Introduction to EFI

Dong Wei & Jason Reasor

HP WORLD 2003
Solutions and Technology Conference & Expo

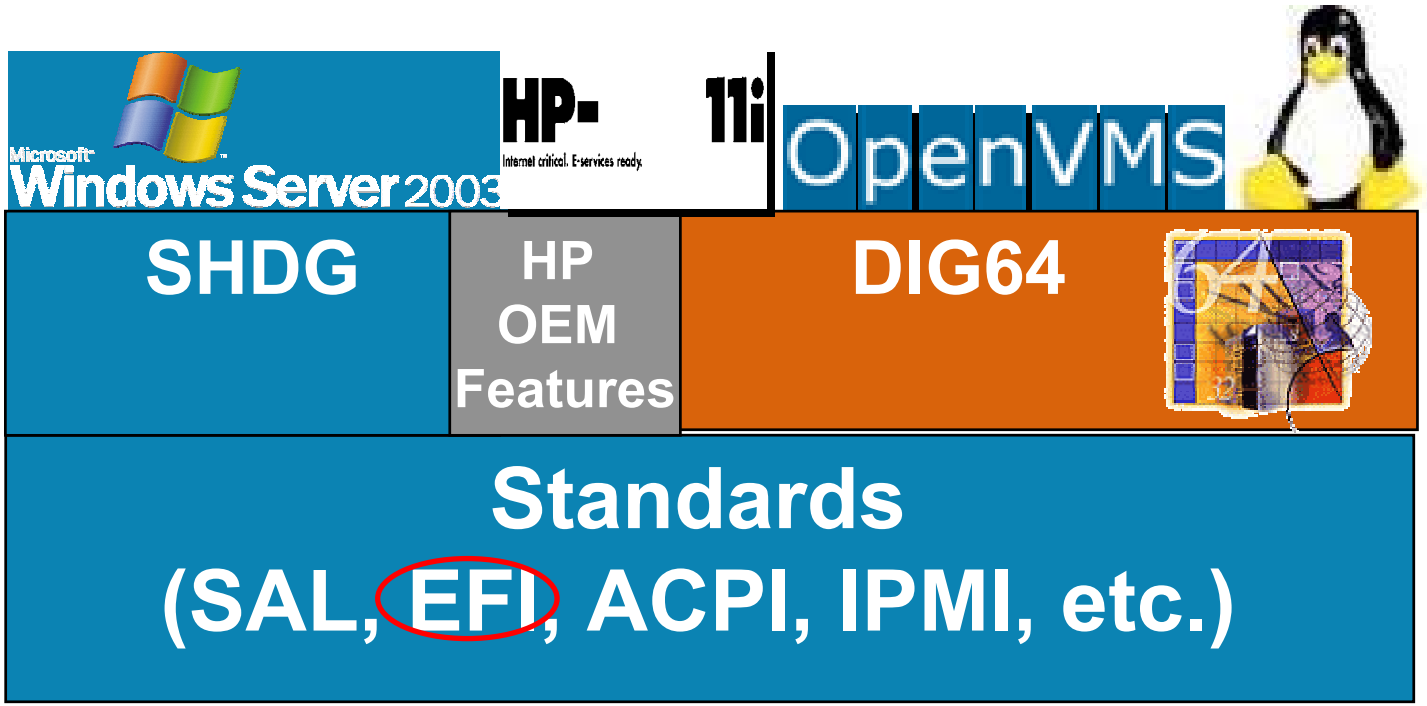
Objectives

- Provide an overview of EFI and how it fits in with the rest of the firmware
- Demonstrate the user interfaces EFI provides, concentrating on changes made by HP

Agenda

- Overview of EFI
- Overview of firmware initialization
- User interfaces
- Demonstration

