



Hewlett Packard
Enterprise

UEFI Shell User Guide for HPE ProLiant Gen10 Servers and HPE Synergy

Abstract

This document details how to access and use the Unified Extensible Firmware Interface (UEFI) Shell that is embedded in the system ROM of all ProLiant Gen10 servers and HPE Synergy compute modules. It is for the person who installs, administers, and troubleshoots servers and storage systems.

Part Number: 881332-003
Published: March 2019
Edition: 1

Notices

The information contained herein is subject to change without notice. The only warranties for Hewlett Packard Enterprise products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. Hewlett Packard Enterprise shall not be liable for technical or editorial errors or omissions contained herein.

Confidential computer software. Valid license from Hewlett Packard Enterprise required for possession, use, or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

Links to third-party websites take you outside the Hewlett Packard Enterprise website. Hewlett Packard Enterprise has no control over and is not responsible for information outside the Hewlett Packard Enterprise website.

Acknowledgments

Microsoft® and Windows® are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

Intel®, Itanium®, Pentium®, Intel Inside®, and the Intel Inside logo are trademarks of Intel Corporation in the United States and other countries.

UEFI® is a registered trademark of the UEFI Forum, Inc.

Linux® is the registered trademark of Linus Torvalds in the U.S. and other countries.

Contents

Using the UEFI Shell	6
Commands and capabilities.....	6
Accessing the Embedded UEFI Shell.....	6
Accessing the UEFI Shell from a serial console connection.....	7
UEFI Shell command reference.....	8
Custom Hewlett Packard Enterprise Shell commands.....	8
Command-line syntax examples.....	8
Command-line completion.....	8
Getting started with the UEFI Shell.....	8
Controlling command output.....	9
Viewing output that scrolled off the screen.....	9
Limiting output to one screen at a time.....	9
Enabling or disabling global page breaks.....	10
Displaying detailed output.....	10
Displaying summary output.....	11
Suppressing the confirmation prompt.....	11
Displaying command help.....	12
Displaying all command help.....	12
Displaying help for a specific command.....	12
Displaying all help for commands beginning with a specific character.....	13
Updating the firmware from the UEFI Embedded Shell	15
Switching to a file system from the Shell.....	15
Using common setup and configuration commands	15
UEFI Shell commands.....	16
acpiview.....	16
ahsdownload.....	17
alias.....	19
attrib.....	20
base64.....	21
boot.....	22
cd.....	23
cls.....	24
comp.....	26
compress.....	27
connect.....	28
cp.....	29
date.....	31
dblk.....	32
devices.....	32
devtree	34
dh.....	35
diags.....	37
disconnect.....	38
dmem.....	39
dmpstore.....	39
drivers.....	40
echo.....	42
edit.....	43

eficompress.....	43
efidecompress.....	44
exit.....	44
ftp.....	45
fwupdate.....	46
getmtc.....	47
goto.....	47
hash.....	47
help.....	48
ifconfig.....	49
ifconfig6.....	50
imlview.....	51
ipmctl.....	52
load.....	52
ls/dir.....	53
map.....	56
memmap.....	58
mkdir.....	62
mode.....	62
mv.....	63
openinfo.....	64
parse.....	64
partitions.....	65
pause.....	65
pci.....	66
ping.....	67
ping6.....	68
ramdisk.....	68
reconnect.....	70
reset.....	71
restclient.....	71
rm/del.....	73
secboot.....	74
sermode.....	76
set.....	77
setsize.....	79
shift.....	79
smbiosview.....	80
stall.....	80
sysconfig.....	81
sysconfig attributes.....	85
sysinfo.....	85
tftp.....	86
time.....	87
timezone.....	88
tlsconfig.....	89
touch.....	91
type.....	91
unload.....	92
ver.....	93
vol.....	93
Webclient.....	94
Running and editing UEFI Shell scripts.....	97
Methods for invoking scripts.....	97

UEFI Shell Script Auto-Start configuration in the System Utilities.....	97
Manually invoking a Shell script.....	97
Exporting and importing settings to files.....	97
Editing Shell scripts.....	97
Sample UEFI Shell scripts.....	98
Application source code script.....	98
Start-up script	99
UEFI Programming Model.....	103
UEFI Shell command status codes.....	108
Websites.....	109
Support and other resources.....	110
Accessing Hewlett Packard Enterprise Support.....	110
Accessing updates.....	110
Customer self repair.....	111
Remote support.....	111
Warranty information.....	111
Regulatory information.....	112
Documentation feedback.....	112

Using the UEFI Shell

The system BIOS in this server includes an Embedded UEFI Shell in the ROM. Based on the UEFI Shell specification, the Shell environment provides an API and a CLI that enables scripting, file manipulation, and obtaining system information. The Shell also runs other UEFI applications. These features enhance the capabilities of the UEFI System Utilities. Access to the Embedded UEFI Shell is enabled by default.

Commands and capabilities

The following features are available in the UEFI Shell:

- Configuration commands
- Scripting
 - nsh files with standard scripting constructs, including `if`, `else`, `endif`, `shift`, and `for/endfor`
 - An `echo` command
 - A `startup.nsh` auto start file similar to `Autoexec.bat`
 - A standard format output (`-sfo`) option for several commands that enables you to parse comma-separated output using a `parse` command.
- File manipulation
 - The ability to read any FAT16 and FAT32 file.
 - Standard file operations commands, such as `md`, `cd`, `cp/copy`, `del`, `dir/ls`, `attrib`, and `touch`
 - File editing (`edit`) and viewing (`type`)
 - Input and output redirection from and to consoles and files

Accessing the Embedded UEFI Shell

NOTE: UEFI Shell requires BIOS administrator authorization. If the BIOS administrator password has been set, the server will prompt for the password to be entered prior to launching the UEFI Shell and UEFI Shell startup scripts. For information on setting the administrator password, see **sysconfig** or the Server Security section of the UEFI System Utilities user guide.

Procedure

1. Ensure that the Embedded UEFI Shell is enabled. The default setting is enabled.
2. Optional: Add the **Embedded UEFI Shell** to the **UEFI Boot Order** list.
3. Optional: Enable UEFI Shell script auto execution.
4. Change the **Embedded UEFI Shell** entry in the **UEFI Boot Order** list.

You can access the Shell:

- During server POST, if the Embedded UEFI Shell is added to the UEFI Boot Order list, press **F11** (Boot Menu) in the POST screen. This option is available only if the Embedded UEFI Shell is added to the UEFI Boot Order list.
- From the **System Utilities** screen, select **Embedded Applications > Embedded UEFI Shell** and press **Enter**.
- Using a serial console connection.

Accessing the UEFI Shell from a serial console connection

Prerequisite

Your terminal software must use a Unicode character set (for example: UTF-8).

NOTE: When accessing the Shell from a serial console, English is the only available language for input and output. The Virtual Serial Port (VSP) does not support the Browser Selection GUI mode.

Procedure

1. Boot the server.
2. Open a connection in an SSH client application using the iLO IP address of the server.
3. Leave the SSH port as **22**.
4. At the login prompt, enter your user name and password.
The `hpiLO->` prompt appears.
5. Enter `vsp`, and then press **Enter** to open the virtual serial port.
The `Shell>` prompt appears.
6. Enter the commands required to complete your task.
7. Enter `exit` and exit the Shell.

Example

```
login as: username
@<ip address>'s password: password
User: logged-in to <server path> / <server MAC address>
iLO 5 Standard 1.30 at January 16 2017
Server Name:
Server Power: On

hpiLO-> vsp

Virtual Serial Port Active: COM2

Starting virtual serial port.
Press 'ESC (' to return to the CLI Session.

Shell>
```

UEFI Shell command reference

The following information provides an overview of the UEFI Shell commands, a description of each command, its syntax, and examples of how to use it.

Custom Hewlett Packard Enterprise Shell commands



This symbol indicates a command that is a custom Hewlett Packard Enterprise addition to the commands provided by the UEFI Shell specification.

Command-line syntax examples

The following lists examples of how to interpret command syntax.

exit

Enter **exit**.

date [*mm/dd*/{*yy*|*yyyy*}] [-*sfo*]

To display the current date, do one of the following:

- Enter **date** with no optional parameters.
- Enter **date** followed by **-sfo** to specify standard formatted output. To set a specific date on the system, enter **date** followed by the date parameters in one of the following formats:
 - *mm/dd/yy*
 - *mm/dd/yyyy*

This syntax indicates that the month (*mm*) and day (*dd*) parameters are optional, but either a two-digit year (*yy*) or four-digit year (*yyyy*) parameter is required for setting a date.

eficompress *infile outfile*

Enter **eficompress** followed by the filename for the uncompressed input file as the *infile* parameter, and the compressed output file as the *outfile* parameter.

Command-line completion

The following command-line completion keys simplify entering and repeating commands.

- 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.

Getting started with the UEFI Shell

You can use basic UEFI Shell commands for:

- Controlling command output
- Displaying command help
- Switching to a file system from the Shell
- Using common setup and configuration commands

Controlling command output

Many UEFI Shell commands support optional parameters for controlling how the output displays on the screen. The following sections explain these options and provide examples of how to use them.

Viewing output that scrolled off the screen

When the shell outputs too much information for the screen, you can scroll up and down.

Procedure

1. Press one of the following to display more output.
 - **PgUp**: Scroll up a page.
 - **PgDown**: Scroll down a page.

Limiting output to one screen at a time

Procedure

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

Example

```
Shell> devtree -b
Ctrl[03] Fv(6522280D-28F9-4131-ADC4-F40EBFA45864)
Ctrl[04] Fv(770BF9B6-8AFE-4F4C-85E5-893FC3D2606C)
Ctrl[05] Fv(27A72E80-3118-4C0C-8673-AA5B4EFA9613)
Ctrl[06] MemoryMapped(0xB,0xFFD40000,0xFFD6FFFF)
Ctrl[07] Fv(5A515240-D1F1-4C58-9590-27B1F0E86827)
Ctrl[08] Fv(5E2363B4-3E9E-4203-B873-BB40DF46C8E6)
Ctrl[09] Fv(CDBB7B35-6833-4ED6-9AB2-57D2ACDDF6F0)
Ctrl[5F] PciRoot(0x0)
Ctrl[195] PciRoot(0x0)/Pci(0x0,0x0)
Ctrl[196] PciRoot(0x0)/Pci(0x2,0x0)
Ctrl[197] PciRoot(0x0)/Pci(0x2,0x0)/Pci(0x0,0x0)
Ctrl[198] PciRoot(0x0)/Pci(0x2,0x1)
Ctrl[199] PciRoot(0x0)/Pci(0x2,0x2)
Ctrl[19A] PciRoot(0x0)/Pci(0x2,0x3)
Ctrl[19B] PciRoot(0x0)/Pci(0x3,0x0)
Ctrl[19C] PciRoot(0x0)/Pci(0x3,0x1)
```

```

Ctrl[19D] PciRoot(0x0)/Pci(0x3,0x2)
Ctrl[19E] PciRoot(0x0)/Pci(0x3,0x3)
Ctrl[19F] PciRoot(0x0)/Pci(0x4,0x0)
Ctrl[1A0] PciRoot(0x0)/Pci(0x4,0x1)
Ctrl[1A1] PciRoot(0x0)/Pci(0x4,0x2)
Ctrl[1A2] PciRoot(0x0)/Pci(0x4,0x3)
Ctrl[1A3] PciRoot(0x0)/Pci(0x4,0x4)
Ctrl[1A4] PciRoot(0x0)/Pci(0x4,0x5)
Ctrl[1A5] PciRoot(0x0)/Pci(0x4,0x6)
Ctrl[1A6] PciRoot(0x0)/Pci(0x4,0x7)
Ctrl[1A7] PciRoot(0x0)/Pci(0x5,0x0)
Ctrl[1A8] PciRoot(0x0)/Pci(0x5,0x1)
Ctrl[1A9] PciRoot(0x0)/Pci(0x5,0x2)
Ctrl[1AA] PciRoot(0x0)/Pci(0x5,0x4)
Press ENTER to continue or 'Q' break:

```

Enabling or disabling global page breaks

To enable or disable global output pagination, set or clear the `pagebreak` environment variable. This command creates a UEFI environmental variable that is retained until a server reboot.

