

Windows NT Embedded Step-by-Step

Sean D. Liming

Annabooks
San Diego

Windows NT Embedded Step-by-Step

by
Sean D. Liming

PUBLISHED BY

Annabooks
12860 Danielson Court
Poway, CA 92064
USA

858-391-5600
<http://www.annabooks.com>

Copyright © Sean D. Liming 2000

All rights reserved. No part of the contents of this book may be reproduced or transmitted in any form or by any means without the prior written consent of the publisher, except for the inclusion of brief quotations in a review.

Printed in the United States of America
ISBN 0-929392-68-X
First Printing May 2000

Information provided in this publication is derived from various sources, standards, and analyses. Any errors or omissions shall not imply any liability for direct or indirect consequences arising from the use of this information. The publisher, authors, and reviewers make no warranty for the correctness or for the use of this information, and assume no liability for direct or indirect damages of any kind arising from technical interpretation or technical explanations in this book, for typographical or printing errors, or for any subsequent changes.

The publisher and authors reserve the right to make changes in this publication without notice and without incurring any liability.

All trademarks mentioned in this book are the property of their respective owners. Annabooks has attempted to properly capitalize and punctuate trademarks, but cannot guarantee that it has done so properly in every case.

Table of Contents

CHAPTER 1 INTRODUCTION

- 1.1 WHAT IS WINDOWS NT EMBEDDED?
- 1.2 DEVELOPMENT OVERVIEW
- 1.3 EMBEDDED SYSTEMS SUPPORT
- 1.4 WHAT ABOUT THE SIZE?
- 1.5 NTE SUMMARY
- 1.6 ABOUT THIS BOOK
- 1.7 DEVELOPMENT SYSTEM REQUIREMENTS
 - 1.7.1 *Hardware Requirements:*
 - 1.7.2 *Software Requirements:*
- 1.8 TARGET SYSTEM REQUIREMENTS
- 1.9 THE TARGET FOR THE EXERCISES
- 1.10 DEVELOPMENT SETUP

CHAPTER 2 TARGET DESIGNER – CREATING NTE IMAGES

- 2.1 TARGET DESIGNER OVERVIEW
 - 2.1.1 *Target Designer Breakdown*
 - 2.1.2 *Menu and Toolbars*
- 2.2 TARGET DESIGNER STEP-BY-STEP
- 2.3 TDSB BREAKDOWN - COMPONENTS AND CAPABILITIES
 - 2.3.1 *System Role*
 - 2.3.2 *Core OS*
 - 2.3.3 *HAL*
 - 2.3.4 *Devices*
 - 2.3.5 *Network*
 - 2.3.6 *Windows Services*
 - 2.3.7 *Management*
 - 2.3.8 *Applications*
- 2.4 EXERCISE 1A – BUILDING AN IMAGE
 - 2.4.1 *Creating the Configuration*
- 2.5 BUILD IMAGE DIALOG OPTIONS
- 2.6 PREPARE TARGET AND REMOTE DOWNLOAD
 - 2.6.1 *Option 1. Second Hard Drive*
 - 2.6.2 *Option 2. Remote Download*
 - 2.6.3 *Preparing the Target*
- 2.7 EXERCISE 1B - REMOTE DOWNLOAD
 - 2.7.1 *Create the Mungeboot Disk*
 - 2.7.2 *Creating the Remote Recover Boot Disk*
- 2.8 EXERCISE 1C – EXPORTING AND IMPORTING CONFIGURATIONS
- 2.9 EXERCISE 1D - BOOTING FROM AN NTFS PARTITION

2.10 SUMMARY

CHAPTER 3 EMBEDDED FEATURES

- 3.1 HEADLESS SUPPORT
 - 3.1.1 *NULL Video Component*
 - 3.1.2 *NULL Keyboard and NULL Mouse*
- 3.2 READ-ONLY BOOT SUPPORT
 - 3.2.1 *Write Filter Component*
 - 3.2.2 *El Torito Component*
 - 3.2.3 *No Page File Component*
- 3.3 SOLID-STATE MEDIA SUPPORT
 - 3.3.1 *DiskOnChip Driver*
 - 3.3.2 *ATA (PCMCIA, PC Card) Flash Disks*
- 3.4 REMOTE MANAGEMENT
 - 3.4.1 *Serial/Modem Remote Management*
 - 3.4.2 *Graphical Remote Management*
 - 3.4.3 *Telnet*
 - 3.4.4 *Shutdown Utility*
- 3.5 ERROR RECOVERY
 - 3.5.1 *System Message Interception*
 - 3.5.2 *Blue Screen*
- 3.6 SID CLONING
- 3.7 EXERCISE 2 - HEADLESS SYSTEM
 - 3.7.1 *Build Image*
 - 3.7.2 *Prepare HyperTerminal on the Host Machine*
 - 3.7.3 *File Transfers*
 - 3.7.4 *Transferring a File*
- 3.8 EXERCISE 3 – EMBEDDED FEATURES: DISKONCHIP, COMMAND SHELL, SHUTDOWN UTILITY, AND THE WRITE FILTER
 - 3.8.1 *Hardware for Exercise*
 - 3.8.2 *Building Image*
 - 3.8.3 *Running the Image*
- 3.9 SUMMARY