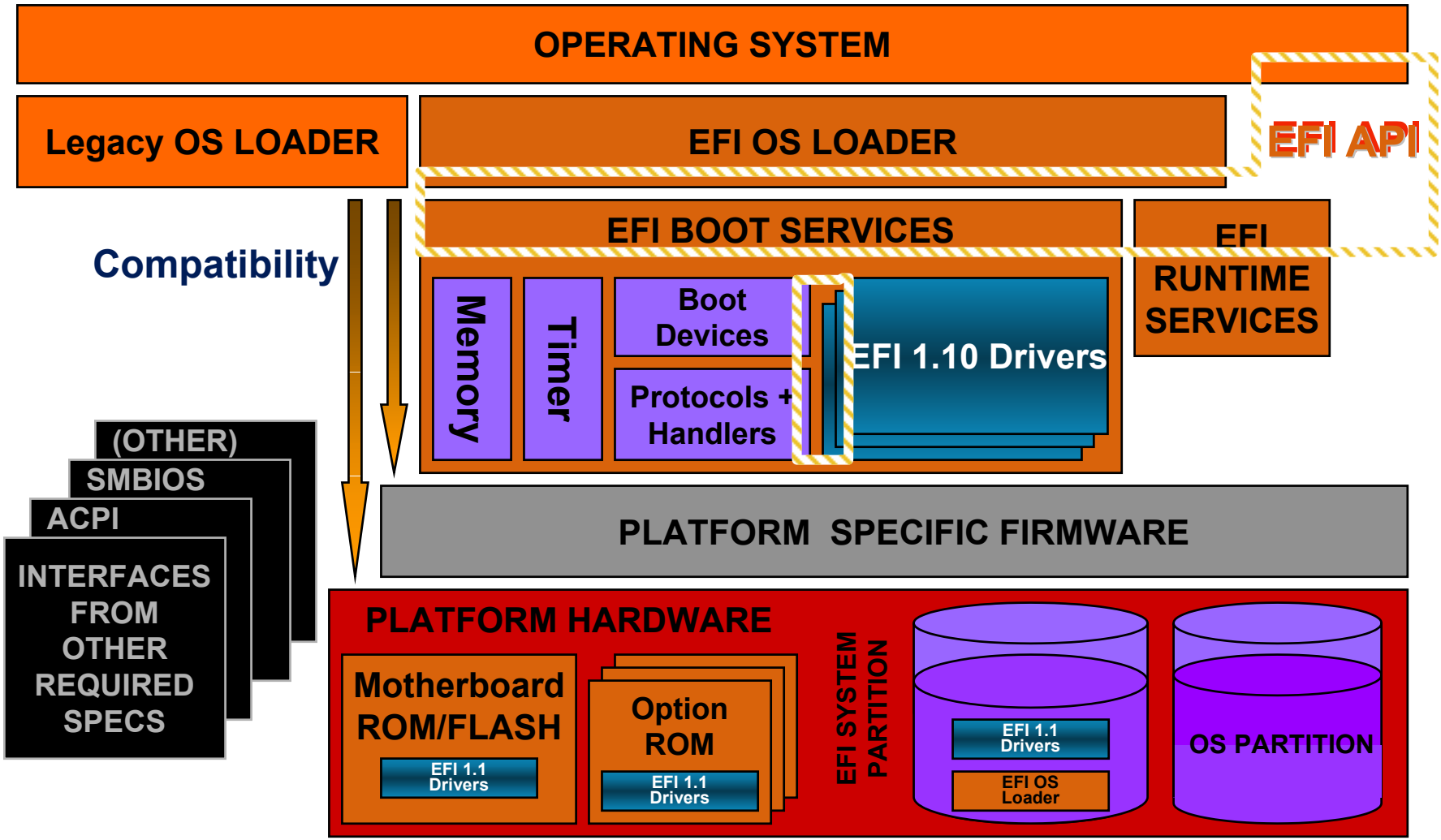
Itanium Platform Architecture



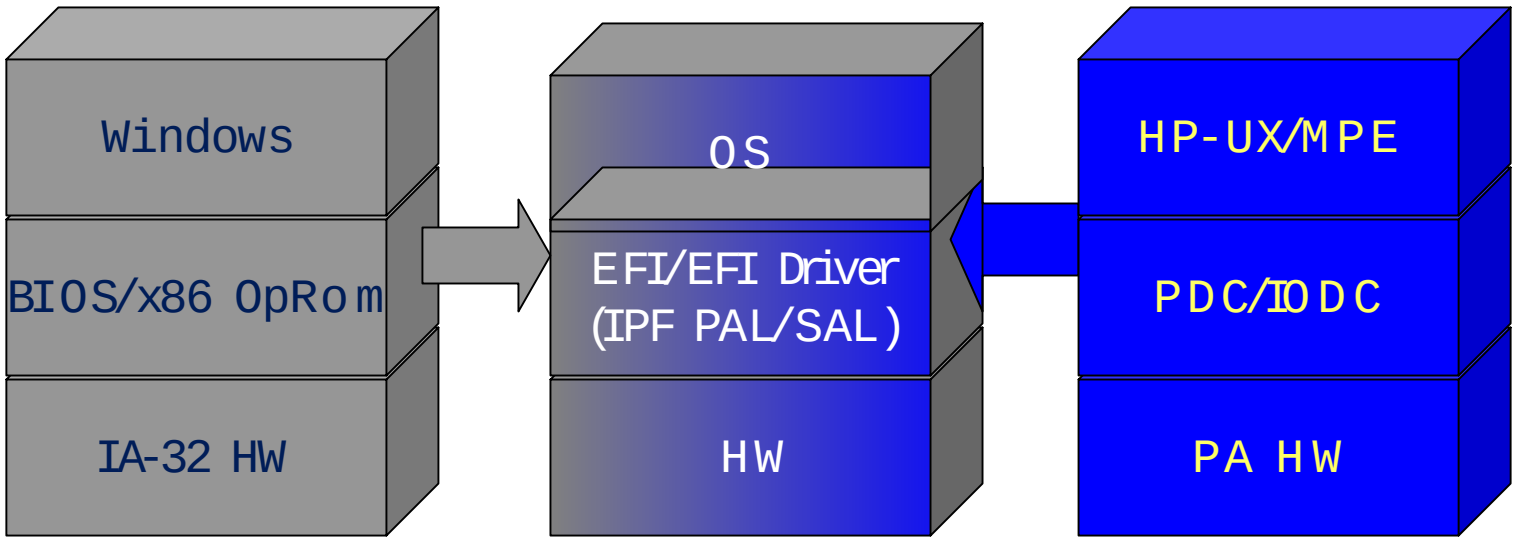
EFI

- Specification is owned by Intel. HP is a major consultant.
- EFI 1.02 defines OS/FW Boot Services that replace BIOS INT calls
- EFI 1.10 defines the Device Driver Model that replaces IA-32 Legacy Option ROM
- IPF equivalent of HPPA's IODC, LIF, and ISL
- Processor Architecture Agnostic
- Intel considers this the BIOS replacement for the next 20 years for all Intel Architectures (e.g., IA-32, IPF)
- UGA to replace VGA: UGA EBC Driver rather than BIOS INT 10h

EFI 1.02 and 1.10 (Diagram)



