



Embedded Computer Systems

Developer's Getting Started Guide

for



ADS document # 411010-1104B

Applied Data Systems

9140A Guilford Road
Columbia MD 21046
301-490-4007

www.applieddata.net

© 2002 ADS

(This page intentionally blank)

Revision History

LTR	DESCRIPTION	DATE	BY
-	Create Date	20-Aug-02	ctacke
1	First Release	9-Sept-02	ctacke
2	Added section for backing up REG/HWT	13-Jan-03	ctacke
A	Added graphical ActiveSync connection steps. Updated boot switch settings. Made other corrections and clarifications. Reorganized sections	24-Jun-03	ctacke
B	Updated BitsyX Boot switch settings, added switch clarifications. Added notes for NK.BIN v. NKLOAD.BIN	10-Sep-03	ctacke

(This page intentionally blank)

Table of Contents

Table of Contents	iii
1 Getting Started	1
1.1 Connectivity Software.....	1
1.2 Development Tools	2
1.3 Development System Contents	3
1.3.1 Hardware	3
1.3.2 Cables	3
1.3.3 Software.....	3
1.4 Booting the System	4
2 The Development Environment	5
2.1 The Debug Port	5
2.2 Connecting to the Debug Port	5
2.3 Deciphering the Debug Output	5
2.3.1 Boot Cause.....	6
2.3.2 Switch Positions	6
2.3.3 Boot ROM Version.....	6
2.3.4 Flash Detection.....	6
2.3.5 Display Information.....	6
2.3.6 CE Start Location	6
2.3.7 ADSmartIO Version Information	6
2.3.8 Windows CE Version	6
2.3.9 ADS Build Number	7
2.4 Connecting With Microsoft ActiveSync	7
2.4.1 RS232 Serial and USB Connections.....	7
2.4.1.1 Graphical Example of making an RS232 Serial Connection	8
2.4.2 Ethernet	17
2.5 Persisting ActiveSync Partnerships.....	17
2.6 Debugging with eMbedded Visual Tools or Visual Studio.NET 2003	17
2.7 Device-side TCP/IP Settings.....	17
3 Customizing Device Behavior and Settings.....	18
3.1 Changing Boot Behavior.....	18
3.1.1 Boot Switch Settings	19
3.2 The Debug Menu	20

3.2.1	Erasing Blocks of Flash Memory	21
3.2.2	Backing up ADSLOAD.REG and ADSLOAD.HWT	21
3.3	ADS System Files	23
3.3.1	Updating the Files.....	23
3.3.2	NKLOAD.BIN / NK.BIN.....	23
3.3.3	ADSLOAD.HWT	23
3.3.4	ADSLOAD.REG	24
3.3.4.1	Altering ADSLOAD.REG for a new CE Image.....	24
3.3.4.2	Persisting Device TCP/IP Settings	25
3.3.4.3	ObjectStorePages: Memory Allocation	25
3.3.4.4	ActiveSync Settings.....	26
3.3.5	ADSLOAD.EXE	26
3.3.6	ADSLOAD.BMP.....	26
3.4	Changing the Boot Code	27
3.5	Changing the ADSmartIO Firmware	27
3.6	The ADS Persistent Registry	28
3.7	Persistent Storage.....	29
4	What Next?	30
4.1	ADS Windows CE Developer's Guide	30
4.2	ADS On the Web	30

1 Getting Started

This guide is designed to get the Windows CE developer going using an ADS development system. The content is divided into four “levels”: Section 1 covers the development tools you’ll need, a device inventory and how to boot the system. Section 2 discusses several topics to familiarize you with the development system and the general system architecture. Section 3 goes into more detail providing information on how to modify and customize your CE device and finally Section 4 provides information on additional resources that will help you get the most out of your system.

1.1 Connectivity Software

Before beginning you should download and install the following software on your development PC:

- [Microsoft ActiveSync version 3.7](#)

ActiveSync provides a TCP/IP based transport layer that allows your development PC to communicate with your Windows CE device. ActiveSync can be used to connect to the device through an RS232 serial, a USB or an Ethernet connection. In the first two cases, ActiveSync will “assign” an IP address to both the device (192.168.55.101) and the development PC (192.168.55.100).

- Terminal Software ([HyperTerminal](#))

All ADS products have the ability to send information off-board through a serial port known as the “debug port” (see Section 2.2). To view this information you will need some form of terminal software.

1.2 Development Tools

The development tools you will need to install on your development PC depend on two factors: the version of CE you are using and the development language you intend to use. The table below outlines the tools you will need.

Development Language	OS Version				Development Tools				
	3.0	4.0	4.1	4.2	eMbedded Visual Tools 3.0	eMbedded Visual C++ 4.0	Visual Studio 2003 and Add-on Pack	HPC 2000 SDK	ADS SDK
eVB	✓				✓				
C/C++ †	✓	✓	✓	✓	✓	✓		✓	✓
VB.NET			✓	✓			✓		
C#			✓	✓			✓		

† For C/C++ development with CE 3.0, you must use eMbedded Visual Tools 3.0 and the HPC 2000 SDK. With CE 4.x you must use eMbedded Visual C++ 4.0 and an ADS SDK. Additionally, CE 4.1 requires eVC SP1 and CE 4.2 requires eVC SP2.

- [Microsoft eMbedded Visual Tools 3.0](#)

Microsoft provides development tools for Windows CE for free. For CE 3.0, you can use either eMbedded Visual Basic (eVB) or eMbedded Visual C++ (eVC) which are part of the eMbedded Visual Tools (eVT) and can be downloaded directly from Microsoft's web site.

- [HPC 2000 SDK](#)

eVT is shipped with three SDKs – the Pocket PC, The HPC Pro and the Palm-Sized PC. None of these are compatible with ADS development systems running CE 3.0 or later. To develop for ADS development systems with CE 3.0 you will need to install the HPC 2000 SDK. It is available as a free download from the Microsoft web site.

- [Microsoft eMbedded Visual C++ 4.0](#)

If you are using a development system with CE 4.x on it you will need to download and install eMbedded Visual C++ 4.0 as well as the latest service packs. Neither eVB 3.0 nor eVC 3.0 can be used with CE 4.x, nor is eVC 4.0 backward compatible with CE 3.0 devices.

- [Microsoft Visual Studio 2003](#)

If you are using CE 4.1 or later, you have the option to use managed code in C# or VB.NET. These are available as part of Visual Studio 2003. The device-side runtimes are known as the .NET Compact Framework or .NETcf.

- [Microsoft Visual Studio 2003 Add-on Pack](#)

As-shipped by Microsoft Visual Studio 2003 cannot connect to a CE 4.x device for debugging support. You must install the Visual Studio 2003 Add-on pack to get debugging support.

- [Applied Data Systems SDK](#)

To be able to debug applications on CE 4.1 and later ADS systems using eMbedded Visual C++, you must also install an ADS Software Development Kit (SDK). Different platforms require a different SDK, but all are available for download from the link above.

1.3 **Development System Contents**

ADS offers application ready development systems that include the following:

1.3.1 **Hardware**

- Single-board computer with on-board SDRAM and Flash – typically 32MB of each.
- Color flat-panel display with backlight inverter
- Analog-resistive touch panel
- 5/12V power supply
- CompactFlash™ ATA card with PCMCIA Adapter

1.3.2 **Cables**

Each ADS development system contains the following cables:

- LCD Display Cable
- Touch panel cable
- Backlight cable
- Power adapter
- DB9 F/F null modem cable
- Adapter cable(s) for serial port connections

1.3.3 **Software**

The following items are pre-installed on your development system:

- Windows CE Image (CE 3.0 or CE.NET)
- REG and HWT configuration files for display and device RAM/flash configuration
- ADSLOAD.EXE (dummy/filler app)
- ADSLOAD.BMP ADS logo for CE desktop

These files are described in greater detail in Section 3.3.

1.4 ***Booting the System***

To boot your ADS development system, simply plug the included power supply into the power connector. The backlight and display should power on and any LEDs on the system board will illuminate.

Shortly after power is applied, the ADS logo will appear on the display followed by the Windows CE desktop.

Note

New systems will boot to a touchscreen calibration window before displaying the Windows CE desktop. Calibration data is then saved to the ADSLOAD.REG configuration file (detailed in Section 3.3). On subsequent startups the calibration screen will not appear.

You can track the progress of the boot sequence, plus get information such as the build version and display settings by attaching the device's debug port to your development PC through a terminal program such as Hyperterminal. See section 2.1 for more details on the debug port.

2 The Development Environment

Once you have inventoried and booted your ADS CE device, the next step is to set up your development environment. This section outlines setting up your environment..

2.1 The Debug Port

By default ADS Windows CE builds use serial port 3 as the debug port. Your applications can display messages about the state of the operating system or application on this port using the RETAILMSG or DEBUGMSG macros. The ADS boot loader displays its progress and status on this port at boot time.

