = 1000;
hBEPDevice = CreateFile(L"BEP1:",
GENERIC_READ | GENERIC_WRITE,
FILE_SHARE_READ | FILE_SHARE_WRITE,
NULL,
OPEN_EXISTING,
FILE_ATTRIBUTE_NORMAL,
NULL);
WriteFile(hBEPDevice,&bsBeep,sizeof(bsBeep),&dwNumberofBytesRead,0);
CloseHandle(hBEPDevice);
```

ACCESSING THE BEEP DRIVER VIA THE BEEP.DLL

You can access the beep driver via the Beep.DLL. Parameters that you can define include frequency, volume and duration (in milliseconds).

PLAYTONE

This function enables you to access the beep driver for various purposes, such as notifying an operator that a malfunction occurred or that a process has finished. Parameters that you can define include frequency, volume and duration (in milliseconds).

SYNTAX

```
bool PlayTone(
int iFrequency,
int iVolume,
int iDurationMS
);
```

PARAMETERS

Values	Description
iFrequency	Specifies the beep frequency. Range: 56 to 20000 KHz.
iVolume	Specifies the beep volume. Range: 1 to 100.
iDurationMS	Specifies the beep duration in milliseconds Range 1 to 10000 (10 seconds).

RETURN VALUES

Values	Description
TRUE	Indicates success.
FALSE	Indicates failure.

CHAPTER 9

INCORPORATING THE EYE•WARE.DLL

ABOUT THE FC-2500 CAMERA

The FC-2500 uses a high performance 5.17 megapixel image sensor to digitally capture images. The camera contains an image sensor which measures light. On the sensor is a grid of tiny light-sensitive diodes, called photosites or pixels. The more pixels a camera has the more detail it can capture and the larger photographs can be without becoming blurry or "grainy."

When you capture an image, the image sensor records the amount of light that falls on each pixel. The more light that hit the pixels, the greater the electrical charge recorded. Since the image sensor only records light intensity and not color. The camera must then perform a process called interpolation, which involves placing red, green and blue filters over the pixels to calculate the color. The camera's processor then compresses stores and the image.

IMAGE TYPES

The JETTcamera.DLL can create the following image files types:

- **JPG**—this file format compresses images and offers the best trade-off between quality, speed and portability.
- **BMP**— Bitmap files create an accurate representation of the image and are useful if you need to do custom post-processing on the FC-2500 or send the image to a portable printer.

IMAGE RESOLUTIONS

The FC-2500 camera has four available resolutions:

- 320 x 200 (HF)
- 640 x 480 (VGA)
- 1280 x 960 (SXVGA)
- 2048 x 1944 (QSXGA)

HF and VGA modes are low-quality modes that you can use to generate quick photographs. The size of HF and VGA JPG images is usually under 500 KB. SXGA and QSXGA images are far better quality but are very large (up to 3.5 MB per JPG, up to 11.5 Mb per BMP). -

ABOUT EYE•WARE AND THE EYE•WARE DLL

Two Technologies' eye•WARE is an optional "wedge" standalone program allows you to easily take photographs or capture bar codes and pass data to a text field that has focus in an application.

The eye•WARE DLL (TwoTechEyeWARELibraryCE.DLL) is written in.NET Compact Framework v 2.0, and allows you to simulate keypress events, read certain camera properties, and detect specific camera events.

EYE•WARE SETUP UTILITIES

Eye•WARE has two setup utilities that create XML configurations files that define the behavior of the camera for photograph taking ands scanning operations.

- The **eye•SEE Setup** utility allows you to create and save custom camera configurations as an XML file. CWCONF.CT2 is the default file name. Each time you start Eye•WARE, it will load CWCONF.CT2. Custom files that you create must be loaded programmatically.
- The **eye•D Setup** utility allows you to create and save custom barcode configurations as an XML file. BWCONF.BT2 is the default file name. Each time you start Eye•WARE, it will load BWCONF.BT2. Custom files that you create must be loaded programmatically.

For more information about using eye•WARE, please refer to the *eye•WARE User's Guide*.

INTEGRATION NOTES

When writing an application, you will need to use Visual Studio 2005 and open a Windows CE 5.0 Smart Device Application and add the eye•WARE DLL as a resource.

- To allow a user to scan barcode characters into 3rd party programs, eye•WARE will always attempt to place the text for a decoded barcode in the last field that had focus before the barcode key press event was detected. The scan result will always go where you tell is to go, not where you want it to go.

  For example, if a button had focus when a scan session starts, that button box will receive the decoded characters, even if it cannot display the characters. As a result, you should wait for a successful scan, and then redirect the ScanResult property to the desired textbox.
- You cannot take a photograph while the auto-focus motor is running.
- You can combine keypad key presses and key press events. For example, you can map one button on the keypad to turn on the viewfinder, lasers and LEDs, and a second button or a timer event to send a second camera event and take the photograph.
- Eye•WARE relies on two methods to take photographs or scan barcodes: Single-Press mode and Double-Press mode.
  - Single-Press mode will allow users to turn on viewfinder, lasers and LEDs (provided those options were selected in the configuration file), perform an auto-focus, and then take a photograph with one key press.
  - Double-Press mode will allow users to turn on viewfinder, lasers and LEDs (provided those options were selected in the configuration file) and perform an auto-focus with one key press and take a photograph with a second key press.
- By default, photographs taken with eye•WARE are stored to the My Documents folder on the FC-2500, which exists in RAM and provides the fastest method of storing photographs. However, this folder will not retain photographs if the FC-2500 experiences a power failure or remove power from the FC-2500.
- As a precautionary measure, you should periodically use a background process or a program (such as Two Technologies, File Transfer Utility) to move the files to the SystemCF folder
- The eye•WARE DLL will always attempt to start eye•WARE when the eye•WARE object is initialized, but can overload the eye•WARE constructor with FALSE to prevent eye•WARE from starting when the object is initilized.
- The eye•WARE object only swaps out configuration files when a camera or barcode event is detected, or when the object is destroyed. This allows you to set a configuration upon key press or automatically reset the file upon the exit of a program.
- Some camera events are stored in properties, so that you can access events with a timer, or held for evaluation. The Capture event equates to the EventStatus property. The Timeout event equates to the Timeout property.

- The ScanResult property will return the estimated distance in inches between the unit and the target barcode in the event of a barcode scan failure. This can be used to prompt the user to move inward or backward from the target barcode based on the minium and maximum read distance from eyeD setup.

EYE•WARE DLL ENUMERATIONS

Enumerations are a method of storing groups of named constants within your code using an integer value from a built in integer type. The .NET CF automatically assigns values for the members starting at 0.

***Note**: Unless otherwise specified all declarations, enumeration values and examples appear in C#*

ACCESSORY_STATE

This enumeration gets or sets the status of the viewfinder, lamp or laser (also known as accessories).

DECLARATION

```
public enum ACCESSORY_STATE
{
        STATE_ON,
        STATE_OFF,
        STATE_QUERY,
};
```

ENUMERATION VALUES

Enumeration	Values	Description
STATE_ON	0	Turns the accessory on
STATE_OFF	1	Turns the accessory off
STATE_QUERY	2	Gets the current status of the accessory

ACTION_STATUS

This enumeration gets the result of the last keyboard event.

DECLARATION

```
public enum ACTION_STATE
{
        NONE
        SUCCESS,
        FAILURE,
        PREPARE_SUCCESS,
        PREPARE_FAILURE,
        CAPTURE_FAILURE,
        CAPTURE_SUCCESS
};
```

ENUMERATION VALUES

Enumeration	Values	Description
NONE	0	No current action
SUCCESS	1	Not used at this time
FAILURE	2	Not used at this time
PREPARE_SUCCESS	3	Accessories turned on successfully
PREPARE_FAILURE	4	An accessory failed to turn on
CAPTURE_FAILURE	5	The photograph or barcode scan failed
CAPTURE_SUCCESS	6	The photograph or barcode scan succeeded

BARCODE_STATUS

This enumeration gets the status of the last barcode attempt.

DECLARATION