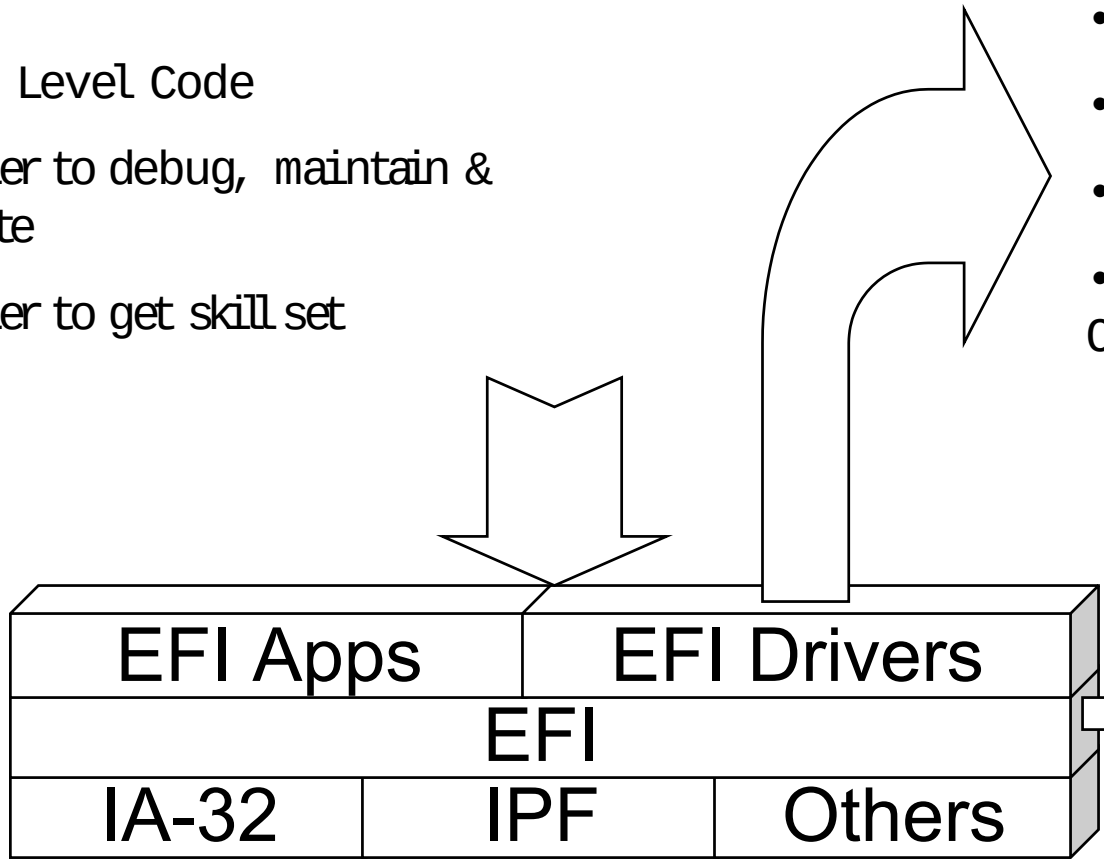
Heritages



Foundation for Multi-OS and Legacy Free Support

EFI Benefits

- High Level Code
- Easier to debug, maintain & validate
- Easier to get skill set



- No Collision
- No Space Limitation
- Support Speedy Boot
- Provide some drivers to OS