2.2 Connecting to the Debug Port

To connect to your development system's debug port, follow these steps:

1. Connect a null modem cable from an open serial port on your development PC to your development system's serial port 3.
2. Open a terminal program on your PC with direct communication with the serial port. Use the setting 38400 baud, no parity, 8 data bits, 1 stop bit (38400,8,N,1) with no handshaking.
3. To test, press the reset on your development system with the boot switches set for Normal Boot. You should get readable output in your terminal program.

2.3 Deciphering the Debug Output

The debug output during the boot sequence provides a lot of information about the system configuration. This section describes what each piece means.

The following is an example of the debug output of a successful boot of a Bitsy. Line numbers have been added here for reference and do not appear in an actual debug output.

```

1 Processor Initialization Table read.
2 Waiting for memory clocks to stabilize.
3 Memory initialization complete.
4 Determining boot cause.
5 Reset detected.
6
7 ** Bitsy Loader initializing(0x3). **
8
9 ADS Bitsy Rom Ver. 1.75 (32 MB DRAM)
10 (Build at 12:56:12 on Jun 3 2002)
11
12 Flash Memory Detected: Intel StrataFlash (32MB)
13
14 Reading Flash Header Size...(0xd50 bytes)
15 Reading Flash ...(from 0x40004 to 0x40d54)
16 Reading hardware register values from flash.
17
18 This is Color, 8 Bit, Active Panel.(Width = 640, Height = 480)
19 8 BPP Color.
20
21 Booting from Flash.
22 Reading Flash Header Size...(0xd85a70 bytes)
23 Reading Flash ...(from 0x140004 to 0xec5a74)
24
25 CE starts From DRAM(0xc1200000)
26 Ethernet Debug Services disabled.
27 Jumping to 0xc1200000.
28 Windows CE Kernel for ARM Built on May 31 2002 at 14:35:27
29 ProcessorType=0b11 Revision=8

```

```
30 sp_abt=ffff5000 sp_irq=ffff2800 sp_undef=ffffc800
31 Windows CE Image was built at 16:54:25 on Aug 16 2002.
32 Sp=ffffc7d0
33 SMARTIO Detected : PartType=0x8535 Firmware=0x1021 Revision=9
34 UCB1200 Detected : ID = 0x1004
35
36
37 Microsoft Windows CE Ver. 4.10 (build 908)
38
39 ADS BITSY Windows CE Ver.(3.99)
```

2.3.1 Boot Cause

Lines 4 and 5 display the cause of the boot. Typical values will be either “Reset Detected” or “Wakeup Detected”.

2.3.2 Switch Positions

Line 7 displays a hexadecimal value in parentheses (in this sample 0x3). This value is the binary representation of the positions of the boot switches. In this case it says that both switches are in the OFF position.

Some platforms report their switch settings with inverse logic (switches in the ON position are reported as zero bits, OFF switches are reported as one bits) so a BitsyX with both switches in the OFF position will report a switch setting of 0x3. However, a Graphics Master reporting a switch setting of 0xD means that switches 1, 3, and 4 are ON.

2.3.3 Boot ROM Version

Lines 9 and 10 display the build date, version and the targeted RAM size for the installed boot loader code.

2.3.4 Flash Detection

Line 12 displays the type and amount of flash memory on the board.

2.3.5 Display Information

Lines 18 and 19 display information about the installed display driver including color depth, resolution and whether the panel is active or passive.

2.3.6 CE Start Location

Lines 25 and 27 give the starting address for Windows CE. This is set in ADSLOAD.HWT.

2.3.7 ADSmartIO Version Information

Lines 33 and 34 provide information about what revision of the ADSmartIO firmware is installed on the device.

2.3.8 Windows CE Version

Line 37 displays the version of Windows CE Kernel on which the image was built (i.e. 3.0, 4.0, 4.1, 4.2, etc.).

2.3.9 ADS Build Number

Line 39 displays the ADS-specific build number of the image loaded on the device.

2.4 **Connecting With Microsoft ActiveSync**

Microsoft ActiveSync is used to communicate between the ADS CE device and your development PC. ActiveSync allows you to debug applications directly on the device with eMbedded Visual Tools, browse the device's file system, browse and edit the device's registry, plus several other features.

ActiveSync consists of, among other libraries, a PC application (wcesconn.exe) and a device application (repllog.exe) that communicate with one another. This section will discuss how to make them connect and communicate.

See the ActiveSync documentation and help files for more information on ActiveSync's capabilities including a full set of PC-side APIs for programmatically communicating with the CE device.

2.4.1 RS232 Serial and USB Connections

Making a serial connection with ActiveSync is straightforward.

Note

Serial ActiveSync requires the DTR line be connected on the connecting device. Some ADS products do not have DSR/DTR lines on all COM ports (for example the BitsyPlus does not have it on COM3).

You can usually overcome this and get an ActiveSync connection by simply looping the PC's DTR line back to its DCD and DSR lines.

You can also use a serial cable with built-in loopbacks (the cable drawing is available at [Topic 612](#)) to achieve this.

First, make sure that your Windows CE device is set for communication using the correct connection (serial, USB or Ethernet). This is done through the Control Panel using the *Communications* applet in CE 3.0 or the *PC Connection* applet in CE.NET (see Section 2.4.1.1).

Next ensure that ActiveSync on the development PC allows the desired connection by selecting the *File | Connection Settings* menu item.

When you have the PC and the CE device configured to connect using your desired port and protocol, connect the two with a cable (if necessary). For RS232 you will need a null modem cable.

Connecting the cable *may* automatically launch ActiveSync on the PC and put it into a listen state. This is signified by the ActiveSync "wheel" icon turning green and spinning.

If ActiveSync on the PC does not start listening automatically, you must manually start it. This is done by selecting *File | Get Connected....*, from the ActiveSync menu and then clicking the *Next* button.

Once ActiveSync on the PC is listening, repllog.exe on the device must contact it. This process *may* start automatically, depending on the device you are using, and you will see a dialog box pop up on the device that says "Connecting to [selected port name]."

If the connection manager on the device does not start automatically, you must launch it manually. This can be done by navigating to the Windows folder on the device and double-tapping repllog.exe. Alternatively you can select repllog.exe from the *Start | Run* dialog.

Once the connection has been established, ActiveSync will prompt you and ask if you wish to establish a partnership with the device. Establishing a partnership is not necessary unless you want to synchronize data between the PC and the device (see the ActiveSync documentation for more information) or if you later want to make an ActiveSync connection over Ethernet.

You can suppress the partnership dialog, forcing ActiveSync to always connect as Guest by adding the following DWORD registry key to your development PC:

```
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows CE Services  
"GuestOnly"=dword:1
```

Set the key's value to 1 to always connect as Guest. Setting the key's value to 0 or removing the key will return ActiveSync to the state where it prompts for a partnership upon connection.

If you have further difficulties connecting to your device, more information can be found online at <http://www.applieddata.net/support/>.

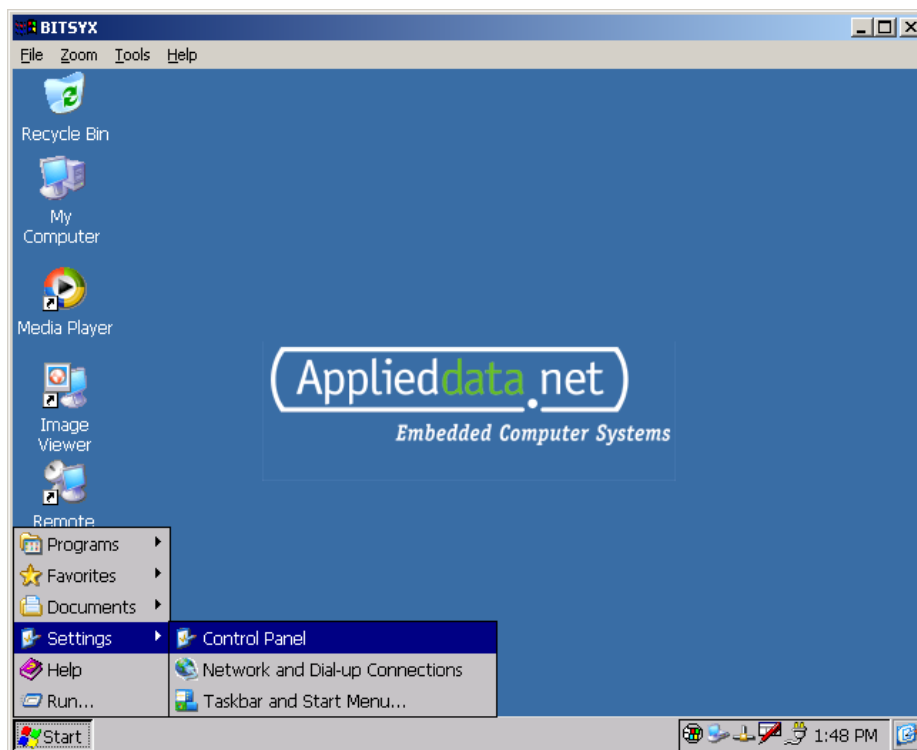
2.4.1.1 Graphical Example of making an RS232 Serial Connection

The following example illustrates how to create an ActiveSync connection with an ADS CE.NET device using an RS232 serial connection.

