



# ***EDK I Build EFI Shell and Debug Lab***

Intel Corporation  
Software and Services  
Group



# Agenda

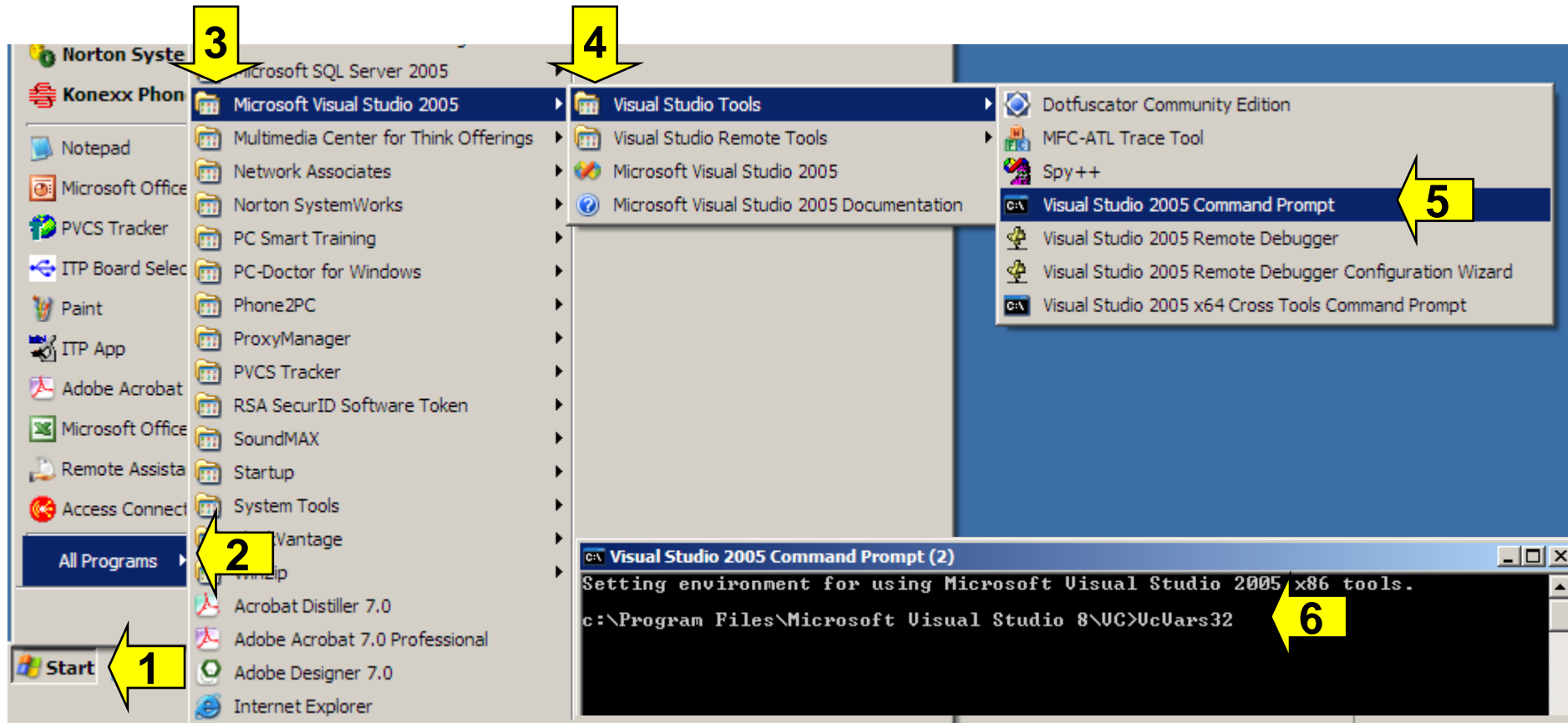
- Generating the Build EDK Lab
- EFI Shell Basics
- Build Results Lab
- EFI Shell – Command Line Tools Lab



- System requirements:
  - Microsoft Windows XP
  - 256MB+ System Memory
  - 500MB+ Free Space on Hard Drive
  - Visual Studio 2005 (Ver 8.0) Professional or VS .NET 2003
- Create a directory on C: \Fw
- Download the EDK and EFI-Shell from Tianocore.org
  - Go to Documents & files
    - Releases
      - Official releases and download the latest Release .zip
- CD Directory NT32BuildLab contains EDK and EFI-Shell
  - Unzip the “EDK\_1.05.Zip” to C:\Fw
  - Unzip the “EfiShell 1.05.Zip” to C:\Fw\Edk\Other\Maintained\Application



- Use the Visual Studio command prompt to setup the proper compiler environment
- Run VcVars32 at the CMD Window with VS 2005



# NT32 Build Lab

- `CD C:\FW\Edk\Sample\Platform\Nt32\Build`
- Edit the Config.env file for VS 2005 Only change the following line to “YES”
  - `USE_VC8 = YES`
- Add the EFI Shell to the Build
  - Edit or Notepad `C:\Fw\Edk\Sample\Platform\Nt32\Build\nt32.dsc` check
    - Add the following line in the Shell EFI Library section under “[Libraries.Platform]”  
`Other\Maintained\Application\Shell\Library\EfiShellLib.inf`
    - Search for `other\Maintained\Application\UefiShell\Bin\Shell.inf` replace with (or comment)  
`#Other\Maintained\Application\UefiShell\Bin\Shell.inf`  
`Other\Maintained\Application\Shell\Shellfull.inf`
  - or copy from `CD Nt32BuildLab\Nt32LAB.dsc` to  
`C:\Fw\Edk\Sample\Platform\Nt32\Build\nt32.dsc`
- Build the NT32
  - `set EDK_SOURCE=C:\FW\EDK`
  - `CD C:\FW\Edk\Sample\Platform\Nt32\`
  - `Build`
- Wait for Build to finish