- Portable
- Fully Realized Flexibility

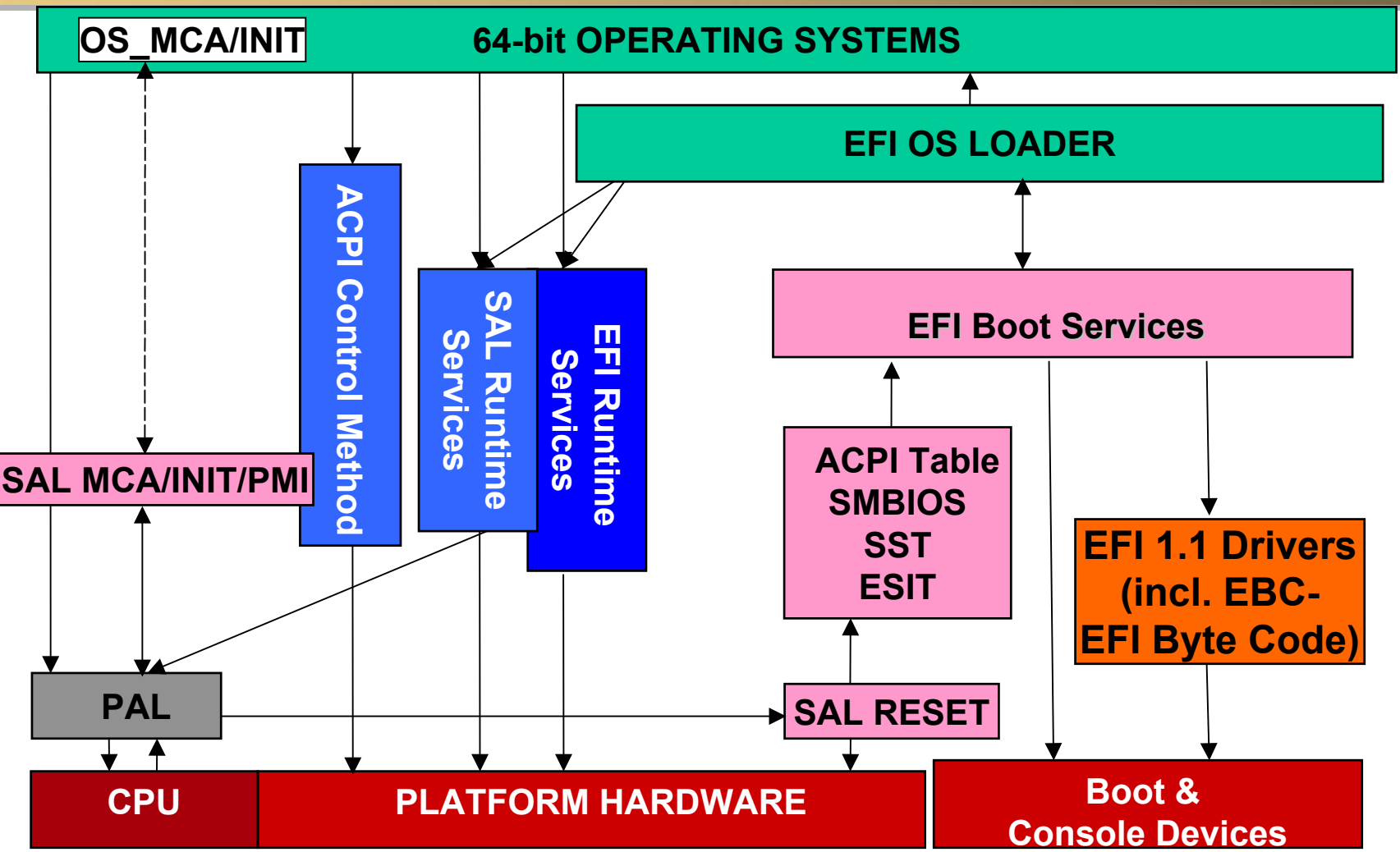
Firmware Initialization

- Firmware initialization is comprised of three major code paths:
  - PAL – provided by Intel, low level cpu initialization
  - SAL – provided by the platform vendor, platform initialization
  - EFI – original source provided by Intel, modified by the platform vendor
- EFI is the last code executed in the firmware initialization path

Firmware Initialization - EFI

- Consoles are connected
- I/O drivers loaded
  - Native EFI
  - EBC
- Devices behind the cards that were initialized are enumerated
- The Boot Manager is launched
- If autoboot is enabled, and a valid boot path exists, the system will attempt to boot

Firmware Initialization



Firmware Initialization

- After the system is initialized, EFI provides the interfaces with which the user interacts
  - Boot Manager
    - Menu based interaction
  - EFI Shell
    - Command line interface
- The user can move back and forth between the two interfaces

Boot Manager

- Menu based interface
- Arrow keys used to traverse menus
- Used primarily for options related to booting an OS or loading an EFI application