```
public enum BARCODE_STATE
{
        BARCODE_UNKOWN
        BARCODE_SUCCESS
        BARCODE_BINARIZATION_ERROR
        BARCODE_CANNOT_DETECT
        BARCODE_CANNOT_DECODE
        BARCODE_LAMP_FAILURE
        BARCODE_LASER_FAILURE
        BARCODE_VF_FAILUREe
        BARCODE_ROI_FAIL
        BARCODE_VF_PHOTOGRAPH_FAIL
        BARCODE_DEMOSAIC_FAIL
        BARCODE_LASER_FOCUS_FAIL
        BARCODE_UNIT_TOO_FAR_FROM_TARGET
        BARCODE_UNIT_TOO_CLOSE_TO_TARGET
        BARCODE_EXPOSURE_ADJUST_FAILED
        }
```

ENUMERATION VALUES

Enumeration	Values	Description
BARCODE_UNKOWN	0	The last barcode is in an unknown status
BARCODE_SUCCESS	1	Barcode capture was a success
BARCODE_BINARIZATION_ERROR	2	Memory Error
BARCODE_CANNOT_DETECT	3	Cannot detect barcode
BARCODE_CANNOT_DECODE	4	Cannot decode barcode
BARCODE_LAMP_FAILURE	5	Lamp failed to activate
BARCODE_LASER_FAILURE	6	Laser failed to activate
BARCODE_VF_FAILURE	7	Viewfinder failed to activate
BARCODE_ROI_FAIL	8	Cannot enable or disable the ROI/Spot meter
BARCODE_VF_PICTURE_FAIL	9	Failed to capture frames in the viewfinder
BARCODE_DEMOSAIC_FAIL	10	Could not demosaic image
BARCODE_LASER_FOCUS_FAIL	11	Laser focus failed
BARCODE_UNIT_TOO_FAR_FROM_TARGET	12	FC-2500 is too far from the target barcode
BARCODE_UNIT_TOO_CLOSE_TO_TARGET	13	FC-2500 is too close to the target barcode
BARCODE_EXPOSURE_ADJUST_FAILED	15	Exposure adjust failed

CAMERA_STATUS

This enumeration gets the status of the camera.

DECLARATION

```
public enum ACTION_STATE
{
        CAMERA_ERROR
        CAMERA_READY
        CAMERA_HOLD_STEADY
        CAMERA_TAKING_PICTURE
        CAMERA_GENERATING_PROOF
        CAMERA_SAVING
        CAMERA_POST_PROCESS
        CAMERA_READY_TO_SAVE
        CAMERA_SHUTTING_DOWN
        CAMERA_BOOT
        CAMERA_DECODING
        CAMERA_REDUCING_IMAGE
        CAMERA_DISPLAYING_IMAGE
};
```

ENUMERATION VALUES

Enumeration	Values	Description
CAMERA_ERROR	0	The camera is in an error state (check event log)
CAMERA_READY	1	The camera is ready to take a photograph
CAMERA_HOLD_STEADY	2	The camera is about to take a photograph
CAMERA_TAKING_PICTURE	3	The camera is taking a photograph
CAMERA_GENERATING_PROOF	4	Deprecated in this version
CAMERA_SAVING	5	The camera is saving a file to RAM or System CF card
CAMERA_POST_PROCESS	6	Deprecated in this version
CAMERA_READY_TO_SAVE	7	The camera has taken a photograph and is ready to save. It can do nothing else until you save or delete the photograph.
CAMERA_SHUTTING_DOWN	8	The camera is powering down
CAMERA_BOOT	9	The camera is booting
CAMERA_DECODING	10	The camera is decoding a barcode
CAMERA_REDUCING_IMAGE	11	Reserved for future use
CAMERA_DISPLAYING_IMAGE	12	Reserved for future use

EYE•WARE DLL PROPERTIES

Properties are members that provide a flexible mechanism to read, write, or compute the values of private fields. You can use Properties as though they are public data members, but they are actually special methods called accessors. This enables you to access data easily while still providing the safety and flexibility of methods.

***Note**: Unless otherwise specified all declarations and examples appear in C#*

AUTOFOCUSMOTORACTIVE

This property holds the status of the AutoFocus motor as either running (true) or idle (false). You cannot take a photograph while the auto focus motor is running.

DECLARATION

```
using TwoTecheyeWARELibraryCE;

eyeWARE MyEyeWARE = new eyeWare( );
bool bFocusActive;
bFocusActive =MYEyeWARE.AutoFocusMotorActive;
```

BARCODECONFIGFILELOCATION

This property holds the full path of the barcode configuration file.

DECLARATION

```
using TwoTecheyeWARELibraryCE;

eyeWARE MyEyeWARE = new eyeWare( );
MYEyeWARE.BarcodeConfigFileLocation;
```

EXAMPLE

```
MYEyeWARE.BarcodeConfigFile = "\\SystemCF\\eyeWARE\\newconfigfile.b2t";
```

BARCODESTATUS

This property holds the status of the last barcode attempt.

DECLARATION

```
using TwoTecheyeWARELibraryCE;

eyeWARE MyEyeWARE = new eyeWare( );
BARCODE_STATUS MyStatus;
MyStatus =MYEyeWARE.Status;
```

CAMERACONFIGFILELOCATION

This property holds the full path of the camera configuration file.

DECLARATION

```
using TwoTecheyeWARELibraryCE;

eyeWARE MyEyeWARE = new eyeWare( );
MYEyeWARE.CameraConfigFileLocation;
```

EXAMPLE

```
MYEyeWARE.CameraConfigFile = "\\SystemCF\\eyeWARE\\newconfigfile.c2t";
```

DIRECTORYFULL

This property contains the current directory condition, where TRUE indicates that the directory is full. FALSE indicates that the directory still has available space.

Because eye•WARE does not know where it will save a file until a camera event or a camera button press occurs, "Directory Full" detection only works when eye•WARE enters a photograph taking session which usually happens on the first press.

If the location is full, the "Memory Full" message will display (if enabled) and the DirectoryFull property is set to TRUE and will stay TRUE until another camera event or button press discovers that the directory location has available space in which case the DirectoryFull property is set to FALSE.

DECLARATION

```
using TwoTecheyeWARELibraryCE;

eyeWARE MyEyeWARE = new eyeWare( );
bool DirectoryFull;
bDirectoryFull t =MYEyeWARE.DirectoryFull;
```

EVENTSTATUS

This property holds the last action performed by a barcode or camera key press event. This will record button events or events sent from the DLL.

DECLARATION

```
using TwoTecheyeWARELibraryCE;

eyeWARE MyEyeWARE = new eyeWare( );
ACTION_STATUS LatestEvent;
LatestEvenet = MYEyeWARE.EventStatus;
```

FILENAME

This property holds the filename for the photograph output file. You must include your own file extension (BMP or JPG) in the filename.

DECLARATION

```
using TwoTecheyeWARELibraryCE;

eyeWARE MyEyeWARE = new eyeWare( );
MYEyeWARE.Filename;
```

EXAMPLE

```
MYEyeWARE.Filename = "MyPic.JPG";
```

FILEPATH

This property holds the directory path for the output file.

DECLARATION

```
using TwoTecheyeWARELibraryCE;

eyeWARE MyEyeWARE = new eyeWare( );
MYEyeWARE.FilePath;
```

EXAMPLE

```
MYEyeWARE.FilePath = "\\SystemCF\\eyeWARE\\MyPics";
```

FocusTableIndex

Use this property to define the manual focus range. It stores the current array index of the manual focus table as read from the SystemCF\eyeWARE\FocusTable.txt file and will return a rounded down value closest to the true motor position.

For example, if the true focus motor position is 37 and the FocusTable.txt file lists steps 0, 10, 20, 30, 40, 50, the FocusTableIndex will read an index of 4 because 37 is closest to 40.

If no focus table is detected, a default 16-step table with values ranging from 0 to 150 will be substituted.; -1 indicates that the motor position cannot be determined.

***Note**: After issuing a motor command, do not call the FocusTableIndex property on the next line. If you call the motor position in the middle of a move, you will get the motor's current location, but not the motor's final location. It is best to keep motor position information on a timer, or check the motor position property approximately two seconds after issuing any motor position commands.*

Declaration

```
using TwoTecheyeWARELibraryCE;

eyeWARE MyEyeWARE = new eyeWare( );
Int FocusTableIndex;
i FocusTableIndex =MYEyeWARE. FocusTableIndex;
```

ScanResult

This property holds the decoded character values of a barcode as a string. This property will always hold the last successful value.

Declaration

```
using TwoTecheyeWARELibraryCE;

eyeWARE MyEyeWARE = new eyeWare( );
string sMyCaptureBarcode;
sMyCaptureBarcode= MyEyeWARE.Scan Result;
```

Status

This property holds the current state of the camera.

Declaration

```
using TwoTecheyeWARELibraryCE;

eyeWARE MyEyeWARE = new eyeWare( );
CAMERA_STATUS MyStatus;
MyStatus =MYEyeWARE.Status;
```

Timeout

This property holds is the current timeout condition. TRUE indicates that the unit has timed out. FALSE indicates that the unit has not timed out or that eye•WARE is not engaged in a scan or camera session.

Declaration