CHAPTER 4 COMPONENT DESIGNER - DEVELOPING KDFS

- 4.1 COMPONENT DESIGNER OVERVIEW
- 4.2 TO KDF OR NOT TO KDF
- 4.3 COMPONENT DESIGNER STEP-BY-STEP
- 4.4 TSDS RULES
- 4.5 GATHERING KDF INFORMATION
- 4.6 TARGET DESIGNER EXTENSIONS (TDX)
- 4.7 EXERCISE 4 - APPLICATION KDF FOR MSPAINT
 - 4.7.1 *Application KDF Basics*
 - 4.7.2 *Create KDF Structure*
- 4.8 EXERCISE 5 - DRIVER KDF FOR NETWORK CARD, HIVE FILE METHOD

- 4.8.1 *Device Driver KDF Creation Basics*
- 4.8.2 *Prepare Target to Add Network Support*
- 4.8.3 *Capture Registry Entries for KDF*
- 4.8.4 *Creating the KDF on the Host Machine*
- 4.8.5 *Build Image with Network support*
- 4.9 UNDERSTANDING THE REGISTRY
 - 4.9.1 *Registry Hierarchy*
 - 4.9.2 *Registry Patterns*
 - 4.9.3 *Sample KDFs*
- 4.10 EXERCISE 6 - DEVELOPING THE "HELLO WORLD" TDX
 - 4.10.1 *TDX Overview*
 - 4.10.2 *SAMPLE.TDX*
 - 4.10.3 *Exercise 6 Overview*
 - 4.10.4 *Create the New Project*
 - 4.10.5 *Project Setup*
 - 4.10.6 *Create a Registry Information Class*
 - 4.10.7 *Create the Properties Dialog*
 - 4.10.8 *Add the Test Key Definition*
 - 4.10.9 *Add the Entry Points*
 - 4.10.10 *Create Test KDF*
- 4.11 SUMMARY

CHAPTER 5 EXPLORING OTHER COMPONENTS AND FEATURES

- 5.1 EXERCISE 7 - REMOTE ADMINISTRATION, GRAPHICAL AND TELNET
 - 5.1.1 *Building the Image*
 - 5.1.2 *Graphical Remote Administration - Connection*
 - 5.1.3 *Telnet Connection*
- 5.2 EXERCISE 7B - GRAPHICAL REMOTE ADMINISTRATION ON A HEADLESS SYSTEM
 - 5.2.1 *Build the Image*
- 5.3 EXERCISE 8 - INTERNET INFORMATION SERVER 3.0 AND FRONTPAGE 2000
 - 5.3.1 *Build the Image*
- 5.4 EXERCISE 8B - HEADLESS WEB SERVER
- 5.5 EXERCISE 9 - DUAL NETWORK CARDS
 - 5.5.1 *Build an Image with Realtek 8029 Support*
 - 5.5.2 *Capture 8139A Second Instance Registry Keys*
- 5.6 EXERCISE 10 - CREATING A BOOTABLE CD-ROM SYSTEM
 - 5.6.1 *CD-ROM Software*
 - 5.6.2 *El Torito and WriteFilter Components*
 - 5.6.3 *Exercise Overview*
 - 5.6.4 *Creating the Image*
 - 5.6.5 *Preparing the Target Drive*
 - 5.6.6 *Creating the CD*
- 5.7 EXERCISE 10B - DATA / APPLICATIONS ON THE ISO-9660 TRACK.
- 5.8 EXERCISE 11 - ERROR RECOVERY
 - 5.8.1 *Part 1 – Event Log*

