The UEFI Shell

The UEFI Shell

The UEFI shell consists of two parts:

1. a set of APIs
2. a command-line interface.

- The set of APIs abstract the command line and file I/O aspects of the system. For example, the command-line APIs allow UEFI Shell programs to read the command line. There are also a variety of APIs that deal with the shell environment, such as getting the current setting for the PATH or other meaningful environment variables.
- The command-line processor is the logic that parses whatever a user or script sends to it and is essentially the interpreter of the “shell language” that the UEFI Shell

What is the UEFI Shell

- The UEFI or EFI Shell is a small program similar in functionality to a DOS command line or the Linux terminal commands.
- The UEFI Shell is unusual in that it is not a shell that is a client of an operating system, but is actually considered a BIOS extension. This puts the shell on par with components that traditionally would be launched prior to an operating system such as an add-in device's option ROM.
- The underlying feature set and size are so important that in some cases the shell may actually be contained in the platform's FLASH device or preloaded.

What can it do

The EFI shell gives you the tools to :

- Access connected devices
- Read and write to those devices
- Perform simple tasks such as listing a directories contents
- Run scripts
- Change and update the NVRAM (boot order etc.).
- Run or start EFI programs

Starting the UEFI Shell

- If the UEFI Shell is not contained in the platform's FLASH device, but preloaded it will be in the EFI partition.
- The default location is in the EFI partition in \EFI\TOOLS generally with the name shellx64.efi.
- Depending on how the system has been set up the EFI shell can be started via a boot manager, selected as a boot option in the UEFI BIOS, or in some other fashion.

The basics

- Colours are used to emphasize and help identify items:
- Mostly text is gray and emphasized items are white.
- The colour yellow is used to identify:
  - File systems, block devices and the prompt, i.e. Shell> , fs0:
- When displaying a directory listing of a file systems the colour gray is used to indicate that the item is a file, and the colour blue a directory, and yellow and executable item (scripts are also executable)
- File/Path names are generally not case sensitive, although some commands are!

The basics

- There are no drive letters, but “file system numbers” fs0:, fs1:
- The Shell only directly supports FAT and FAT32 file systems.
- Besides file systems there are also “Block devices” BLK0, BLK1 etc.
- A block device may refer to a complete disk!
- These interfaces are constructed during the UEFI Shell environment’s initialization.
- Both the forward slash and the backslash can be used when specifying a pathname.
- Ctrl-C can be used to terminate an action.

The basics

- EFI environmental variables are case sensitive.
- You can add or remove directories from the path using the “set” command.

  Example to set path:

  set path fs0:\syscfg
- The path setting is valid until next reboot.
- To clear something out of the path, type:

  set –d (item to clear)
- To see the contents of a variable type echo %variable name%

The basics

- EFI, like DOS and Linux, can run programs that are listed in the environmental path from the root. You can add or remove directories from the path using the “set” command.
- Example to set path:

  set path fs0:\syscfg
- where path is the directory holding the utility in the root of your first USB drive. Warning!: This command is case sensitive!
- The path setting is valid until next reboot!
- To clear something out of the path, type:

  set –d (item to clear)

The basics -keys

- up arrow—Scrolls backward through the list of previously entered commands.
- down arrow—Scrolls forward through the list of previously entered commands.
- TAB—Completes file names in the command line. Type at least one character, and then press the TAB key to complete the file name. If more than one possibility exists, press the TAB key again to view all possibilities. No shift TAB round robin
- PgUp—Scrolls up a page. PgDown—Scrolls down a page.

The basics - screen

To limit screen output

- Enter the –b option after a command.
- Review the initial output screen, then do either of the following:
  - To continue to the next screen, press Enter.
  - To quit the display, press Q.

Example

```
devtree -b
```

The basics - switching to a file system

Procedure

- Using HDD, USB, virtual USB, attach a FAT16 or FAT32 formatted file system.
- Use the map –r command to refresh file system mappings.
- Enter one of the fsx file systems available, i.e. fs0 or fs1, then Enter.