To enable global output page breaks:

```
fs0:\> set -v pagebreak 1
```

To disable global output page breaks:

```
fs0:\> set -v pagebreak 0
```

Displaying detailed output

Shell commands supporting this option include:

- `dh`
- `help`
- `map`

Example

```

Shell> dh -v -b
01: LoadedImage
02: Decompress
03: UnknownDevice DevicePath Fv(6522280D-28F9-4131-ADC4-F40EBFA45864) UnknownDevice
04: UnknownDevice DevicePath Fv(770BF9B6-8AFE-4F4C-85E5-893FC3D2606C) UnknownDevice
05: UnknownDevice DevicePath Fv(27A72E80-3118-4C0C-8673-AA5B4EFA9613) UnknownDevice
06: UnknownDevice DevicePath Fv(29A72E80-7BFE-4101-8459-AB5B3EFA4271) UnknownDevice
07: UnknownDevice DevicePath Fv(5A515240-D1F1-4C58-9590-27B1F0E86827) UnknownDevice
08: UnknownDevice DevicePath Fv(5E2363B4-3E9E-4203-B873-BB40DF46C8E6) UnknownDevice
09: UnknownDevice DevicePath Fv(CDBB7B35-6833-4ED6-9AB2-57D2ACDDF6F0) UnknownDevice
0A: UnknownDevice UnknownDevice
0B: ImageDevicePath LoadedImage
0C: UnknownDevice Pcd
0D: ImageDevicePath LoadedImage
0E: UnknownDevice UnknownDevice
0F: ImageDevicePath LoadedImage
10: UnknownDevice ImageDevicePath LoadedImage
11: UnknownDevice UnknownDevice UnknownDevice ImageDevicePath LoadedImage

```

```
12: ImageDevicePath LoadedImage
13: UnknownDevice
14: UnknownDevice ImageDevicePath LoadedImage
15: UnknownDevice
16: UnknownDevice
17: ImageDevicePath LoadedImage
18: ImageDevicePath LoadedImage
19: UnknownDevice ImageDevicePath LoadedImage
1A: ImageDevicePath LoadedImage
1B: UnknownDevice
1C: ImageDevicePath LoadedImage
1D: UnknownDevice
Press ENTER to continue or 'Q' break:
```

Displaying summary output

Example

```
Shell> ver
Embedded UEFI Shell v2.1
(C) Copyright 1982–201i Hewlett Packard Enterprise Development LP
UEFI v2.60 (HPE, 0x00012000)
HPE ProLiant System ROM U31 v1.40 (01/16/2018)

Shell> ver -t
```

Suppressing the confirmation prompt

Shell commands supporting this option include:

- cp
- rm/del

Example

```
fs0:\> rm -q test\temp*
removing fs0:\test\temp1\temp1.txt
- [ok]
removing fs0:\test\temp1\boot\nshell.efi
- [ok]
removing fs0:\test\temp1\boot
- [ok]
removing fs0:\test\temp1
- [ok]
removing fs0:\test\temp2\temp2.txt
- [ok]
removing fs0:\test\temp2
- [ok]
```

Displaying command help

Displaying all command help

Example

```
Shell> 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.
cls           - Clears standard output and optionally changes background
              color.
comp          - Compares the contents of two files on a byte for byte basis.
compress      - Compresses and decompresses files to and from zip files
              using MiniZip.
connect       - Binds a driver to a specific device and starts the driver.
cp           - Copies one or more files or directories to another
              location.
date          - Displays and sets the current date for the system.
devices       - Displays the list of devices managed by UEFI drivers.
devtree       - Displays the UEFI Driver Model compliant device tree.
dh           - Displays the device handles in the UEFI environment.
dmem          - Displays the contents of system or device memory.
drivers       - Displays the UEFI driver list.
echo          - Controls script file command echoing or displays a message.
edit          - Displays a full screen editor for ASCII or UCS-2 files.
eficompress   - Compresses a file using UEFI Compression Algorithm.
efidecompress - Decompresses a file using UEFI Decompression Algorithm.
else          - Identifies the code executed when 'if' is FALSE.
endfor        - Ends a 'for' loop.
endif         - Ends the block of a script controlled by an 'if' statement.
exit          - Exits the UEFI Shell or the current script.
for           - Starts a loop based on 'for' syntax.
Press ENTER to continue or 'Q' break:
```

Displaying help for a specific command

Use any of the following options.

- `help commandname`
- `? commandname`
- `commandname -?`

Example

```
Shell> help ls -b
Lists a directory's contents or file information.
```

```
LS [-r] [-a[attrib]][-sfo][file]format:
-r      - Displays recursively (including subdirectories)
-a      - Displays only those files with the attributes of type attrib.
         If no attributes are listed, all files will be listed. If -a is
         not specified, all non-system and non-hidden files will be
         listed.
-sfo    - Displays information in Standard-Format Output.
```

attrib - File attribute list:
 a - Archive
 s - System
 h - Hidden
 r - Read-only
 d - Directory
file - Name of file or directory (wildcards are permitted)

NOTES:

1. This command lists directory contents or file information. If no file name or directory name is specified, the current working directory is assumed.
2. The contents of a directory are listed if all of the following are true:
 - If option -r is not specified
 - If no wildcard characters are specified in the file parameter
 - If file represents an existing directory
3. In all other cases, the command functions as follows:
 - All files/directories that match the specified name are displayed.
 - The -r flag determines whether a recursive search is performed.
 - The option flag -a[attrib] tells the command to display only those files with the attributes that are specified by [attrib].

Displaying all help for commands beginning with a specific character

Example

```
Shell> help a*
```

Displays, creates, or deletes UEFI Shell aliases.

ALIAS [-d|-v] [alias-name] [command-name]
-d - Delete an alias. command-name must not be specified.
-v - Make the alias volatile.
alias-name - Alias name
command-name - Original command's name or path.

NOTES:

1. This command displays, creates, or deletes aliases in the UEFI Shell environment.
2. An alias provides a new name for an existing UEFI Shell command or UEFI application. Once the alias is created, it can be used to run the command or launch the UEFI application.
3. There are some aliases that are predefined in the UEFI Shell environment. These aliases provide the MS-DOS and UNIX equivalent names for the file manipulation commands.
4. Aliases will be retained even after exiting the shell unless the -v option is specified. If -v is specified the alias will not be valid after leaving the shell.

EXAMPLES:

- * To display all aliases in the UEFI Shell environment:
 Shell> alias
- * To create an alias in the UEFI Shell environment:
 Shell> alias shutdown "reset -s"

- * To delete an alias in the UEFI Shell environment:
Shell> alias -d shutdown
- * To add a volatile alias in the current UEFI environment, which has a star * at the line head. This volatile alias will disappear at next boot.
Shell> alias -v fs0 floppy

Displays or changes the attributes of files or directories.

ATTRIB [+a|-a] [+s|-s] [+h|-h] [+r|-r] [file...] [directory...]

```
+a|-a      - Set or clear the 'archive' attribute
+s|-s      - Set or clear the 'system' attribute
+h|-h      - Set or clear the 'hidden' attribute
+r|-r      - Set or clear the 'read-only' attribute
file       - File name (wildcards are permitted)
directory  - Directory name (wildcards are permitted)
```

NOTES:

1. Four attribute types are supported in the UEFI file system:
 - Archive [A]
 - System [S]
 - Hidden [H]
 - Read only [R]
2. If a file (in general meaning) is a directory, it is also shown to have the attribute [D].
3. If any file in the file list that is specified in the command line does not exist, attrib will continue processing the remaining files while reporting the error.
4. If no attributes parameters are specified, the current attributes of the specified files or directories will be displayed.
5. If no files or directories are specified, the command applies to all files and sub-directories within the current directory.

EXAMPLES:

- * To display the attributes of a directory:
Shell> :\> attrib fs0:\
- * To display the attributes of all files and sub-directories in the current directory:
fs0:\> attrib *
- * To add the system attribute to all files with extension '.efi':
fs0:\> attrib +s *.efi
- * To remove the read only attribute from all files with extension '.inf':
fs0:\> attrib -r *.inf

Updating the firmware from the UEFI Embedded Shell

Procedure

1. Access the System ROM Flash Binary component for your server from the Hewlett Packard Enterprise Support Center (<http://www.hpe.com/support/hpesc>).
2. Copy the binary file to a USB media or iLO virtual media.
3. Attach the media to the server.
4. Boot to the UEFI Embedded Shell.
5. To obtain the assigned file system volume for the USB key, enter `map -r`.
6. Change to the file system that contains the System ROM Flash Binary component for your server. Enter one of the `fsx` file systems available, such as `fs0:` or `fs1:`, and press **Enter**.
7. Use the `cd` command to change from the current directory to the directory that contains the binary file.
8. Flash the system ROM by entering `fwupdate -d BIOS -f filename`.
9. Reboot the server. A reboot is required after the firmware update in order for the updates to take effect and for hardware stability to be maintained.

Switching to a file system from the Shell

To switch from the Shell to a file system before executing commands requiring file input or output:

Procedure

1. Using HDD, USB, or iLO virtual USB, attach an FAT16 or FAT32 formatted file system.
2. Use the `map -r` command to refresh file system mappings.
3. Enter one of the `fsx` file systems available, such as `fs0:` or `fs1:`, and then press **Enter**.

The prompt changes to `fsx>`, where `x` is the number of the file system selected. Files can now be accessed and written to any writable files in the specified file system.

Example

```
Shell> map -r
Shell>fs0:
fs0:\>
```

NOTE: Output examples use the `fs0:\>` prompt to show where a file system would be accessed from the Shell.

Using common setup and configuration commands

How do I enter the Embedded UEFI Shell?

See [Accessing the Embedded UEFI Shell](#).

How do I determine if a server has UEFI boot options?

See [sysconfig](#).

How do I configure the time zone?

See [timezone](#).

How do I create a temporary RAM disk from the Embedded UEFI Shell?

See [ramdisk](#).

How do I dump the Integrated Management Log (IML) to a selected file on a RAM disk from the Embedded UEFI Shell, and confirm that the file is on the RAM disk?

See [imlview](#).

How do I reset one or all mappings between user-defined names and a device handles?

See [map](#).

How do I display all devices that are compliant with the UEFI Driver Model?

See [devtree](#).

How do I exit the Embedded UEFI Shell back to the System Utilities?

See [exit](#).

UEFI Shell commands

The following describes the components of each command listing and then lists UEFI Shell commands in alphabetical order.

Prerequisite

Syntax

Command syntax, including required and optional parameters.

Description

Brief description of how the command is used.

Options

Description of syntax parameters and variables.

Usage

Detailed description of command usage.

Examples

One or more examples of command usage.

NOTE: User input is indicated in **bold** in command usage examples.

Output details

Descriptions of the command display fields, when applicable.

acpiview

Syntax

```
acpiview [-s table signature | -a] [-p] [-d] [-v] [-b] [-n] ID [-o] name [-t] name
```

Description



Displays and disassembles Advanced Configuration and Power Interface (ACPI) tables.

Options

-s *table signature*

Displays the specified ACPI table. Except for the RSDP descriptor, the *table signature* must be a defined value in the ACPI spec. Use RSDP instead of RSD PTR.

-a

Decodes and displays all ACPI static tables.

-p

Decodes and displays all ACPI static tables, and disassembles the AML code in the descriptor tables.

-d

Dumps ACPI table hexadecimal data while decoding tables.

-v

Displays ACPI table verbose data while decoding tables.

-n *ID*

Displays the descriptor table specified by an OEM table *ID*.

-b

Displays one screen at a time.

-o *name*

-t *name*

Dumps the binary ACPI data for descriptor tables to the file named.