- 5.8.2 *Part 2 – Automatic Reply*
- 5.9 EXERCISE 12 - SCSI KDF – LOCAL METHOD, AND SCSI SUPPORT
 - 5.9.1 *Target Platform Hardware*
 - 5.9.2 *Build up the Target Platform*
 - 5.9.3 *SCSI Driver Installation*
 - 5.9.4 *Creating the SCSI KDF*
 - 5.9.5 *Implement the New SCSI KDF*
 - 5.9.6 *SCSI KDF Summary*
- 5.10 EXERCISE 13 - MULTIPROCESSOR SYSTEM
 - 5.10.1 *Required Hardware*
 - 5.10.2 *Exercise Overview*
 - 5.10.3 *Creating the Pinball KDF*
 - 5.10.4 *Test Pinball on a Single CPU System (Intel Pentium III Evaluation Board)*
 - 5.10.5 *Test Pinball on a Multiprocessor System (Intel Pentium III / 840 Development Kit)*
 - 5.10.6 *Summary*

CHAPTER 6 DEPLOYMENT STRATEGIES

- 6.1 SYSTEM TESTS
- 6.2 MANUFACTURING - DISK DUPLICATORS, SIDS, NETWORK ADDRESSES, AND COMPUTER NAMES
 - 6.2.1 *Security ID (SID)*
 - 6.2.2 *Computer Names and Network Addresses*
 - 6.2.3 *SystemPrep Tool*
 - 6.2.4 *Write Protected Boot Partitions and El Torito CD-ROMs*
 - 6.2.5 *Duplication*
 - 6.2.6 *Manufacturing Summary*
- 6.3 EXERCISE 14 - SYSTEM CLONING
 - 6.3.1 *Image without System Cloning*
 - 6.3.2 *Image with System Cloning*
- 6.4 SOURCE CONTROL
- 6.5 FIELD UPDATE ISSUES
 - 6.5.1 *Easy Restore*
 - 6.5.2 *Worst Case Situation*
- 6.6 HIGH AVAILABILITY

CHAPTER 7 RESOURCES

- 7.1 THIRD PARTY TOOLS
- 7.2 REFERENCES & RECOMMENDED READING
- 7.3 COMPONENT LIST

Chapter 3 EMBEDDED FEATURES

As part of the TDSB, Microsoft has included some embedded components that are not part of the standard Windows NT release. These components allow Windows NT to be deployed in a variety of applied computing systems.

3.1 Headless Support

This enables Windows NT to be used in devices that boot and run without a mouse, keyboard, or display device. Many embedded systems do not expose either a traditional user interface (e.g., Windows-based or DOS-based PC) or, in many instances, any local user interface whatsoever. Windows NT requires a display driver to interface between the graphics sub-system and the video hardware. All currently available video display drivers assume and rely on the existence of underlying video hardware. Microsoft has included a NULL video driver component that eliminates the need for video hardware.

3.1.1 NULL Video Component

The NULL Video Driver acts as a stubbed driver. GUI calls are made to the video driver, but the driver does not access any video hardware. The video information is kept in memory and an acknowledgement is returned that the video command was completed successfully. Exercise 6b shows that you can access a headless system video interface remotely.

3.1.2 NULL Keyboard and NULL Mouse

The NULL Keyboard and NULL Mouse drivers are dependent on each other. When you select the NULL Keyboard, you are asked for the NULL mouse component, and vice-versa. These drivers are similar to the NULL video, but operate in reverse since these are input drivers. Connection status is returned "okay", faking NT into believing that there is a mouse and keyboard present.

TIP: Most PC platforms expect a video card and keyboard. The BIOSes in these platforms will halt if a device like a VGA card and keyboard are missing. Make sure that the

BIOS can turn off detection of the video card and/or keyboard if you want to use the board in a headless system.

3.2 Read-Only Boot Support

Many embedded devices use ROM to support solid state operation, which lowers the unit cost and improves reliability. Windows NT Embedded supports a variety of read-only media for boot-strapping itself in a manner that is transparent to the applications and system binaries that access the media (e.g., El Torito support with the write-through filter component allows you to boot from CD-ROMs).

3.2.1 Write Filter Component

The Write Filter driver allows any writes to a partition to be redirected to a RAM disk. The driver allows partitions and whole disks to act as read-only media. The filter can be used for protecting the life of flash devices such as DiskOnChip or ATA flash devices. Any files written to a partition will be saved to memory, and while the system is still running the files can be transferred, opened, and deleted just like any file. When you shut down the system the files will be gone. The component's properties must be modified to list all of the read-only partitions. This will allow you to write-protect a partition via software. You do not need to list any read-only partitions, such as CD-ROMs.