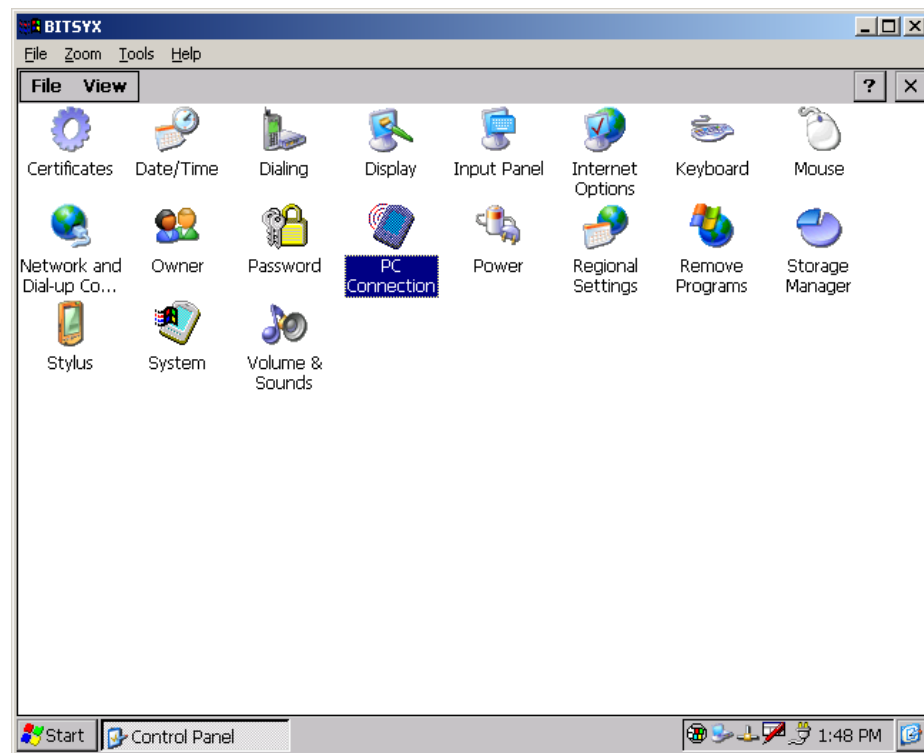
Step1: Selecting a connection

From the device's PC Connection Control Panel Applet, choose the connection you wish to use for ActiveSync.

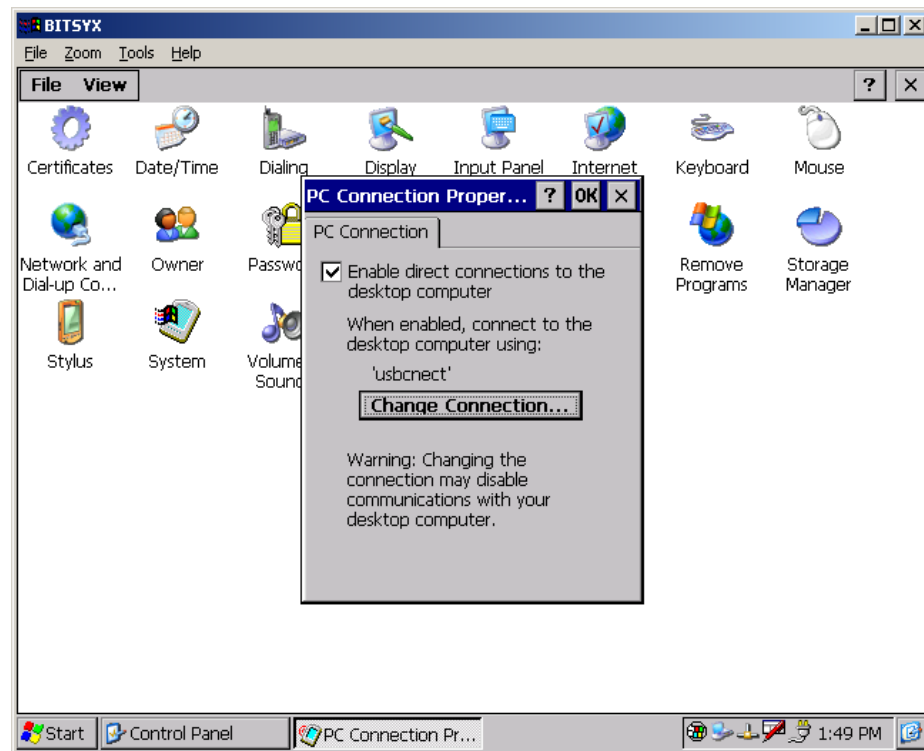
First select *Control Panel* from Settings on the Start Menu.



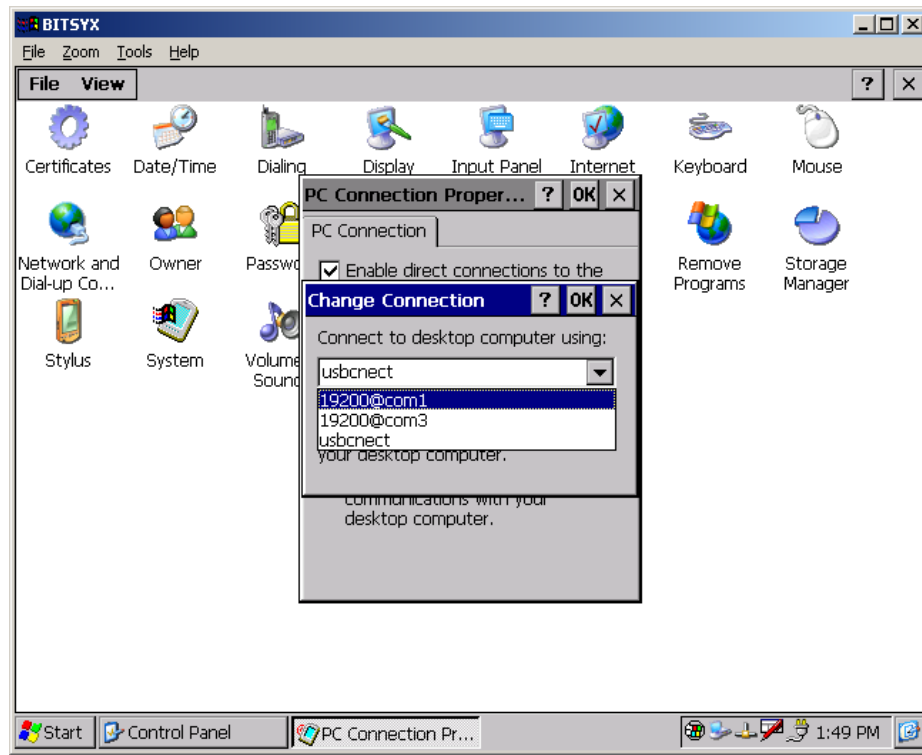
Next select and run the *PC Connection* applet.



From the applet, tap the *Change Connection* button.

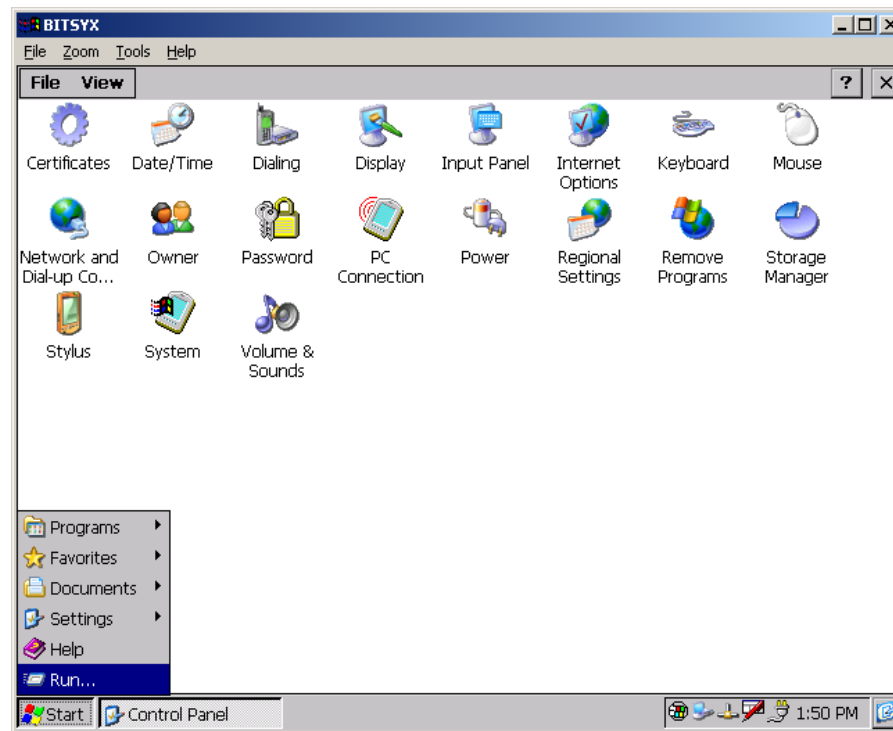


Now select the specific port you wish to use for the connection. In this example we're using COM1.