# Agenda

- Generating the Build EDK Lab
- **EFI Shell Basics**
- Build Results Lab
- EFI Shell – Command Line Tools Lab



- Interactive way to use UEFI code in system
- Has command line prompt and Scripting
  - Is similar to DOS and Linux\* shell but not EXACTLY - its own unique syntax
- Is an EFI executable in itself
- Knows only about EFI file systems that are FATxx
- Shell is a Sub-project on EFI Development Kit (EDK) on the [EFI and Framework Open source Community Website](#)
  - Shell programs
  - users' guide
  - EFI Shell Source
- EDK has Binary UEFI Shell for processors IA32, Intel® 64, IA-64





WEB Site: <http://www.tianocore.org>

Project: EFI-Shell

Documents & files : “EFI Shell Getting Started Guide”

efi-shell: Documents & files - Microsoft Internet Explorer

Address <https://efi-shell.tianocore.org/servlets/ProjectDocumentList>

Google Search

My pages Projects Home openCollabNet

Projects >> efi-shell

Project tools  
Project home  
Discussions  
Documents & files  
Wiki  
Subversion  
Project metrics  
CUBIT  
Search  
This project  
Go

efi-shell  
Documents & files

- efi-shell (1)
  - Documents (3)
  - Member Documents (0)
- Releases (0)
  - Development Snapshots (11)
  - Official Releases (6)
  - Snapshot/Release Notes (10)

Files

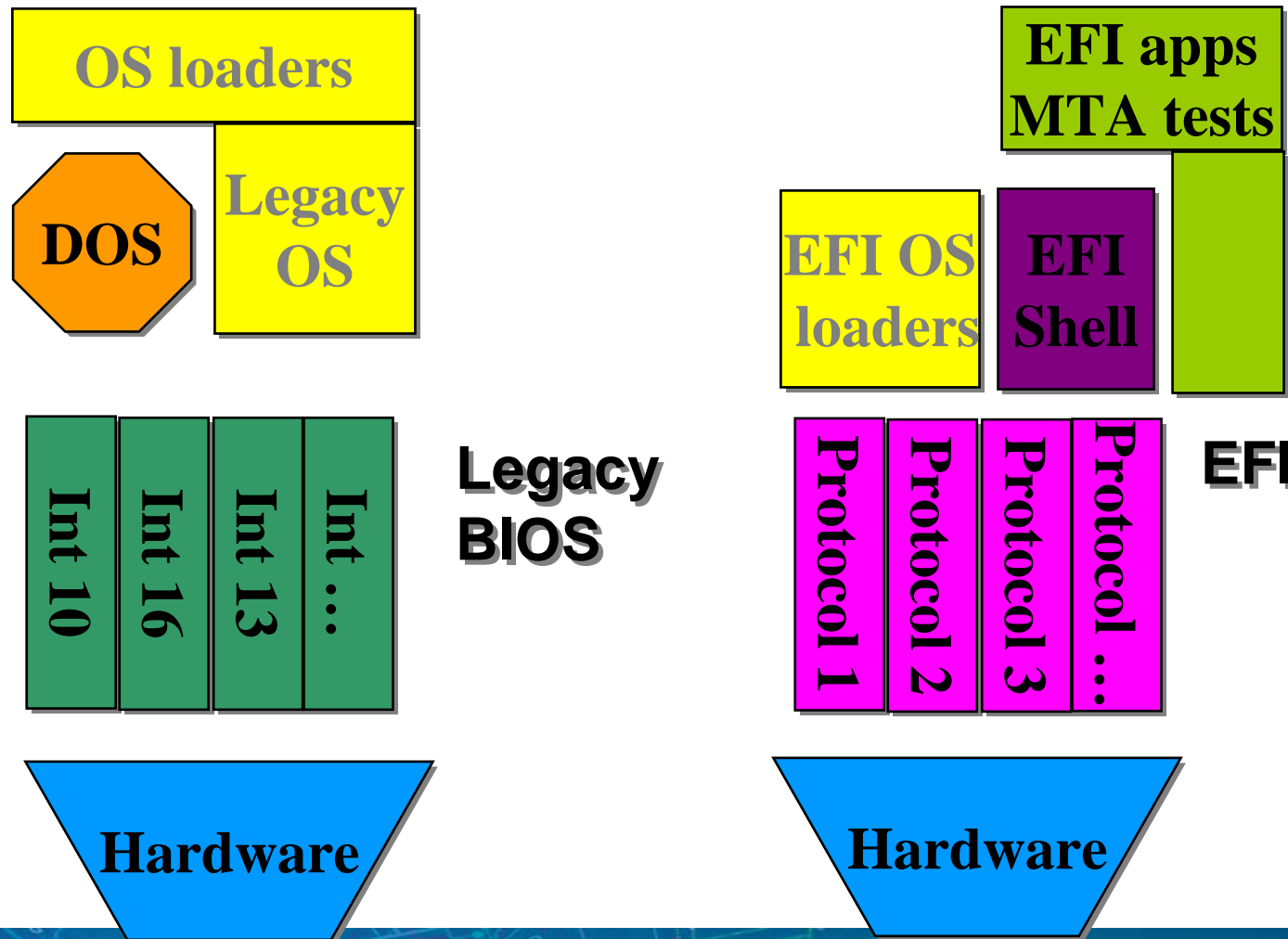
Filter this list  Filter

Name	Status	Modified by	Size
EFI Shell Getting Started Guide.pdf	Draft	lfeishe on Friday, July 1, 2005 at 8:09:32 PM	140.93 kB