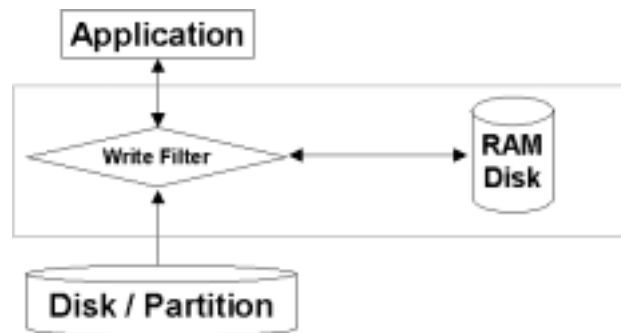


Figure 3.1: Write Filter Architecture

3.2.2 El Torito Component

The El Torito driver allows OEMs to build bootable NT CDs. The El Torito component must be used with the write filter and the No Page file components.

3.2.3 No Page File Component

NT Embedded allows you to select the virtual memory page file size from 0 to 512 MB. The No Page file can be used to reduce the constant read/writes to disks. The No Page file can extend the life of flash disks if read/write access is required. You may need to increase the amount of RAM in your system to improve performance. You should profile your application(s) to determine what memory resources are required.

3.3 Solid-State Media Support

Due to environmental factors such as shock and vibration, many embedded devices use bootable non-volatile storage media with no moving parts to increase mean time between failure (MTBF) ratings and to improve overall system robustness. NT Embedded supports solid state disks and flash memory.

3.3.1 DiskOnChip Driver

The M-Systems' DiskOnChip has become a popular device for many embedded systems. Almost all of the single board computer manufacturers support DiskOnChip. Windows NT Embedded supplies the DiskOnChip driver to allow read-write access for DiskOnChip 2000 up to 144MB.

3.3.2 ATA (PCMCIA, PC Card) Flash Disks

NT can boot from ATA flash disks and Compact flash cards as long as the system's BIOS makes these devices look like regular hard drives.

3.4 Remote Management

In an embedded system, access to the operating system and/or application is often through a remote system of some type, since a user interface on the device itself may not exist or may not be practical to access. Windows NT Embedded provides both character-mode and graphical management solutions using serial, dial-up, and network connections.

3.4.1 Serial/Modem Remote Management

The first of the character mode accesses, Serial/Modem Remote management allows for the connection over a telephone line or through a direct cable connect. Hyperterm can be used as the interface to the remote system. Files and other basic administration functions can be performed.

3.4.2 Graphical Remote Management

Microsoft supplies NetMeeting 3.0 within NT Embedded. CD 1 contains a full release. You can connect to a remote system over a Network-TCP/IP connection, and take control of the remote desktop. Graphical Remote Management allows you to operate a remote system as if you were in the same room. Transfer of files, installing files, and administration can all be performed.

3.4.3 Telnet

Telnet is a popular method to access remote systems. Windows NT Embedded provides a simple Telnet host. You can connect remotely, and use FTP to transfer files. FTP client is supplied in the TCP/IP utilities component.

3.4.4 Shutdown Utility

The shutdown utility is one of the most popular utilities taken from the NT resource kits. The shutdown utility allows you to remotely shut down and/or reboot a headless system. The help for the shutdown utility shows the following parameters:

Usage: SHUTDOWN [/?] [\\Computer] [/L] [/A] [/R] [/T:xx] ["Msg"] [/Y] [/C]

/?	Shows this screen.
\\Computer	Specifies a remote computer to shut down.
/L	Specifies a local shutdown.
/A	Aborts a system shutdown. This is only possible during the timeout period. If this switch is used, all others are ignored.
/R	Specifies that the machine should reboot after shutdown.
/T:xx	Sets the timer for system shutdown in seconds (20 sec. default).
"Msg"	Specifies an additional message
/Y	Answer all following questions with "yes"
/C	Forces running applications to close.

ATTENTION: If you use the /C parameter NT ignores the applications option to save data which may have changed. You will see no File-Save dialog box, because NT will force the application to close. This will result in a loss of all data not previously saved !!!

3.5 Error Recovery

Robustness and fault-resilience have always been design criteria for Windows NT. For example, Windows NT possesses characteristics such as protected virtual memory and

object-based security. Many embedded applications have more extensive error recovery requirements since they are often used in 24/7 applications in which operator intervention and assistance is not practical. Windows NT Embedded solves the issues associated with "blue screens," and system messages normally displayed on the console that require user interaction. Not only can these errors be logged, but they can be used to trigger application software events.

3.5.1 System Message Interception

Though headless, the NULL video driver is still processing graphical information. If a computer that uses DHCP client to get a TCP/IP address is cut off from the network, a dialog box appears to notify the user. In a headless system, the user is unaware of the box, and has no way to close it. Redirecting error dialogs to a text log is ideal for headless systems. NT Embedded includes a system message intercept component that can respond to system messages.

3.5.2 Blue Screen

Notice that when you shut down an NT target the system reboots. The auto-rebooting feature allows systems to recover if a fault occurs. If a configuration built by Target Designer fails, the system reboots by default. You can edit the AutoReboot registry value to display a Stop error, commonly called "blue screen."

