MEDC

Microsoft®
Mobile & Embedded
DevCon 2006

Windows XP Embedded Boot Options (Melbourne)

David Jones
RMIT University
Melbourne, Victoria, Australia
300: Developing Windows Embedded Devices

Agenda

- Microsoft Windows XP Embedded Features
- Creating an XP Embedded Image
- Boot up Overview
- Windows XP Embedded Boot Options
- Booting from:
  - Hard drive
  - Compact Flash
  - Other devices:
    - Solid-state drives
    - CD-ROM
    - Network
    - Hibernated state
- Boot Performance
- Conclusions

Windows XP Embedded Boot Options
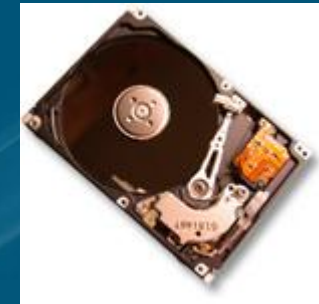
Can boot from a variety of media

...cluding diskless systems

As well as CF and other Flash devices

And even USB

Windows XP Embedded Features

Key features of Windows XP Embedded

- Componentized version of Windows XP Professional
  - Same binaries as Windows XP Professional
  - Fully compatible protocols
  - Support for all Windows Device Drivers
    - Without modification/wrappers
  - Runs desktop Windows XP applications
    - Subject to resources, without modification
    - Full Win32 and Microsoft .NET API
      - .NET2
  - Embedded boot options
    - For example, from Compact Flash
  - Windows XP Embedded Service Pack 2: Security and boot options

Windows XP Embedded Service Pack 2: Changes to the Embedded-enabling Features

Feature or area	Description
Windows application compatibility macro components	Increase application compatibility between your run-time image and applications in areas of multimedia, networking, shell, Windows Core, and Windows Management Instrumentation (WMI).
Generic Device Driver Support component	Quickly add support for one or more device classes to your run-time image.
Enhanced Write Filter (EWF)	**Reduce boot time for your EWF-protected run-time image by using a hibernation file.**
Minlogon	Implement a single-user logon environment that supports standby and hibernation.
Windows XP Embedded documentation	Read the latest Help documentation for expanded information about security and servicing, as well as more how-to topics. Component Help is now available for every component in the Windows Embedded Studio component database and includes detailed information about dependencies, resources, and interfaces.
New deployment options	**– Hibernate Once Resume Many (HORM)** **– Remote Boot.**

Some Windows XP Embedded Scenarios

- Robust diskless system but want to use existing Windows XP–supported hardware peripherals
  - Windows XP Embedded uses same peripherals as desktop Windows XP
  - Windows XP Embedded can be booted from solid state devices. (For example, CF.)
- Stateless system (always boots to the same state) with quick boot
  - Windows XP Embedded with CF and EWF (RAM Reg mode)
- System of many stateless and diskless webpads
  - Windows XP Embedded with remote boot
- Embedded device that runs desktop .NET applications and services
  - Windows XP Embedded runs same apps as desktop Windows XP

Creating an XP Embedded Image

Microsoft Windows Embedded Tools–1

- Target Analyser
  - Interrogates target system for devices
    - Ta.exe, runs under DOS
    - Tap.exe, runs under XP
      - Can run on existing desktop XP installation (best)
      - Can run from PXE boot using first XPE CD
- Component Designer
  - Imports output from target analyser to create platform component
  - Can also import XP device installation files
- Component Database
  - Stored in SQL 2000/5

Microsoft Windows Embedded Tools-2

- Target Designer
  1. Specify image
     - Which platform
     - Choose a macro
     - Add other devices
     - Fine tune settings
  2. Resolve dependencies
     - Can auto-resolve
     - Repeat until all OK.
  3. Build Image
     - Generates image within file structure for deployment
     - Not a bound single file like nk.bin (CE)
       - Can create a single bound image file (.sdi)
         - Eg. Use for network boot

MEDC Microsoft Mobile & Embedded DevCon 2006

Deployment

1. Suitably prepare target media
   - Create partition, set as active partition, reboot and format
2. Copy XPE build files to media
   - Must copy hidden/system files as well
   - Can insert media into dev. machine and copy, eg. CF
   - Can do network copy
3. Run First Boot Agent (FBA)
   - Completes the run-time image build process by running a sequence of tasks on the target system
     - Eg. Plug and Play device detection, security installation, and network configuration.
   - Can add commands to FBA and specify when they will run during the FBA process

Creating an Windows XP Embedded Image

demo

David Jones
RMIT University
Melbourne, Victoria, Australia
300: Developing Windows Embedded Devices

A quick look at the Microsoft Windows Embedded Tools

Overview of Boot Up

Boot Devices

- Need to read boot information and code
- Need to read operating system configuration and code
- Both may be same media or different