# Analogy to Old DOS: BIOS



- Execute preboot programs
  - Setup
  - operating system install
  - Test
  - disk utilities
  - Driver Diagnostics, Configurations
- Move files around between the hard disk, floppy disk, CD-ROM, USB flash devices, and so on
- Load a preboot EFI driver in the system (has an .efi suffix), examples:
  - LAN stack tcpip drivers
  - Update old drivers in flash
  - New drivers for plugin cards
- Shell.efi verses Shell\_full.efi
  - Shell.efi smaller to fit in Flash
  - Shell\_full.efi Richer commands



Can manipulate EFI system fatxx partition only where boot loader and EFI application are

```
Shell> map
Device mapping table
fs0   : Acpi(PNP0A03,1)/Pci(1F|0)/Pci(2|0)/
       Scsi(Pun0,Lun0)/HD(Part1,Sig8983DFE0-F474-01C2-507B-
       9E5F8078F531)
blk0  : Acpi(PNP0A03,0)/Pci(1F|1)/Ata(Primary,Slave)
blk1  : Acpi(PNP0A03,0)/Pci(1F|1)/Ata(Primary,Master)
blk2  : Acpi(PNP0A03,1)/Pci(1F|0)/Pci(2|0)/Scsi(Pun0,Lun0)
blk3  : Acpi(PNP0A03,1)/Pci(1F|0)/Pci(2|0)/
       Scsi(Pun0,Lun0)/HD(Part1,Sig8983DFE0-F474-01C2-507B-
       9E5F8078F531)
blk4  : Acpi(PNP0A03,1)/Pci(1F|0)/Pci(2|0)/
       Scsi(Pun0,Lun0)/HD(Part2,Sig898D07A0-F474-01C2-F1B3-
       12714F758821)
blk5  : Acpi(PNP0A03,1)/Pci(1F|0)/Pci(2|0)/
       Scsi(Pun0,Lun0)/HD(Part3,Sig89919B80-F474-01C2-D931-
       F8428177D974)
```



fs0 :

Acpi(PNP0A03,1)/Pci(1F|0)/Pci(2|0)/Scsi(Pun0,Lun0)/HD(Part1  
,  
Sig8983DFE0-F474-01C2-507B-9E5F8078F531)

Fs0:

Acpi(PNP0A03,1)

Pci(1F|0)/Pci(2|0)

Scsi(Pun0,Lun0)

HD(Part1,Sig8983DFE0-F474-01C2-507B-9E5F8078F531)



# *EFI Shell Commands*

## *Help ?*

dh

help ?

map

mount

load

unload

loadbmp

nshell

ver

memmap

bcfg

Dblk

alias

dmem

dmpstore

err

guid

pci

mm

reset

stall

getmtc

hexedit

Setsize

Set

drivers

devtree

devices

connect

disconnect

openinfo

reconnect

drvcfg

drvdiag

loadpcirom



# *Main EFI Shell Commands For EFI / UEFI Drivers*

- **dh**
  - Displays handles in the EFI environment
- **map**
  - Displays or defines mappings
- **drivers**
  - Displays drivers and attributes in database
- **connect / disconnect**
  - Start and Stop managing a driver
- **drvcfg**
  - Run the driver's Configuration program
- **drvdiag**
  - Run the Driver's Diagnostic program



# Agenda

- Generating the Build EDK Lab
- EFI Shell Basics
- Build Results Lab
- EFI Shell – Command Line Tools Lab





# Setting Up The Environment

- Local disks & folders can be mounted as file systems under the NT32 environment
- Window names can be modified
- The `System.cmd` in the Build directory Examples:
  - Map a 1.44MB Floppy:  
`set EFI_WIN_NT_PHYSICAL_DISKS=a:RW;2880;512`
  - Set the window name:  
`set EFI_WIN_NT_UGA="UGA Window 1"`
  - Map a system directory as a file system under EFI NT32:  
`set EFI_WIN_NT_FILE_SYSTEM=c:\virtual` (needs to a `mkdir c:\virtual` before you `nmake run`)  
Or set it to `.` to point to the current build dir where all the `.efi` files are at. This will allow you to run the `.efi` files from `FS0`:
- Run `System.cmd` before running `nmake run` to get file systems