Boot Manager

- First level menu: Boot Manager Menu
  - Displays boot options
  - Entrypoint for EFI shell
  - Boot maintenance menu
  - Autoboot timeout
    - time period before boot option list is traversed

Boot Manager

- Second level menu: Boot Maintenance Menu
  - Configure boot options
  - Select console devices
  - Reset
  - Exit back to primary boot manager menu

Boot Manager

- Boot Maintenance Menu: Boot Options
  - Boot from a file
  - Add boot options
  - Delete boot options
  - Change boot order
  - Manage “BootNext” setting
  - Set auto boot time out

Boot Manager

- Boot Maintenance Menu: Select Console Devices
  - Select active console output devices
  - Select active console input devices
  - Select active console error devices

EFI Shell

- Command line interface
- Not UNIX, not DOS
- Provides a platform for a user to:
  - Get information on the system
  - Boot an OS
  - Install an OS
  - Execute batch scripts
  - Launch EFI applications
  - Load EFI drivers
  - Manage files and system variables

EFI Shell

- Shell invocation
  - Automatically execute the “startup.nsh” file if it exists
  - Wait for command input from console
- EFI commands
  - See references section for more information and user guides

EFI Shell

- File systems
  - EFI understands FAT filesystems
  - All disk partitions (FAT or not) will be displayed as “blkX” devices
  - FAT partitions will be displayed as “fsX” devices
  - A user can traverse “fsX” file systems just like he would traverse a filesystem under UNIX or DOS
  - File systems can be seen from the shell by issuing the “map” command

EFI Applications

- Compiled to run in the EFI environment using the EFI developer's toolkit
- Named using the ".efi" extension
- Example: "ifconfig.efi"
- To execute the application enter the name without the .efi extension as well as any parameters the application requires
- OS loaders
  - HP-UX: hpux.efi
  - VMS: vms_loader.efi
  - Windows: ia64ldr.efi
  - Linux: elilo.efi

EFI Drivers

- Compiled to load in the EFI environment using the EFI developer's toolkit
- Named using the ".efi" extension
- Example: "tcpipv4.efi"
- To load the driver, use the "load" command and the driver name

EFI Drivers

- PCI card drivers are located on the card
  - Native EFI
  - EBC
- At boot EFI will load and connect:
  - Drivers on cards connected to the core cell
  - Drivers on cards that have boot paths associated with them
- If a card does not have an EFI driver, the devices attached to the card can not be used from the EFI Shell
- EFI drivers have nothing to do with OS drivers
  - If a card does not have an EFI driver, the OS can still use it

Scripts

- Batch scripts
  - text file containing a sequence of commands and / or comments
  - named using the ".nsh" extension
  - can execute shell commands and EFI applications
  - comments begin with #
  - example: netsetup.nsh

```
cd \efi\tools
load tcpipv4.efi
# set the ip address for the machine "beat_ibm"
ifconfig sni0 inet 15.99.80.20 netmask 255.255.255.0
route add default 15.99.80.254
cd \
```

POSSE - Background

- Pre-OS System Environment
- Common firmware user interface for all HP manufactured IPF servers and workstations
- EFI shell from Intel used as a base
- Integrate PA firmware interface functionality into the IPF environment
- Designed and implemented across several HP labs

POSSE - Background

- Designed to make the EFI shell code common across all HP servers and workstations
- There are different firmware bases within HP, so POSSE provides an abstraction layer to the commands so they can gather the appropriate information from the core firmware

POSSE - Commands

- Command categories
  - boot – boot related commands
  - configuration – retrieving and updating system information
  - device – device, driver, and handle related commands
  - filesystem – filesystem related commands
  - memory – memory related commands
  - shell – basic shell navigation and customization
  - scripts – EFI shell script commands

POSSE – Boot Commands

- autoboot – view or set autoboot timeout variable
- bcfg – displays/modifies the driver/boot configuration
- boottest – view or set speedyboot bits
- lanboot – boot over the LAN
- reset – reset the system
- tftp – trivial file transfer protocol