You will need to edit the auto-reboot registry value in order to prevent the auto-reboot from occurring.

- Select the configuration that you want to change.
- On the toolbar, click the **Registry View** button.
- Locate the registry path: HKLM\System\ControlSet001\Control\CrashControl.
- Right click the **AutoReboot** value and select modify.
- In the **DWORD Editor** dialog box, change the setting from 1 (reboot on failure) to 0 (stop responding or "hang" on failure).

Note: When set to 0 (stop responding on failure), a failing target displays a Blue Screen (aka Blue Screen of Death - BSOD) and does not reboot.

3.6 SID Cloning

Each system must have a unique Security ID if it is going to be on a network. The SID is created from the Product Identification Number that identifies a licensed copy of Windows NT and other values within the system. Windows NT Embedded includes a low-volume cloning solution to change the SID for each image created by Target Designer.

The SID cloning utility will change the SID for the platform upon the first boot. The HKEY_USERS key can be used to verify that the SID has changed from one copy of the image to another. The image must be cloned to create individual images for read only boot partitions (i.e., El Torito, write filter boot partitions).

3.7 Exercise 2 - Headless System

Microsoft has added features to help OEMs run Windows NT in non-traditional desktop systems. Being able to build a system without a VGA card or mouse/keyboard support is a key feature for many embedded developers. Microsoft has created NULL-VGA, NULL-Mouse, and NULL-Keyboard drivers to allow you to build headless NT systems. In this exercise, we will build a headless system and connect to the system via Serial Remote Administration and Hyperterm. The exercise will also have a reduced footprint with the selection of the Minimal OS and No Page File.

Keep the keyboard, mouse, and video display attached to the target platform. Even though this is a headless image, the goal is to prove that the system is truly headless by seeing what comes out on the display and what you can or cannot type in on the keyboard.

3.7.1 Build Image

1. Add a new configuration under the NTE Class collection.
2. Rename the configuration to Exercise2.
3. Select the following components. (This time nothing is in bold, walk the tree from the top down):

Capability	Component	Size	Check Box
System / Platform / SystemRole	Workstation System	0	
System / Platform / User Identification	Automatic Login	0	
System / Platform / Core OS	Minimal	7.87MB	
System / Platform / HAL	Standard HAL	958KB	
System / Platform / Virtual Memory	No Page File	0MB	
System / Devices / Storage / Floppy Disk	Floppy	18KB	

System / Devices / Storage / Fixed Disk	EIDE	0	
System / Devices / Storage / Fixed Disk	SCSI Disk	15KB	
System / Devices / Storage / CD-ROM	EIDE CD-ROM	0	
System / Devices / Storage / CD-ROM	SCSI CD-ROM	22KB	
System / Devices / Storage / File Systems	FAT	139KB	
System / Devices / Display / Display Drivers	Null VGA	99KB	
System / Devices / Keyboard / Keyboard Layout	US Keyboard Layout	9KB	
System / Devices / Keyboard / Keyboard Drivers	Null Keyboard Driver	4KB	
System / Devices / Keyboard / Keyboard Input Local	English (US) Input Locale	7KB	
System / Devices / Mouse	Null Mouse	0	
System / Devices / BUS Devices SCSI / SCSI Adapters	EIDE SCSI Adapter	0	
System / Devices / BUS Devices SCSI / SCSI Drivers	EIDE SCSI Driver	27KB	
System / Devices / Serial Ports	COM1	0	
System / Management / Management Applications / Remote Administration	Serial Console Administration	143KB	
System / Management / Management Services	Event Logging	377KB	
System / Shared System Components	Console Administrator	56KB	

System / Shared System Components	Display	25KB	
System / Shared System Components	Keyboard Drivers	9KB	
System / Shared System Components	Keyboard Locale	135KB	
System / Shared System Components	Mouse Common	9KB	
System / Shared System Components	SCSI	58KB	
System / Shared System Components	Serial Ports	44KB	
System / Shared System Components	Virtual Memory	0	
Applications / Shell	Command Shell	0	
Applications / Utilities	Windows NT Command Prompt	262KB	

4. View the Properties of the Serial Admin Component. Make sure that the properties are set to the following:

Port should be set to 1.
 Baud Rate: 19200
 Data bits: 8
 Parity: None
 Stop Bits: 1

5. Run a dependency check on the configuration. If there are any missing dependencies or conflicts, double check the list above with the components selected.
6. Build the image. A dialog box will indicate that the x:\EmbeddedOS directory is not empty. Click YES to process and erase old contents.
7. Once the image is built, use a download method to download the new OS to the target. See Exercise 1b for the Remote Recover method.