# NT32 Build Lab Continued

```
C:\ Visual Studio 2005 Command Prompt
bKb\Dxe\makefile all
nmake -nologo -f K:\Edk\Sample\Platform\Nt32\uefi\IA32\Sample\Bus\Usb
bMassStorage\Dxe\makefile all
nmake -nologo -f K:\Edk\Sample\Platform\Nt32\uefi\IA32\Sample\Bus\Usb
bMouse\Dxe\makefile all
nmake -nologo -f K:\Edk\Sample\Platform\Nt32\uefi\IA32\Sample\Univers
Network\SnpNt32\Dxe\makefile all
nmake -nologo -f K:\Edk\Sample\Platform\Nt32\uefi\IA32\Other\Maintain
Application\Customize\HelloWorldx\makefile all
nmake -nologo -f K:\Edk\Sample\Platform\Nt32\uefi\IA32\Other\Maintain
Universal\Disk\FileSystem\EnhancedFat\Dxe\makefile all
k:\edk\Sample\Platform\Nt32\uefi\Tools\GenFvImage -I FURECOVERY.inf
GenFvImage - Tiano Firmware Volume Generation Utility. Version 0.1

copy /b Fv\FvRecovery.fv + /b Fv\NvStorage.fv /b Fv\FvRecovery.fd
Fv\FvRecovery.fv
Fv\NUSTORAGE.fv
1 file(s) copied.

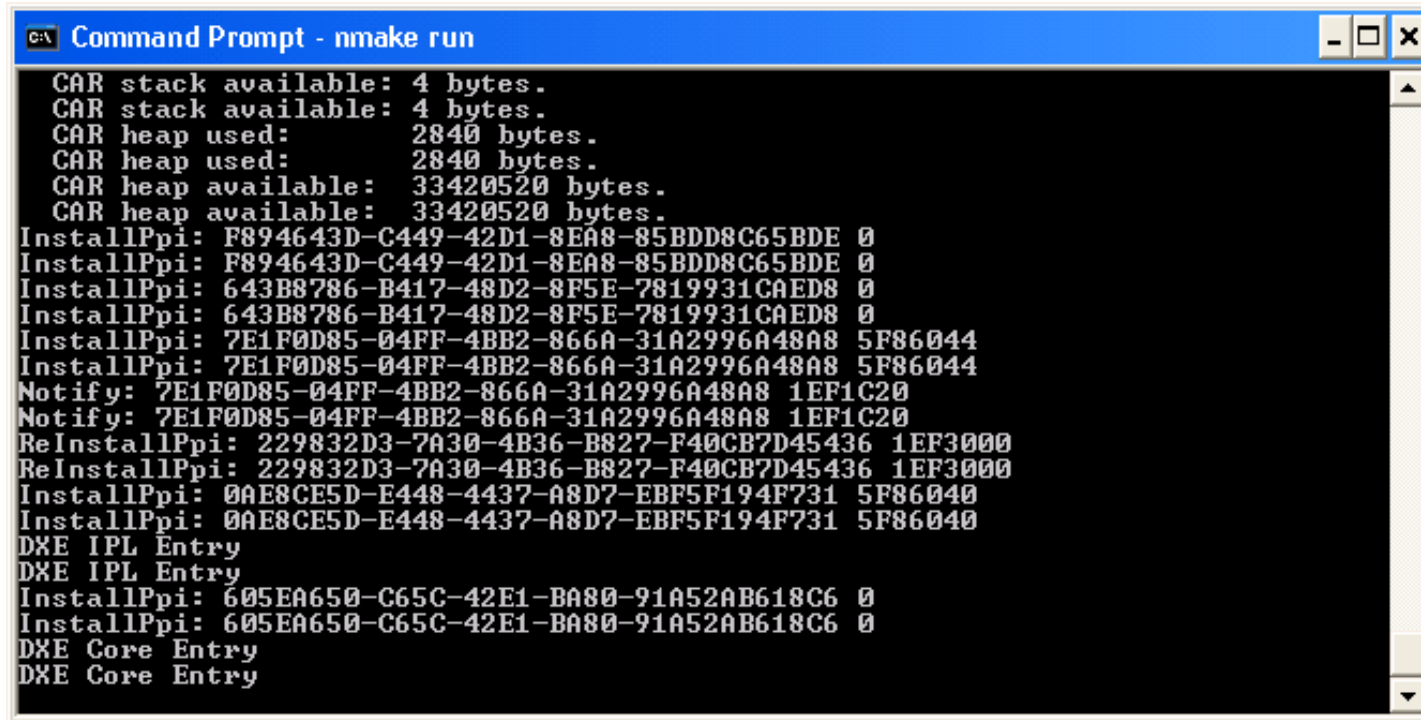
Start Time 14:51:29.95
End Time 14:54:01.15

K:\Edk\Sample\Platform\Nt32>
```