POSSE – Configuration Commands

- cpuconfig – deconfigure or reconfigure cpus
- date – displays or set the date
- dimmconfig – deconfigure or reconfigure dimms
- err – displays or changes the error level
- errdump – view or clear logs
- fru – view fru data

POSSE – Configuration Commands

- info – display hardware information
- monarch – view or set a monarch processor
- palproc – make a PAL call
- rootcell – view of set the root cell
- salproc – make a SAL call
- search – connect drivers on a cell or pci slot
- time – display or set the time
- ver – display the version information

POSSE – Info Commands

- info sys – display system information
- info cpu – display cpu information
- info mem – display memory information
- info io – display io information
- info chiprev – display ASIC revisions
- info cache – display cache information
- info fw – display firmware revision information
- info boot – display boot information

POSSE – Device Commands

- connect – binds a driver to a device
- dblk – hex dump of blkio devices
- devices – display devices managed by EFI drivers
- devtree – display tree of devices
- dh – dump handle info
- disconnect – disconnects driver from device

POSSE – Device Commands

- drivers – display list of drivers
- drvcfg – invoke the driver config protocol
- drvdiag – invokes the driver diagnostics protocol
- guid – dump known guids
- lanaddress – display core io MAC address
- load – load EFI driver
- loadpcirom – load PCI option ROM image into memory

POSSE – Device Commands

- map – map short name to device path
- openinfo – display the open protocols for given handle
- pci – display PCI devices or PCI configuration space
- reconnect – reconnects driver to a device
- unload – unload a protocol image

POSSE – Filesystem Commands

- attrib – display or change the attributes of files or directories
- cd – updates the current directory
- comp – compares the contents of two files
- cp – copies one or more files/directories to another location
- eficompress – compress infile and write to outfile
- efidecompress – decompress infile and write to outfile
- ls – display a list of files and subdirectories

POSSE – Filesystem Commands

- mkdir – creates directory
- mount – mount a filesystem on a block device
- rm – delete one or more files or directories
- setsize – set the size of a file
- touch – update time of file or directory with current time
- type – display the contents of a file
- vol – displays volume information of the file system

POSSE – Memory Commands

- default – set the default nvram values
- dmem – dump memory or memory mapped IO
- dmpstore – display all EFI variables
- memmap – display the memory map
- mm – memory modify
- pdt – view or clear pdt

POSSE – Shell Commands

- alias – view or edit alias settings
- cls – clear the screen
- exit – exit EFI shell
- getmtc – display current monotonic counter value
- help or ? – displays help
- set – set or get environment variable
- xchar – toggle extended character features

POSSE – Script Commands

- echo – echo text to stdout or toggle script echo
- else – script-only: use with IF THEN
- endfor – script-only: delimiter for FOR loop construct
- endif – script-only: delimiter for IF THEN construct
- for – script-only: loop construct
- goto – script-only: jump to label location in script
- if – script-only: IF THEN construct
- input – take user input, place in efi variable
- pause – script-only: prompt to quit or continue
- stall – stall the processor for some microseconds

Terminology

- ACPI – Advanced Configuration and Power Interface
- DIG64 – Developer's Interface Guide for 64-bit Intel Architecture-based Servers (Dell, Fujitsu-Siemens, HP, Intel, IBM, NEC)
- EBC – EFI Byte Code
- EFI – Extensible Firmware Interface
- IPF – Itanium Processor Family
- PAL – Processor Abstraction Layer
- PDT – Page De-allocation Table
- SAL – System Abstraction Layer
- SHDG – Microsoft Server Hardware Design Guide

References

- EFI Website
  - http://developer.intel.com/technology/efi/efi.htm
- ACPI Website
  - http://www.acpi.info
- DIG64 Website
  - http://www.dig64.org

Interex, Encompass and HP bring you a powerful new HP World.