When the target boots it will stop at the NTDETECT blue screen. The NULL video driver is loaded, but video data is not going to the monitor. A mouse icon is not available so the mouse will not work. Try typing on the keyboard and verify that nothing happens. Now reboot the target without the video card, keyboard, and mouse attached (your BIOS should be able to disable keyboard and video checking at POST).

3.7.2 Prepare HyperTerminal on the Host Machine

8. Make sure that the NULL modem cable is connected between your host system and your target (COM1).
9. In your host machine, open a HyperTerminal session and setup the session. When the New Connection Dialog pops-up, call the session NTE Serial Connect.
10. In the “Connect to” dialog, make sure that the “connect using” is set to COM1 (or COM2).
11. Set the COM port settings to match the above setting for Serial Remote Administration.

When the session is open and connected, you will be asked for a user name and password. The User and Password are case sensitive.

User name: **Administrator**
Domain:
Password: **Administrator**



Figure 3.2: Logging into Remote Target



Figure 3.3: Welcome Screen – Login successful

3.7.3 File Transfers

Once you have a connection, you can perform DOS commands to view directories and copy and delete files. You can also copy files from one machine to another. HyperTerminal can send files to other terminals, using various protocols. The Serial Remote Administration Service implements ZMODEM protocol. To specify the protocol HyperTerminal uses for sending or receiving files, choose ZMODEM as the current protocol.

Sending Files from Host to Target Machine Using HYPER TERMINAL

To send a file from the host to the target machine, use “Transfer/Send File” from the menu or press the *send* icon. You can specify a file to send. As the file is sent to the target, a send progress dialog will be opened.

You can also cancel sending a file in the middle. The file will be saved in the current directory (of the *cmd.exe* shell) and will **replace** any existing file on the target machine.

For the *cmd.exe* shell to be ready to receive files, the line buffer should be empty and no application should be running in the console of the *cmd.exe* shell. (The HyperTerminal actually activates *rz.exe* in the *cmd.exe* shell before starting file transfer. The *rz.exe* utility is the utility on the target responsible for file receiving.)

If the send process continues uninterrupted and no error is reported, the file is guaranteed to reach its destination correctly (by CRC). If a file transfer is interrupted, a partial file may reside in the target machine. The file name will be the same as in the host

machine. (Path is stripped from the file.) More than one file can be sent this way (using wildcards).

Sending a File from Target to Host Machine - SZ.EXE Utility

Before sending a file from the target to the host machine, define in the HyperTerminal the target directory on the host for the files to arrive at (Transfer/Receive File on the menu). All received files will be saved in this directory. Files received will not replace existing files (instead a Number will be attached to the file name). HyperTerminal ignores the file path.

Activate the *sz.exe* utility to send a file by entering the following:

```
Sz.exe /s<FileName>
```

HyperTerminal will notice that a file is being sent and will open a file transfer progress Dialog. The file transmission can be halted at any time by pressing the Cancel button. File transmission is protected by CRC and the file is guaranteed to arrive correctly from target to host if no error is reported.

Sz.exe utility options:

Sz.exe has many options. A brief description follows ([] indicates default values):

Option	Description
/sSourceFileName	Source file name (can include a full path). No Space between the 's' and the file name.
/tTargetFileName [SourceFileName]	Name of the sent file on host (HyperTerminal will ignore the path).
/xEscapeControlCodes : + or [-]	Specify whether binary chars will be sent within the ZMODEM stream.
/cCrcType: [32] or 16	Specify the type of CRC used for transfer (32 bit or 16 bit CRC).
/oOverWrite:	Specify the file overwrite behavior for HyperTerminal to use.
/iIncludePath : [+] or -	Specify if path is to be sent to HyperTerminal (HyperTerminal will ignore path information).
/eEolConvert: + or [-]	Specify if Eol characters are to be converted before being sent.
/mXferMethod : WINDOW or [STREAMING]	Specify the transfer method of data packets (STREAMING has better throughput but WINDOW is better for noisy lines).

3.7.4 Transferring a File

12. Make sure that you are connected to your target system.

13. Create a text file using NotePad or WordPad on the desktop. Save the file to your Host's C drive root.
14. In HyperTerminal, select Send File from the Transfer menu.