- Electromechanical
  - Floppy
  - Hard drive
- Solid State
  - Microdrive
  - DiskOnChip
- Network
  - Remote boot

- Flash
  - Compact Flash
  - USB
    - Memory Stick
    - uDoc
- CD-ROM
  - El Torito

Classifications of Boot Media

- Read/Write vs. Read-only media
  - Typical storage is read/write. For example, hard drive.
  - For Flash ROM/CF with limited write cycles better to use as read-only, except for updates and occasional save state
- Fixed vs. Removable
  - Can the OS storage be removed?
  - Physically/electrically
- Stateless vs. Stateful
  - Is it a requirement that the device boots to the same pristine state? (Stateless)
    That is, discard transient state at shutdown
- Local vs. Remote media
  - OS on local storage
  - OR get it from network server
    - Need network ROM

Read-write and Read-only Media

- Read/write
  - Hard drive
  - Embedded Disk
  - uDoc
  - (Compact Flash)
- Read-only Media
  - Compact Flash
  - CD-ROM
  - Network

Removable Vs. Fixed Media

- Fixed Media:
  - Hard drive, solid-state hard drive
- Removable media
  - CF, USB Memory
- Removable media
  - Can't be partitioned (with Windows tools)
  - Can't be made active partition disk
  - Hard to get to boot
- CF can be marked as fixed and used as fixed in True IDE mode.

Booting with Windows XP Embedded

Windows XP Embedded Boot Modes

- Hard drive
  - Read/write
  - Overlays
  - HORM
- .OR Diskless:
  - Solid State Devices
    - Solid State Disk
    - Flash ROM
    - Compact Flash
    - USB
  - Bootable CD
  - Network

Booting With Windows XP Embedded in 5 Seconds

- Media needs to be configured to boot using boot.ini file
  - Boot partition needs to be marked as active
  - Boot.ini points to boot partition and OS directory
- Need operating systems files on boot media
  - Not a single binary file like nk.bin with Windows CE*
    - That is, no boot loader program
  - Created with directory structure in Microsoft Windows Embedded System build process
  - Can do xcopy of files (/s /h) to media if it can be placed on development system
    - * Or use SDI: Bind files into one file that then can be deployed and unpacked
- System needs to be configured (that is, in BIOS) to boot from that media

A Typical Boot.ini File

```
[boot loader]
timeout=0
default=multi(0)disk(0)rdisk(0)partition(1)\WINDOWS
[operating systems]
multi(0)disk(0)rdisk(0)partition(1)\WINDOWS="Microsoft
Windows XP Embedded" /fastdetect
```

- This will boot Windows XP Embedded from partition 1 on hard drive(media) 0 with OS files in \Windows

Hard Drive: Read-write Mode

- Can build a Windows XP Embedded system to boot from hard drive on a standard desktop system

- If it runs Windows XP it will boot Windows XP Embedded

- Can multi-boot system to desktop Windows XP as well as Windows XP Embedded
  - TAP interrogation of system can be more thorough when done on an existing Windows XP desktop

- Can build an Windows XP Embedded system in read-write HD mode on Virtual PC and VMWare emulators.