Usage

Include one of the *-s*, *-a*, or *-p* parameters.

The *-p* option, with no other options specified, dumps, and then parses all ACPI tables.

If *-v*, *-d*, or *-o* are not specified, verbose mode is automatically enabled.

The *-n* option requires a full 8-character OEM table ID string. To include spaces, use quotation marks ("").

The *-o* option requires a writable file system.

Examples

To dump and parse all ACPI tables:

```
fs0:\> acpiview -p
```

To dump and parse only the DSDT:

```
fs0:\> acpiview -p -s DSDT
```

To dump and parse the SSDT table with OEM table ID `PCISSDT`:

```
fs0:\> acpiview -p -s SSDT -n PCISSDT
```

To dump all ACPI tables with no parsing of AML:

```
fs0:\> acpiview -a
```

ahsdownload

Syntax

```
ahsdownload[-n filename][[-u serialnumber][[-s startdate][[-e enddate][[-case
casenumber][[-name contactname][[-email emailaddress][[-phone phonenumber][[-
company companyname][[-email emailaddress][[-a][[-l][[-q]
```

Description



Downloads an Active Health System (AHS) log file used for problem resolution.

Options

-n *filename*

Specifies a name for the AHS file.

-u *serialnumber*

Specifies a system serial number.

-s *startdate*

Specifies the start date (in YYYY-MM-DD format) for retrieving data.

-e *enddate*

Specifies the end date (in YYYY-MM-DD format) for retrieving data.

-case *casenumber*

Adds a support case number to the customer information file.

-name *contactname*

Adds a contact name to the customer information file.

-phone *phonenumber*

Adds a phone number to the customer information file.

-email *emailaddress*

Adds an email address to the customer information file.

-a

Downloads all data from the AHS log.

-l

Displays AHS download date range information.

-q

Performs the download in quiet mode without user prompts.

Usage

AHS monitors and records changes in the server hardware and system configuration. The log files can assist you in diagnosing problems and delivering rapid resolution.

AHS does not collect information about your operations, finances, customers, employees, partners, or data center (for example, IP addresses, host names, user names, and passwords are not collected). By downloading and sending the AHS data to Hewlett Packard Enterprise, you agree to have Hewlett Packard Enterprise use the data for analysis, technical resolution, and quality improvements.

This command mounts an AHS partition, downloads AHS log files, and bundles them into an `.ahs` package file.

If a start date and end date are not specified, by default AHS data related to the last seven days is downloaded. If a start date without an end date is specified, AHS uses the current date as an end date for the download. If a start date and end date are specified, only AHS data in that date range, including start

and end dates, is downloaded. If a file name is not specified, a default file name is generated. For instance, `HPE_XXXXXXXXXX_20140821.ahs`.

Examples

To display available AHS download date range information:

```
fs0:\> ahsdownload -l
```

To download an AHS file to the default location with a log spanning seven days (the default range):

```
fs0:\> ahsdownload
```

To download an AHS file with a specific date range:

```
fs0:\>ahsdownload -n ahstestfile.ahs -s 2015-04-02 -e 2015-04-07
```

To download an AHS file to the default location with a complete log and customer information:

```
fs0:\> ahsdownload -a -case 1234 -name Joe -phone 123-456-7890  
-email ahstest@testing.com -company MyCompany
```

To upload the entire AHS file to a network location:

```
fs0:\> ahsdownload -n http://www.example.com/upload/ahstestfile.ahs
```

alias

Syntax

```
alias [-d | -v] [alias-name] [command-name]
```

Description

Displays, creates, or deletes aliases in the UEFI Shell environment.

Options

-d

Deletes an alias. Command name must not be specified.

-v

Makes an alias volatile.

alias-name

Specifies an alias name.

command-name

Specifies the original file name or path.

Usage

An alias provides a new name for an existing UEFI Shell command or UEFI application. Once the alias is created, it can be used to run the command or launch the UEFI application.

There are some aliases that are predefined in the UEFI Shell environment. These aliases provide the MS-DOS and UNIX equivalent names for the file manipulation commands.

Aliases are retained even after exiting the shell unless the `-v` option is specified. If `-v` is specified, the alias is not valid after leaving the Shell.

Examples

To display all aliases in the UEFI Shell environment:

```
Shell> alias  
md : mkdir
```

```
rd : rm
myguid : guid
```

To create an alias in the UEFI Shell environment:

```
Shell> alias myguid guid
Shell> alias
md : mkdir
rd : rm
myguid : guid
```

To delete an alias in the UEFI Shell environment:

```
Shell> alias -d myguid
Shell> alias
md : mkdir
rd : rm
```

To add a volatile alias in the current UEFI environment, which has a star * at the line head. This volatile alias disappears at the next boot:

```
Shell> alias -v fs0 floppy
Shell> alias
md : mkdir
rd : rm
* fs0 : floppy
```

attrib

Syntax

```
attrib[+a|-a][+s|-s][+h|-h][+r|-r][file...][directory...]
```

Description

Displays, sets, or changes the attributes of files or directories.

Options

[+a|-a]

Sets or clears the `archive` attribute.

[+s|-s]

Sets or clears the `system` attribute.

[+h|-h]

Sets or clears the `hidden` attribute.

[+r|-r]

Sets or clears the `read-only` attribute.

file...

Specifies the file name. Wildcards are permitted.

directory...

Specifies the directory name. Wildcards are permitted.

Usage

The following four attribute types are supported in the UEFI file system:

- Archive—A
- System—S
- Hidden—H
- Read only—R

If a file is a directory, it is also shown to have the attribute D.

If any file in the file list that is specified in the command line does not exist, `attrib` continues processing the remaining files while reporting the error.

If no file or directory is specified, all of the files in the current directory are displayed.

If no attribute is specified, the attributes of the files are displayed.

Examples

To display the attributes of a directory:

```
fs0:\> attrib fs0:\
attrib: D fs0:\
```

To display the attributes of all files and subdirectories in the current directory:

```
fs0:\> attrib *
\attrib: AS fs0:\serial.efi
attrib: DA fs0:\test1
attrib: A HR fs0:\bios.inf
attrib: A fs0:\VerboseHelp.txt
attrib: AS fs0:\IsaBus.efi
```

To add the `system` attribute to all files with extension `.efi`:

```
fs0:\> attrib +s *.efi
```

To remove the `read-only` attribute from all files with extension `.inf`:

```
fs0:\> attrib -r *.inf
\attrib: A H fs0:\bios.inf
```

base64

Syntax

```
base64 value value...
```

```
base64 -d value value...
```

Description



Provides base64 encoding and decoding.

Options

value

Specifies one or more arguments to encode or decode.

-d

Decodes arguments. Each result is displayed on a new line.

Examples

To encode one argument:

```
Shell> base64 TestValue
```

To decode one argument:

```
Shell> base64 -d VGVzdFZhbHVl
```

To encode multiple arguments:

```
Shell> base64 Value1 Value2
```

To decode multiple arguments:

```
Shell> base64 -d VmFsdWUx VmFsdWUy
```

boot

Syntax

```
boot[-d[-sfo]][-a num]|-r  
[-n num | -all | -pxe | -iscsi | -fcoe | -url | -http]
```

Description



Boots or displays UEFI boot options.

Options

-d

Displays UEFI boot options in order.

-n

Boots a specific UEFI boot option.

-a

Enables a specific UEFI boot option.

-i

Disables a specific UEFI boot option.

num

Specifies an option number to boot. This is a four-digit hex value for each option.

-all

Boots all UEFI boot options in order.

-pxe

Boots all UEFI PXE boot options in order.

-iscsi

Boots all UEFI PXE boot options in order.

-fcoe

Boots all UEFI FCoE boot options in order.

-url

Boots all UEFI URL boot options in order.

-http

Boots all UEFI HTTP boot options in order.

-r

Refreshes UEFI boot options.

-sfo

Displays information in standard formatted output.

Usage

This command cannot display or boot Legacy BIOS boot options. You can use it to configure a seamless transition from one UEFI boot target to another (such as a downloaded OS image) without the need for a reboot.

Examples

To display all UEFI boot options in order:

```
Shell> boot -d
```

To refresh boot options in the Boot Order List, enabling newly-added or removed devices to take effect:

```
Shell> boot -r
```

To refresh boot options and then display all UEFI boot options in order:

```
Shell> boot -r -d
```

To boot UEFI boot options in order:

```
Shell> boot -all
```

To boot UEFI PXE boot options in order:

```
Shell> boot -pxe
```

To boot the UEFI boot option with option number 0004:

```
Shell> boot -n 0004
```

To boot UEFI HTTP boot options in order:

```
Shell> boot -http
```

cd

Syntax

```
cd[path]
```

Description

Displays or changes the current directory.

Options

path

Specifies the relative or absolute directory path.

Usage

If a file system mapping is specified, the current working directory is changed for that device. Otherwise, the current working directory is changed for the current device.

If *path* is not present, the current working directory (including file system mapping) is displayed to standard output.

The following table describes the conventions that are used to refer to the directory, its parent, and the root directory in the UEFI Shell environment.

Table 1: Directory name conventions

Convention	Description
.	Current directory.
..	Parent of the current directory.
\	Root of the current file system.

The current working directory is maintained in the environment variable `%cwd%`.

Examples

To change the current file system to the mapped `fs0` file system:

```
Shell> fs0:
```

To change the current directory to subdirectory `efi`:

```
fs0:\> cd efi
```

To change the current directory to the parent directory (`fs0:\`):

```
fs0:\efi> cd ..
```

To change the current directory to `fs0:\efi\tools`:

```
fs0:\> cd efi\tools
```

To change the current directory to the root of the current `fs` (`fs0`):

```
fs0:\efi\tools> cd \  
fs0:\>
```

NOTE: Changing volumes with `cd` does not work. For example:

```
fs0:\efi\tools> cd fs1:\
```

First enter `fs1:`, and then `cd` to the directory you want.

To move between volumes and maintain the current path:

```
fs0:\> cd \efi\tools  
fs0:\efi\tools> fs1:  
fs1:\> cd tmp  
fs1:\tmp> cp fs0:*. * .
```

This sequence copies all of the files in `fs0:\efi\tools` into the `fs1:\tmp` directory.

cls

Syntax `cls[background][foreground] | [-sfo]`

Description Clears the console output and optionally changes the background and foreground color.

Options

background

Specifies a new background color from the following options:

- 0—Black
- 1—Blue
- 2—Green
- 3—Cyan
- 4—Red
- 5—Magenta
- 6—Yellow
- 7—Light gray

foreground

Specifies a new foreground color from the following options:

- 0—Black
- 1—Blue
- 2—Green
- 3—Cyan
- 4—Red
- 5—Magenta
- 6—Yellow
- 7—Light gray
- 8—Dark gray
- 9—Light blue
- 10—Light green
- 11—Light cyan
- 12—Light red
- 13—Light magenta
- 14—Yellow
- 15—White

-sfo

Displays current console color settings in Standard Format Output

Usage If background color is not specified, or if background and foreground colors are not specified, the colors do not change. When `-sfo` flag is specified, console output is not cleared. The console displays current console foreground and background attribute settings.

Examples

To clear the standard output without changing the background or foreground color:

```
fs0:\> cls
```

To clear the standard output and change the background color to cyan:

```
fs0:\> cls 3 To clear the standard output and change the background to black and the foreground to white:
```

```
fs0:\> cls 0 15
```

comp

Syntax

```
comp[-b] file1 file2
```

Description

Compares the contents of two files on a byte for byte basis.

Options

-b

Displays one screen at a time.

file1

Specifies the first file name. Directory names or wildcards are not permitted.

file2

Specifies the second file name. Directory names or wildcards are not permitted.

Usage

This command displays up to 10 differences between the two files. For each difference, up to 32 bytes from the location where the difference starts are dumped. The UEFI Shell exits immediately if the lengths of the compared files are different.