Build Time stamp

- After BUILD is complete
  - CD Uefi
  - System
  - Nmake Run





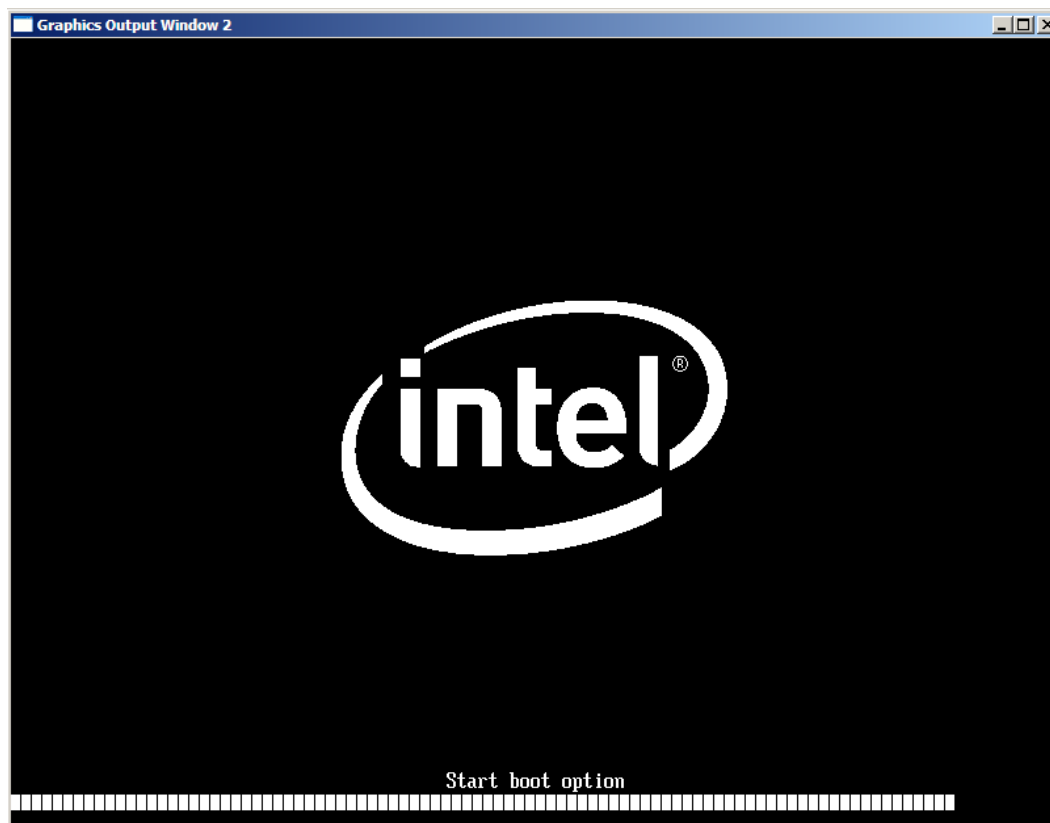
```
Command Prompt - nmake run
CAR stack available: 4 bytes.
CAR stack available: 4 bytes.
CAR heap used: 2840 bytes.
CAR heap used: 2840 bytes.
CAR heap available: 33420520 bytes.
CAR heap available: 33420520 bytes.
InstallPpi: F894643D-C449-42D1-8EA8-85BDD8C65BDE 0
InstallPpi: F894643D-C449-42D1-8EA8-85BDD8C65BDE 0
InstallPpi: 643B8786-B417-48D2-8F5E-7819931CAED8 0
InstallPpi: 643B8786-B417-48D2-8F5E-7819931CAED8 0
InstallPpi: 7E1F0D85-04FF-4BB2-866A-31A2996A48A8 5F86044
InstallPpi: 7E1F0D85-04FF-4BB2-866A-31A2996A48A8 5F86044
Notify: 7E1F0D85-04FF-4BB2-866A-31A2996A48A8 1EF1C20
Notify: 7E1F0D85-04FF-4BB2-866A-31A2996A48A8 1EF1C20
ReInstallPpi: 229832D3-7A30-4B36-B827-F40CB7D45436 1EF3000
ReInstallPpi: 229832D3-7A30-4B36-B827-F40CB7D45436 1EF3000
InstallPpi: 0AE8CE5D-E448-4437-A8D7-EBF5F194F731 5F86040
InstallPpi: 0AE8CE5D-E448-4437-A8D7-EBF5F194F731 5F86040
DXE IPL Entry
DXE IPL Entry
InstallPpi: 605EA650-C65C-42E1-BA80-91A52AB618C6 0
InstallPpi: 605EA650-C65C-42E1-BA80-91A52AB618C6 0
DXE Core Entry
DXE Core Entry
```

- Start Emulation
  - Emulation runs as Windows application
  - Reset vector code for desktop/server replaced with Windows application startup code
  - Startup code initializes framework and calls dispatcher to start loading drivers