- Can allow stateful system
  - All changes are preserved between boots

- OR can use stateless
  - Disk Overlays (EWF Disk Mode)
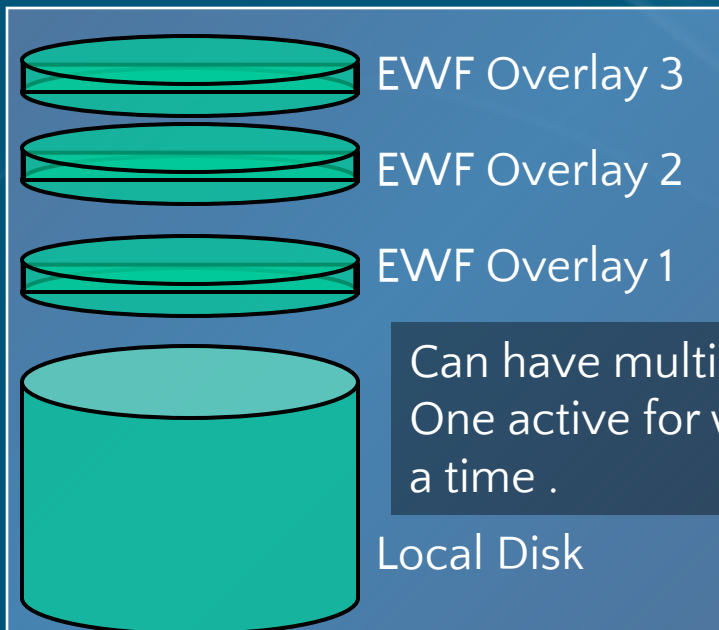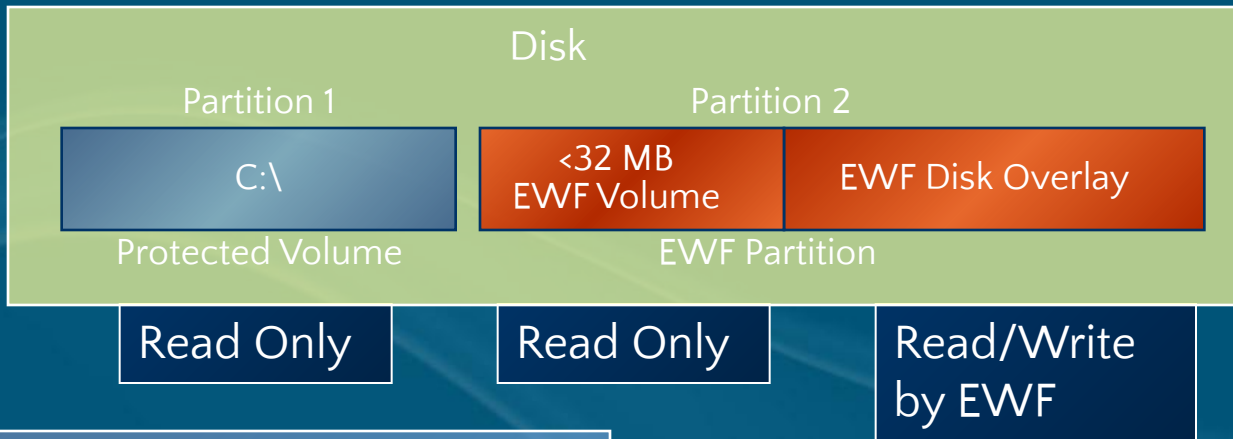
Enhanced Write Filter (EWF) Overview

- Protects a volume from write access
  - Enables boot from read-only media such as CDROM and flash
  - Can save write cycles to other media where desirable:
    - CF, Network
- EWF Overlay
  - Protects contents of volume from writes by redirecting to alternative media
    - Eg Read/write storage, RAM
  - Like a transparency overlay
  - Can be consolidated into the volume or discarded
- EWF Volume
  - Stores static EWF information about the volume being protected
  - Not the transient overlay data
  - Can be on protected volume, in registry

EWF Modes

Mode	Overlay Location	EWF Volume Location	Notes
Disk	On disk	On disk in unpartitioned space	Useful where state is required to maintained between boots
RAM	In RAM	On disk in unpartititioned space	Useful for "stateless" systems
RAM Reg	In RAM	In system registry	Useful where disk has only one partition

•Note: EWF Volume is created by First Boot Agent for first two modes.

Disk Overlays

Disk

Partition 1	Partition 2	
C:\	<32 MB EWF Volume	EWF Disk Overlay
Protected Volume	EWF Partition	

Read Only | **Read Only** | **Read/Write by EWF**
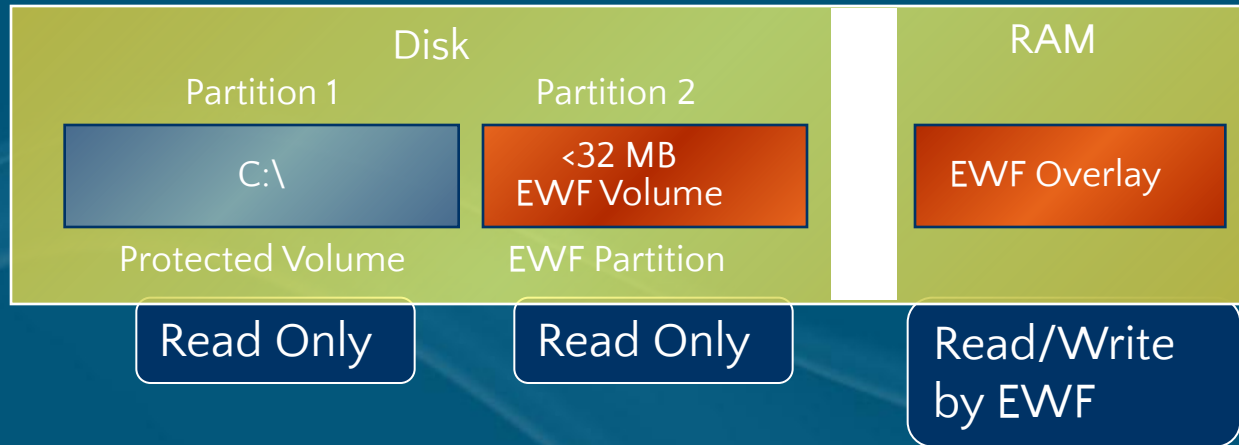
EWF Overlay 3