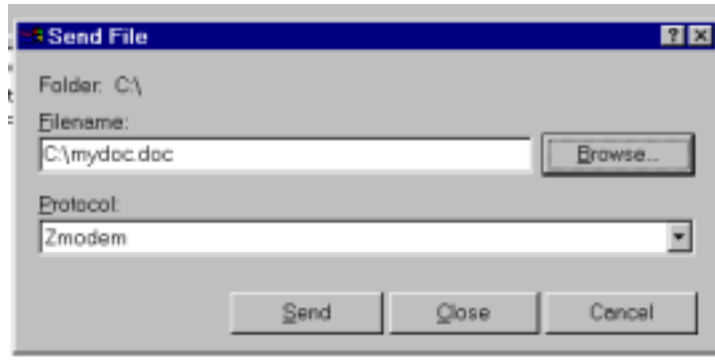


Figure 3.4: Transfer File from Host to Target

15. Locate and send the text file you created in Step 2 to your target machine. You will see a transfer dialog appear indicating the status of the transfer. The file will be located in the target's C drive root.
16. Now let's transfer the file back to your host machine. In the command window type the following:
`C:>sz /smydoc.doc` (where mydoc.doc is the name of your text file) Once you hit return the file transfer will begin.

```
EMBEDDED FILE TRANSMITTER

/sSourceFileName
/tTargetFileName [SourceFileName]
/aAspectsControlCodes : + or [-]
/cCrcType : [32] or 16
/oOverwrite :
    NONE      None
    N_L      Never or longer
    CRC      CRC differs
    APPEND    Append to file
    [ALWAYS] Overwrite always
    L_D      Length or date differs
    NEVER    Never overwrite
/iIncludePath : [+] or -
/eMailCommand : + or [-]
/mXferMethod : WINDOW or [STREAMING]

C:\>az /mydoc.doc

C:\>
```

Figure 3.5: Transfer File back to Host

3.8 Exercise 3 – Embedded Features: DiskOnChip, Command Shell, Shutdown Utility, and the Write Filter

In the last exercise, you experimented with a few of the embedded features that Windows NT Embedded provides. In this exercise, we will continue to explore a few more features.

M-Systems' DiskOnChip is a popular device supported by many Embedded PC manufacturers. Flash technology has taken off dramatically over the past few years, resulting in small flash disks that are economical, viable, and robust enough for most embedded systems. OEMs are concerned about hard drive corruption, which can happen if a system is unsafely shut down. Windows NT can boot from a read-only partition. The write filter component directs all writes to a RAM disk, thus protecting a boot partition from data corruption.

Windows NT allows you to replace the explorer shell with an alternative shell. NT Embedded also includes some of the popular utilities that are found in the resource kits. The shutdown utility is very helpful in headless systems, allowing remote administrators to connect over a modem, access a remote system, and shutdown/restart a system.

In this exercise, we will implement a combination of these embedded components. The DiskOnChip will be write-protected to prove that drives can be made read-only. The command shell will be used to run a batch file to start other NT applications, and to shut down the system when the last application is closed.

3.8.1 Hardware for Exercise

The AAEON 5894 has support for the DiskOnChip 2000. You may choose to use another platform as long as the DiskOnChip is supported.

3.8.2 Building Image

17. Add a new configuration under the NTE Class collection.
18. Rename the configuration to Exercise3.
19. Select the following components:

Capability	Component	Size	Check box
System / Platform / SystemRole	Workstation System	0	
System / Platform / User Identification	Automatic Login	0	
System / Platform / Core OS	Minimal	7.87MB	
System / Platform / HAL	Standard HAL	958KB	
System / Platform / Virtual Memory	No Page File	0MB	
System / Devices / Storage / Floppy Disk	Floppy	18KB	
System / Devices / Storage / Fixed Disk	DiskOnChip	31KB	
System / Devices / Storage / Fixed Disk	EIDE Disk	0	
System / Devices / Storage / Fixed Disk	SCSI Disk	15KB	
System / Devices / Storage / File Systems	FAT	139KB	
System / Devices / Storage / Storage Filter Drivers	Write Filter	17KB	
System / Devices / Display / Display Drivers	VGA	10KB	
System / Devices / Keyboard / Keyboard	US Keyboard Layout	9KB	

Layout			
System / Devices / Keyboard / Keyboard Drivers	PC/AT Keyboard	29KB	
System / Devices / Keyboard / Keyboard Input Local	English (US) Input Locale	7KB	
System / Devices / Mouse	Logitech	29KB	
System / Devices / BUS Devices SCSI / SCSI Adapters	EIDE SCSI Adapter	0	
System / Devices / BUS Devices SCSI / SCSI Drivers	EIDE SCSI Driver	27KB	
System / Devices / Serial Ports	COM1	0	
System / Shared System Components	Display	25KB	
System / Shared System Components	Keyboard Drivers	9KB	
System / Shared System Components	Keyboard Locale	135KB	
System / Shared System Components	Mouse Common	9KB	
System / Shared System Components	SCSI	58KB	
System / Shared System Components	Serial Ports	44KB	
System / Shared System Components	Virtual Memory	0	
Applications / Shell	Command Shell	0	
Applications / Utilities	Windows NT Command Prompt	262KB	
Applications / Utilities	Shutdown Utility	34KB	
Applications / Accessories	Calculator	124KB	
Applications / Accessories	Notepad	225KB	