Example:

Shell> map –r (gives a list of all devices)

Shell>fs0:

fs0:\>

The basics -help

- The help command can be used to list all commands :

Example:

```
help –b
alias       - Displays, creates, or deletes UEFI Shell aliases.
attrib      - Displays or changes the attributes of files or
directories.
boot        - Boots or displays boot options.
cd          - Displays or changes the current directory.
:
:
:
```

- Or a specific command :

```
echo help
```

The basics -help

ECHO [-on|-off]
ECHO [message]

-on - Enables display when reading commands from script files.
-off - Disables display when reading commands from script files.
message - Specifies a message to display.

NOTES:

1. The first form of this command controls whether script commands are displayed as they are read from the script file. If no argument is given, the current "on" or "off" status is displayed.
2. The second form prints the given message to the display.

EXAMPLES:

* To display a message string of 'Hello World':
  fs0:\> echo Hello World

The basics – redirection & piping

- Redirection can be achieved using the following :
  - > ␣ redirect standard output to target location
  - >> ␣ redirect standard output, appending to target location
  - 2> ␣ redirect error output to target location
  - 2>> ␣ redirect error output, appending to target location
- Target location can be either a traditional file on some non-volatile media or it can be a volatile environment variable!
- Piping is accomplished as follows:
  - | ␣ Pipe output of a command to another program in UCS-2 format.
  - |a ␣ Pipe output of a command to another program in ASCII format.

Shell commands – commonly used

ls (or dir)	Lists the directory contents.
cd	Changes the directory.
cp	Copies one or more files/directories
move	Moves one or more files/directories
rm	Deletes one or more files or directories.
map	Displays, resets, or deletes mappings
map –r	Commonly used to refresh mapped drives.
edit	Starts a basic text editor

Shell commands – commonly used

set	Displays/create/delete EFI environmental variables.
cls	Clears the screen / change the background color.
echo	Displays results on the screen.
help	Displays help information.
exit	Leaves the EFI shell and returns to the BIOS utility.
reset	Resets the system with a warm reboot or shutdown.
bcfg	Set/Change boot configuration

UEFI Shell Scripting

- UEFI Shell scripts are interpreted programs (usually with the extension .nsh)
- The UEFI Shell searches for shell scripts first in the current directory and then in the directories specified by the path environment variable. Shell scripts are carriage return delimited lists of shell commands that are executed (by default) from first to last.
- Normally when starting the shell, the script startup.nsh will be executed!

Scripting commands

- echo – Outputs text to the standard output device (“@” prevents echoing echo)
- exit – Terminates the currently executing script
- for…endfor – Repeatedly executes a block of script commands
- goto – Continues execution with the specified label
- if…else…endif – Conditionally executes a block of script commands
- shift – Shifts positional command-line parameters

Scripting commands - Example ‘findimage.nsh’

```
@echo -off
mode 80 25
;foundimage section is simply to locate the correct drive
cls
if exist .\ipmi.efi then
goto FOUNDIMAGE
endif
if exist fs0:\ipmi.efi then
fs0:
echo Found Update Packages on fs0:
goto FOUNDIMAGE
endif
if exist fs1:\ipmi.efi then
fs1:
echo Found Update Packages on fs1:
goto FOUNDIMAGE
endif
```

Scripting commands - Example ‘findimage.nsh’

```
if exist fs2:\ipmi.efi then
fs2:
echo Found Update Packages on fs2:
goto FOUNDIMAGE
endif
echo "Unable to find Update Packages".
echo "Please mount the drive with the update package".
echo ""
goto END
:FOUNDIMAGE
echo ========================================================
echo "This utility will allow you to update the BIOS and firmware."
echo "If you want to do all these functions, press Enter. If not, hit 'q'."
echo "if you want to update single components like BMC/BIOS separately,"
echo "please go to the EFIscripts folder and execute the specified scripts"
echo "Do NOT reboot during this process until you see Updates completed."
```