EWF Overlay 2

EWF Overlay 1
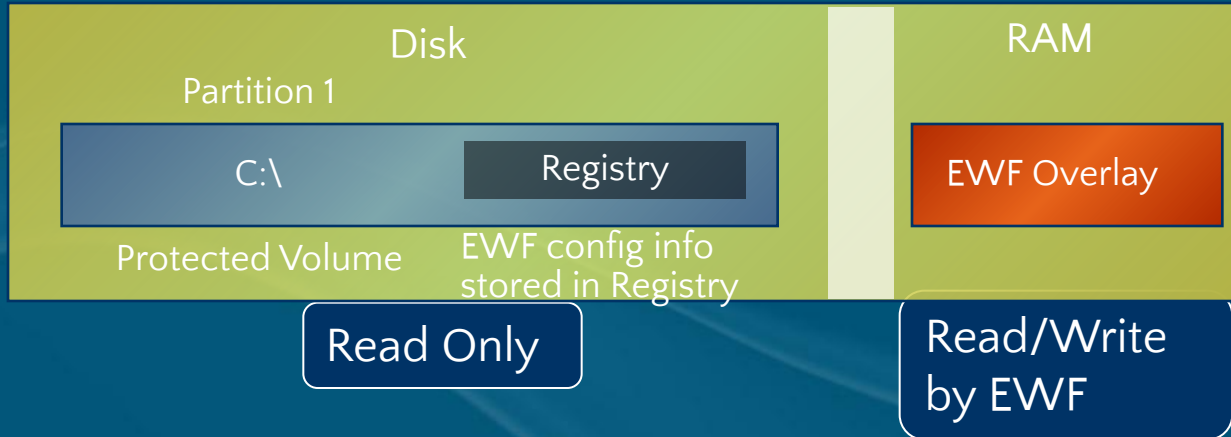
Local Disk

Can have multiple overlays. One active for write at a time .

Changes to C drive will appear to the OS as being to C drive but will actually be written to EWF Disk Overlay

RAM Overlay

Disk		RAM
Partition 1	Partition 2	
C:\	<32 MB EWF Volume	EWF Overlay
Protected Volume	EWF Partition	
Read Only	Read Only	Read/Write by EWF

- When system is shutdown, overlay in RAM is lost
- Useful for stateless system
- Can commit overlay to protected volume
- Disk must be partition-able
  - Not CF in removable mode, not CDROM
- Single overlay only

RAM Reg Overlay

Disk		RAM
Partition 1		
C:\	Registry	EWF Overlay
Protected Volume	EWF config info stored in Registry	

Read Only

Read/Write by EWF

- EWF configuration is skipped in FBA
- EWF is added to registry after FBA and enabled
- Useful:
  - Where media is not partition–able
  - For removable media such as CF, USB
  - Minimizing CF write cycles
- Can commit Overlay to disk if it is writeable

Note on EWF API

- There is an API for programmatic use of EWF overlays
- eg. Sample app:
- http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnXPesp1/html/EWFAPI.asp
- EWF API is part of Windows XP Embedded install:
  - Include EWF API Component in Windows XP Embedded build
  - Use EWFAPI.DLL, EWFAPI.LIB, EWFAPI.H in programming
  - EWFAPI.LIB on Windows XP Embedded SP2 is incorrect file
    - Update: (May 2003 Q818822):
- http://msdn.microsoft.com/embedded/downloads/xp/impQFE/default.aspx
- May also need Power Management API:
  - Shutdown, Restart, etc. functionality
  - XPEPM.LIB, XPEPM.DLL, XPEPM.H
  - In VALUEADD\MSFT\XPEPM folder on Disk 1

MEDC Microsoft Mobile & Embedded DevCon 2006

Booting Windows XP Embedded from a Hard Drive

demo

David Jones
RMIT University
Melbourne, Victoria, Australia
300: Developing Windows Embedded Devices