Examples

To compare two files with different lengths:

```
fs0:\> comp bios.inf legacy.inf
Compare fs0:\bios.inf to fs0:\legacy.inf
Difference #1: File sizes mismatch
[difference(s) encountered]
```

To compare two files with the same contents:

```
fs0:\> comp bios.inf rafter.inf
Compare fs0:\bios.inf to fs0:\rafter.inf
[no difference encountered]
```

To compare two files with the same length but different contents:

```
fs0:\> comp bios.inf bios2.inf
Compare fs0:\bios.inf to fs0:\bios2.inf
Difference #1:
File1: fs0:\bios.inf
00000000: 5F * *
File2: fs0:\bios2.inf
00000000: 33 *3*
Difference #2:
File1: fs0:\bios.inf
0000000C: 00 00 00 00 *....*
File2: fs0:\bios2.inf
0000000C: 25 32 03 03 *%2..*
[difference(s) encountered]
```

compress

Syntax

```
compress[-z][-s][-ex][-c1 0-9][-o output_file][file...][-sfo]
```

```
compress[-u][-s][-ex][-p password][-od output_directory][-i input_file][-sfo]
```

Description



Compresses and decompresses files to and from zip files using MiniZip, reducing data transferred over the network.

Options

-z

Compresses the specified files into a zip file.

-u

Decompresses files from the input zip file.

-s

Displays information in silent mode, with no output messages.

-ex

Excludes the path from the file name when adding files to or extracting files from the zip file.

-c1 0-9

Selects a compression level: 0 compresses faster; 9 (default) compresses better.

-o *output_file*

Specifies an output file.

file...

Specifies one or more files to zip.

-p *password*

Specifies a password for the zip file.

-od *output_directory*

Specifies a directory where files from the zip file are to be extracted.

-i *input_file*

Specifies an input file.

-sfo

Displays information in standard formatted output.

Usage

This command is useful for reducing network load. It enables you to transfer multiple compressed files while still maintaining optimal performance. If an *output_file* name is not specified for the zip operation, the name of the first file/directory without the extension is used as the output zip file name. Password encryption for the zip operation is not supported. Decryption of a password protected zip file is supported. Existing files are overwritten for both the zip and unzip operations.

Examples

To execute a simple zip operation:

```
fs0:\> compress -z a.txt b.txt c.txt
```

To create a zip file and provide the name of the output zip file:

```
fs0:\> compress -z -o zipfile.zip a.txt b.txt c.txt
```

To create a zip file with faster compression logic:

```
fs0:\> compress -z -cl 0 -o zipfile.zip a.txt b.txt c.txt
```

To execute a simple unzip operation:

```
fs0:\> compress -u -i zipfile.zip
```

To unzip files to a specific directory:

```
fs0:\> compress -u -od newdir -i zipfile.zip
```

To unzip a password protected zip file to specific directory:

```
fs0:\> compress -u -p password -od newdir -i zipfile.zip
```

To unzip all files from a zip file to a root directory:

```
fs0:\> compress -u -ex -i zipfile.zip
```

connect

Syntax

```
connect [ [devicehandle] [driverhandle] | [-c] | [-r] ]
```

Description

Binds a driver to a specific device and starts the driver.

Options

devicehandle

Specifies a device handle in hexadecimal format.

driverhandle

Specifies a driver handle in hexadecimal format.

-c

Connects only the console devices described in UEFI Shell environment variables and related devices.

-r

Connects console devices recursively.

Usage

If a *devicehandle* is not specified, all device handles in the current system are the default.

If *driverhandle* is not specified, all matched drivers are bound to the specified device. If *driverhandle* is specified, it is given highest priority on connecting the specified devices. If the **-r** option is specified, all handles are recursively scanned to see if any loaded or embedded driver matches the specified device. Additionally, if more device handles are created during the binding, these handles are also checked to see if a matching driver can bind to the specified devices. This process is repeated until no more drivers are able to connect to any devices.

If the **-r** option is not specified, newly-created device handles are not bound further to any drivers. If only a single handle is specified and the handle has an `EFI_DRIVER_BINDING_PROTOCOL`, the handle is assumed to be a driver handle. Otherwise, it is assumed to be a device handle. If no parameters are

specified, the command attempts to bind all proper drivers to all devices without recursion and each connection status is displayed. Output redirection is not supported for `connect -r` usage.

Examples

To connect all drivers to all devices recursively:

```
Shell> connect -r
```

To display all connections:

```
Shell> connect
```

To connect drivers with 0x17 as highest priority to all the devices they can manage:

```
Shell> connect 17
```

To connect all possible drivers to device 0x19:

```
Shell> connect 19
```

To connect drivers with 0x17 as highest priority to device 0x19:

```
Shell> connect 19 17
```

To connect console devices described in the UEFI Shell environment variables:

```
Shell> connect -c
```

cp

Syntax

```
cp[-r][-q]src src...[dst]
```

Description

Copies one or more source files or directories to a destination.

Options

-r

Creates a recursive copy.

-q

Creates a quiet copy (with no prompt).

src src...

Specifies a source file or directory name. Wildcards are permitted.

dst

Specifies a destination file or directory name. Wildcards are not permitted. If not specified, the current working directory is assumed to be the destination. If more than one directory is specified, the last is always assumed to be the destination.

Usage

If the source is a directory, the `-r` flag must be specified. If `-r` is specified, the source directory is recursively copied to the destination (which means that all subdirectories are copied). If a destination is not specified, the current working directory is assumed to be the destination.

If any target file (not directory) already exists, a prompt appears, asking you to confirm replacing the file. The following choices are available:

- `Yes`—Replaces the file.
- `No`—Does not replace the file.
- `All`—Replaces the existing files in all subsequent cases.
- `Cancel`—Does not replace any existing files in all subsequent cases.

If there are multiple source files/directories, the destination must be a directory.

If an error occurs, the copying process stops immediately.

When executing in a script, the default is `-q`.

When copying to another directory, the directory must already exist.

Examples

To display the contents of the current directory:

```
fs0:\> ls
Directory of: fs0:\          06/18/01 01:02p <DIR>  512    efi
                06/18/01 01:02p <DIR>  512    test1
                06/18/01 01:02p <DIR>  512    test2
                06/13/01 10:00a 28,739    IsaBus.efi
                06/13/01 10:00a 32,838    IsaSerial.efi
                06/18/01 08:04p 29        temp.txt
                06/18/01 08:05p <DIR>  512    test
                3 File(s) 61,606 bytes
                4 Dir(s)
```

To copy a file in the same directory and change the file name:

```
fs0:\> cp temp.txt readme.txt
copying fs0:\temp.txt -> fs0:\readme.txt
- [ok]
```

To copy multiple files to another directory:

```
fs0:\> cp temp.txt isaBus.efi \test
copying fs0:\temp.txt -> fs0:\test\temp.txt
- [ok]
copying fs0:\isaBus.efi -> fs0:\test\IsaBus.efi
- [ok]
```

To copy multiple directories recursively to another directory:

```
fs0:\> cp -r test1 test2 boot \test
copying fs0:\test1 -> fs0:\test\test1
copying fs0:\test1\test1.txt -> fs0:\test\test1\test1.txt
- [ok]
copying fs0:\test2 -> fs0:\test\test2
copying fs0:\test2\test2.txt -> fs0:\test\test2\test2.txt
- [ok]
copying fs0:\boot -> fs0:\test\boot
copying fs0:\boot\shell.efi -> fs0:\test\boot\shell.efi
- [ok]
```

To see the results of the above operations:

```
fs0:\> ls \test
Directory of: fs0:\test          06/18/01 01:01p <DIR>      512
                06/18/01 01:01p <DIR>          0          ..
                01/28/01 08:21p <DIR>      512          test1
```

```

01/28/01 08:21p <DIR>          512          test2
01/28/01 08:21p <DIR>          512          boot
01/28/01 08:23p                29
temp.txt
01/28/01 08:23p                28,739      IsaBus.efi
      2 File(s)                28,828 bytes
      5 Dir(s)
Shell>

```

date

Syntax

```
date [mm/dd{yy|yyyy}] [-sfo]
```

Description

Displays or sets the current date for the system.

Options

mm

Specifies the month of the date to be set (1-12).

dd

Specifies the day of the date to be set (1-31).

yyyy

Specifies a four-digit year date.

-sfo

Specifies a standard formatted output display.

Usage

If no parameters are specified, the current date displays. If a valid month, day, and year are specified, the system's date is updated. Rules are:

- Except for numeric characters and /, all other characters in the argument are invalid. The Shell reports an error if the number is in the wrong month/date/year range.
- A space before or after the numeric character is not allowed. Inserting a space into the number is invalid.
- The year range is greater than or equal to 1998. Two numeric characters indicate the year. Numbers below 98 are regarded as 20xx, and numbers equal to or above 98 are regarded as 19xx. 00 means 2000. For example:

```

Shell> date 8/4/97
Shell> date
8/04/2097
Shell>
Shell> date 8/4/98
Shell> date 08/04/1998
Shell>

```

The range of valid years is 1998–2099.

Examples

To display the current date in the system:

```
fs0:\> date  
06/18/2001
```

To set the date with a long year format and display it:

```
fs0:\> date 01/01/2050  
fs0:\> date  
01/01/2050
```

To set the date with a short year format and display it:

```
fs0:\> date 06/18/01  
fs0:\> date  
06/18/2001
```

dblk

Syntax

```
dblk device[lba] [blocks] [-b]
```

Description

Displays one or more blocks from a block device.

Options

-b

Display one screen at a time.

device

Block device name.

lba

Index of the first block to be displayed (a hexadecimal number).

blocks

Number of blocks to be displayed (a hexadecimal number). The default is 1. If larger than 0x10, only 0x10 are displayed.

Examples

To display one block of blk0, beginning from block 0:

```
fs0:\> dblk blk0
```

To display one block of fs0, beginning from block 0x2:

```
fs0:\> dblk fs0 2
```

To display 0x5 blocks of fs0, beginning from block 0x12:

```
fs0:\> dblk fs0 12 5
```

devices

Syntax

```
devices [-b] [-lxxx] [-sfo]
```


Description

Displays a list of devices managed by UEFI drivers.

Options

-b

Displays one screen at a time.

-lxxx

Displays devices in a specific language. For a list of possible code options, see the UEFI specification.

-sfo

Displays information in standard formatted output.

Example

To display all devices compliant with the EFI Driver Model:

```
Shell> devices
C   T   D
T   Y   C   I
R   P   F   A
L   E   G   G   #P   #D   #C   Device Name
==  =  =  =  ==  ==  ==  =====
20  R  -  -  -  1  13  VenHw(58C518B1-76F3-11D4-BCEA-0080C73C8881)
3D  D  -  -  3  -  -  Primary Console Input Device
3E  D  -  -  3  -  -  Primary Console Output Device
64  B  -  -  1  6  2  UGA Window 1
65  B  -  -  1  6  2  UGA Window 2"
66  B  -  -  1  1  1  EFI_WIN_NT_SERIAL_PORT=COM1
67  B  -  -  1  1  1  COM1
68  B  -  -  1  4  2  PC-ANSI Serial Console
69  D  -  -  1  -  -  EFI_WIN_NT_SERIAL_PORT=COM2
6E  D  -  -  1  -  -  EFI_WIN_NT_PHYSICAL_DISKS=e:RW;262144;512
6F  D  -  -  1  -  -  EFI_WIN_NT_CPU_MODEL=Intel(R) Processor Model
70  D  -  -  1  -  -  EFI_WIN_NT_CPU_SPEED=3000
71  D  -  -  1  -  -  EFI_MEMORY_SIZE=64
72  D  -  -  1  -  -  EFI_MEMORY_SIZE=64
```

Output details

The following table describes the possible output for this command.