```
using TwoTecheyeWARELibraryCE;

eyeWARE MyEyeWARE = new eyeWare( );
bool bUnitTimedOut;
bUnitTimedOut =MYEyeWARE.AutoFocusMotorActive;
```

EYE•WARE DLL METHODS

A *method* is a code block containing a series of statements. In C#, every executed instruction is done so in the context of a method.

***Note**: Unless otherwise specified all examples appear in C#*

ACCESSORY_STATE DISPLAY

This method changes the viewfinder state on and off, or gets the current viewfinder status. It will also return the new state.

SYNTAX

```
ACCESSORY_STATE Display(ACCESSORY_STATE Action)
```

PARAMETERS

Values	*Description*
ACCESSORY_STATE	Gets or sets the status of the viewfinder
Action	Specifies of the following: STATE_ON, STATE_OFF or STATE_QUERY

RETURN VALUES

Values	*Description*
STATE_ON	Turns the viewfinder on
STATE_OFF	Turns the viewfinder off

EXAMPLE

```
using TwoTecheyeWARELibraryCE;

eyeWARE MyEyeWARE = new eyeWARE( );
if (MyEyeWARE.Display(eyeWARE.ACCESSORY_STATE.STATE_QUERY) ==
eyeWARE.ACCESSORY_STATE.STATE_OFF)
{
MessageBox.Show("Viewfinder off");
}
```

ACCESSORY_STATE LASER

This method changes the laser state on ,off, or gets the current laser status and returns the new state.

SYNTAX

ACCESSORY_STATE Laser(ACCESSORY_STATE Action)

PARAMETERS

Values	Description
ACCESSORY_STATE	Gets or sets the status of the laser
Action	Specifies of the following: STATE_ON, STATE_OFF or STATE_QUERY

RETURN VALUES

Values	Description
STATE_ON	Turns the laser on
STATE_OFF	Turns the laser off

EXAMPLE

```
using TwoTecheyeWARELibraryCE;

eyeWARE MyEyeWARE = new eyeWARE( );
if (MyEyeWARE.Laser(eyeWARE.ACCESSORY_STATE.STATE_QUERY) ==
eyeWARE.ACCESSORY_STATE.STATE_ON)
{
MessageBox.Show("Laser on");
}
```

ACCESSORY_STATE LAMP

This method changes the lamp (the four white LEDs) state from on and off, or gets the current lamp status. The lamp will be toggled with a brightness level of the last known state, or the default state of 50% if no key press events have been detected, it will return the new state.

SYNTAX

ACCESSORY_STATE Lamp(ACCESSORY_STATE Action)

PARAMETERS

Values	Description
ACCESSORY_STATE	Gets or sets the status of the lamp
Action	Specifies of the following: STATE_ON, STATE_OFF or STATE_QUERY

RETURN VALUES

Values	Description
STATE_ON	Turns the lamp on
STATE_OFF	Turns the lamp off

EXAMPLE

```
using TwoTecheyeWARELibraryCE;

eyeWARE MyEyeWARE = new eyeWARE( );

if (MyEyeWARE.Lamp(eyeWARE.ACCESSORY_STATE.STATE_QUERY) ==
eyeWARE.ACCESSORY_STATE.STATE_ON)
          {
           MessageBox.Show("Lamp on");
         }
```

BARCODEEVENT

This method sends a barcode event key press command to the eye•WARE engine. The command returns false if the camera status is not CAMERA_READY.

SYNTAX

```
bool BarcodeEvent (void)
```

PARAMETERS

None

RETURN VALUES

Values	Description
TRUE	Indicates success.
FALSE	Indicates failure.

EXAMPLE

```
using TwoTecheyeWARELibraryCE;

eyeWARE MyEyeWARE = new eyeWARE( );

if (MyEyeWARE.AutoFocusMotorActive == false)
        {
        MyEyeWARE.BarcodeEvent();
        }
```

CAMERAEVENT

This method sends a camera event key press command to the eye•WARE engine. The command returns false if the camera status is not CAMERA_READY.

SYNTAX

```
bool CameraEvent(void)
```

PARAMETERS

None

RETURN VALUES

Values	Description
TRUE	Indicates success.
FALSE	Indicates failure.

EXAMPLE

```
using TwoTecheyeWARELibraryCE;

eyeWARE MyEyeWARE = new eyeWARE( );

if (MyEyeWARE.AutoFocusMotorActive == false)
        {
        MyEyeWARE.CameraEvent( );
        }
```

CANCEL

This method cancels a double-press session, stops the timeout clock and turns the status of any accessories to STATUS_OFF.

SYNTAX

```
bool Cancel(void)
```

PARAMETERS

None

RETURN VALUES

Values	Description
TRUE	Indicates success.
FALSE	Indicates failure.

EXAMPLE

```
using TwoTecheyeWARELibraryCE;

eyeWARE MyEyeWARE = new eyeWARE( );

MyEyeWARE.Cancel( );
```

SAVEIMAGE

This method sends a save command to the eye•WARE engine which instructs it to save a photograph in memory to a file. This method will return false if the status is not CAMERA_READY_TO_SAVE.

SYNTAX

```
bool SaveImage(void)
```

PARAMETERS

None

RETURN VALUES

Values	Description
TRUE	Indicates success.
FALSE	Indicates failure.

EXAMPLE

```
using TwoTecheyeWARELibraryCE;

eyeWARE MyEyeWARE = new eyeWARE( );

MyEyeWARE.SaveImage( );
```

DELETEIMAGE

This method sends a delete command to the eye•WARE engine which instructs it to delete a photograph from memory. This method will return false if the status is not CAMERA_READY_TO_SAVE.

SYNTAX

```
bool DeleteImage(void)
```

PARAMETERS

None

RETURN VALUES

Values	Description
TRUE	Indicates success.
FALSE	Indicates failure.

EXAMPLE

```
using TwoTecheyeWARELibraryCE;

eyeWARE MyEyeWARE = new eyeWARE( );

MyEyeWARE.DeleteImage( );
```

MANUALFOCUSBACK

This method moves the focus back one step in the manual focus table. If no focus table is detected, a default 16-step table with values ranging from 0 to 150 will be substituted.

SYNTAX

```
void SaveImage(void)
```

PARAMETERS

None

RETURN VALUES

Values	Description
TRUE	Indicates success.
FALSE	Indicates failure.

EXAMPLE

```
using TwoTecheyeWARELibraryCE;

eyeWARE MyEyeWARE = new eyeWARE( );

MyEyeWARE.SaveImage( );
```

MANUALFOCUSFORWARD

This method moves the focus forward one step in the manual focus table. If no focus table is detected, a default 16-step table with values ranging from 0 to 150 will be substituted.

SYNTAX

```
void SaveImage(void)
```

PARAMETERS

None

RETURN VALUES

Values	Description
TRUE	Indicates success.
FALSE	Indicates failure.

EXAMPLE

```
using TwoTecheyeWARELibraryCE;

eyeWARE MyEyeWARE = new eyeWARE( );

MyEyeWARE.SaveImage( );
```

MANUALFOCUSTOINDEX

This method moves the focus to a specfic place in the manual focus table. If no focus table is detected, a default 16-step table with values ranging from 0 to 150 will be substituted. Passing a -1 for the iFocusTableIndex will force the focus motor to go to the hyperfocal point.

SYNTAX

```
void SaveImage(int iFocusTableIndex)
```

PARAMETERS

None

RETURN VALUES

Values	Description
TRUE	Indicates success.
FALSE	Indicates failure.

EXAMPLE

```
using TwoTecheyeWARELibraryCE;

eyeWARE MyEyeWARE = new eyeWARE( );

MyEyeWARE.SaveImage( );
```

StartEngine

This method starts the eye•WARE engine if not started. The function will return FALSE if the eyeWARE engine does not exist.

Syntax

```
bool StartEngine( )
```

Parameters

None

Return Values

Values	Description
TRUE	Indicates success.
FALSE	Indicates failure.

Example

```
using TwoTecheyeWARELibraryCE;

eyeWARE MyEyeWARE = new eyeWARE( );

MyEyeWARE.StartEngine ( );
```

StopEngine

This method shuts down the eyeWARE engine when started.

Syntax

```
void SaveImage(void)
```

Parameters

None

Return Values

Values	Description
TRUE	Indicates success.
FALSE	Indicates failure.

Example

```
using TwoTecheyeWARELibraryCE;

eyeWARE MyEyeWARE = new eyeWARE( );