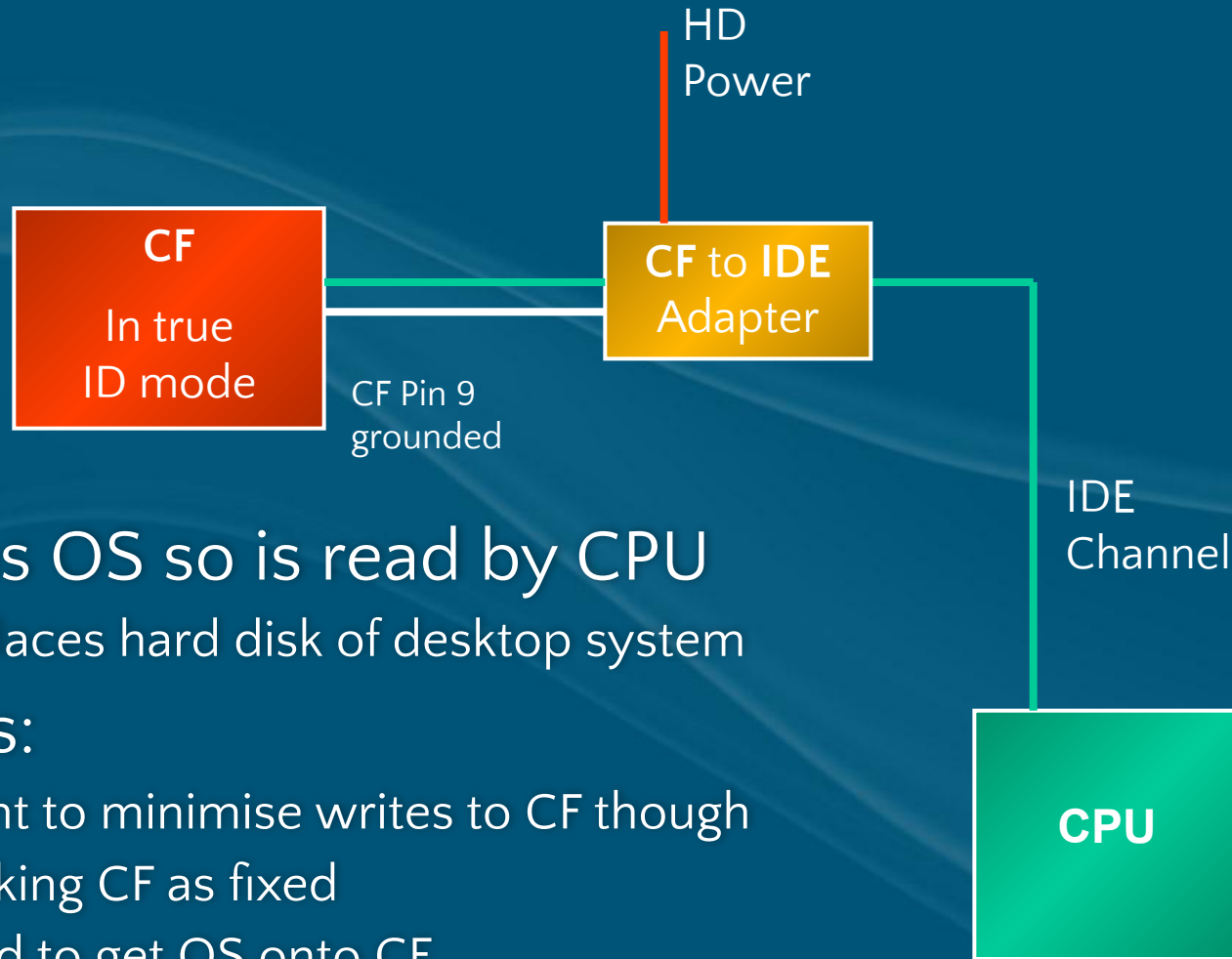*Demonstrate booting a preconfigured Windows XP Embedded from a hard drive:* with overlays.

CF Types

- Consumer Grade
  - Meant for use in consumer devices as removable bulk data storage
  - That is, cameras, PDAs, phones
  - 300K read-writes
- OEM
  - For use in embedded systems
  - Wear-levelling
    - Hardware algorithm than avoids continuous rewrites to same location.
- Industrial grade
  - Wider temperature ranges
- Consumer grade CF are less expensive

CF Issues

- Pros and Cons
  - 👍Solid state
    - 👍 Robust
  - 👍 Removable
  - 👍 Can be adapted for IDE (CF-IDE adapter)
    - 👎 May need to mark as non-removable in IDE mode
      - 👎 Need vendor utility
  - 👎Limited write cycles (300K)
    - 👍 Wear-levelling in (some ?) OEM/Industrial grade CF
  - 👎 Slower than diskonchip because of lack of DMA

- SanDisk
  - SanDisk is inventor of CF
  - They provide consumer grade and OEM CF
  - Not industry grade CF (Temperature range aspect)

- Flash Drives that boot Windows XP Embedded
  - http://www.seanliming.com/flashhelp.html

Using CF in an Embedded System

HD
Power

CF
In true
ID mode

CF Pin 9
grounded

CF to **IDE**
Adapter

IDE
Channel

CPU

- CF has OS so is read by CPU
  - Replaces hard disk of desktop system
- Issues:
  - Want to minimise writes to CF though
  - Marking CF as fixed
  - Need to get OS onto CF
    - USB CF Adapter

True IDE Mode: Fixed Disk Mode

- When pin 9 of CF is low it boot up the CF enters a true IDE mode
- With an adapter can be used in an IDE bus for storage (read/write)
- Windows XP will not allow low-level formatting with non-fixed disks
  - fdisk and diskpart don't recognise it
  - So can't mark it as the active partition/disk
  - So can't boot from it with Windows XP Embedded
- Need to mark the CF as fixed
  - Need a utility for that
  - Reversible
- Note: CompactFlash ORG requires CFs to be delivered in removable mode
- If using CF with single partition (RAM Reg mode) you may not need marked as fixed.