Table 2: Output details—`devices` command

Column	Description
CTRL	Handle number of the device.
TYPE	Device type. Options are: <ul style="list-style-type: none">• R—Root controller• B—Bus controller• D—Device controller
CFG	Configuration Protocol support status: <ul style="list-style-type: none">• Y—Yes• N—No
DIAG	Diagnostics Protocol support status: <ul style="list-style-type: none">• Y—Yes• N—No
#P	Number of parent controllers for this device.
#D	Number of this type of devices.
#C	Number of child controllers produced by this device.
Device Name	Name of the device from the Component Name Protocol.

devtree

Syntax

```
devtree [-b] [-d] [-lxxx] [devicehandle]
```

Description

Displays the tree of devices compliant with the UEFI Driver Model.

Options

-b

Displays one screen at a time.

-d

Displays the device tree using device paths.

-lxxx

Displays the device tree in a specific language. For a list of possible code options, see the UEFI specification.

devicehandle

Displays the device tree below a specified handle.

Usage

By default, device names that are retrieved from the Component Name Protocol. If the option `-d` is specified, the device paths are printed instead.

Example

To display the tree of all devices compliant with the UEFI Driver Model one screen at a time:

```
Shell> devtree -b
devtree -b fs0:\
Ctrl[04] Fv(770BF9B6-8AFE-4F4C-85E5-893FC3D2606C)
Ctrl[05] Fv(27A72E80-3118-4C0C-8673-AA5B4EFA9613)-directories in the current
Ctrl[06] MemoryMapped(0xB,0xFFD40000,0xFFD6FFFF)
Ctrl[07] Fv(5A515240-D1F1-4C58-9590-27B1F0E86827)
Ctrl[08] Fv(5E2363B4-3E9E-4203-B873-BB40DF46C8E6)
Ctrl[09] Fv(CDBB7B35-6833-4ED6-9AB2-57D2ACDDF6F0)extension '.efi':
Ctrl[5F] PciRoot(0x0).efi
Ctrl[195] PciRoot(0x0)/Pci(0x0,0x0)
Ctrl[196] PciRoot(0x0)/Pci(0x2,0x0)from all files with extension '.inf':
Ctrl[197] PciRoot(0x0)/Pci(0x2,0x0)/Pci(0x0,0x0)
Ctrl[198] PciRoot(0x0)/Pci(0x2,0x1)
Ctrl[199] PciRoot(0x0)/Pci(0x2,0x2)
Ctrl[19A] PciRoot(0x0)/Pci(0x2,0x3)
Ctrl[19B] PciRoot(0x0)/Pci(0x3,0x0)
Ctrl[19C] PciRoot(0x0)/Pci(0x3,0x1)
Ctrl[19D] PciRoot(0x0)/Pci(0x3,0x2)
Ctrl[19E] PciRoot(0x0)/Pci(0x3,0x3)
Ctrl[19F] PciRoot(0x0)/Pci(0x4,0x0)
Ctrl[1A0] PciRoot(0x0)/Pci(0x4,0x1)
Ctrl[1A1] PciRoot(0x0)/Pci(0x4,0x2)
Ctrl[1A2] PciRoot(0x0)/Pci(0x4,0x3)
Ctrl[1A3] PciRoot(0x0)/Pci(0x4,0x4)
Ctrl[1A4] PciRoot(0x0)/Pci(0x4,0x5)
Ctrl[1A5] PciRoot(0x0)/Pci(0x4,0x6)
Ctrl[1A6] PciRoot(0x0)/Pci(0x4,0x7)
Ctrl[1A7] PciRoot(0x0)/Pci(0x5,0x0)
Ctrl[1A8] PciRoot(0x0)/Pci(0x5,0x1)
Ctrl[1A9] PciRoot(0x0)/Pci(0x5,0x2)
Ctrl[1AA] PciRoot(0x0)/Pci(0x5,0x4)
Ctrl[1AB] PciRoot(0x0)/Pci(0x6,0x7)
Press ENTER to continue or 'Q' break:
```

dh

Syntax

```
dh[-lxxx][handle|-p prot_id][-d][-b][-v][-sfo]
```

Description

Displays the device handles in the UEFI environment.

Options

-lxxx

Displays device handles in a specific language. For a list of possible code options, see the UEFI specification.

handle

Displays the handle for a specific device.

-p prot_id

Displays protocol information associated to handles. If not specified, all protocols are displayed.

-d

Displays UEFI driver model-related information.

-b

Displays one screen at a time.

-v

Displays verbose information.

-sfo

Displays information in standard formatted output.

Usage

If a handle number is specified, the details of all the protocols that are associated with that device handle are displayed. Otherwise, the `-p` option can be used to list the device handles that contain a specific protocol. If neither `-p prot_id` nor `handle` is specified, all handles are displayed.

Examples

To display all handles one screen at a time:

```
Shell> dh -b
```

```
Handle dump
```

```
1: Image (DXE Core)
2: FwVol FwFileSys FwVolBlk DevPath (MemMap (11:1B50000-
  1D4FFC8) )
3: Image (Ebc)
4: DevPath (MemMap (11:1CA0000-1CB0000) )
5: Image (WinNtThunk)
6: WinNtThunk DevPath (..76F3-11D4-BCEA-0080C73C8881) )
7: Image (WinNtBusDriver) DriverBinding
```

```
...
```

To display detailed information about handle 0x30:

```
Shell> dh 30 -v
```

```
Handle 30 (01AF5308)
```

```
  IsaIo
```

```
    ROM Size.....: 00000000
```

```
    ROM Location...: 00000000
```

```
    ISA Resource List :
```

```
      IO : 000003F8-000003FF Attr : 00000000
```

```
      INT : 00000004-00000000 Attr : 00000000
```

```
  dpath
```

```
    PNP Device Path for PnP
```

```
    HID A0341D0, UID 0x0
```

```
    Hardware Device Path for PCI
```

```
PNP Device Path for PnP
HID 50141D0, UID 0
AsStr: 'Acpi (PNP0A03,0)/Pci (1F|0)/Acpi (PNP0501,0)'
```

To display all handles associated with the `diskio` protocol:

```
Shell> dh -p diskio
Handle dump by protocol 'Diskio'
 15: DiskIo BlkIo DevPath(..i(3|1)/Ata(Secondary,Master))
 16: DiskIo BlkIo DevPath(..,1)/PCI(0|0)/Scsi(Pun0,Lun0))
 44: DiskIo BlkIo Fs DevPath(..ABD0-01C0-507B-9E5F8078F531)
    ESP
 45: DiskIo BlkIo Fs DevPath(..i(Pun0,Lun0)/HD(Part4,SigG0))
    ESP
 17: DiskIo BlkIo DevPath(..PCI(3|1)/Ata(Primary,Master))
```

To display all handles associated with the `Image` protocol and break when the screen is full:

```
Shell> dh -p Image -b
Handle dump by protocol 'image'
 1: Image (DXE Core)
 5: Image (WinNtThunk)
 7: Image (WinNtBusDriver) DriverBinding
 8: Image (Metronome)
 A: Image (IsaBus) DriverBinding
 B: Image (WinNtConsole) DriverBinding
 ...
```

Output details

The following table describes the possible output for this command.

Table 3: Output details—`dh` command

Column	Description
<code>Driver Name</code>	Name of driver producing the handle.
<code>Controller Name</code>	Name of controller producing the handle.
<code>Handle number</code>	Integer number of the handle.
<code>Device Path</code>	Device path associated with the handle.
<code>Protocol Identifiers</code>	Semicolon-delimited list of protocol identifiers or GUIDs.

diags

Syntax

```
diags [-console]
```

Description



Launches the Embedded Diagnostics tool.

Options

-console

Launches Embedded Diagnostics in text mode.

Examples

To launch Embedded Diagnostics:

```
fs0:\> diags
```

To launch Embedded Diagnostics in text mode:

```
fs0:\> diags -console
```

disconnect

Syntax

```
disconnect devicehandle [driverhandle][childhandle][-r]
```

```
disconnect [-r] [-nc]
```

Description

Disconnects one or more drivers from the specified devices.

Options

devicehandle

Specifies a device handle in hexadecimal format.

driverhandle

Specifies a driver handle in hexadecimal format. If not specified, the device specified by *devicehandle* is disconnected.

childhandle

Specifies a child handle of a device in hexadecimal format. If not specified, all child handles of the device specified by *devicehandle* are disconnected.

-r

Disconnects all drivers from all devices, and then reconnects consoles.

-nc

Does not reconnect console devices.

Usage

This command does not support output redirection.

Examples

To disconnect all drivers from all devices, and then reconnect console devices:

```
Shell> disconnect -r
```

To disconnect all drivers from all devices, including console devices:

```
Shell> disconnect -r -nc
```

To disconnect all drivers from device 0x28:

```
Shell> disconnect 28
```

To disconnect driver 0x17 from device 0x28:

```
Shell> disconnect 28 17
```

To disconnect driver 0x17 from controlling the child 0x32 of device 0x28:

```
Shell> disconnect 28 17 32
```

dmem

Syntax

```
dmem[-b] [address] [size] [-MMIO]
```

Description

Displays the contents of system or device memory.

Options

-b

Displays one screen at a time.

address

Displays memory contents from a specific starting address (in hexadecimal format).

size

Displays memory contents of a specific size (in hexadecimal format).

-MMIO

Displays memory mapped contents using the `EFI_PCI_ROOT_BRIDGE_IO_PROTOCOL`.

Usage

If *address* is not specified, the contents of the EFI System Table are displayed. Otherwise, memory starting at the *address* is displayed. If *size* is not specified, the display defaults to 512 bytes. If `-MMIO` is not specified, main system memory is displayed. Otherwise, device memory is displayed through the use of the `EFI_PCI_ROOT_BRIDGE_IO_PROTOCOL`.

Example

To display memory contents from 1af3088 with a size of 16 bytes:

```
Shell> dmem 1af3088 16
```

```
Memory Address 000000001AF3088 16 Bytes
```

```
01AF3088: 49 42 49 20 53 59 53 54-00 00 02 00 18 00 00 00 *IBI SYST.....*
```

```
01AF3098: FF 9E D7 9B 00 00 *.....*
```

dmpstore

Syntax

```
dmpstore[-b] [-d] [-all| [variable] [-guid guid] [-sfo]
```

```
dmpstore[-all| [variable] [-guid guid] [-s file]
```

```
dmpstore[-all| [variable] [-guid guid] [-l file]
```

Description

Manages all UEFI variables.

Options

-b

Displays one screen at a time.

-guid *guid*

Specifies the GUID of the variables to display in standard text format. If not specified, and **-all** is not specified, the `EFI_GLOBAL_VARIABLE` GUID is assumed.

-sfo

Displays information in Standard-Format Output.

-all

Displays all variables, including those with a different GUID than `EFI_GLOBAL_VARIABLE`.

-d

Deletes variables.

-s

Saves variables to a file.

-l

Loads and sets variables from a file.

variable

Specifies a variable name. This can be a literal name or a pattern as specified in the `MetaMatch()` function of the `EFI_UNICODE_COLLATION2_PROCOOL`.

The variable value is printed as a hexadecimal dump.

Examples

To dump all variables with the GUID `EFI_GLOBAL_VARIABLE`:

```
Shell> dmpstore
```

To dump all variables, regardless of GUID or name:

```
Shell> dmpstore -all
```

To dump the `path` variable with the GUID `158DEF5A-F656-419C-B027-7A3192C079D2`:

```
Shell> dmpstore path -guid 158DEF5A-F656-419C-B027-7A3192C079D2
```

To save all variables, regardless of GUID or name, to a file named `VarDump.txt`:

```
Shell> dmpstore -all -s VarDump.txt
```

To delete the `BootOrder` variable with the GUID `EFI_GLOBAL_VARIABLE`:

```
Shell> dmpstore -d BootOrder
```

drivers

Syntax

```
drivers [-lxxx] [-sfo]
```

Description

Displays a list of information for drivers that follow the UEFI Driver Model.

Options

-lxxx

Displays drivers in a specific language. For a list of possible code options, see the UEFI specification.