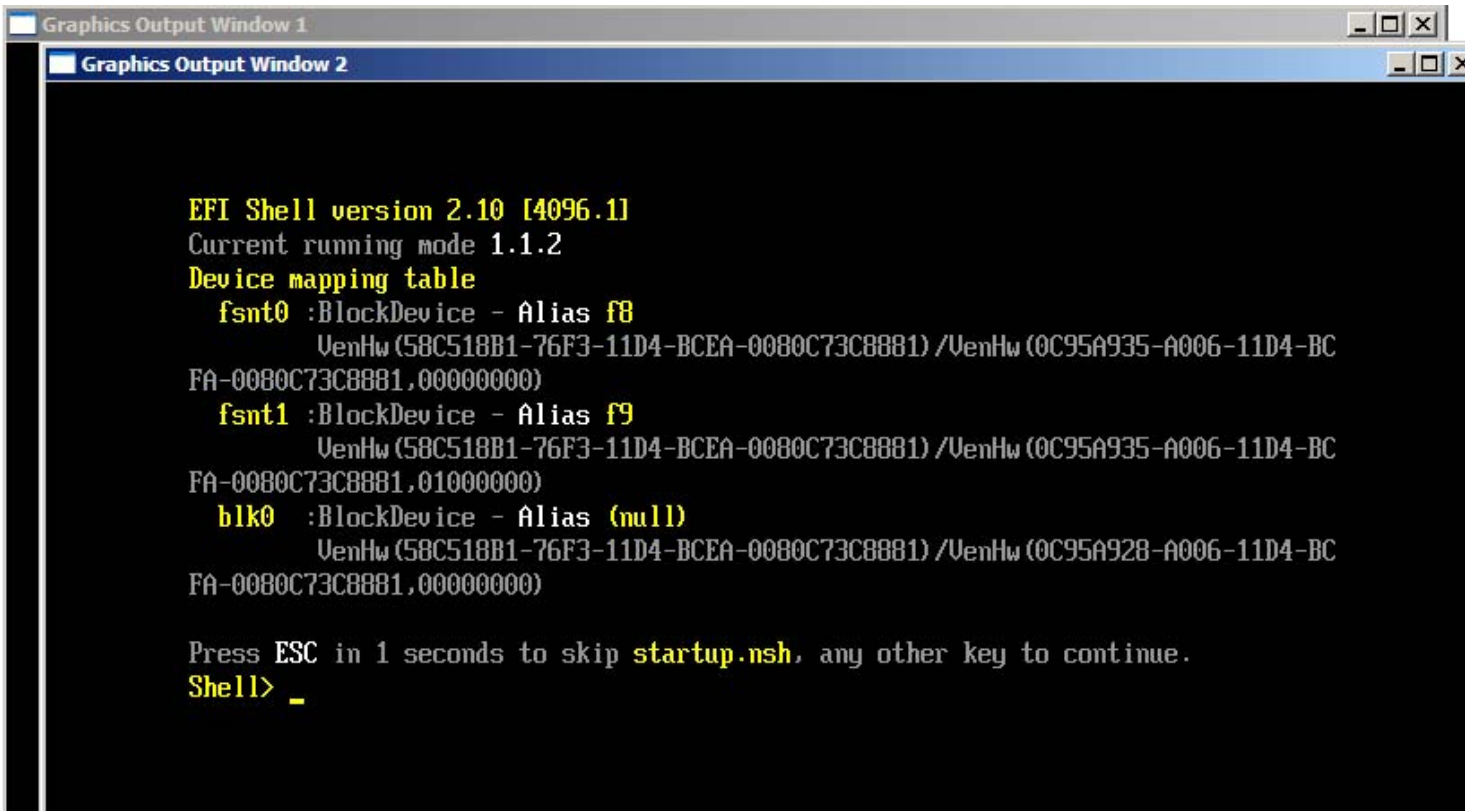


## *NT32 Emulated Boot*

- Splash screen displayed
  - Windows drivers simulate output to screen, input from keyboard and disk operations to Hard Drive



# Boot to the EFI Shell



```
Graphics Output Window 1
Graphics Output Window 2

EFI Shell version 2.10 [4096.11]
Current running mode 1.1.2
Device mapping table
  fsnt0 :BlockDevice - Alias f8
          VenHw(58C518B1-76F3-11D4-BCEA-0080C73C8881)/VenHw(0C95A935-A006-11D4-BC
FA-0080C73C8881,00000000)
  fsnt1 :BlockDevice - Alias f9
          VenHw(58C518B1-76F3-11D4-BCEA-0080C73C8881)/VenHw(0C95A935-A006-11D4-BC
FA-0080C73C8881,01000000)
  blk0  :BlockDevice - Alias (null)
          VenHw(58C518B1-76F3-11D4-BCEA-0080C73C8881)/VenHw(0C95A928-A006-11D4-BC
FA-0080C73C8881,00000000)

Press ESC in 1 seconds to skip startup.nsh, any other key to continue.
Shell> _
```

- Load the EFI Shell
- Switch to fsnt0: or F8: for the local disk
  - Mapped to EFI\_WIN\_NT\_FILE\_SYSTEM



# Agenda

- Generating the Build EDK Lab
- EFI Shell Basics
- Build Results Lab
- **EFI Shell – Command Line Tools Lab**