CF Scenarios

- Use an OEM/Industrial grade CF
  - Get utility from CF manufacturer to mark CF as fixed
    - Confidentiality issues
    - If using, Mode may not need to mark as fixed.
  - Implement as storage device for operating system
  - Implement enhanced write filter to minimise (or totally inhibit) writes to CF
    - Wear levelling also with OEM/Industrial grade CFs
  - Provide embedded systems with CFs

- Use Consumer grade CF
  - The utility will work with only some CFs
    - Even within same batch
  - OK for development/teaching purposes

Booting from CF with Windows XP Embedded RAM Reg Mode

- Hardware needs
  - CF-IDE adapter
  - CF marked as fixed
  - CF-USB adapter
- Key Windows XP Embedded Components
  - EWF Manager
  - Enhanced Write Filter
  - EWF NTLDR
  - EWF Registry keys
    - Create as XPe component
    - Next slide
- Other
  - Disable "Start EWF Enabled
  - Disable the FBA DLL/COM

- Prepare CF as bootable drive
  - Boot with Windows XP Embedded Disk 1
- Copy files to disk
  - Use CF-USB adapter on dev machine
  - Xcopy /s /h
    - Get hidden files
- Boot and let FBA run
- When finally booted run EWFMGR on device and see that EWF is running in RAM Reg mode

EWF on CF Registry keys

```
Windows Registry Editor Version 5.00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\ewf]
"ErrorControl"=dword:00000001
"Group"="System Bus Extender"
"Start"=dword:00000000
"Type"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Class\
{71A27CDD-812A-11D0-BEC7-08002BE2092F}]
"UpperFilters"="Ewf"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\ewf\
Parameters]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\ewf\
Parameters\Protected]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\ewf\
Parameters\Protected\Volume0]
"VolumeID"="{1EA414D1-6760-4625-8CBE-4F9F85A48E15}"
"Type"=dword:00000001
"ArcName"="multi(0)disk(0)rdisk(0)partition(1)"
```

Booting Windows XP Embedded from CF

demo

David Jones
RMIT University
Melbourne, Victoria, Australia
300: Developing Windows Embedded Devices

Boot a device in RAM Reg mode from CF

Booting from Other Devices

Embedded Disks

- Can have IDE interface
- Plug in in place of a hard drive
- More robust than HD
- Faster than CF
- Can be used as read-writeable HD
- No need for overlays
- No "set fixed mode" utility required
- Rewrite 1 million +
- 10 MB/s read/write transfer rate
- Lower capacity than CF

Solid State Disk Demos

- The previous Disk Overlay Demo used an EmbedDisk.

- The Hibernate Once Resume Many Demo also uses and EmbedDisk

USB

- Many embedded SBC only have USB 1.1
  - Not fast enough for boot process
  - Must have USB 2
- USB (2) Memory sticks are in not suitable XPe boot devices
  - USB is enabled late in boot process
  - USB devices are not fixed
- M-Systems solution:
  - uDoc

USB: M-Systems uDoc

- M-Systems uDoc (USB DiskOnChip)
  - Windows XP Embedded can be configured to boot f
    - Loads early in boot process
  - Can be connected to USB socket on board
  - No need for EWF
    - Wear levelling technology
  - http://www.m-sys.com/site/en-US/Products/M-Module/M-Module/Products_/uDiskOnChip.htm
  - Easy to transfer image files (USB host adapter for module)
  - USB 2.0 Boot must be supported in BIOS