-sfo

Displays in a standard formatted output table.

Example

To display the driver list:

```
Shell> drivers
          T  D
D          Y C I
R          P F A
V  VERSION E G G #D #C DRIVER NAME IMAGE NAME
=== ===== = = = == == =====
39 00000010 D - - 1 - Platform Console Management Driver ConPlatform
3A 00000010 D - - 1 - Platform Console Management Driver ConPlatform
3B 00000010 B - - 1 1 Console Splitter Driver ConSplitter
3C 00000010 ? - - - - Console Splitter Driver ConSplitter
3D 00000010 B - - 1 1 Console Splitter Driver ConSplitter
3E 00000010 ? - - - - Console Splitter Driver ConSplitter
42 00000010 D - - 1 - UGA Console Driver GraphicsConsole
43 00000010 ? - - - - Serial Terminal Driver Terminal
44 00000010 D - - 1 - Generic Disk I/O Driver DiskIo
45 00000010 D - - 1 - FAT File System Driver Fat
48 00000010 ? - - - - ISA Bus Driver IsaBus
49 00000010 ? - - - - ISA Serial Driver IsaSerial
4C 00000010 B - - 1 1 PCI Bus Driver PciBus
55 00000010 D X X 1 - Windows Block I/O Driver WinNtBlockIo
56 00000010 ? - - - - Windows Text Console Driver WinNtConsole
57 00000010 ? - - - - Windows Serial I/O Driver WinNtSerialIo
58 00000010 D - - 1 - Windows Simple File System Driver WinNtSimpleFileSystem
59 00000010 B - - 1 3 Windows Bus Driver WinNtBusDriver
5F 00000010 D - - 1 - Windows Universal Graphics Adapter WinNtUga
```

Output details

The following table describes possible output for this command.

Table 4: Output details—drivers command

Column	Description
DRV	Integer handle of the driver.
VERSION	Version number of the driver.
TYPE	Driver type. Possible values are: <ul style="list-style-type: none"> B—Bus driver D—Device driver

Table Continued

Column	Description
CFG	Configuration Protocol Support status: <ul style="list-style-type: none"> • Y—Yes • N—No
DIAG	Driver Protocol support status: <ul style="list-style-type: none"> • Y—Yes • N—No
#D	Number of devices this driver is managing.
#C	Number of child devices this driver has produced.
DRIVER_NAME	Name of the driver from the Component Name Protocol.
IMAGE_NAME	Device path from which the driver was loaded.

echo

Syntax

```
echo [-on | -off]
```

```
echo message
```

Description

Controls whether script commands are displayed as they are read from the script file, and prints the given message to the display.

Options

-on

Enables the display when reading commands from script files.

-off

Disables the display when reading commands from script files.

message

Specifies a message to display.

Usage

The first form of this command controls whether or not script commands display as they are read from the script file. If no argument is given, the current `on` or `off` status displays. The second form of the command prints the specified message to the display.

This command does not change the value of the environment variable `lasterror`.

Examples

To display a message string of Hello World:

```
fs0:\> echo Hello World
Hello World
```

To turn command echoing on:

```
fs0:\> echo -on
```

To execute HelloWorld.nsh, and display when reading lines from the script file:

```
fs0:\> HelloWorld.nsh
+HelloWorld.nsh> echo Hello World
Hello World
```

To turn command echoing off:

```
fs0:\> echo -off
```

To display the current echo setting:

```
fs0:\> echo
Echo is off
```

edit

Syntax

```
edit [file]
```

Description

Edits an ASCII or UCS-2 file in full screen mode.

Options

file

Specifies the name of file to be edited. If none is specified, an empty file is created with a default file name.

Usage

This command supports both UCS-2 and ASCII file types.

Example

To edit the `shell.log` file:

```
fs0:\> edit shell.log
```

eficompress

Syntax

```
eficompress infile outfile
```

Description

Compresses a file using the EFI Compression Algorithm, and writes the compressed form to a new file.

Options

infile

Specifies the filename for the uncompressed input file.

outfile

Specifies the filename for the compressed output file.

Example

To compress a file named `uncompressed` to file named `compressed`:

```
fs0:\> eficompress uncompressed compressed
```

efidecompress

Syntax

```
efidecompress infile outfile
```

Description

Decompresses a file using the EFI Decompression Algorithm, and writes the decompressed form to a new file.

Options

infile

Specifies the filename for the compressed input file.

outfile

Specifies the filename for the decompressed output file.

Example

To decompress a file named `compressed` to file named `uncompressed`:

```
fs0:\> eficompress compressed uncompressed
```

exit

Syntax

```
exit[/b] [exit-code]
```

Description

Exits the UEFI Shell or the current script.

Options

`/b`

Indicates that only the current UEFI Shell script should be terminated. Ignored if not used within a script.

exit-code

If exiting a UEFI Shell script, specifies the value placed into the environment variable `lasterror`. If exiting an instance of the UEFI Shell, specifies the value returned to the caller. If not specified, 0 is returned.

Example

To exit the UEFI Shell:

```
fs0:\> exit
```

ftp

Syntax

```
ftp host[port][-b]
```

Description



Connects to FTP servers for network file transfers.

Options

host

Specifies a server IPv4 address or host name.

port

Specifies a server FTP port.

-b

Enables page break for sub-commands.

Usage

This command launches an interactive shell for network file transfer (FTP) operations. Only IPv4 addresses are supported. Press **ESC** or **Ctrl-C** to cancel a file transfer.



IMPORTANT: You do not need to use `ifconfig` on a network interface if you plan to run `webclient` or `ftp` over the same interface because these interface and IP address settings are automatically selected by the **Pre-Boot Network Settings** configured in the System Utilities.

If the interface used by `ftp` and `webclient` happens to be configured by `ifconfig`, that setting is erased and, instead, the System Utilities **Pre-Boot Network Settings** menu is applied on the interface when the commands are run.

Examples

To connect to an FTP server at a specified IP address:

```
fs0:\> ftp 192.168.1.20
```

To connect to an FTP server using a host name:

```
fs0:\> ftp ftp.hpe.com
```

To connect to an FTP server using an IP address, the user name `user` and a password of `pass`:

```
fs0:\> ftp 192.168.1.20
User (192.168.1.20):user
Password:pass
Login successful.
```

To connect to an FTP server at a specified IP address and enable page break in sub-commands:

```
fs0:\> ftp 192.168.1.20 -b
```

fwupdate

Syntax

```
fwupdate -l[-sfo] fwupdate -d device | -a [i image]-f file[-q]
```

Description



Updates firmware components, including the system BIOS, NICs, and storage cards.

In addition to the `fwupdate` command, the **Firmware Update** option in the System Utilities, and the FWUPDATE utility, the images in the Online Flash Components for Windows, Linux, and VMware operating systems are available for updating the system firmware.

This command checks the system and provides a choice (if more than one exists) of available firmware revisions.

Options

-l

Lists the devices with firmware update capability and their revisions.

-d *device*

Specifies a device name.

-i *image*

Specifies the image to update.

-f *file*

Specifies a firmware image file to update.

-q

Displays information in quiet mode, skipping all confirmations.

-a

Updates all applicable NVDIMMs with the image specified by `-f`.

-sfo

Displays information in standard formatted output.

Usage

Some devices have more than one firmware image. For example, a network adapter can have firmware images for PXE boot code, and NCSI firmware. In such cases, you can update images separately.

Examples

To list devices and their firmware revisions:

```
fs0:\> fwupdate -l
* [BIOS] System ROM - P92 v1.00 (05/09/2014)
  * [NIC.LOM.1.5] Embedded LOM 1 : HPE Ethernet 1Gb 4-port 331i Adapter - NIC
    1. BOOT CODE - 5719-v1.38
    2. MBA - 16.6.0\
    3. ASF - N/A
    4. NCSI - 1.2.43
    5. UMP - N/A
    6. VPD - N/A
    7. CCM - 7.10.31
```

To update the system BIOS firmware:

```
fs0:\> fwupdate -d BIOS -f P92_1.00_03_22_2014.signed.full
```

To upload BIOS firmware from a network location:

```
fs0:\> fwupdate -d BIOS -f http://www.example.com/BIOS/P92_1.00_03_22_2014.signed.full
```

getmtc

Syntax

```
getmtc
```

Description

Gets the MTC (monotonic counter) value from Boot Services and displays it.

Usage

This command displays the current monotonic counter value. The lower 32 bits increment every time this command is executed. Every time the system is reset, the upper 32 bits is incremented and the lower 32 bits is reset to 0.

goto

Syntax

```
goto label
```

Description

Goes to a label in a script.

hash

Syntax

```
hash [option] file...
```

Description



Computes a hash of one or more files.

Options

option

Specifies a hash option. Valid vales are:

- -md5—MD5 digest
- -sha1—SHA1 (160-bit) digest
- -sha256—SHA256 digest
- -sha384—SHA384 digest
- -sha512—SHA512 digest

file...

Specifies one or more files to hash.

Example

To compute the SHA256 message digest of a file:

```
Shell> hash -sha256 file
```

help

Syntax

```
help[cmd|pattern|special] [-usage] [-v] [-section sectionname] [-b]
```

Description

Displays the list of commands that are built into the UEFI Shell.

Options

cmd

Specifies the command for which to display help.

pattern

Specifies the pattern to use for displaying command help.

special

Displays a list of the special characters used in the shell command line.

-usage

Displays usage information for the command. This display is the same display as specifying **-section:NAME** and **-section:SYNOPSIS**.

-v

Displays verbose information.

-section *sectionname*

Displays the specified section of the help information.

-b

Displays the help output one screen at a time.

Usage

If no options are specified, each command is displayed along with a brief description of its function. If **-v** is specified, all help information for the specified commands appears. If **-section** is specified, only the help section specified appears (see Examples). If **-usage** is specified, the command, a brief description, and the usage appears.

The help text is gathered from UCS-2 text files found in the directory where the shell or shell command executable is located. The files have the name *command-name.man*, where *command-name* is the name of the shell command. The files follow a subset of the MAN page format, as described Examples.

If no option is specified, only the **NAME** section of the page appears.

Examples

To display the list of commands in the UEFI Shell and break after one screen:

```
Shell> help -b
```

```
alias          - Displays, creates, or deletes UEFI Shell aliases.
attrib        - Displays or changes the attributes of files or directories.
cd            - Displays or changes the current directory.
cls          - Clears standard output and optionally changes background color.
comp         - Compares the contents of two files on a byte for byte basis.
```


To display help information about the shell command `ls` (use any of the following syntax options):

```
Shell> help ls
Shell> ? ls
Shell> ls -?
```

To display the list of commands starting with the character `p`:

```
Shell> help p*
pause - Prints a message and suspends for keyboard input
```

ifconfig

Syntax

```
ifconfig[-r [name]][-l[name]] ifconfig[-s name dhcp|[static IPaddress mask gateway]| dns IP]
```

Description

Modifies the default IP address of the UEFI IPv4 network stack.

Options

-r

Renews the interface configuration and sets the DHCP policy.

-l

Lists the configuration.

-s

Sets the configuration.

name

Specifies an adapter name. For example, `eth0`.

dhcp

Specifies that DHCP4 is to dynamically request IPv4 addresses for all or a specific interface.

static IPaddress

Specifies a static IPv4 address in four integer values, each between 0 and 255, separated by periods.

mask

Specifies a subnet mask in four integer values, each between 0 and 255, separated by periods.

gateway

Specifies a default gateway in four integer values, each between 0 and 255, separated by periods.

dns IP

Specifies a DNS server address.

permanent

Specifies that the configuration is permanent (not one-time only).

Usage

! **IMPORTANT:** You do not need to use `ifconfig` on a network interface if you plan to run `webclient` or `ftp` over the same interface. The interface and IP address settings are selected automatically by the **Pre-Boot Network Settings** configured in the System Utilities.

If the interface used by `ftp` and `webclient` is configured by `ifconfig`, that setting is erased and, instead, the System Utilities **Pre-Boot Network Settings** menu is applied on the interface when the commands are run.