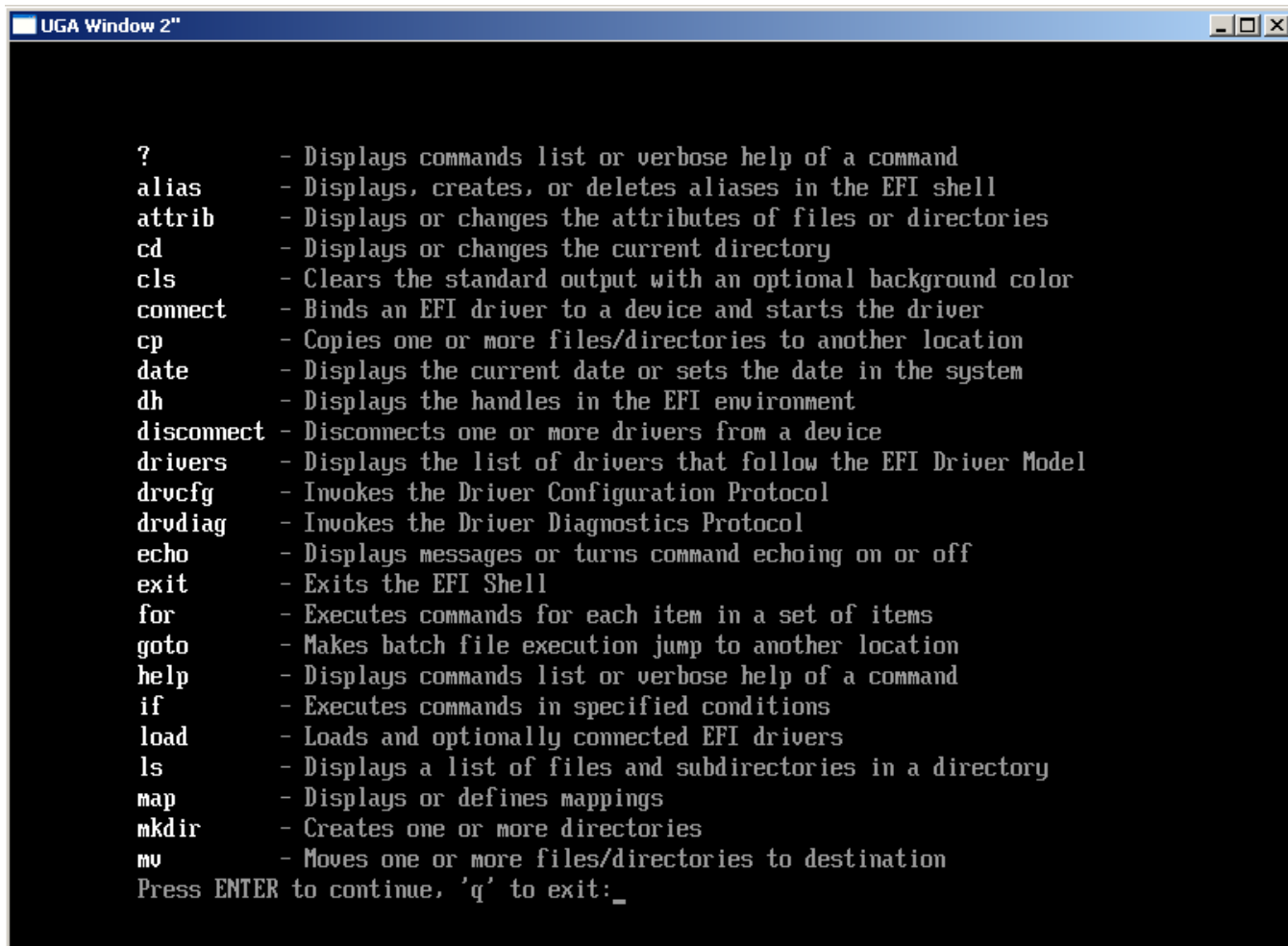
# *EFI Shell Command*

- The following EFI Shell commands are useful for Debugging EFI applications
  - Help
  - MM
  - Mem
  - Memmap
  - Drivers
  - Devices
  - Devtree
  - Dh
  - Load





- Help -b



```
UGA Window 2"

?          - Displays commands list or verbose help of a command
alias      - Displays, creates, or deletes aliases in the EFI shell
attrib     - Displays or changes the attributes of files or directories
cd         - Displays or changes the current directory
cls        - Clears the standard output with an optional background color
connect    - Binds an EFI driver to a device and starts the driver
cp         - Copies one or more files/directories to another location
date       - Displays the current date or sets the date in the system
dh         - Displays the handles in the EFI environment
disconnect - Disconnects one or more drivers from a device
drivers     - Displays the list of drivers that follow the EFI Driver Model
drvcfg     - Invokes the Driver Configuration Protocol
drvdiag    - Invokes the Driver Diagnostics Protocol
echo       - Displays messages or turns command echoing on or off
exit       - Exits the EFI Shell
for        - Executes commands for each item in a set of items
goto       - Makes batch file execution jump to another location
help       - Displays commands list or verbose help of a command
if         - Executes commands in specified conditions
load       - Loads and optionally connected EFI drivers
ls         - Displays a list of files and subdirectories in a directory
map        - Displays or defines mappings
mkdir      - Creates one or more directories
mv         - Moves one or more files/directories to destination
Press ENTER to continue, 'q' to exit:_
```



# ***mm – Displays or Modifies memory, I/O or PCI resources***

Usage:

**MM Address [Width 1|2|4|8] [;MMIO | ;MEM | ;IO | ;PCI] :Value -n**

- |         |   |
|---------|---|
| Address | - Starting address for MMIO, MEM, IO or PCI   |
| Width   | - Size accessed in bytes (1, 2, 4 or 8)   |
| ;MMIO   | - MMIO Range (0 – 0xFFFFFFFF_FFFFFFFF)  |
| ;MEM    | - Memory Range (0 – 0xFFFFFFFF_FFFFFFFF)  |
| ;IO     | - IO Address Range (0 – 0xFFFF)   |
| ;PCI    | - PCI Config Address (0x000000ssbbddffrr)<br>ss = SEG, bb = BUS, dd = DEV, ff = FUNC, rr = REGISTER |
| Value   | - Value to write  |
| -n      | - Non-interactive mode  |



# ***mm – Displays or Modifies memory, I/O or PCI resources***

*continued*

## **Notes:**

- 1. MEM type is the default**
- 2. In interactive mode type a hex value to modify, enter 'q' or '.' to exit.**
- 3. Use the PCI command to discover the PCI device before using MM to modify PCI configuration space.**
- 4. Use '-n' mode inside of shell script files (\*.nsh)**
- 5. Not all PCI registers are writable. PCI option will not do read-modify write. MM will only write the value posted.**



# ***mem – Displays Contents of System Memory or Device Memory***

Usage:

**MEM [-b] [address] [size] [;MMIO]**

- b - Display one screen at a time
- address - Starting Address (hex) to display. This needs to be an even address boundary for the processor the command is run on.
- size - Number of bytes to display (hex)
- ;MMIO - Memory mapped I/O. Turns on any bits required to force memory access across out to the PCI bus

**Note: Run w/o args to see the system table entry pointer and all other system table pointer addresses.**



# ***memmap – Displays Memory Map Maintained by EFI***

Usage:

**MEMMAP [-b]**

-b                      - Display one screen at a time

Note:

Use the EFI specification to lookup the memory type.