- See: 300   Building Windows XP Embedded Devices for USB Boot

Windows XP Embedded CD-ROM Boot

- Use RAM Reg mode
- Develop El Torito Bootable CD-ROM
- CD-ROM appears to be writeable via overlay in RAM
- Configure system to boot from CDROM
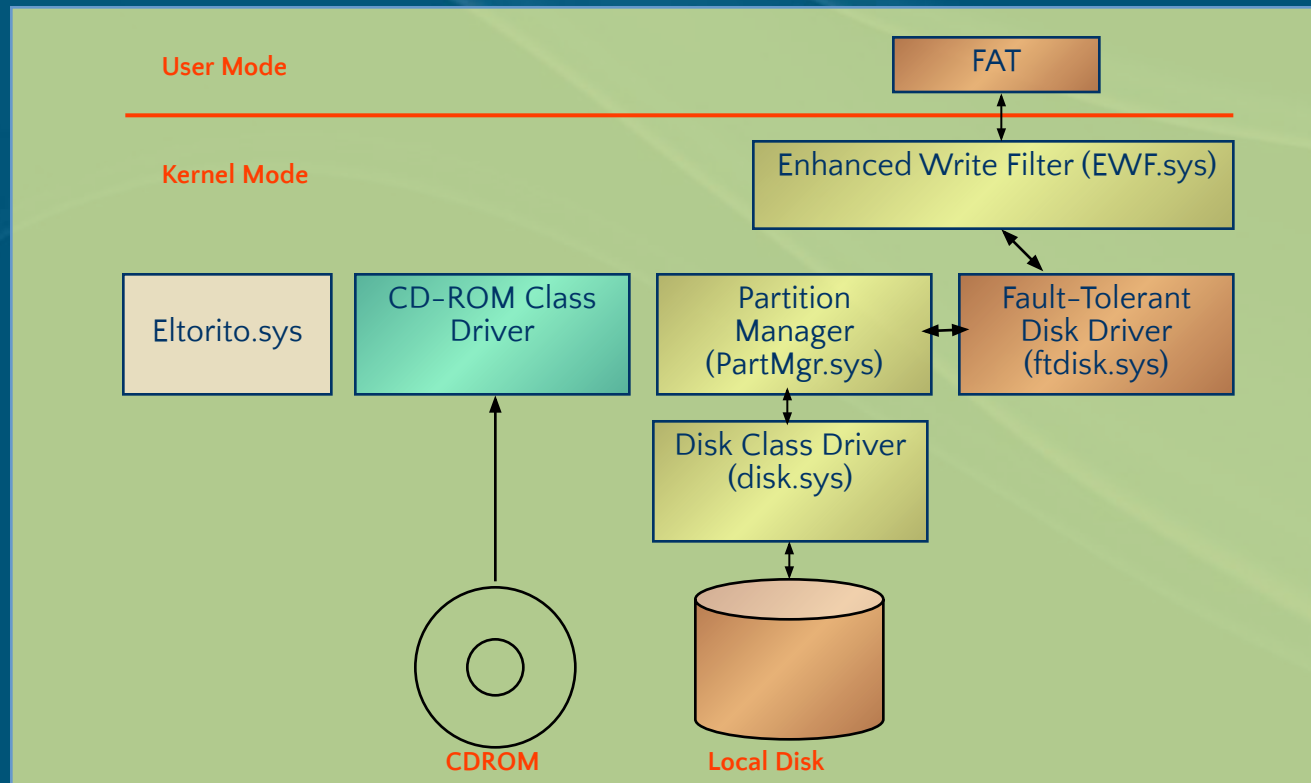
Booting From CD-ROM

- Slower boot than solid state
- Easy deployment/update to multiple systems
- System integrity
- Can use hard disk-less system
- General slow depending upon image size and CD-ROM speed
- Useful as a system tool

CD-ROM Boot: How To

- Create image
  - Need EWF components
    - NTLDR, Registry, Management console
  - Need CDFS
  - Need El Torito Component
  - Some others needed
- Place HD and CDROM in IDE channel 1
- FBA on target as <500M hard drive partition
  - Need an El-Torito CD in drive during FBA
- Have 2nd partition for ISO image (900M)

- Swap the HD and CDROM drive letters
  - Run ETPrep
- Generate Boot CD ISO
  - Reboot with Windows XP Embedded CD1 into PXE
  - Create ISO file
  - Run HD2ISO
  - Copy ISO file to development machine
  - Burn CD
- Remove HD
- Place CD in target and boot

Booting Windows XP Embedded from CD-ROM

demo

David Jones
RMIT University
Melbourne, Victoria, Australia
300: Developing Windows Embedded Devices

Using a pre-existing image

Windows XP Embedded HORM Boot

- Hibernate Once Resume Many
  - Resume from same hibernated state
  - Quick boot
  - Stateless system
  - Normally when Windows boots, the state information in the hibernation file is deleted by zeroing out the first page of the hibernation file
  - However, by using EWF, you can persist that state information from boot to boot
  - Reuse the same hibernated state

Booting from HORM

- Fast boot of system to known state
- Easily extended/updated
    - Disable HORM
    - Reboot and add changes
    - Recapture
- Adds robustness to system
- Adds security
- Use EWF with Hibernation:
    - Develop EWF system
    - Boot
    - Run required applications without closing
    - Hibernate
    - Persist same hibernation state after boot
        - Normally deleted
    - Volume remains protected

HORM Boot: How to

- Create image. Needs
  - EWF:
    - RAM Overlay Mode, API, NTLDR, Management Console
  - Power Management
  - Computers
    - One of: ACPI PC, Standard PC, ACPI Uniprocessor
    - Setting: Enable hibernation
  - Must have target's specific video driver
- Build image
- Create and place a file:
  - resmany.dat in root
  - Tells EWF that this is a HORM setup
    - Enables reuse of same hiberfile.sys file.