MyEyeWARE.SaveImage( );
```

EYE•WARE DLL EVENTS

An event is a notification by an application of a specific occurance.

Note: Unless otherwise specified all examples appear in C#

CANCELED

This event passes a boolean value when eye•WARE gets a request to cancel a double-press photograph

SYNTAX

```
void MyEyeWARE_Canceled(e)
```

RETURN VALUES

Values	Description
TRUE	Indicates success.
FALSE	Indicates failure.

EXAMPLE

```
using TwoTecheyeWARELibraryCE;

void MyEyeWARE_Canceled(object sender, CanceledEventArgs e)
        {
        lblCanceled.Text = e.Canceled.ToString();
        }
```

CAPTURE

This event passes a boolean value when the eyeWARE's photograph or barcode session has timed out. The boolean value e.Timeout is set to TRUE, although the state of the event is not needed to evaluate if a timeout has occoured.

This event is thrown when eye•WARE has performed an action and is reporting on the success or failure of that action. The description of the action is inside the e.CaptureAction event object. This event equates to an ACTION_STATUS, and will be thrown upon the activation of the accessories, and the successful or unsucessful capture of a photograph or barcode.

SYNTAX

```
void MyEyeWARE_Capture(e)
```

E.CAPTUREACTION RETURN VALUES

Enumeration	Values	Description
NONE	0	No current action
SUCCESS	1	Not used at this time
FAILURE	2	Not used at this time
PREPARE_SUCCESS	3	Accessories turned on successfully
PREPARE_FAILURE	4	An accessory failed to turn on
CAPTURE_FAILURE	5	The photograph or barcode scan failed
CAPTURE_SUCCESS	6	The photograph or barcode scan succeeded

EXAMPLE

```
using TwoTecheyeWARELibraryCE;

void MyEyeWARE_Capture(object sender, CaptureEventArgs e)
        {
        lblEventBar.Text = e.CaptureAction.ToString();
        }
```

FINISHEDSAVING

This event passes a boolean value for the e.FinishedSaving() save or delete event object. The event will return TRUE when eye•WARE is instructed to and completes a save file operation and return FALSE when eye•WARE is instructed to delete a file and completes a delete file operation.

SYNTAX

```
void MyEyeWARE_FinishedSaving(e)
```

RETURN VALUES

Values	Description
TRUE	Indicates success.
FALSE	Indicates failure.

EXAMPLE

```
using TwoTecheyeWARELibraryCE;

void MyEyeWARE_FinishedSaving(object sender, FinishedSavingEventArgs e)
        {
        lblFinishedSaving.Text = e.FinishedSaving.ToString();
        }
```

TIMEOUT

This event passes a boolean value when the eyeWARE's photograph or barcode session has timed out. The boolean value e.Timeout is set to TRUE, although the state of the event is not needed to evaluate if a timeout has occoured.

SYNTAX

```
void MyEyeWARE_Timeout(e)
```

RETURN VALUES

Values	Description
TRUE	Indicates success.
FALSE	Indicates failure.

EXAMPLE

```
using TwoTecheyeWARELibraryCE;