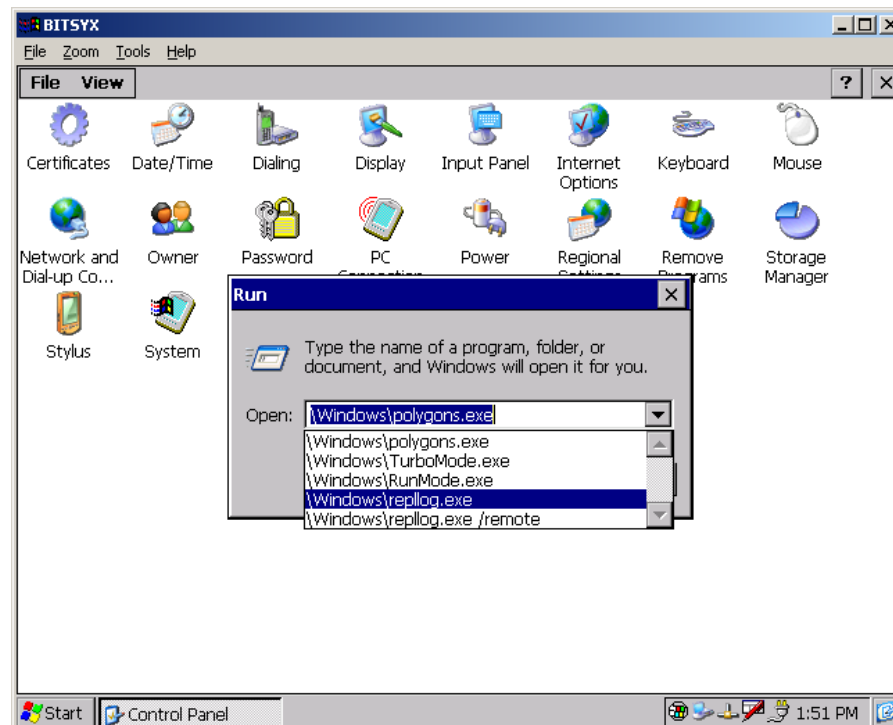


Step 2 – Running the ActiveSync device client: repllog.exe

Once a connection is selected, run the device's ActiveSync Client. From the *Start* menu, select *Run*.



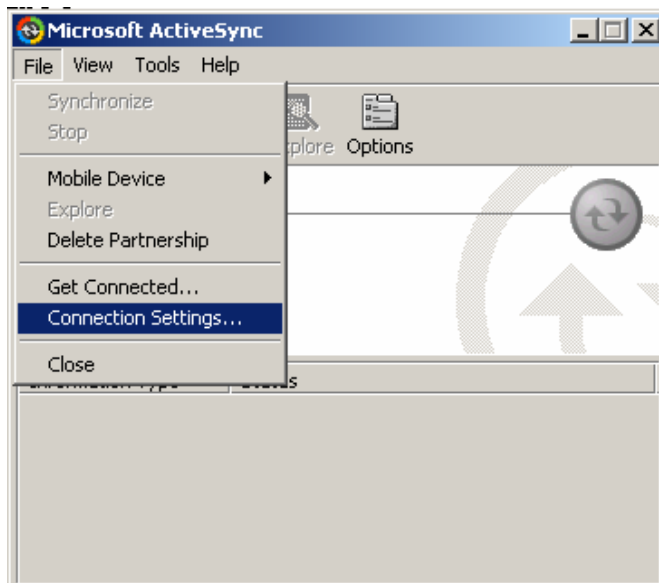
The device will display the *Run* dialog. From the dropdown select “repllog.exe” or type it in manually.



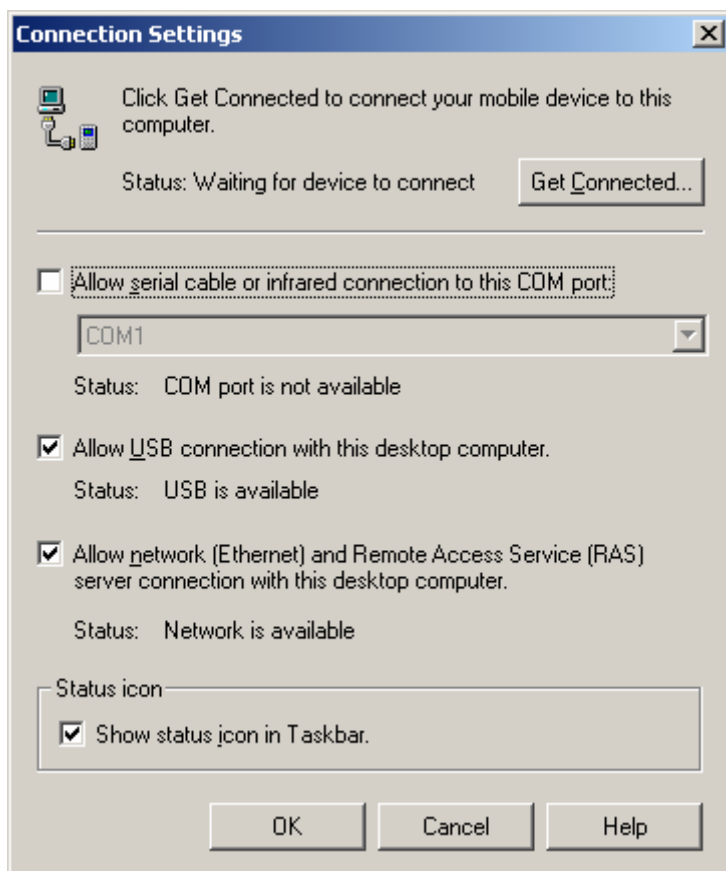
Before tapping the *OK* button on the device move to the next Step.

Step 3 – Preparing ActiveSync on the PC

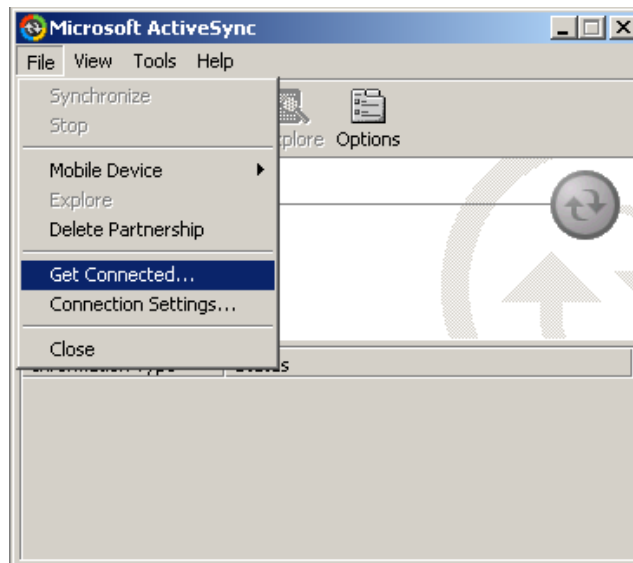
On your development PC, open ActiveSync and from the menu select *File | Connection Settings....*



Ensure that the “Allow Serial cable...” checkbox is selected and that the PC’s COM port you wish to use for this connection is selected in the drop-down box.



On your development PC, open ActiveSync and from the menu select *File | Get Connected....*



This will display a connection Wizard.