The bcfg (boot configuration) command

- This command manages the boot and driver options stored in UEFI variables and can :
  - Display the current boot (order) or device variables
  - Add a new boot or device variable
  - Delete a boot or device variable
  - Modify a boot or device variable
  - Move an option
  - Assign a hot key to a boot option

Using bcfg – display boot order

```
Shell> bcfg boot dump -b
Option: 00. Variable: Boot0003
  Desc     - AN Launcher
  DevPath  - HD(1,GPT,33A1D84B-3B2E-4090-A766-0CFEFD223D1B,0x800,0x32000)/\EFI\BOOT\LAUNCHER.EFI
  Optional- N
Option: 01. Variable: Boot0001
  Desc     - ubuntu
  DevPath  - HD(1,GPT,33A1D84B-3B2E-4090-A766-0CFEFD223D1B,0x800,0x32000)/\EFI\ubuntu\shimx64.efi
  Optional- Y
Option: 02. Variable: Boot0002
  Desc     - Windows Boot Manager
  DevPath  - HD(1,GPT,33A1D84B-3B2E-4090-A766-0CFEFD223D1B,0x800,0x32000)/\EFI\Microsoft\Boot\bootmgfw.efi
  Optional- Y
Option: 03. Variable: Boot2001
  Desc     - EFI USB Device
  DevPath  -
  Optional- Y
Option: 04. Variable: Boot2002
  Desc     - EFI DVD/CDROM
  DevPath  -
  Optional- Y
Option: 05. Variable: Boot0000
  Desc     - test
  DevPath  - PciRoot(0x0)/Pci(0x17,0x0)/Sata(0x0,0x0,0x0)/HD(1,GPT,33A1D84B-3B2E-4090-A766-0CFEFD223D1B,0x800,0x32000)/\EFI\Boot\LAUNCHER.EFI
  Optional- N
Option: 06. Variable: Boot2003
  Desc     - EFI Network
  DevPath  -
  Optional- Y

Shell> _
```

Using bcfg – display boot order

bcfg dump –b

```
Option: 00.     Variable: Boot0003
  Desc          - AN Launcher
  DevPath       - HD(1, GPT, 33A1D84B-3B2E-4090-A766-0CFEFD223D1B, 0X800,
                  0x32000) /\EFI\BOOT\LAUNCHER.EFI
  Optional      - N
Option: 01.     Variable: Boot0001
  Desc          - AN Launcher
  DevPath       - HD(1, GPT, 33A1D84B-3B2E-4090-A766-0CFEFD223D1B, 0X800,
                  0x32000) /\EFI\ubuntu\shimx64.efi
  Optional      - N
```

Using bcfg - dump explained

Option:	This is the boot order number (0 is the first)
Variable:	The variable name
Desc	The description of the boot item
DevPath	The device type, i.e. HD / USB etc. Device number, Partition GUID e.g. 33A1D84B-3B2E-4090-A766-0CFEFD223D1B, start block, size, pathname of item i.e. /\EFI\BOOT\LAUNCHER.EFI
Optional	either Y (Yes) or N (No)

Using bcfg – add an item to boot menu

bcfg boot add option1 full-pathname “description".

Example:

bcfg boot add 0 fs0:EFI\BOOT\LAUNCHER.EFI “AN Launcher".

Where

boot add	- is the option to add to the boot
option1	- position to place item (zero is very first item)
pathname	- the drive & the full pathname of the item to add
"AN Launcher “	- the description (optional)

Using bcfg – move an item in boot menu

bcfg boot mv option1 option2

Example:

bcfg boot mv 3 7 (if items where a, b, c, d, e, f, g, h, result is a, b, d, e, g, c, h)

Where

boot mv	- is the option to move an item in the boot menu
option1	- the position of the item to move
option2	- the position to move the item to

Using bcfg – delete an item in boot menu

bcfg boot rm option1

Example:

bcfg boot rm 3 (if items where a, b, c, d, e, f, g, h, result is a, b, d, e, g, h)

Where

boot mv	- is the option to move an item in the boot menu
option1	- the position of the item to move
option2	- the position to move the item to