20. Create a Batch file at the root of your host's C:\ drive and call it **MYBATCH.BAT**.

21. Edit the MYBATCH.BAT file to contain the following:

```
Echo off  
CLS  
Echo Hello World  
Start calc.exe  
Start /wait notepad.exe  
Shutdown /L /Y /C
```

The Start command will launch applications Calculator and Notepad. The WAIT option pauses the batch file from continuing until Notepad is closed. The system will shut down within 20 seconds.

22. In Target Designer click on the Exercise3 configuration, and select files view.

23. Right click in the files view pane and select Add File.

24. Use the browse button, and select the Mybatch.bat file.

25. In the add file dialog, set the target directory to \.



Figure 3.6: Add a File to the Configuration within Target Designer

26. Click OK, and the file will be added to the list in a different color.

Target Name	Target Path	Size	Source Name	Source Path
mybatch.bat		32	mybatch.bat	C:
ntdetect.com		26816	ntdetect.com	\j386
ntldr		156496	ntldr	\j386
vga850.fon	\winnt\fonts	5232	vga850.fon	\j386

Figure 3.7: File Added Listed in a Different Color

27. Change the properties of the command shell component:
 Cmd.exe /k c:\mybatch.bat runs the script and keeps command window open.



Figure 3.8: Running a Batch File upon Boot-up

28. Right click on the Write Filter component and select properties.
29. Add the DiskOnChip driver to the list of read-only drives:
\Device\Harddisk0\Partition1

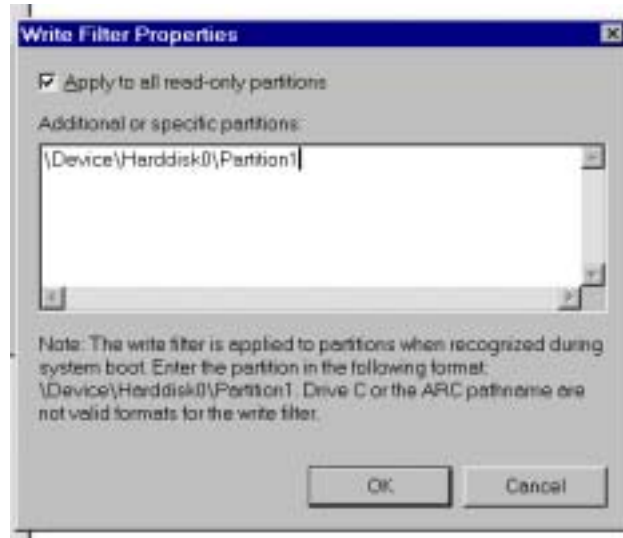


Figure 3.9: Adding a Partition to the Write Filter

Warning: Make sure that you spell the partition format information correctly. Capital letters are a must. The write-filter and Target Designer will not catch format errors.

30. Run a dependency check on the configuration. If there are any missing dependencies or conflicts, double check the list above with the components selected.
31. Build the image. A dialog box will indicate that the x:\EmbeddedOS directory is not empty. Click YES to process and erase old contents.
32. Once the image is built, use a download method to download the new OS to the target's hard drive, and copy MYBATCHE.BAT to the DiskOnChip. See Exercise 1b for the Remote Recover method.

3.8.3 *Running the Image*

When the system boots, Hello World will echo to the command window screen, and Calculator and Notepad applications will run.

33. Create a file using notepad and save the file to the DiskOnChip. Notice that the file can be re-opened and moved around like any file, until you shut down the system.

34. Close Notepad and watch the system shut down.
35. Reboot the target.
36. Try to open the file using note pad. Notice that the file is not on the DiskOnChip.

The write filter can also be used on hard drive partitions to protect sensitive data from being overwritten.

Running a batch file from the command shell is handy for a headless system. The batch file can be used to start the primary application, or you can use a batch file to run the NET.EXE utility to share printer and file resources on a network.

3.9 Summary

The embedded features added to Windows NT allow it to reside in non-desktop systems. NTE can exist as a headless system with a minimal footprint. Exercise 3 introduced a method to run applications from the command line in batch mode. Later exercises will introduce other embedded features such as graphical remote administration and CD-boot.