void MyEyeWARE_Timeout(object sender, TimeoutEventArgs e)
        {
        lblEventBar.Text = "Timeout";
        }
```

CHAPTER 10

KEYBOARD MAPPING

INTRODUCTION

Kbdtool.exe, designed to run on your development system, is a command line utility that creates a key map file. This key map file will remap the current FC-2500 keypad configuration externally (outside of an application), when the unit boots up.

During the boot sequence, the FC-2500 searches in the Windows folder in ascending alphanumeric order for existing key map files (identified by their ".RMT" extension). When the FC-2500 encounters a file of this type, it checks the key map ID number. If the ID number contained in the key map file matches the number stored in the FC-2500's hardware configuration block, the FC-2500 uses that value in that file to map to the keypad.

After creating your RMT file, you must copy it to the SystemCF folder on the FC-2500 and deploy it to the Windows folder during boot up. See Using FileCopy for more information.

The default keypad template file (**FC250055.TXT**) is included with the FC-2500 Developer's CD.

SYNTAX

`KBDTOOL filename.ext`

Option	Description
filename.ext	Specifies the name of the file containing the keypad template. The file name must follow MS-DOS 8.3 naming conventions. Default file name: 55 key Joystick Keypad = **FC250055.TXT**

EXAMPLE

The following example syntax executes kbdtool.exe using a FC250055.txt as it argument to create the file, FC250055.rmt for the 55 key keypad.

`KBDTOOL FC250055.txt`

ERRORS MESSAGES

Message	Description
Unable to open filename.ext.	KBTOOL cannot find the specified file. The specified file is named incorrectly
Unable to parse to scan code 'XXX' on NNN	The entry (XXX) is not a valid keyword on the specified line (NNN)
Invalid line NNN	The entry specified on line NNN is either misspelled, not allowed or not formatted correctly

ALLOWABLE KEY MAP VALUES

The following table lists the allowable values and the names of allowable values that you can map to a keypad.

Table 10-1: Allowable Key Map Values

A	V	CARET	F9	NUMPAD1	SEMICOLON
B	W	CLEAR	F10	NUMPAD2	SHIFT
C	X	COLON	F11	NUMPAD3	SLASH
D	Y	COMMA	F12	NUMPAD4	SPACE
E	Z	CONTROL	FUNCTION	NUMPAD5	STAR
F	(	DELETE	HASH	NUMPAD6	SUBTRACT
G	)	DOLLAR	HOME	NUMPAD7	TAB
H	[	DOUBLEQUOTE	INSERT	NUMPAD8	TILDA
I	]	DOWN	LEFT	NUMPAD9	UNDERLINE
J	{	END	KEY0	PAGEDOWN	UP
K	}	EQUAL	KEY1	PAGEUP	USER_DEF1
L	<	ESCAPE	KEY2	PAUSE	USER_DEF2
M	>	EXCLAMATION	KEY3	PERCENT	USER_DEF3
N	ADD	F1	KEY4	PERIOD	USER_DEF4
O	ALT	F2	KEY5	PIPE	USER_DEF5
P	AMPERSAND	F3	KEY6	PRINT	USER_DEF6
Q	AT	F4	KEY7	QUESTION	USER_DEF7
R	BACKQUOTE	F5	KEY8	QUOTE	USER_DEF8
S	BACKSLASH	F6	KEY9	RETURN	USER_DEF9
T	BACKSPACE	F7	NUMLOCK	RIGHT	USER_DEF10
U	CAPSLOCK	F8	NUMPAD0	SCROLL	WINMENU

***Notes**: Scan codes "USER_DEF1" through "USER_DEF10" can produce some proprietary action, such as backlight adjustment, display rotation, etc. For each user-defined key-function (except as noted below), you must supply the appropriate program or DLL and any parameters via the Hot Keys applet, which can be found in the Control Panel. Hot Key 1 corresponds to USER_DEF1, Hot Key 2 corresponds to USER_DEF2, etc.*

The number in COLS must always be set to five regardless of the actual number of columns.

Do not change the ID number; it must match the number stored in the hardware configuration block.

55–KEY KEYPAD MAP VALUES

The following table lists the default values for a 55-key keypad, where:

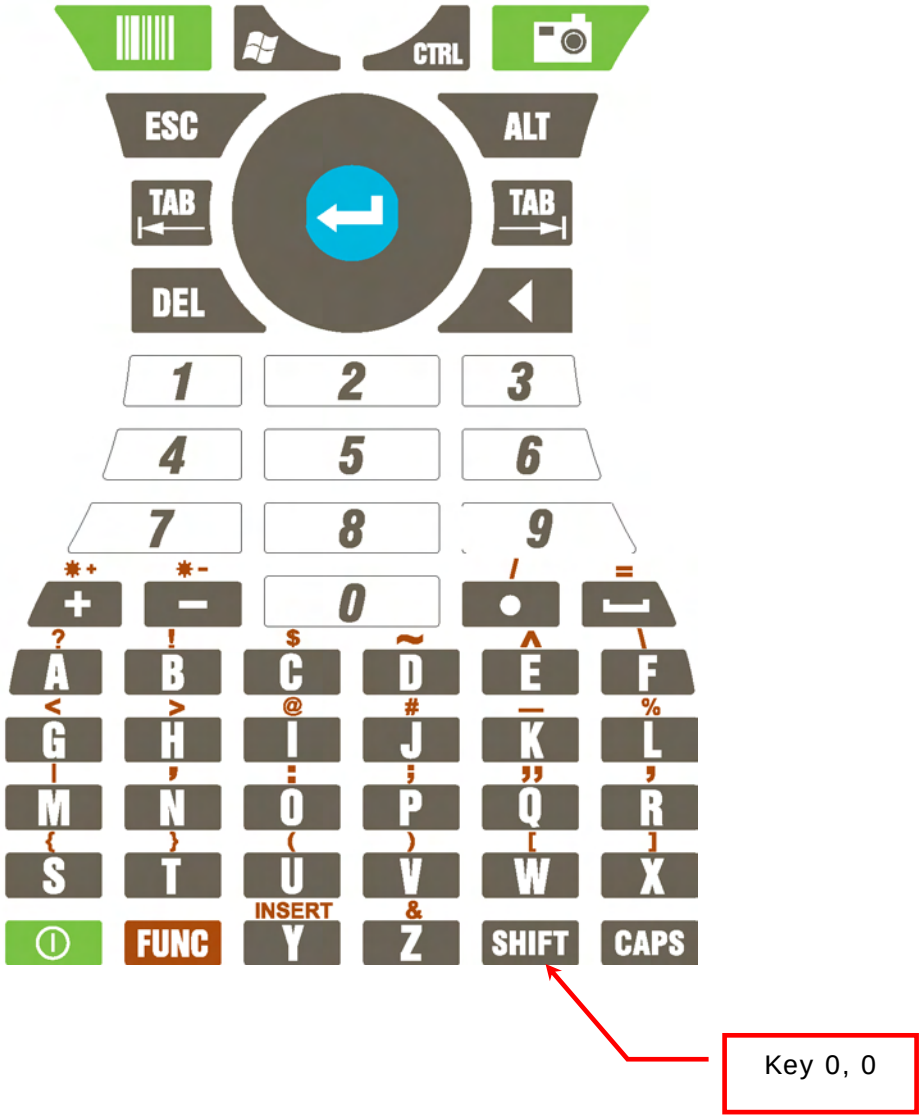
- Row 0, Column 0 is the key second to the right on the bottom row (SHIFT)
- The keyword "FUNCTION" is used for the "2nd" key
- The keyword "RETURN" is used for the "ENTER" key
- "USER_DEF1" defines the CAMERA TRIGGER (take photograph)
- "USER_DEF2" defines Barcode scan trigger
- "USER_DEF3" defines the Backlight-INCREASE key
- "USER_DEF4" defines the Backlight-DECREASE key
- "WINMENU" produces the Windows "Start" menu

Table 10-2: 55–Key Keypad Map Values

Row	Column	Default	FUNC	Shift
0	0	SHIFT	N/A	N/A
0	1	FUNCTION	N/A	N/A
0	2	Y	INSERT	N/A
0	3	Z	AMPERSAND	N/A
0	5	CAPSLOCK	N/A	N/A
1	0	S	{	N/A
1	1	T	}	N/A
1	2	U	(	N/A
1	3	V	)	N/A
1	4	W	[	N/A
1	5	X	]	N/A
2	0	M	PIPE	N/A
2	1	N	COMMA	N/A
2	2	O	COLON	N/A
2	3	P	SEMICOLON	N/A
2	4	Q	DOUBLEQUOTE	N/A
2	5	R	QUOTE	N/A
3	0	G	<	N/A
3	1	H	>	N/A
3	2	I	AT	N/A
3	3	J	HASH	N/A
3	4	K	UNDERLINE	N/A
3	5	L	PERCENT	N/A
4	0	A	QUESTION	N/A
4	1	B	EXCLAMATION	N/A
4	2	C	DOLLAR	N/A
4	3	D	TILDA	N/A
4	4	E	CARET	N/A

Row	Column	Default	FUNC	Shift
4	5	F	BACKSLASH	N/A
5	0	ADD	USER_DEF3	N/A
5	1	SUBTRACT	USER_DEF4	N/A
5	2	KEY0	N/A	N/A
5	3	N/A	N/A	N/A
5	4	PERIOD	SLASH	N/A
5	5	SPACE	EQUAL	N/A
6	0	KEY7	N/A	COLON
6	1	KEY4	N/A	SEMICOLON
6	2	KEY8	N/A	INSERT
6	3	KEY5	N/A	CAPSLOCK
6	4	KEY9	N/A	HOME
6	5	KEY6	N/A	N/A
7	0	KEY1	N/A	CLEAR
7	1	DELETE	N/A	SCROLL
7	2	KEY2	N/A	PAGEUP
7	3	N/A	N/A	N/A
7	4	KEY3	N/A	PAUSE
7	5	BACKSPACE	N/A	N/A
8	0	BACKTAB	N/A	BACKTAB
8	1	LEFT	N/A	ALT
8	2	N/A	N/A	N/A
8	3	DOWN	N/A	BACKSLASH
8	4	N/A	N/A	N/A
8	5	TAB	N/A	N/A
9	0	ESCAPE	N/A	WINMENU
9	1	N/A	N/A	N/A
9	2	UP	N/A	USER_DEF3
9	3	N/A	N/A	N/A
9	4	RIGHT	N/A	COMMA
9	5	ALT	N/A	N/A
10	0	B	N/A	ADD
10	1	USER_DEF2	N/A	SUBTRACT
10	2	WINMENU	N/A	EQUAL
10	3	CONTROL	N/A	BACKSPACE
10	4	USER_DEF1	N/A	SPACE
10	5	C	N/A	N/A
11	3	RETURN	N/A	N/A

Figure 6-1: 55-Key Keypad and the first key



CHAPTER 11

LAUNCHING FILES AT STARTUP & BACKING UP FILES AT SUSPEND

INTRODUCTION

Because the FC-2500 does not store files copied into RAM memory permanently, you must store the files on the System CF card and copy them into RAM, when you power on the unit. To accomplish this task, you can use one of two startup files: AUTOEXEC.BAT or FILECOPY.F2C or the ColdBootCopy utility within TTStartup.

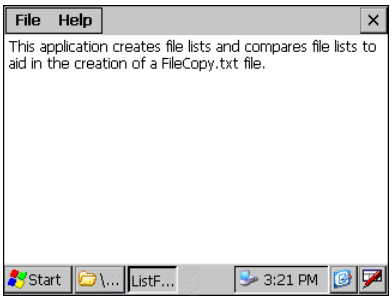
You may also find that you want to backup files when the unit Suspends (or turns off), and restore them upon resume (or turning the device back on). This can be done with the SuspendBackup utility within TTStartup.

First we will cover how to list the different files on the FC-2500, then we will get into how to launch files at startup with AUTOEXEC.BAT or FILECOPY.F2C and finally, we'll cover TTStartup which allows you to copy files at startup and backup files upon suspend.

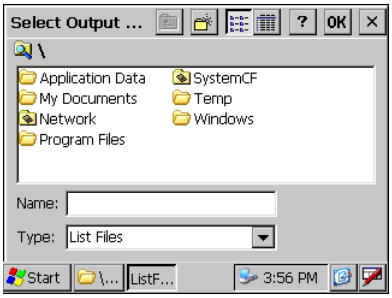
TRACKING SELF-INSTALLED FILES

ListFiles.exe is a utility program that enables you to compare the number of files before and after the installation of self-extracting software on a FC-2500. An output file, which contains the differences, shows the path and names of the added files, enables you to verify the components of the installed software as well as their location. You can also incorporate listed files in the output file with the contents of FileCopy.F2C to launch the installed software at boot up.

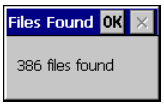
1. On the FC-2500, navigate to the Windows folder and double-tap **ListFiles.exe**. The ListFiles.exe dialog box appears.



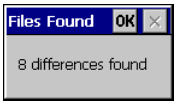
2. On the menu bar, tap **File** and select **New File List**. The Select Output File dialog box appears.



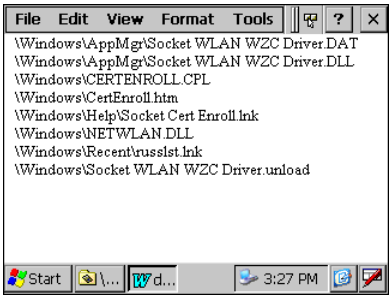
3. Enter the name of the output file (such as before.txt) and tap **OK** (you should copy the file to either System CF or a compact flash card to permanently store the file).ListFiles.exe will then display the total number of files found on the FC-2500.



4. Tap **OK** to close the Files Found dialog.
5. Exit ListFiles.exe.
6. Copy and install the new software on the FC-2500.
7. After successfully installing the software, restart ListFiles.exe and following Steps 2 through 4 create another output file with a different file name (such as after.txt).
8. On the ListFiles menu bar, tap **File** and select **Compare Lists**. The Select Small List dialog box appears.
9. Navigate to the folder that contains the file created in Steps 2 through 4 (i.e., before.txt), select it and tap **OK**. The Select Larger List dialog box appears.
10. Navigate to the folder that contains the file created in Step 7 (i.e., after.txt), select it and tap **OK**. The Select Output File dialog box appears.
11. Enter the name of the output file (such as diff.txt) that will contain the list of differences and tap **OK**. ListFiles.exe will then display the total number of difference found between the first and second files.



12. Tap **OK** to close the Files Found dialog.
13. Exit ListFiles.exe.
14. Navigate to the folder that contains the output file (i.e., diff.txt) and double-tap it to view the contents. For example:



USING FILECOPY

You can create an ASCII text input file to automatically copy files and create folders when booting up the FC-2500. During the boot up process, the FC-2500 looks in the SystemCF folder for the FileCopy.F2C file, and if found, opens the file and then parses and executes its contents.

When the file copy function executes, it creates the FCLog.txt file in the SystemCF folder. This log file will contain any errors encountered during the execution of the FileCopy.F2C file.

FILECOPY COMMANDS

Each line in the FileCopy.F2C file must begin with one of the following command line arguments:

Function	Command	Arguments
Copy File	copy	<\path\source_file> <\path\file_name>
Make Directory	md or mkdir	<\path\directory>
Comments	;	

If a file or directory name includes one or more spaces, the whole path must appear within quotes. For example:

```
copy    \systemcf\helloworld.exe  "\program files\helloworld.exe"
```

NOTES

1. The "F2C" file extension applies to FC-2500s with OS Versions 0.73 and above. Previous versions must still use the "txt" file extension.
2. You cannot copy files from a CF card inserted in the compact flash card slot
3. Do not insert a slash after the directory name when using the md or mkdir commands.
4. FileCopy does not support the use of wildcards

EXAMPLE

In the example below, the first line does not require quotes since neither the source path nor the destination path include a space character. However, the second and third lines do require quotes because the folder name "My App" contains a space character.

```
; install helloworld app

copy    \systemcf\mfcce.dll         \windows\mfcce400d.dll
md      "\My Apps\"
copy    \systemcf\helloworld.exe  "\ My Apps\helloworld.exe"
```

Note: *It is highly recommended that you do not use FileCopy and TTStartup simultaneously.*

USING AUTOEXEC.BAT

AUTOEXEC.BAT is an ASCII text file that you can create to execute DOS commands that can run programs, create folders, transfer files, etc. While FILECOPY.F2C executes during boot up and is seamless, AUTOEXEC.BAT executes in a Command Prompt window immediately after Windows CE .NET loads when the file is stored in the SystemCF folder. For a list of available DOS commands you can use, open the Command Prompt on the FC-2500 and type, "Help."

You can change the way the resulting command window displays by editing the following registry key: as follows:

```
[HKEY_LOCAL_MACHINE\Software\Two Technologies\RunScript]
"HideCmd"=dword:1 ; 0: don't hide CMD window, 1: send the CMD window to the taskbar.
```

USING TTSTARTUP

TTSTARTUP OBJECTIVE

The purpose of this tool is to perform multiple tasks including the:

- Initial loading of specific files at boot-up.
- Backing up of files when the JETT is suspended.
- Restoring of backed up files during subsequent booting of the device.

INITIAL LOADING OF FILES AT COLD BOOT

When the device boots, the stream driver, **TTStartup.dll**, will examine the "**\SystemCF\TTStartup\ColdBootCopy**" folder, and if found, will copy all files contained within that folder into folders with the same name. Existing populated folders will not be affected aside from files being added. New folders will be placed in the root.

Example:

- Source path:
  - **"\SystemCF\TTStartup\ColdBootCopy\Windows\Quality.exe"**,
- Destination path:
  - **"\Windows\Quality.exe"**.

This allows multiple files and folders to get copied / created in a volatile ram-based file system, over-writing files with the same name that may be present there.

The registry key, **`FileCopy`**, when set to zero, disables the file-copy functionality. When set to one (1), enables the file-copy functionality. The default is one (1).

Note: *It is highly recommended that you do not use FileCopy and TTStartup simultaneously.*

SUSPEND BACKUP

When the device is suspended, the text based script file:

"**\SystemCF\TTStartup\SuspendBackup\SuspendBackup.txt**", is parsed to determine which files need to be copied to non-volatile storage. Only the source file's path is listed within the script file. When a suspend occurs, should this file be present, it will be parsed and all files listed will be copied into the folder:

"**\SystemCF\TTStartup\SuspendBackup\nnnn\FileNamePath**", where *nnnn* represents a number indicating the n-th backup, a value that increments by one each time the unit suspends. This allows multiple copies of data to be preserved, such that if an error occurs, causing an erroneous backup of a data file, previous versions of the file may exist to provide backup.

The quantity of backup sessions to preserve is customer selectable, determined by the registry key **`BackupQty`**. Should this registry key be absent, a default value of three (3) will be used. This initial implementation will force the total quantity of backup up sessions to be less than or equal to the quantity **`BackupQty`**, such that the oldest backup session will be deleted before a new one is created if the **`BackupQty`** has been reached.

The registry key, **`Backup`**, when set to zero, disables the suspend-backup functionality. When set to one (1), enables the suspend-backup functionality. The default is zero (0).

The TTStartup.dll will use a very high launch order value. This will cause the driver to load last, and as such, it will be the first to be called during the shut-down process. This allows all the other drivers to be available during the suspend process. GWES may have already been shut down, so no graphical APIs may be used in this context.

As the SuspendBackup.txt file is parsed, a log file will be generated to record any errors as well as status information concerning the backup session. Counter values will be recorded and stored into the file, providing session restore information to the restore selection dialog.

The name of the log file will be:

"**\SystemCF\TTStartup\SuspendBackup\9999\Logfile.txt**".

POSSIBLE ISSUES USING THE SUSPEND BACKUP METHOD

- A data file the user desires to have backed up was opened without file-read-share permissions, preventing any other program from opening or copying the file.
- When the application uses CreateFile() to open the data file, the **`dwShareMode`** parameter needs to include the **`FILE_SHARE_READ`** flag.
- The application fails to perform a file flush following each update of data. Thus, the current data hasn't been written to the data file yet, but is still in a file buffer. (The TTStartup driver will implement a fix for this problem by requesting a file system flush before it begins to copy data files.)

RESTORE OF BACKUP DATA

When the unit boots up (warm or cold), the SuspendBackup.txt file is checked to determine if any restore action is needed. Should the file exist, and one or more session restore directories are present, a dialog will be presented to the user asking if they wish to restore the latest (newest) set of data files, some other set of session data files (should more than one set be present), or skip the session restore process altogether. If more than one set is present, the user will be given a choice of which set to restore.

The registry key, **`Restore`**, when set to zero, disables the session restore functionality. When set to one (1), enables the session restore functionality.

The default is zero (0).

SUSPEND NOTIFICATION

When the TTStartup.dll driver receives notification from the power manager (PM), that the device is being suspended, it will notify any drivers or applications waiting on a named event, that a suspend action is currently in process. Should a program wish to perform some minimal functionality at this time, a "suspend-event" thread should wait on this named event, such that when this event is signaled, the thread will be allowed to proceed and perform the actions desired. This method should be used very cautiously, since it introduces latency to the suspend process. The event name is: **`EVENT_TTStartup_Suspend`**

NOTES

You will need to configure Session Backups and Session Restore for the program to work. To do this:

- Click **Start>Settings>Control Panel>Startup**
- Then select Enable Session Backups and Session Restore
- You may then select the amount of sessions to backup, the default option is 3

You need to create the folder structure and .txt file for TTStartup. The tree should be placed directly inside the **\SystemCF** folder.

EXAMPLE

- TTStartup
    - ColdBootCopy
    - SuspendBackup
        - SuspendBackup.txt

You then need to add the paths to each of the files you want to backup within the suspendbackup.txt file

EXAMPLE OF SUSPENDBACKUP.TXT:

List of files to backup to **\SystemCF\TTStartup\SuspendBackup** directory:

- **\My Documents\My Data\DataFile.bin**
- **\My Documents\My Data\OrdersFilled.pdq**

RULES FOR SUSPENDBACKUP.TXT SCRIPT FILES:

- 8-bit ASCII only (escaped ASCII may be used for UNICODE, supporting languages, such as Chinese)
- File names will be parsed from the first column (column 1) up to either a carriage return (0x0D) or line feed character (0x0A) is found.
- Comment lines may be included and will begin with a semi-colon character ';' in column 1.
- When the unit is resumed, no restore action is taken.
- Since the display of any dialog within a windows context requires the majority of system components to be loaded and running, the restore process with the display of this dialog will be performed near the end of the boot process.
- Two Technologies suggests that great care and thought be used in determining what files should be backed up with this utility. Every file takes a finite amount of time to write, and when a unit is in the process of suspending, there may only be a small window of time in which to write the files If the total size of the files becomes excessively large, it is possible to create conditions similar to pulling the battery while the unit is running or shutting down, possibly corrupting a file system.

CHAPTER 12

PROGRAMMING THE FC-2500

WARM BOOTING FROM AN APPLICATION

We recommend achieving a warm boot by invoking the warmboot.exe utility.

The method that you use depends upon which development environment that you are using. Here are samples of code written in C#.

For Microsoft® Visual Studio® 2005 and .NET 2.0:

```
System.Diagnostics.Process.Start( "WARMBOOT.EXE" , "YES" );
```

For Microsoft® Visual Studio® 2003 and .NET 1.1:

Note: *If you are using Visual Studio® 2003 and .NET 1.1 then you need a reference to TwoTechLibraryCE.dll.*

```
TwoTechLibraryCE.Utility.Functions TTFunc = new TwoTechLibraryCE.Utility.Functions();
TTFunc.OpenExternalProgram( "WARMBOOT.EXE", "YES" );
```

For more information on working with applications, see our extranet support page at: http://www.2t.com. You will need a Gem Partner login to gain access to this information.

CHAPTER 13

TROUBLESHOOTING & FAQ'S

BASIC FAQS

MY FC-2500 DOES NOT RESPOND WHEN I PRESS THE POWER BUTTON.

- Is the unit in Suspend mode?
- If battery-powered, check the batteries.
- Are all cables connected properly:
  - Is the power supply plugged into an active AC outlet?
  - Is the power connector securely plugged into the FC-2500?

I CHANGED MY SYSTEM SETTINGS, BUT WHEN I TURN ON THE FC-2500, MY SETTINGS ARE GONE.

- You must save the registry after making any system or configuration changes.

I TRANSFERRED FILES TO THE FC-2500 FROM MY HOST COMPUTER, BUT WHEN I TURN ON THE FC-2500 MY TRANSFERRED FILES ARE MISSING.

- To store transferred files permanently, you must file copy the files into internal flash memory or to USB flash drive.

  Occasionally, transferred files can be hidden from view, double-tap My Computer, select Options from the View menu and clear all boxes.

I CANNOT CONNECT TO THE DEVELOPMENT SYSTEM USING ACTIVESYNC.

- Did you install ActiveSync using the Administrator account?
- Check the cable connections.
- Check the serial communications configuration.
- Make sure the correct COM port is available.
- In ActiveSync, check the Connection Settings for the connection type you are using (USB, Serial or Ethernet).

THE SCREEN IS TOO LIGHT OR TOO DARK.

- Adjust the brightness via the brightness control in the Control Panel.

THE STYLUS IS NOT RESPONDING PROPERLY.

- The screen is not calibrated correctly to interpret the screen taps. You need to recalibrate the screen.

THE FC-2500 ACTS SLOWLY.

- The unit may be short of program memory or storage memory.
- Increase the amount of storage or program memory through the System control in the Control Panel.
- You can also delete any unnecessary files.

I GET LITTLE OR NO SOUND FROM THE FC-2500.

- Adjust the volume and sound properties via the Volume and Sound control in the Control Panel.

THE FC-2500 GOES INTO AUTO-SUSPEND AFTER A SHORT PERIOD OF INACTIVITY.

- As a default, the device will auto-suspend after five minutes of inactivity while running on batteries. Configuration is necessary in the Control panel applet named "Power" if you want to have the unit auto-suspend when running on AC power.
- Adjust the power management properties via the Power control in the Control Panel.

NO SOUND IS HEARD WHEN YOU TAP THE TOUCH SCREEN OR PRESS A KEY.

- Volume setting is low or turned off.
- Check the volume slider in the Volume & Sound properties dialog box in the Control Panel.

WHY CAN'T MY FC-2500 ACCESS MY "KNOWN GOOD" USB FLASH DRIVE?

- It is possible that you are not using a proper OTG USB cable to connect your USB Flash Drive to the FC-2500. An easy way to verify this is by examining the end of the cable that connects to the FC-2500 and looking for the letter "A". If the A is present, then OTG should work, and you should see the OTG HOST icon in the System Tray on the FC-2500. If it does not have an "A" then it may not be an OTG ready cable. Another surefire way to tell is to look at the plastic inside the "A" plug; it should be white if it is an OTG cable. What makes the OTG USB cable different from older USB cables is the appearance of a sensing or "ID" pin which, in the A connector, is shorted to ground. In the "B" connector, this pin is floating. The omission of this wire is what makes older USB cables non-OTG compliant.
- Although unlikely, it is also possible that the USB Flash drive is not OTG compliant, or will not function in an OTG based system. However, most Flash Drives do work with the FC-2500.

INTEGRATOR FAQS

This section provides answers to some frequently asked questions when integrating eye•WARE into your application.

WHY DO I GET "\WINDOWS\RECENT\" AS THE DEFAULT PHOTOGRAPH FILE PATH VALUE INSTEAD OF \MY DOCUMENTS\?

In eye•WARE versions 1.2.6.7 and above, the registry value to store photograph file path was changed from "FILEPATH" to "DESTPATH." but a deficiency exists where the TwoTechEyeWARELibraryCE.DLL (1.0.6) does not update accordingly. Two Technologies will address this issue in a future release.