The edit command

- The edit command opens a window in which the specified file can be edited.
- The following hot keys are used in the edit window:
  - Ctrl-E/Ctrl-W open/close help
  - Ctrl-G F1 Go to line
  - Ctrl-S F2 Save File
  - Ctrl-Q F3 Exit
  - Ctrl-F F4 Search
  - Ctrl-R F5 Search/Replace
  - Ctrl-K F6 Cut line
  - Ctrl-U F7 Paste line
  - Ctrl-O F8 Open File
  - Ctrl-T F9 File Type(Ascii/Unicode)

The edit command

- The edit window

EFI tools

- There are a small number of tools such as a python 3 implementation
- One of the most useful tools available is a memory test metest86
- The file can be downloaded from:
  - https://www.memtest86.com

Shell commands - 1

Command	Description
alias	Displays, creates, or deletes aliases (can alias commands, drives, and executables)
attrib [+|-][a|s|h|r] file dir	Displays or changes the attributes of files or directories
bcfg	Display / change / delete / modify / boot or device variables
break	Executes a debugger break point
cd	Changes the directory
cls	Clears the screen and can change background color
comp file1 file2	Compares the contents of two files up to a maximum of 10 differences
connect	Binds an EFI driver to a device and starts the driver
cp [-r][-q] src file [dst]	Copies one or more files/directories to another location -r – Copies all recursively
date	Displays the current date or sets the date in the system
dblk	Displays the contents of blocks from a block device
devices	Displays the list of devices being managed by EFI drivers
devtree	Displays the tree of devices that follow the EFI Driver Model
dh	Displays the handles in the EFI environment
disconnect	Disconnects one or more drivers from a device
dmem	Displays the contents of memory
dmpstore	Displays all NVRAM variables
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
OpenInfo	Displays the protocols on a handle and the agents

Shell commands - 2

Command	Description
edit	Edits an ASCII or UNICODE file in full screen
EfiCompress	Compresses a file
EfiDecompress	Decompresses a file
err	Displays or changes the error level
exit	Exits the EFI Shell
getmtc	Displays the current monotonic counter value
goto	Makes batch file execution jump to another location
guid	Displays all the GUIDs in the EFI environment
help [-b]	Displays commands list or verbose help of a command. –b – Displays one page at a time
hexedit	Edits with hex mode in full screen
load	Loads EFI drivers (e.g., Load ipmi.efi)
LoadBmp –w(seconds)	Displays a Bitmap file onto the screen
LoadPciRom	Loads a PCI Option ROM image from a file
ls [-b|-r|-a]	Display a list of files
map [-r|-v|-d]	Displays, resets, or deletes mappings with verbose option
memmap	Displays the memory map
mkdir dirname	Creates a directory at current location
mm	Displays or modifies MEM/IO/PCI
mode [row col]	Displays or changes console output
mount BlkDevice [sname]	Mounts a file system on a block device. The mounted names are lost at next map –r.
mv src dest	Moves one or more files/directories to destination
OpenInfo	Displays the protocols on a handle and the agents

Shell commands - 3

Command	Description
pause	Prints a message and suspends for keyboard input. Options are q to quit, any other key resume script.
pci	Displays PCI devices or PCI function configuration space
reconnect	Reconnects one or more drivers from a device
reset [-w|-s] [string]	Resets the system with warm reboot or complete shutdown. Can pass a string to the reset service.
rm [-q] file|dir	Deletes one or more files or directories -q – Does not prompt for a confirmation
set [-d|-v|-b] [sname [valu	Displays, deletes, changes, or creates environment variables
setsize	Sets the size of a file
stall	Stalls the processor for some microseconds
time	Displays the current time or sets the time of the system
touch [-r] filename	Sets the time and date of a file to the current time and date
type [-a|-u] file	Displays the contents of a file (ASCII or Unicode)
unload	Unloads a protocol image
ver	Displays the version information
vol [fs] [VolLabel]	Displays volume information of the file system

Questions?

Thank You