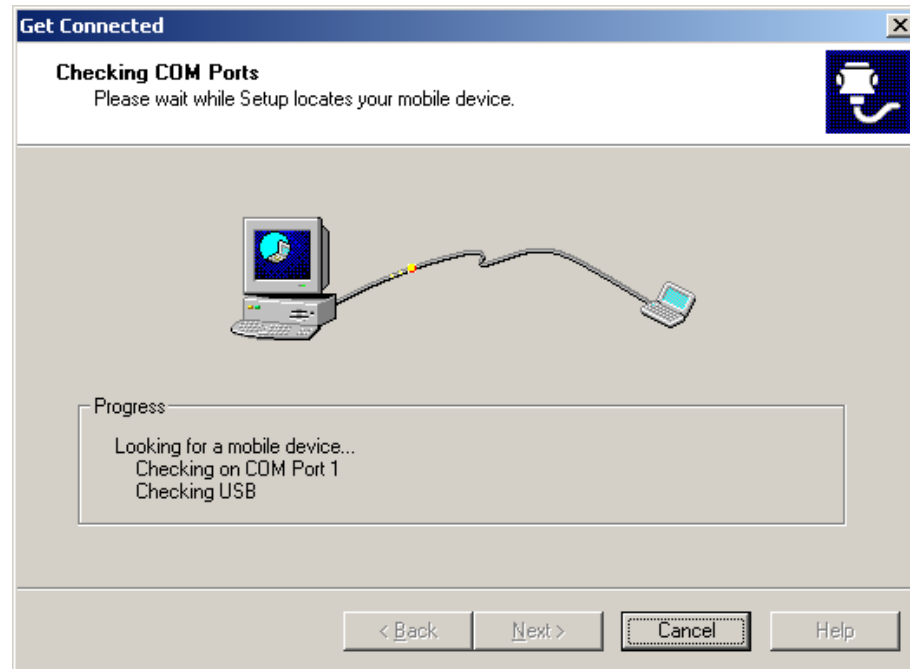


Before clicking *Next* move to the next step.

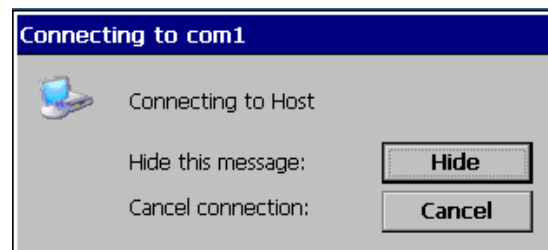
Step 4 – Making the Connection

Now that both the device and PC are ready to connect, it is time to begin the actual connection process. While this isn't complicated, it does require a bit of synchronization to prevent either the device or PC from timing out.

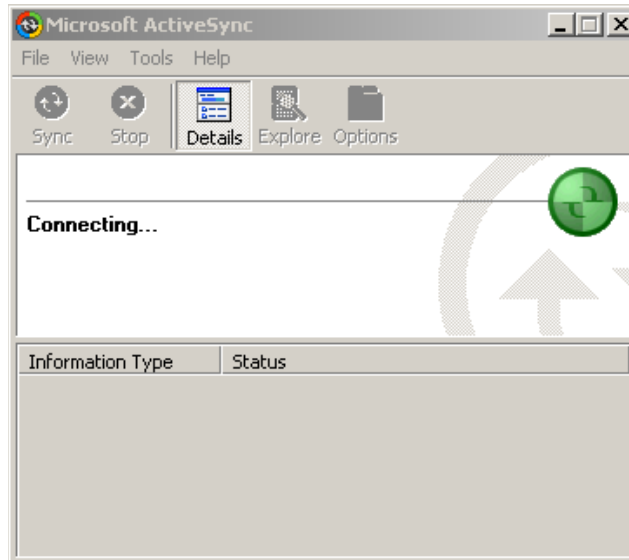
First, on the PC, click *Next*. You will get a new window showing that the PC is looking for devices.



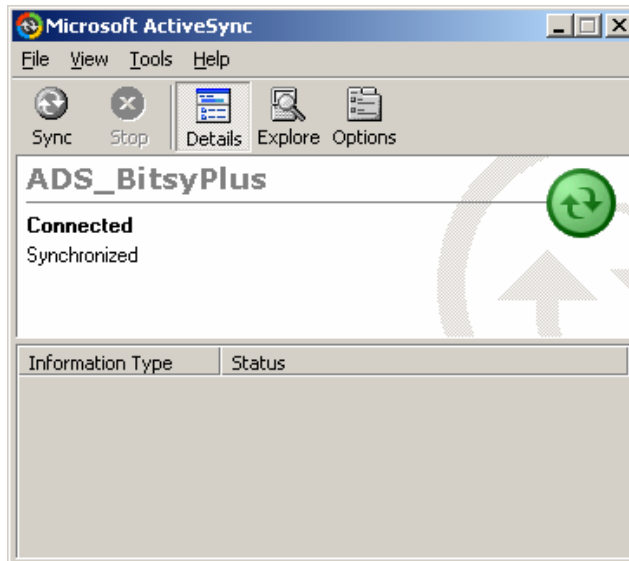
Now on the CE device's Run dialog tap *OK*. You will get a Connecting dialog as the device searches for a host.



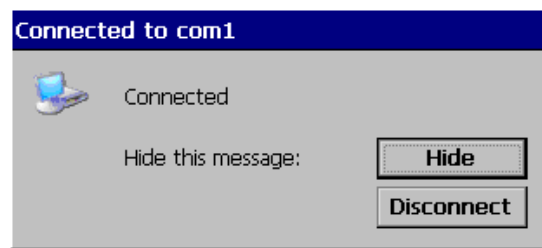
When the PC and CE device successfully find one another other, the PC the connection wizard will disappear and ActiveSync will display “Connecting...” in the application window.



A short time later, once the devices have established a connection ActiveSync on the PC will display the connected device's name.



The CE device will display a dialog that describes how it is connected.



The PC and the CE device also display tray icons in the lower right of their respective desktops to show that they are connected. On the PC it is a green “wheel” icon.



On the CE device it is a PC and monitor icon.



2.4.2 Ethernet

To make an ActiveSync Ethernet (TCP/IP) connection between your device and PC, you must first create a partnership using an RS232, USB or IrDA connection (see section 2.4.1). A partnership is not required for any other Ethernet use (such as web browsing from the device), only for an ActiveSync Ethernet connection.

Once a partnership has been established between the device and your development PC, make sure that the device is connected to your network and has a valid IP address. Note: being "connected" can be as simple as using a crossover Ethernet cable directly between the device and the development PC.

Next run "repllog.exe /remote" from the *Start | Run...* dialog. An additional dialog may appear in which you must choose the computer name of your development system. Once a connection is established, the Connection Status dialog will appear on the device (it may be minimized to an icon in the tray).

2.5 ***Persisting ActiveSync Partnerships***

When your development system loses power or is reset all information stored in RAM, such as registry settings and applications (including ActiveSync partnership information) is lost. To avoid having to repeat the process of creating a partnership you can modify your boot registry to include your partnership information.

More information on how to persist your device registry, including ActiveSync partnership information, can be found in section 3.3.4.4.

2.6 ***Debugging with eMbedded Visual Tools or Visual Studio.NET 2003***

eMbedded Visual Tools provides the ability to connect with your development system and provide live debugging information, break points and process attachment through an ActiveSync connection.

To use remote debugging you must have a live ActiveSync connection with the device. While an ActiveSync partnership is not required, we do recommend it so that you can debug over Ethernet. Large amounts of data are transferred between the device and the PC during debugging and Ethernet greatly facilitates speedy debugger responsiveness.

Application development and debugging are further detailed in the Windows CE Developer's Guide.

2.7 ***Device-side TCP/IP Settings***

In some cases the eMbedded Visual Tools' Connection Manager will still be unable to make a connection to the device, even with an active ActiveSync connection. This was particularly common with earlier versions of ActiveSync (3.1 and earlier). If this occurs, try setting the primary WINS address on the development system to the IP address of your development PC.

To set the Primary WINS address on the device, run the *Network* control panel applet. Select your network device/adaptor then click the *Properties* button. The WINS server address is set on the Name Servers tab of this dialog.

You can persist this and other Ethernet settings as described in Section 3.3.

3 Customizing Device Behavior and Settings

Once you have familiarized yourself with your ADS Windows CE device, you will often find the need to customize the device's behavior or settings. This section provides information on performing these customizations.

3.1 *Changing Boot Behavior*

Your ADS development system shipped ready to boot and run with a specific CE image and with specific configuration settings that are typical and sufficient for most developers' needs. It is, however, possible to alter almost every aspect of how the system boots.

ADS development systems have the inherent capability to allow you to change the resident-flash installed CE image, boot directly from an ATA storage device, modify the registry or hardware tables, modify your desktop background image, automatically launch an application at startup and directly manipulate the on-board flash system.

Caution

Altering the boot files or on-board flash of your development system can cause the system to no longer boot or to no longer output to the display properly. In some instances, these modifications may even require that your device be returned to Applied Data Systems for reprogramming.

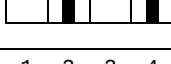
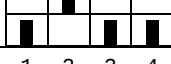
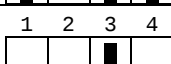
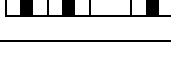
We highly recommend that you fully understand the implications of any alterations prior to making them. A large amount of additional information about the boot files and how to safely modify them can also be found online at <http://www.applieddata.net/support/>.