You can use this command to configure host networks for preboot network access from the Shell. The `-c` option clears the configuration for all or a specified interface, causing the network stack for related interfaces to default back to DHCP.

Examples

To list the configuration for the `eth0` interface:

```
fso:\> ifconfig -l eth0
```

To use DHCP4 to dynamically request the IPv4 address configuration for the `eth0` interface:

```
fso:\> ifconfig -s eth0 dhcp
```

```
fso:\> ifconfig -s eth0 static 192.168.0.5 255.255.255.0 192.168.0.1 permanent
```

To configure a DNS server address for the `eth0` interface:

```
fso:\> ifconfig -s eth0 dns 192.168.0.8 192.168.0.9
```

ifconfig6

Syntax

```
ifconfig6 -r [name] | -l [name]
```

```
ifconfig6 -s name [dad num] [auto | [man [id mac] [host IPv6 gw IPv6] [dns IPv6]]]
```

Description

Displays or modifies the IPv6 configuration for a network interface.

Options

name

Specifies an adapter name, such as `eth0`.

-r name

Reconfigures all a specified interface, and sets the automatic policy. If the specified interface is already set to automatic, the IPv6 configuration is refreshed.

-l name

Lists the configuration of the specified interface.

-s name dad num

Sets the duplicate address detection transmit count of the specified interface.

-s name auto

Sets the automatic policy of the specified interface.

-s name man id mac

Sets the alternative interface ID of the specified interface. The interface must be controlled by a manual policy.

-s name man host IPv6 gw IPv6

Sets the static host IP and gateway address of the specified interface. The interface must be controlled by a manual policy.

-s name man dns IPv6

Sets the DNS server IP addresses of the specified interface. The interface must be controlled by a manual policy.

Examples

To list the configuration for the interface eth0:

```
Shell> ifConfig6 -l eth0
```

To use automatic configuration to request the IPv6 address configuration dynamically for the interface eth0:

```
Shell> ifconfig6 -s eth0 auto
```

To set the DAD transmit count for eth0 under automatic policy:

```
Shell> ifconfig6 -s eth0 auto dad 10
```

To set the alternative interface ID of eth0 under manual policy:

```
Shell> ifconfig6 -s eth0 man id ff:dd:aa:88:66:cc
```

To use the static IPv6 address configuration for the interface eth0, and specify that this configuration survives the network reload:

```
Shell> ifconfig6 -s eth0 man host 2002::1/64 2002::2/64  
gw 2002::3/64
```

To export the third TLS certificate in DER format to a file named `abc.crt`:

```
Shell> tlsconfig -x -i 3 -f abc.crt -t DER
```

To specify TLS cipher algorithms:

```
Shell> tlsconfig -c AES128-SHA256:AES256-SHA256
```

To show the TLS supported ciphers:

```
Shell> tlsconfig -c list
```

To enable certificate validation on a TLS connection:

```
Shell> tlsconfig -k PEER -o HOSTNAME
```

To specify that TLS v1.2 protocol version is used:

```
Shell> tlsconfig -v 1.2
```

imlview

Syntax

```
imlview[-export filename][-sfo][-b][-c][-start_id][-end_id]
```

Description



Displays and exports the Integrated Management Log (IML).

Options

-sfo

Displays information in standard formatted output.

-b

Displays the IML log one page at a time.

-c

Clears IML log entries.

-export filename

Exports the IML log to a specified file.

start_id

Outputs records with IDs that are greater than this specified number.

end_id

Outputs records with IDs that are lesser than this specified number.

Usage

The IML provides a record of historical events that have occurred on the server. Entries in the IML can help you diagnose issues or identify potential issues.

Examples

To change the current file system to the mapped FS0 file system:

```
Shell> fs0:
```

To display the current IML log:

```
FS0:\> imlview
```

To export the IML log to a file named `output.txt`:

```
FS0:\> imlview -export output.txt
```

ipmctl

Syntax

```
ipmctl command [options] [targets] [properties]
```

Description

To manage Intel Optane DC persistent memory modules, use the command `ipmctl`.

-h

Display help for the command.

Usage

For more information about using this command, see the Intel documentation at <https://github.com/intel/ipmctl/tree/master/Documentation/ipmctl>.

load

Syntax

```
load[-nc] file[file...]
```

Description

Loads a UEFI driver into memory.

Options

-nc

Loads the driver, but does not connect the driver.

file

Specifies the image file containing the UEFI driver to load. Wildcards are permitted.

Usage

You can use this command to load multiple files at one time and can use wildcards when specifying files. If **-nc** is not specified, the system attempts to connect the driver to a proper device. It can also cause previously loaded drivers to connect to corresponding devices.

Examples

To load the driver contained in the `Isabus.efi` file:

```
FS0:\> load Isabus.efi
```

To load the drivers contained in the `Isabus.efi` and `IsaSerial.efi` files:

```
FS0:\> load Isabus.efi IsaSerial.efi
```

To load drivers contained in multiple files with `Isa` in the file names:

```
FS0:\> load Isa*.efi
```

To load the driver contained in the `Isabus.efi` file without connecting:

```
FS0:\> load -nc Isabus.efi
```

ls/dir

Syntax

```
ls[-r] [-a[attrib]] [-sfo] [file]
```

Description

Lists the contents of a directory or file information. The `dir` command is an internal alias for this command.

Options

-r

Displays recursively (including subdirectories).

-a attrib

Displays only those files with the attributes specified. If no attributes are specified, all files are listed. If **-a** is not specified, all nonsystem and nonhidden files are listed. The attributes (*attrib*) can be one or more of the following:

- a—Archive
- s—System
- h—Hidden
- r—Read-only
- d—Directory

-sfo

Displays in standard formatted output.

file

Specifies the name of a file or directory. Wildcards are permitted.

Usage

If no file name or directory name is specified, the current working directory is assumed. The contents of a directory are listed if all the following are true:

- Option `-r` is not specified.
- No wildcard characters are specified in the `file` parameter.
- The `file` specified represents an existing directory. In all other cases, the command functions as follows:
 - All files or directories that match the specified name are displayed.
 - The `-r` option determines whether a recursive search is performed.
 - The option `-a attrib` displays only those files with the attributes specified. If more than one attribute is specified, only the files that have all those attributes are listed. If `-a` is followed by nothing, all files or directories are displayed, regardless of their attributes. If `-a` itself is not specified, all files, except system and hidden files, are displayed.

Examples

To hide files by adding the hidden or system attribute to them:

```
fs0:\> attrib +s +h *.efi
ASH      fs0:\IsaBus.efi
ASH      fs0:\IsaSerial.efi
```

To display all files and directories, except the files or directories with the `h` or the `s` attribute:

```
fs0:\> ls
Directory of: fs0:\

06/18/01      09:32p      153      for.nsh
06/18/01      01:02p <DIR>    512      efi
06/18/01      01:02p <DIR>    512      test1
06/18/01      01:02p <DIR>    512      test2
06/18/01      08:04p      29      temp.txt
06/18/01      08:05p <DIR>    512      test
01/28/01      08:24p  r      29      readme.txt
3 File(s)                211 bytes
4 Dir(s)
```

To display files with all attributes in the current directory:

```
fs0:\> ls -a
Directory of: fs0:\

06/18/01      09:32p           153      for.nsh
06/18/01      01:02p <DIR>      512      efi
06/18/01      01:02p <DIR>      512      test1
06/18/01      01:02p <DIR>      512      test2
06/18/01      10:59p           28,739   IsaBus.efi
06/18/01      10:59p           32,838   IsaSerial.efi
06/18/01      08:04p           29       temp.txt
06/18/01      08:05p <DIR>      512      test
01/28/01      08:24p  r           29       readme.txt
5 File(s)
4 Dir(s)
61,788 bytes
```

To display files with read-only attributes in the current directory:

```
fs0:\> ls -ar
Directory of: fs0:\
06/18/01      11:14p  r           29       readme.txt
1 File(s)
0 Dir(s)
29 bytes
```

To display files with an attribute of `s`:

```
fs0:\> ls -as isabus.efi
Directory of: fs0:\
06/18/01      10:59p           28,739   IsaBus.efi
1 File(s)
0 Dir(s)
28,739 bytes
```

To display all in the `fs0:\efi` directory recursively:

```
fs0:\> ls -r -a efi
```

To search for files with the specified type in the current directory recursively:

```
fs0:\> ls -r -a *.efi -b
```

Output details

The following tables describe possible volume (directory) and file information output from this command.

Table 5: Output details—`ls` command (volume information)

Column	Description
Name	Standard volume label.
Total Size	Total number of bytes in the volume.

Table Continued

Column	Description
Read Only status	Read-only status as: <ul style="list-style-type: none"> • True • False
Free Space	Total number of free bytes in the volume.
Block Size	Nominal block size by which files are typically grown, in bytes.

Table 6: Output details—`ls` command (file information)

Column	Description
Name	Complete file name and directory, including the mapped name of the file system.
Logical Size	Size of the file, in bytes.
Physical Size	Size of the file in the volume, including any padding, in bytes.
Attributes	List of file attributes. Possible values are: <ul style="list-style-type: none"> • <code>a</code>—Archive • <code>d</code>—Directory • <code>h</code>—Hidden • <code>r</code>—Read-only • <code>s</code>—System
File Creation Time	Time when the file was created, in the format <code>hh:mm:ss</code> .
File Creation Date	Date when the file was created, in the format <code>dd:mm:yyyy</code> .
File Access Time	Time when the file was accessed, in the format <code>hh:mm:ss</code> .
File Access Date	Date when the file was accessed, in the format <code>dd:mm:yyyy</code> .
File Modification Time	Time when the file was modified, in the format <code>hh:mm:ss</code> .
File Modification Date	Date when the file was modified, in the format <code>dd:mm:yyyy</code> .

map

Syntax

```
map[-d mappedname]
```

```
map[-r|-v|-c|-f|-u|-t type[, type] | mappedname] [-sfo]
```


`map [mappedname | mapping]`

Description

Displays or configures a mapping between a user-defined name and a device handle.

Options

-d

Deletes a mapping.

mappedname

Specifies a mapping name.

-r

Resets a mapping.

-v

Displays verbose information about all mappings.

-c

Shows the consistent mapping.

-f

Shows the normal mapping.

-t

Shows the device mappings, filtered according to the device type. Supported types are:

- `fp`—floppy
- `hd`—hard disk
- `cd`—CD-ROM

Types can be combined by putting a comma between two types. Spaces are not allowed between types.

-sfo

Displays in standard formatted output.

-u

Adds mappings for newly-installed devices and removes mappings for uninstalled devices, but does not change the mappings of existing devices. Preserves user-defined mappings.

handle

Specifies the number of the handle.

mapping

Specifies a new mapped name to assign to a device. The mapping must end with a colon (:).

Usage

The most common use of this command is to create a mapped name for devices that support a file system protocol. Once these mappings are created, the names can be used with all the file manipulation commands.

The UEFI Shell environment creates default mappings for all of the devices that support a recognized file system.

This command can be used to create additional mappings, or it can be used to delete an existing mapping with the `-d` option. If the command is used without any parameters, all of the current mappings are listed. If the `-v` option is used, the mappings are shown with additional information about each device.

The `-r` option resets all the default mappings in a system. This is useful if the system configuration has changed since the last boot.

The `-u` option adds mappings for newly installed devices and remove mappings for uninstalled devices, but does not change the mappings of existing devices. The user-defined mappings are also preserved. A mapping history is saved so that the original mapping name is used for a device with a specific device path if that mapping name was used for that device path last time. The current directory is also preserved if the current device is not changed.

Each device in the system has a consistent mapping. If the hardware configuration has not changed, the device's consistent mappings do not change. If two or more machines have the same hardware configurations, the device's consistent mapping is the same. Use the `-c` option to list all of the consistent mappings in the system.

The mapping consists of digits and characters. Other characters are not allowed.