# *drivers – Displays List of EFI Drivers*

Usage:

## **DRIVERS [-b] [-IXXX]**

- b - Displays one screen at a time
- IXXX - Displays drivers using the ISO 639-2 language specified by XXX

Note:

Run DRIVERS /? to see the display format



## ***devices – Display List of Devices Managed by EFI drivers***

Usage:

### **DEVICES [-b] [-IXXX]**

- b        - Displays one screen at a time
- IXXX    - Displays devices using the ISO 639-2 language  
             specified by XXX

Note:

Run DEVICES /? to see the display format





# ***devtree – displays Tree of Devices Follow the EFI Driver Model***

## Usage:

**DEVTREE [-b] [-d] [IXXX] [DeviceHandle]**

- b                      - Displays one screen at a time
- d                      - Displays device tree using device paths
- IXXX                  - Displays drivers using the ISO 639-2  
language specified by XXX
- DeviceHandle           - Displays device handle below a  
certain handle



## ***dh – Displays handles in EFI***

Usage:

**DH [-b] [-d] [-IXXX] [-v] [handle] [-p prot\_id]**

- |            |  |
|------------|--|
| -b         | - Displays one screen at a time                                  |
| -d         | - Displays device tree using device paths                        |
| -IXXX      | - Displays drivers using the ISO 639-2 language specified by XXX |
| -v         | - Dumps information on all handles                               |
| handle     | - Dumps information on a specific handle                         |
| -p prot_id | - Dumps all handles of a certain protocol                        |



## ***load – Load EFI Drivers***

Usage:

**LOAD [-nc] file [file...]**

- |      |  |
|------|--|
| -nc  | - Load the driver, but do not connect the driver     |
| file | - File that contains the EFI driver (.efi extension) |

Note:

- LOAD can handle multiple files & supports wildcards
- Use the 'UNLOAD' command to unload an EFI driver



## ***stall – Stall the Processor***

Usage:

**STALL microseconds**

Note:

- The 'microseconds' value is in decimal.
- STALL will cause the emulation environment to 'sleep' for the specified period.



# Additional Resources

- Class CD - Documents
  - Framework R8.x Build.pdf describes build tools
- <https://www.TianoCore.org>
  - Website for EFI open source resources
    - EFI Developer Kit (EDK)
      - Nt32 emulation environment
    - EFI shell <https://efi-shell.tianocore.org/>
    - EFI Documentation: <https://efi-shell.tianocore.org/servlets/ProjectDocumentList?folderID=50&expandFolder=50&folderID=0>



# Q & A

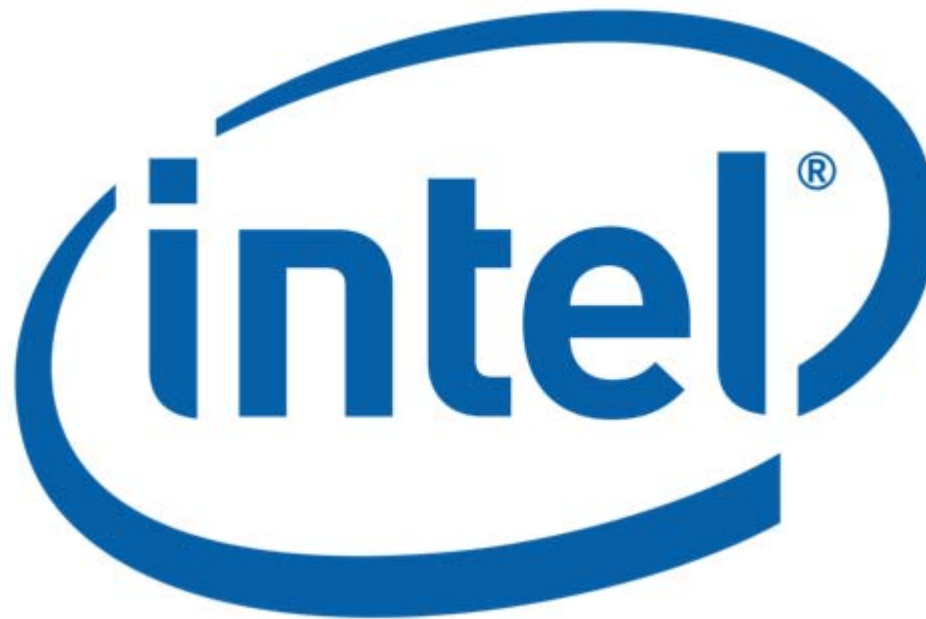


## UEFI / Framework Training 2008

Copyright © 2006-2008 Intel Corporation  
•Other trademarks and brands are the property of their respective owners

Slide 35





## UEFI / Framework Training 2008

Copyright © 2006-2008 Intel Corporation

•Other trademarks and brands are the property of their respective owners

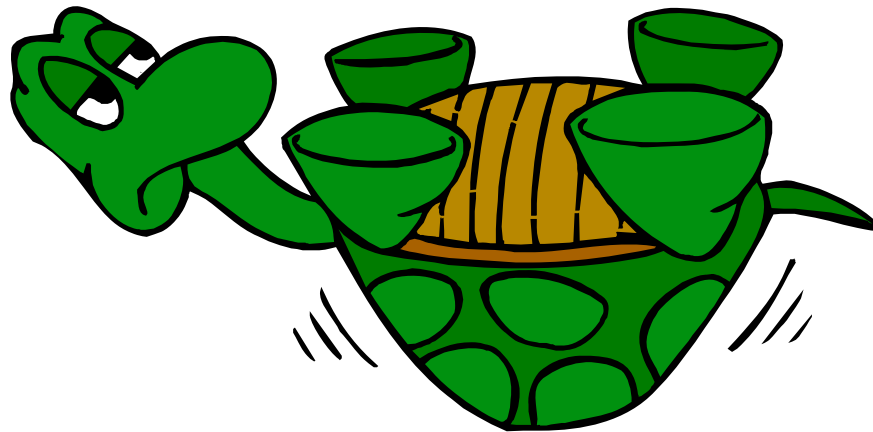
Slide 36





*Back up*

**Back Up**



## UEFI / Framework Training 2008

Copyright © 2006-2008 Intel Corporation  
•Other trademarks and brands are the property of their respective owners

Slide 37