3.1.1 Boot Switch Settings

Your development system's boot behavior is largely affected by the positions of the DIP switches on your board (typically at S1). There are essentially five (5) boot modes:

1. **Normal Boot.** When powered up or reset, the device copies the OS from onboard flash to RAM and then boots. Development systems ship with S1 set for this mode.
2. **Open Box Upgrade.** When powered up or reset, the device copies all ADS boot files from an installed ATA storage card to onboard flash and then presents the debug menu (see below).
3. **Sealed Box Upgrade.** When powered up or reset, the device copies all ADS boot files from an installed ATA storage card to onboard flash then proceeds with a normal boot (see above).
4. **Boot from ATA.** When powered up or reset, the device copies the OS from an installed ATA storage card directly to RAM and boots (onboard flash is unaffected).
5. **Debug Menu.** When powered up or reset the device echoes a Debug Menu (see section 0) to the Debug Port (see section 2.1).

The following table outlines what effect different switch settings have on the boot mode:

	Graphics Client Plus Graphics Master Display Master AGX	Bitsy* Bitsy Plus Advanced Graphics Client	BitsyX
Normal Boot	1 2 3 4 ON  OFF 	1 2 ON  OFF 	1 2 ON  OFF 
Open Box Upgrade	1 2 3 4 ON  OFF 	1 2 ON  OFF 	1 2 ON  OFF 
Sealed Box Upgrade	1 2 3 4 ON  OFF 	1 2 ON  OFF 	1 2 ON  OFF 
Boot from ATA	1 2 3 4 ON  OFF 	1 2 ON  OFF 	Only available through virtual switch in HWT
Debug Menu ‡	1 2 3 4 ON  OFF 	1 2 ON  OFF 	1 2 ON  OFF 
Use Virtual Switch from HWT	N/A	N/A	1 2 ON  OFF 

* Bitsy systems may have a 2 or 4-position DIP switch installed. Only 1 and 2 affect system boot.

‡ BitsyX and AGX systems will display the debug menu even if no image is available on an inserted ATA card

Virtual Boot Switch

The BitsyX and AGX platforms have added a “virtual” boot switch to complement the physical switch and accommodate future feature additions. The virtual switch is set by modifying the following line in the device’s ADSLOAD.HWT file (see Section 3.3.3)

```

;-----
; BootSwitch,
;   9: Load CE files from ATA card if available, then Start CE.
;   5: Read CE image from Flash and Start CE
;   4: Load CE files from ATA card if available, then Start Menu on Debug Port
;   2: Start CE from ATA card if available.
;   Menu on Debug port : Other cases and Failed from above
;-----
0xA1E30000 0x00000009 ; BootSwitch

```

The valid values for the virtual switch setting and their behavior are:

Boot Mode	Virtual Switch Value
Sealed Box Upgrade	0x00000009
Normal Boot	0x00000005
Open Box Upgrade	0x00000004
Boot from ATA	0x00000002
Debug Menu	0x00000000

3.2

The Debug Menu

The debug menu can, in some instances, be useful for troubleshooting or maintaining your development system. Most commonly it is used to erase blocks of flash corrupted during development.

Caution

The debug menu allows direct access to blocks of flash memory. Incorrect use of the debug menu may require that your device be returned to Applied Data Systems for reprogramming.

We highly recommend that you fully understand the implications of any alterations prior to making them. A large amount of additional information about the boot files and how to safely modify them can also be found online at <http://www.applieddata.net/support/>.

To display and use the debug menu, connect your development PC to your device’s debug port (see section 2.1), make sure the debug switch is set for Debug Menu (section 3.1.1) and reset the device. The following is an example of what will be echoed through the debug port:

```

*****
*                               *
*           ADS Bitsy Loader           *
*                               *
*****

1]  Display Register
2]  Set Register
3]  Dump Memory
4]  Set Memory
5]  Erase the block of Flash.
6]  Display Files in the Flash.
7]  Move an Image from Flash to RAM
8]  Display Ethernet Information
9]  Set Up Ethernet Information

```

```

0] Download Image Via Ethernet(ESHELL)
A] Download over Serial (XModem)
B] Execute HWT(Hardware Table) from Flash
C] Start Windows CE @ c1800000
D] Fill Memory with a Pattern.
E] Set MAC(Media Access Control) Address.
F] SA1111 Test.
G] Find PCMCIA Card.

```

3.2.1 Erasing Blocks of Flash Memory

On rare occasion it is useful to fully erase the flash memory of your development system. This is usually associated with a disaster recovery where part of the flash memory has been corrupted or when the flash driver has difficulty recognizing all or part of the flash memory.

You can erase blocks of flash memory using item 5 of the debug menu. Be very cautious, however. On systems where the boot code is stored in flash, such as the Bitsy family of devices, you can easily erase the boot code. The only recovery from erased boot code is to send your development system back to ADS for reprogramming.

After selecting menu item 5, select option 1 "Select by Block Address (Minimum 256KB, 40000h)". You will be prompted for starting and ending addresses between which the system will erase. On systems that have a separate boot ROM, you can safely erase from 0 to 2000000 (six zeros) for 32MB flash systems. On systems that have the boot code held in flash (such as the Bitsy) do not erase anything in the first 2MB of flash. Instead erase from 200000 (five zeros) to 2000000 (six zeros) for 32MB flash systems.

Erasing the flash memory on your development system will require you to reload NK[LOAD].BIN, ADSLOAD.EXE and ADSLOAD.REG (see section 3.3).

3.2.2 Backing up ADSLOAD.REG and ADSLOAD.HWT

Before modifying any of the ADS Boot Files (see Section 3.4) we recommended that you make a backup of your existing ADSLOAD.REG and ADSLOAD.HWT configuration files. This is done in two ways:

1. Using ADSLOAExtract.exe

ADSLOAExtract.exe is a utility that extracts ADSLOAD.REG and ADSLOAD.HWT from the device and saves them in the device's \Temp folder. ADSLOAExtract is available online in the ADS Support Forums at http://www.applieddata.net/forums/topic.asp?TOPIC_ID=964.

2. Using the Debug Menu

The configuration files can also be extracted using any terminal program that allows data capture. The following procedures are for HyperTerminal:

1. Connect to your device's Debug Port as outlined earlier in section 0.
2. In HyperTerminal's program menu select **Transfer | Capture Text** and provide the path and filename for your backup (e.g. "C:\Backups\ADSLOAD.REG").
3. From the debug menu, select option 6, "Display Files in the Flash" and you will get the following menu:

```

1] ADSLOAD.BMP (The BMP File)
2] ADSLOAD.HWT (The hardware register table)
3] ADSLOAD.REG (The registry file)
4] ADSLOAD.EXE (The startup application)
5] NKLOAD.BIN (The WINCE image)

```

4. Choose the file you wish to back up (in this example option 3). The file contents will scroll through on HyperTerminal's main screen while simultaneously writing to the file location selection in Step 3. Note that even though the debug menu has options for binary files, you can only extract the ADSLOAD.REG and ADSLOAD.HWT text files through HyperTerminal.
5. Stop capturing HyperTerminal's output by selecting from the program menu **Transfer | Capture Text | Stop**.
6. Now open the target text file to trim any of the debug menu output from the start and/or end of the file that may have also been captured.

3.3 **ADS System Files**

All ADS devices reserve a section of flash memory to store system files. These files include the CE image, an executable, a bitmap, and files containing information used during the boot process.

These files, described in detail in the following sections, are not visible by the Windows CE file system. They can only be viewed and/or modified using the methods described below.

Note

Before changing your existing device configuration files, it is highly recommended that you back up your existing, known-good configuration files.

3.3.1 **Updating the Files**

You can modify and update your development system by placing new system files (any number in any combination) with the following steps:

1. Place the files in the root directory of an on an ATA card (PCMCIA or Compact Flash)
2. Insert the card in the device
3. Set your boot switches to Sealed Box Upgrade (see section 3.1.1)
4. Press the reset button. You can follow the progress of the modifications and see information about any errors that occur by following the debug output. In addition, devices with 3 LEDs will turn on the amber LED while copying files to flash.

The following sections describe the function of each boot file supported.

3.3.2 **NKLOAD.BIN / NK.BIN**

NKLOAD.BIN or NK.BIN is the Windows CE image. It contains the kernel, system drivers, applications like IE and Media Player and other system files. All ADS systems that use the StrongARM processor use NKLOAD.BIN, all StrongARM systems use NK.BIN.

When uploading a new BIN file to your development system, you must ensure that the BIN file is for your specific platform (i.e. a Bitsy image will not work with a Graphics Master) with the same hardware configurations (i.e. you cannot use a build designed for a system with 64MB RAM on a system with 32MB or RAM). You must also ensure that you modify your ADSLOAD.REG and ADSLOAD.HWT files to match the new image size (see sections 3.3.3 and 3.3.4).

You can download pre-compiled Windows CE images for your ADS development System from the ADS Support Web site at <http://www.applieddata.net/support/>.

3.3.3 **ADSLOAD.HWT**

ADSLOAD.HWT is the hardware table for your development system. This text file defines specific aspects of your hardware that the boot loader and Windows CE need to start up, such as where the image resides in flash and how large it is.