If you directly get the file path from the registry, use "DESTPATH" instead of "FILEPATH" to retrieve photograph file path value.

If you are using functions from TwoTechEyeWARELibraryCE.DLL (1.0.6) to retrieve or set the file path, please contact Technical Support.

WHY CAN'T I ACCESS EYE•WARE REGISTRY SETTINGS WHEN CREATING MY OWN EYE•WARE CONFIGURATION FILE AND SETTING FILE NAME INFORMATION?

You must add the eye•WARE registry settings programmatically or run eye•SEE setup once to have eye•WARE create the registry settings prior to running eye•WARE.

WHY CAN'T I START EYE•WARE THROUGH MY APPLICATION AFTER I UPGRADE THE OPERATING SYSTEM,

You must add the eye•WARE registry settings programmatically or run eye•SEE setup once to have eye•WARE create the registry settings prior to running eye•WARE.

WHY DOES EYE•WARE GENERATE A PROGRAM EXCEPTION WHEN I CHOOSE "DEFAULT NAME ONLY" OR "SENT FROM EYE•WARE API" AS MY "FILE INCREMENT" SETTING AND USE THE SAME FILE NAME REPEATEDLY?

A deficiency exists in eye•WARE that prevents using the same file name repeatedly. Currently, you must use a different file name each time you save a photograph. Two Technologies will address this issue in a future release.

WHY DON'T SAVED PHOTOGRAPHS APPEAR IN THE FILE PATH I SPECIFIED?

When you specify the default save path, make sure you use following format: "\SystemCF\My Pics\" or "\temp\."

APPENDIX A

SPECIFICATIONS

POWER

- Recharge/Line-Power: Topcon AD-11 12V AC/DC converter/power supply
- Rechargeable Battery Type: Topcon BT-66Q Lithium-ion
- Current Rating: 1.2 Amp Maximum
- Voltage: 8.4 Volts
- Capacity: 2500 mAh

DISPLAY

- Color, outdoor readable, transflective, active matrix, liquid crystal display with touch screen
- Landscape orientation
- Resolution: QVGA (320 x 240 pixels)

CPU

- Type: Marvel® PXA270 processor
- Speed: up to 624 MHz (312MHz option is software selectable)
- Operating System: Windows CE 5.0

MEMORY AND MASS STORAGE

- SDRAM: 256 MB
- Internal Compact Flash: 2 GB standard (30 MB reserved for OS)

USER INPUT

- Touch Screen
- Key Pad: 55-key (8-way joystick style) polycarbonate keypad
- Feedback: Tactile and audible

INDICATORS

- 5 Modifier Key/Programmable LEDs
- Charge/Low Battery Indicator (battery-powered units only)

Camera Image System

- Color Image Sensor: 5.17 megapixels (4 megapixels processed)
- Image size in pixels: QSXGA - 2592 x 1944, SXGA – 1280 x 960, VGA – 640 x 480 and HF – 320 x 240
- Resolutions Available: 2592 x 1944 (QSXGA), 1280 x 960 (SXGA), 640 x 480 (VGA) or 320 x 200 (HF)
- Lens: High Quality IR Coated Glass Optics (polarizing lens optional)
- Shutter Speed: Up to 1/500 of a second
- Focal Length: Macro 2.25" to infinity
- Aperture: Automatic, F 2.8 or F 8.0 (software selectable)
- True-Light LED Lighting System: Four white LEDs each with a typical luminous flux of 100 lm.
- Laser Focus System: Two Class IIIa type lasers that operate at 635nm at a radiated power each of less than 3mW.

Lens

- Material: High quality IR coated glass optics
- Shutter speed: Up to 1/500 of a second
- Focus range: Macro 2.25 to infinity
- Effective Focal length: 8.22 mm (equivalent 35mm focal length is 41mm)
- Aperture settings: automatic, f/2.8 and f/8

Auto Focus

- Method: Laser, hyperfocal, manual and ROI (region of interest)
- Focus range: Macro 2.25 to infinity

Connectivity

- DE-9 serial port (RS-232) with auxiliary power out
- USB (mini USB A/B Connector) OTG Host/Client compatible
- Charge Jack
- Bluetooth (Class 2)
- Wireless LAN (802.11b/g)

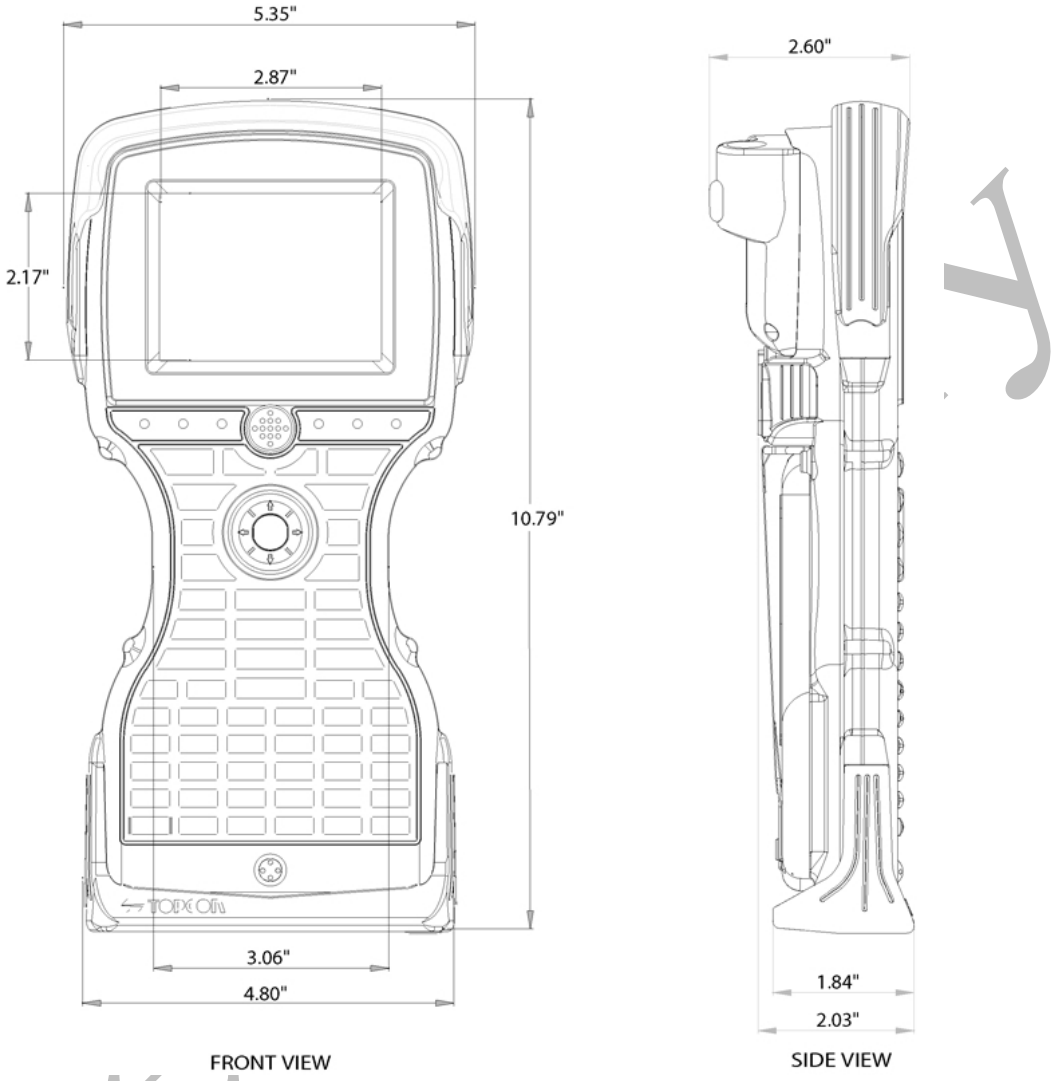
Environmental

- Charging Temperature: 0°C to + 40°C
- Humidity: 5 to 95% Non-condensing
- IP Rating: 67. The unit is tested to EN60529:1991 IP6x Method 13.5 for dust ingress and to EN60529:1991 IPx7Mehtod 14.2.7 for water ingress
- Shock Rating: This product is tested to MIL-STD-810F: January 1st, 2000 Method 516.5 Procedure IV for shock. Test is performed with a drop tester from 122cm (48 inches) onto all faces, edges, and corners.

Physical Dimensions

- Height (H): 10.79 Inches (274 mm)
- Width (W): 4.8 Inches (122 mm)
- Depth (D): 2.6 Inches (66 mm)

Figure A-1: Case Dimensions



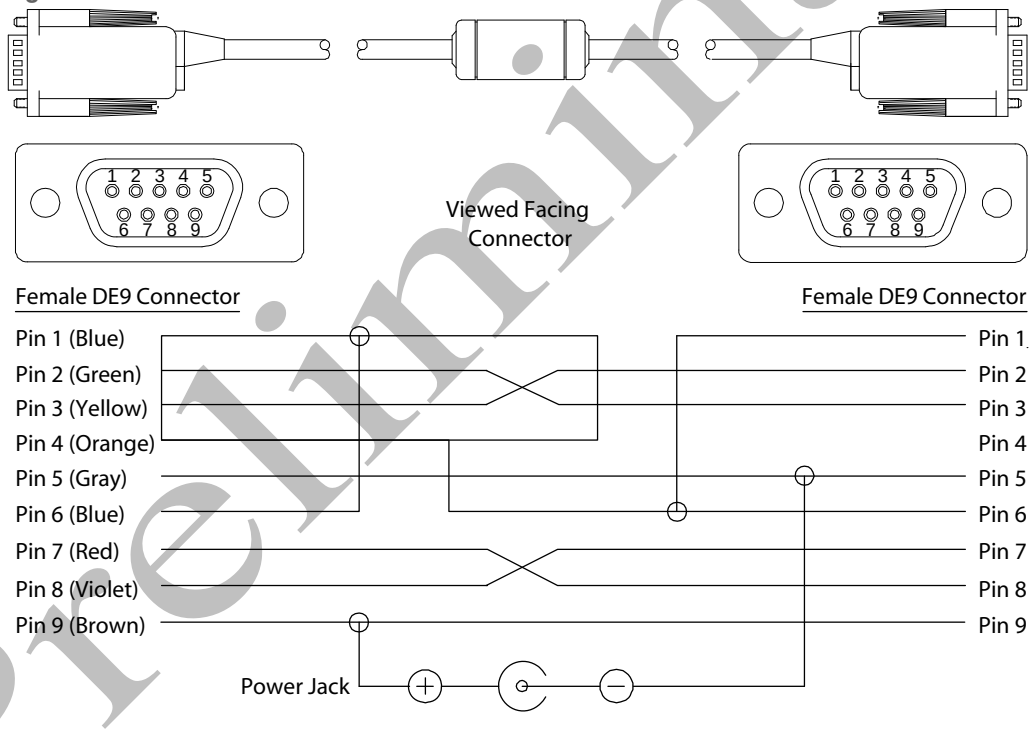
APPENDIX B

SIGNAL AND PIN ASSIGNMENTS

NULL MODEM CABLE

Figure B-5 lists the signal and pin assignments for Two Technologies' 14375 DE-9 Female to DE-9 Female null modem cable. Please note that this cable differs from standard null modem cables in that it uses Pin 9 for input power for the FC-2500.

Figure B-2: DE-9 Female to DE-9 Female Null Modem Cable



INDEX

M

N

O

P

R

S

T

U

W