- Check that hiberfile.sys is in root
  - dir /Ahs c:\hiberfil.sys
- Use EWFMGR and XPEPM to manage HORM:
  - Enable EWF
    - EWFMGR c: –enable
  - To see status
    - EWFMGR c :
  - Hibernate system and enable HORM
    - XPEPM –Hibernate
  - Reboot system from hibernation
    - Can shutdown and restart to same state.
    - XPEPM –shutdown / –restart
  - Can commit changes to HD
    - EWFMGR c: –commit
  - Can commit and stop HORM
    - EWFMGR c: –commitanddisable –live

HORM: Key Issue

- This uses Power Management component
- That only works if you have the correct hardware drivers
  - Some things are missed by TAP via PXE
  - Can get by with other boot configurations but not with Power Management
- Need the correct Windows XP Video driver
- Need various BUS drivers
  - Ref: "Troubleshooting Windows XP Embedded's blue screen "Stop 0x0000007B" error", Sean Liming

    http://www.windowsfordevices.com/articles/AT6380158626.html

MEDC Microsoft® Mobile & Embedded DevCon 2006

Windows XP Embedded (REMOTE) Network Boot

- Boot with device PXE
  - Preboot eXecution Environment
- Image downloaded to RAM from PXE server
  - Uses TFTP
- OS then boots from RAM Disk
- No persist storage required:
  - Diskless system
- Requirements:
  - About double normal RAM
  - PXE compatible BIOS and Network card
  - Remote boot PXE server
    - DHCP, PXE, TFTP

Remote Boot

- Create a Windows XP Embedded image file (post FBA)

- Deploy that to a server
  - Install RBS on Windows Server from Windows XP Embedded disk 2
  - With DHCP

- Target PXE (network) boots.
  - On getting DHCP response RBS uses MAC address to determine which image to return.

- Note: Needs RAM 2 x image size
  - Image file gets downloaded to ram and system is generated from there.

Remote Boot: How to

- Create image and run FBA
  - No special components
  - Use HD system on target
- Create .SDI file from booted image
- Install Microsoft Windows Embedded Remote Boot Manager
  - On a Windows 2003 Server
  - With DHCP on the Server
  - Some Remote Boot Server–DHCP configuring
- Configure boot
  - Place .SDI file on RBS
  - Configure PXE boot based upon MAC address
- Enable Network boot on target and boot

demo

David Jones

RMIT University

Remote Booting of a Windows XP Embedded image

demo

David Jones
RMIT University
Melbourne, Victoria, Australia
300: Developing Windows Embedded Devices

HORM: Hibernate Once Read Many

Boot Performance

Usability

- How much boot time will users suffer?
  - CF boots in about 1 minute
  - Other non CD boots < 1 minute
    - Network would depend upon many factors
- Generally up to one minute to boot is suitable for most scenarios
  - For example, POS at start of day.
  - If stateless system goes down, can bring back up in pristine state within one minute
- Enable Standby for more frequent power downs

Discussion:

Collation of Boot Times
from Demos as Above

Conclusions

- Windows XP Embedded can be booted from a variety of media

- Besides using a standard hard drive, it can be booted from a CD-ROM and solid state media such as Compact Flash and Flash drives

  - Solid state boot media make the system physically more robust

- Windows XP Embedded can be booted from read-only media which has significant reliability and security benefits

- Windows XP Embedded systems can also be booted from a saved hibernation state which facilitates fast boot up

- Network boot of Windows XP Embedded facilitates a network of diskless systems to be used

Resources

MSDN Training Course 2545C:

http://www.microsoft.com/communities/newsgroups/en-us/default.aspx

Search for "Windows XP Embedded" "General Discussion"

Some Books (Search on web):

Windows XP Embedded Step by Step, Beau Cseri, Annabooks/Rtc Books January 2003 Windows XP Embedded Advanced , Sean D. Liming, Annabooks/Rtc Books, October 2003

MSDN-Embedded-XPe Website:

http://msdn.microsoft.com/embedded/windowsXPembedded/default.aspx

MSDN Library:

http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnanchor/html/XPembedded.asp

Resources

XPe Newsgroup:

"Developing Solutions for Microsoft Windows XP Embedded"

Windows XP Embedded Team Blog:

http://blogs.msdn.com/embedded/

Mikehall's Embedded WEBlog:

http://blogs.msdn.com/mikehall/

Flash Drives that boot XPe

http://www.seanliming.com/flashhelp.html

Need developer resources on this subject?

Stop by the MED Content Publishing Team Station in the Microsoft Pavilion or visit the MED Content Publishing Team Wiki site:

http://msdn.microsoft.com/mobility/wiki

Contact:

- David Jones   **RMIT**University
  - RMIT University, Melbourne Victoria, Australia
    - School of Electrical & Computer Engineering
  - davejones@rmit.edu.au
  - http://babbage.sece.rmit.edu.au/embedded

Microsoft®

Your potential. Our passion.™