ADSLOAD.HWT must be less than 256k in size.

ADS systems based on the StrongARM processor require that ADSLOAD.HWT provides information about the size of your CE image and the amount of RAM and flash your system has. XScale devices automatically determine this information, but StrongARM platforms will have a section in ADSLOAD.HWT that looks like this:

```
; Onboard flash set aside for NKLOAD.BIN
;0xA110C401 0x00800000 ; 8MB
0xA110C401 0x00A00000 ; 10MB
;0xA110C401 0x00C00000 ; 12MB

; Location in RAM to load CE image
;0x00B002AD 0xC0800000 ; 8MB CE, 16MB RAM
;0x00B002AD 0xC0600000 ; 10MB CE, 16MB RAM
;0x00B002AD 0xC0400000 ; 12MB CE, 16MB RAM

;0x00B002AD 0xC8800000 ; 8MB CE, 32MB RAM
0x00B002AD 0xC8600000 ; 10MB CE, 32MB RAM
;0x00B002AD 0xC8400000 ; 12MB CE, 32MB RAM

;0x00B002AD 0xC9800000 ; 8MB CE, 64MB RAM
;0x00B002AD 0xC9600000 ; 10MB CE, 64MB RAM
;0x00B002AD 0xC9400000 ; 12MB CE, 64MB RAM
```

The semicolon is seen as a comment marker. Information from a semicolon to the end of the line is ignored. In most cases you will be able to comment and uncomment a couple lines to get the file to match the characteristics of your system.

For example, only the **bold** lines in the above sample will be used by the system.

The first section is the amount of flash to reserve to store NKLOAD.BIN. Make sure that this value is larger than the NKLOAD.BIN image size you are using. In the above example, we are using a 10 MB image.

The next section is the RAM address for loading the image. This value depends both on the size of the image you are using and the amount of available RAM in your system. In the above example, we are using a 10MB image on a system with 32MB of RAM.

For more information on modifying ADSLOAD.HWT visit our online support forums at <http://www.applieddata.net/support/>.

3.3.4 ADSLOAD.REG

ADSLOAD.REG is a text files that contains updates to the core registry that will be applied at boot. It contains settings for things such as the display, the flash disk driver, touch panel and a variety of initial state settings.

ADSLOAD.REG must be less than 256k in overall size.

Below are some examples of useful ADSLOAD.REG entries. More entries can be found online at <http://www.applieddata.net/support/>.

3.3.4.1 Altering ADSLOAD.REG for a new CE Image

When you download the standard version of ADSLOAD.REG, you may need to update it to set the size of your FlashFX Disk. The size you select depends on the amount of installed flash, the size of your CE image and your specific CE device. See the ADS Support Forum Topic found at http://www.applieddata.net/forums/topic.asp?TOPIC_ID=299 for a memory map and more details.

ADS systems based on the StrongARM processor require that ADSLOAD.REG provides information about the size and start address of your FlashFX disk. XScale devices automatically determine this information, but StrongARM platforms will have a section in ADSLOAD.REG that looks like this:

```
[HKEY_LOCAL_MACHINE\Drivers\BuiltIn\FlashFx]
"Start"=dword:08B00000 ; 10MB CE image
"Length"=dword:0500000 ; 5MB flash disk (16MB flash installed)
; "Length"=dword:1500000 ; 21MB flash disk (32MB flash installed)
;
; "Start"=dword:08D00000 ; 12MB CE image
; "Length"=dword:0300000 ; 3MB flash disk (16MB flash installed)
; "Length"=dword:1300000 ; 19MB flash disk (32MB flash installed)
;
; "Start"=dword:08900000 ; 8MB CE image
; "Length"=dword:0700000 ; 7MB flash disk (16MB flash installed)
; "Length"=dword:1700000 ; 23MB flash disk (32MB flash installed)
```

The semicolon is seen as a comment marker, so any line with a leading semicolon is ignored. We have tried to provide all possible configurations, so in most cases you will be able to simply comment and uncomment a couple lines to get the file to match the CE image and flash installed on your system.

For example, only the **bold** lines in the above sample will be used by the system.

The sum of Start and Length values must be less than 0x09000000 for 16MB flash systems and less than 0x0A000000 for 32MB flash systems. The Start value must be above the locations of all system files stored in flash.

3.3.4.2 Persisting Device TCP/IP Settings

The following ADSLOAD.REG entries will persist the settings for the on-board Ethernet adapters on ADS development systems:

```
;-----
; TCP/IP Address info
;-----
[HKEY_LOCAL_MACHINE\Comm\SMC90001\Parms\TcpIp]
"EnabledDHCP"=dword:0 ; 0 = OFF, 1 = ON
"IpAddress"="192.168.0.100" ; Ignored with DHCP
"Subnetmask"="255.255.255. 0" ; Ignored with DHCP
"DefaultGateway"="192.168.0.1" ; Ignored with DHCP
"DNS"="192.168.0.1"
"WINS"="192.168.0.55"
```

You can set these values by modifying and reloading ADSLOAD.REG or you can use the Network Control Panel applets and then persist the registry (see section 3.6).

3.3.4.3 ObjectStorePages: Memory Allocation

Windows CE allows a portion of RAM to be dedicated to file storage (called “object storage” by CE). You can modify this value by adjusting the slider bar in the *Memory* tab of the *System* Control Panel applet.

You can set this value and make the setting persistent by modifying the registry with an entry in ADSLOAD.REG by defining how many 4k pages of memory should be set aside for storage (the remainder being allocated for Program Memory). Values are set in hexadecimal: i.e. “40” allocates 256k, “100” allocates 1MB and “280” allocates 2.5MB.

```
;-----
; Set Storage memory and Program memory
```

```

;-----
[HKEY_LOCAL_MACHINE\Drivers\BuiltIn\MemoryConfig]
"ObjectStorePages"=dword:200 ; allocate 2MB for program memory

```

You can persist this value by modifying and reloading ADSLOAD.REG or you can use the Network Control Panel applets and then persist the registry (see section 3.6).

3.3.4.4 ActiveSync Settings

The recommended method for persisting ActiveSync settings is to persist the CE device registry after creating the partnership, provided your device supports a persistent registry. If your device does not support a persistent registry, you can persist an ActiveSync partnership by following these steps:

1. Establish an ActiveSync partnership with your desktop system (see section 2.4.1).
2. Using \windows\regedit.exe on your ADS WinCE device. Find and copy the values from the following registry keys on your development system:

```

\HKLM\Software\Microsoft\Windows CE Services\Partners\P1
"PName"
"Pid"

```

3. Enter the values from those keys into your new ADSLOAD.REG file so it looks something like this (the actual values will depend on your system):

```

;-----
; ActiveSync partnership info
;-----
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows CE Services\Partners\P1]
"PName"="MyWorkstation"
"Pid"=dword:05af05af

```

4. Update the ADSLOAD.REG file on your system with the changed file by placing the new file on an ATA card a following the steps outlined in section 3.3 above.

3.3.5 ADSLOAD.EXE

ADSLOAD.EXE provides the ability to automatically launch an application at boot time. Simply rename your executable ADSLOAD.EXE and load it onto your system by placing the new file on an ATA card a following the steps outlined in section 3.3 above.

ADSLOAD.EXE can be any compiled executable that you want to run at system startup. Once copied to the development system, this executable will be run whenever the development system comes up from either a hard-reset or a loss of power.

ADSLOAD.EXE is optional, but if it is used, it must be less than 256k in size.

To remove ADSLOAD.EXE you must either format the device's flash disk or replace ADSLOAD.EXE with another executable that does nothing.

3.3.6 ADSLOAD.BMP

ADSLOAD.BMP is the image used as the background or desktop image when the CE desktop is finally launched. Development systems are loaded with the ADS logo, but this image can be anything you like.

This file must be less than 256k in size and either 4-bpp or 8-bpp.

3.4 **Changing the Boot Code**

Some ADS development systems, such as the Bitsy family, store their boot code in onboard flash, while others, such as the Graphics Master and Graphics Client Plus, store their code in a separate socketed part. The boot code initializes the processor registers, reads the ADSLOAD.HWT file and launches the Windows CE image. It also includes functionality to copy system files from a PCMCIA or CF card into the onboard flash and a simple boatload menu.

If you have a system that stores the boot code in Flash, you can download an update to your boot code as ADSLOAD.ROM, available from <http://www.applieddata.net/support/>.

Load this file in the same manner as you do the CE image and configuration files (NK[LOAD].BIN, ADSLOAD.HWT, ADSLOAD.REG, etc) as described in section 3.3.

Caution

If an error occurs in reprogramming the boot code, you may have to return your Bitsy to the factory for reprogramming. Charges may apply. Please take the following precautions when reprogramming your system's boot code:

1. As a check against file corruption, make sure the file you copy to the flash disk is the size and date listed in the readme accompanying the ADSLOAD.ROM file.
2. Test your file loading process with a non-critical file such as ADSLOAD.BMP.
3. Be very careful not to interrupt reprogramming of your Bitsy. (Because the critical portion of the process is fairly quick-- about two or three seconds--it's unlikely you'll accidentally interrupt it).

3.5 **Changing the ADSmartIO Firmware**

AVRPROG.EXE, found in the Windows folder of your ADS development system is an application that allows you to update the ADSmartIO firmware at your site.

To upgrade the ADSmartIO firmware, follow these instructions:

1. Download the desired SMARTIO.BIN firmware file from <http://www.applieddata.net/support/> and place it in the root folder of your CE device (i.e. at the same level as the Windows folder).
2. (Optional but strongly recommended) Set up the debug port to monitor information about the status of the reprogramming.
3. Run the AVRPROG.EXE application. If you are running the backlight from your CE device, the display may go dark while the firmware is being reprogrammed. Programming takes up to 60 seconds.
4. When complete, you should see a message displayed on the screen.
5. (Optional) Reset the system to confirm the version of the ADSmartIO code that was actually installed. See text at the top of the document included with the firmware file for expected debug port output. Note that the part type may vary between 4434 and 8535.

3.6 The ADS Persistent Registry

Some ADS CE Builds have the ability to persist the entire registry to flash RAM. This is very useful for maintaining registry changes through power outages and hard-resets. This method is simpler than the method of persisting registry information through the ADSLOAD.REG file (see section 3.3.4).

To determine if your device supports a persistent registry, open the device's Control Panel. If you have an "ADS Persistent Registry" applet, then your device supports a persistent registry.



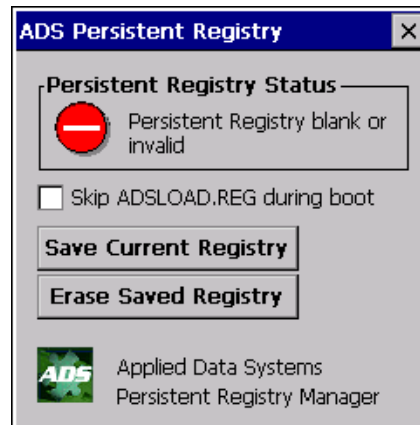
Note

Early CE images that supported the persistent did not include the ADS Persistent Registry Control Panel applet. If your device support the persistent registry but does not have the applet, visit <http://www.applieddata.net/support/> to get a newer CE image.

Using the persistent registry is straightforward. By launching the Control Panel applet, you will see the current status of the persistent registry. As shipped, it will likely be blank and the applet will look like the figure on the right.

You can see that the applet allows you to save the current registry, erase the currently persisted (saved) registry and to skip ADSLOAD.REG during the boot process.

To save the device's current registry at any time, simply tap the "Save Current Registry" button.

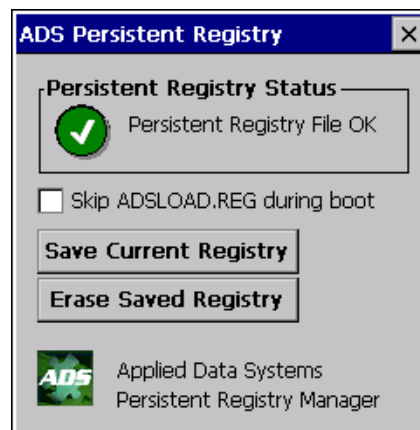


Note

The device registry can also be saved by calling the `RegFlushKey()` API from an application. Keep in mind that any application that calls `RegFlushKey` will save the registry, including installing *any* CAB file.

Saving the registry takes a few seconds. Once your registry has been successfully saved to flash, the applet will notify you and look like the figure to the right.

To erase it at any time, simply tap on the "Erase Saved Registry" button. Currently this function can *only* be performed through the Persistent Registry applet – there is no equivalent API.



The last feature of the persistent registry is the ability to ignore the ADSLOAD.REG file stored in flash. To ignore the configuration file, simply check the checkbox on the applet. The data for this feature is stored in the registry under the following DWORD key:

```
[HKEY_LOCAL_MACHINE/Drivers/BuiltIn/REGI]
SkipRegistryInit
```

Note

Early CE images that supported the persistent registry would *not* erase the persisted registry when the CE image was updated. This can lead to a lot of confusion because the new image will use the old registry and therefore report itself as the image that was replaced.

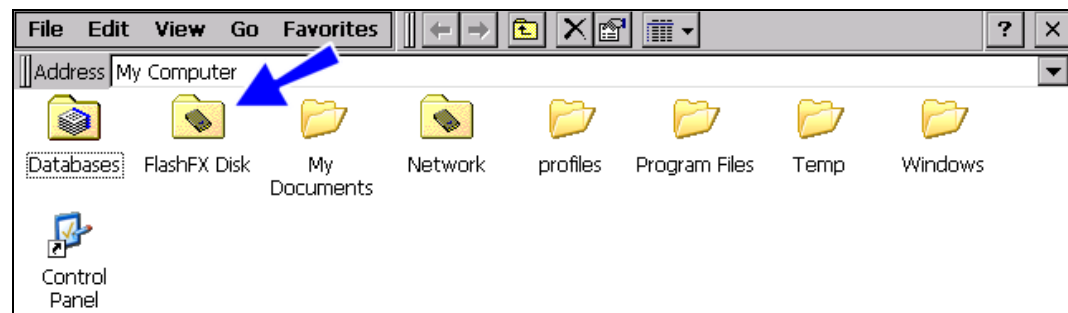
To avoid this problem, it is recommended that you *always* erase the persistent registry before upgrading the image on your device.

3.7

Persistent Storage

On the surface, standard ADS products appear to behave differently than typical commercial PDAs in that when you turn off or reset the ADS device, all of your applications, data and settings are lost. The reason for this is that typical commercial PDAs are never actually turned off, but are put into sleep mode. While in sleep mode, the device RAM is maintained by battery power. ADS systems support this same functionality with the addition of an external battery, and if you would like more information on implementing this, check the power management section of your device's User's Manual.

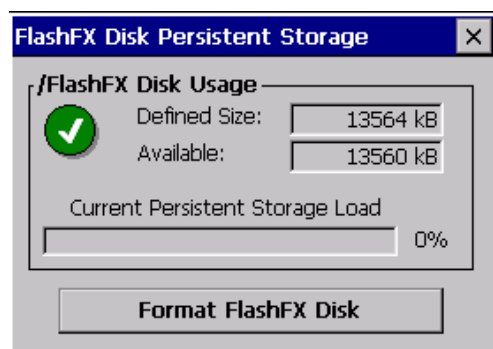
Unlike most commercial PDAs, ADS systems also provide the ability to save data even through a full loss of power. All standard ADS products reserve the area of flash RAM not used for storing system data (i.e. the CE image or configuration files) for persistent file storage. This area is accessed through the "FlashFX Disk" folder in the root directory of your development system and to applications it is no different than any other folder in the CE filesystem.



If your system does not display the "FlashFX Disk" folder it is likely that the FlashFX driver could not find a FAT for it and the disk needs to be formatted. Formatting the FlashFX Disk can be done using the ADS Persistent Storage Control Panel applet by tapping the "Format FlashFX Disk" button.



The ADS Persistent Storage Control Panel applet also shows the maximum amount of persistent storage available through the FlashFX Disk folder, the amount remaining available as well as the "load" or percentage of the folder currently being used for storing data.

**Note:**

Early Windows CE builds did not include the Persistent Storage applet. To format the FlashFX Disk on these systems a utility application is included in the \Windows folder called `erasefxdisk.exe`. Simply run this executable to format your flash disk.

When the device power is cycled or the device is reset, any data stored in the FlashFX Disk folder will remain intact. If you need data to be stored in other folder locations, ADS provides several useful utilities for copying data such as `ADSCopy.exe`. Visit <http://www.applieddata.net/support/> for more information on these utilities.

4 What Next?

This manual is intended as a starting point for Windows CE developers using Applied Data Systems products. ADS provides additional resources beyond this document that we also highly recommend you utilize to get the best possible ADS experience.

4.1 **ADS Windows CE Developer's Guide**

For more in-depth coverage of creating, debugging and deploying applications on your Applied Data Systems Windows CE device, see the [ADS Windows CE Developer's Guide](#). There are separate versions of the Guide depending on your development language, but each version covers creating application projects, how to debug applications in the chosen environment plus tips and suggestions on methods for deploying your finished application.

Note: As of the publication of this revision of the [Developer's Getting Started Guide](#) the [Developer's Guide](#) has been released in preliminary form only.

4.2 **ADS On the Web**

Applied Data Systems has a large online knowledgebase in our Support Forums. We have answers to our most common support questions, sample applications complete with source code, utility applications to help facilitate your solution and electronic copies of all of our product documentation.

Visit <http://www.applieddata.net/support/> to create a user account, read and search existing posts and post your own questions.