This command supports wildcards to delete or show mappings. However, when assigning the mapping, wildcards are forbidden.

Example

To delete a mapping:

```
Shell> map -d devicename
```

Output details

The following table describes possible output for this command.

Table 7: Output details—`map` command

Column Number	Description
1	The name of the table. The name is mappings.
2	Mapped name. The mapped device name.
3	Device Path. The device path that corresponds to the mapped device name.
4	Consistent Name. The consistent mapped name (if any) that is equivalent to <i>mappedname</i> . If <i>mappedname</i> is already a consistent mapped name, this column is empty.

memmap

Syntax

```
memmap [-b] [-sfo]
```

Description

Displays the system memory map.

Options

-b

Displays one screen at a time.

-sfo

Displays standard formatted output in a detailed and a summary table.

Usage

The memory map keeps track of all the physical memory in the system and how it is currently being used.

Example

To display the system memory map:

```
fs0:\> memmap
```

Type	Start	End	# Pages	Attributes
Available	0000000000000000	-0000000000008DFFF	00000000000008E	00000000000000F
Reserved	000000000008E000	-000000000008FFFF	000000000000002	00000000000000F
Available	0000000000090000	-000000000009FFFF	000000000000010	00000000000000F
Available	0000000000100000	-000000000072CFFFF	0000000000072C00	00000000000000F
BS_Data	000000000072D00000	-000000000072DFFFF	0000000000000100	00000000000000F
Available	000000000072E00000	-0000000000837F0FF	00000000000109F1	00000000000000F
LoaderCode	0000000000837F1000	-000000000084030FF	0000000000000840	00000000000000F
Available	000000000084031000	-0000000000847BDFF	000000000000078D	00000000000000F
BS_Data	0000000000847BE000	-000000000084809FF	000000000000004C	00000000000000F
Available	00000000008480A000	-000000000084827FF	000000000000001E	00000000000000F
BS_Data	000000000084828000	-0000000000850B7FF	0000000000000890	00000000000000F
Available	0000000000850B8000	-0000000000850B9FF	0000000000000002	00000000000000F
BS_Data	0000000000850BA000	-0000000000850CCFF	0000000000000013	00000000000000F
LoaderData	0000000000850CD000	-0000000000850D5FF	0000000000000009	00000000000000F
BS_Data	0000000000850D6000	-0000000000861DFFF	000000000000110A	00000000000000F
BS_Code	0000000000861E0000	-0000000000861F9FF	000000000000001A	00000000000000F
RT_Data	0000000000861FA000	-0000000000861FEFF	0000000000000005	80000000000000F
BS_Code	0000000000861FF000	-000000000086958FF	000000000000075A	00000000000000F
BS_Data	000000000086959000	-00000000008695CFF	0000000000000004	00000000000000F
BS_Code	00000000008695D000	-000000000086E41FF	00000000000004E5	00000000000000F
BS_Data	000000000086E42000	-00000000008C842FF	00000000000005A01	00000000000000F
BS_Code	00000000008C843000	-00000000008CDE1FF	000000000000059F	00000000000000F
RT_Data	00000000008CDE2000	-00000000008CDE6FF	0000000000000005	80000000000000F
Reserved	00000000008CDE7000	-00000000008D006FF	0000000000000220	00000000000000F
RT_Data	00000000008D007000	-00000000008D00FFF	0000000000000009	80000000000000F
Reserved	00000000008D010000	-00000000008E00FFF	0000000000001000	00000000000000F
RT_Data	00000000008E010000	-00000000008E020FF	0000000000000011	80000000000000F
BS_Data	00000000008E021000	-00000000008E021FF	0000000000000001	00000000000000F
RT_Data	00000000008E022000	-00000000008E742FF	0000000000000721	80000000000000F
BS_Data	00000000008E743000	-000000000096742FF	0000000000000800	00000000000000F
Available	000000000096743000	-000000000096748FF	0000000000000006	00000000000000F
LoaderData	000000000096749000	-000000000096B42FF	00000000000003FA	00000000000000F
Available	000000000096B43000	-000000000096F12FF	00000000000003D0	00000000000000F
LoaderCode	000000000096F13000	-000000000096F42FF	0000000000000030	00000000000000F
Available	000000000096F43000	-000000000096F43FF	0000000000000001	00000000000000F
BS_Code	000000000096F44000	-000000000097F42FF	0000000000000FFF	00000000000000F
RT_Data	000000000097F43000	-000000000098342FF	0000000000000400	80000000000000F
RT_Code	000000000098343000	-000000000098B42FF	0000000000000800	80000000000000F
Reserved	000000000098B43000	-000000000098C42FF	0000000000000100	00000000000000F
ACPI_NVS	000000000098C43000	-00000000009B042FF	0000000000002400	00000000000000F

```

ACPI_Recl  000000009B043000-000000009B242FFF 00000000000000200 0000000000000000F
BS_Data    000000009B243000-000000009B244FFF 00000000000000002 0000000000000000F
RT_Data    000000009B245000-000000009B2C5FFF 00000000000000081 8000000000000000F
BS_Data    000000009B2C6000-00000000A2C63FFF 0000000000000799E 0000000000000000F
BS_Code    00000000A2C64000-00000000A2C64FFF 00000000000000001 0000000000000000F
BS_Data    00000000A2C65000-00000000A2CFBFFF 00000000000000097 0000000000000000F
Available  00000000A2CFC000-00000000AF7FFFFFF 0000000000000CB04 0000000000000000F
Available  0000000100000000-000000103FFFFFFEFFF 00000000000F3FFFF 0000000000000000F
Reserved   000000000000F0000-000000000000FFFFFF 00000000000000010 00000000000001001
Reserved   00000000AF800000-00000000BFFFFFFF 00000000000010800 00000000000000008
MMIO       00000000C0000000-00000000CFFFFFFF 00000000000010000 80000000000000001
MMIO       00000000FE000000-00000000FE00FFFF 00000000000000010 80000000000000001
Reserved   000000103FFFF000-000000103FFFFFFF 00000000000000001 00000000000000000

```

```

Reserved   :          72,499 Pages (296,955,904 Bytes)
LoaderCode:          2,160 Pages (8,847,360 Bytes)
LoaderData:          1,027 Pages (4,206,592 Bytes)
BS_Code    :           8,696 Pages (35,618,816 Bytes)
BS_Data    :          94,006 Pages (385,048,576 Bytes)
RT_Code    :           2,048 Pages (8,388,608 Bytes)
RT_Data    :           3,014 Pages (12,345,344 Bytes)
ACPI_Recl  :           512 Pages (2,097,152 Bytes)
ACPI_NVMS :           9,216 Pages (37,748,736 Bytes)
MMIO       :          65,552 Pages (268,500,992 Bytes)
MMIO_Port  :            0 Pages (0 Bytes)
PalCode    :            0 Pages (0 Bytes)
Available  :       16,583,958 Pages (67,927,891,968 Bytes)
Persistent:            0 Pages (0 Bytes)

```

```

-----
Total Memory:          65,252 MB (68,422,193,152 Bytes)

```

Output details

The following table describes the possible output for this command.

Table 8: Output details—memmap command

Column	Description
Type	Type of memory. Options are: <ul style="list-style-type: none">• Available• LoaderCode• LoaderData• BootServiceCode• BootServiceData• RuntimeCode• RuntimeData• Reserved• MemoryMappedIO• MemoryMappedIOPortSpace• UnusableMemory• ACPIReclaimMemory• ACPIMemoryNVS• PalCode
Start	Starting address.
End	Ending address.
# Pages	Number of 4 KB pages.
reserved	Reserved memory total size in bytes.
LoaderCode	Loader code total size in bytes.
LoaderData	Loader data total size in bytes.
BS_code	Boot service code total size in bytes.
BS_data	Boot service data total size in bytes.
RT_data	Runtime data total size in bytes.
available	Available memory in bytes.
Total Memory	Total memory size in bytes.

mkdir

Syntax

```
mkdir dir[dir...]
```

Description

Creates one or more new directories.

Option

dir

Specifies one or more names for directories. Wildcards are not permitted.

Usage

If *dir* includes nested directories, parent directories are created before child directories. If the directory already exists, the command exits with an error.

Examples

To create a new directory and display its contents:

```
fs0:\> mkdir rafter
```

```
fs0:\> ls
```

```
Directory of: fs0:\
```

```
06/18/01 08:05p <DIR>    512    test
06/18/01 11:14p  r         29    readme.txt
06/18/01 11:50p <DIR>    512    rafter
      1 File(s)        211 bytes
      2 Dir(s)
```

To create and display multiple directories:

```
fs0:\> mkdir temp1 temp 2
```

```
fs0:\> ls
```

```
Directory of: fs0:\
```

```
06/18/01 08:05p <DIR>    512    test
06/18/01 11:14p  r         29    readme.txt
06/18/01 11:50p <DIR>    512    rafter
06/18/01 11:52p <DIR>    512    temp1
06/18/01 11:52p <DIR>    512    temp2
      1 File(s)        211 bytes
      4 Dir(s)
```

mode

Syntax

```
mode[col row]
```

Description

Displays or changes the mode for the console output device.

Options

col

Specifies the number of columns.

row

Specifies the number of rows.

Usage

When entered without any parameters, this command shows the list of modes that the standard output device currently supports. The command can then be used with the *row* and *col* parameter to change the number of rows and columns on the standard output device.

NOTE: The display is cleared every time the *mode* command is used to change the currently selected display mode.

Examples

To display all available modes on standard output and the current selected mode (indicated by an *):

```
Shell> mode
Available modes on standard output
  col 80 row 25 *
  col 80 row 50
  col 80 row 43
  col 100 row 100
```

To change the current mode setting to an 80 X 50 text mode display:

```
Shell> mode 80 50
Available modes on standard output
  col 80 row 25
  col 80 row 50 *
  col 80 row 43
  col 100 row 100
```

mv

Syntax

```
mv src...[dst]
```

Description

Moves one or more files to a destination within a file system.

Options

src...

Specifies a source file or directory name. Wildcards are permitted.

dst

Specifies a destination file or directory name. Wildcards are permitted. If not specified, the current working directory is assumed to be the destination. If there is more than one argument in the command line, the last one is always considered the destination.

Usage

This command does not support moving between file system volumes. If the destination is an existing directory, the sources are moved into that directory. Otherwise, the sources are moved to the destination as if the directory has been renamed. If a destination is not specified, the current directory is assumed to be the destination.

Attempting to move a read-only file or directory results in an error. Moving a directory that contains read-only files is allowed. You cannot move a directory into itself or its subdirectories. You cannot move a directory if the current working directory is itself or its subdirectories.

If an error occurs, the remaining files or directories are still moved.

Example

To rename a file:

```
fs0:\> mv IsaBus.efi Bus.efi
moving fs0:\IsaBus.efi -> \Bus.efi
- [ok]
```

openinfo

Syntax

```
openinfo handle[-b]
```

Description

Displays the protocols and agents associated with a handle.

Options

-b

Displays one screen at a time.

handle

Displays open protocol information for specified handle.

parse

Syntax

```
parse filename tablename column[-i instance][-s instance]
```

Description

Retrieves a value from a specified record that was output in standard formatted output.

Options

filename

Specifies a source file name.

tablename

Specifies a table name to be parsed.

column

Specifies a one-based column index for determining which value from a particular record to parse.

-i instance

Starts parsing with the *nth* instance of the specified *tablename*, after the specified instance of *ShellCommand*. If not present, all instances are returned.

-s instance

Starts parsing with the *nth* instance of the specified *ShellCommand* table. If not present, 1 is assumed.

Usage

This command enables the parsing of data from a file containing data output from a command that used the `-sfo` parameter. Because the standard formatted output has a well-known means of parsing, this command is intended to be used as a simplified means of having scripts consume such constructed output files and use this retrieved data in the logic of the scripts being written for the UEFI Shell.

Examples

The following data is contained in a temporary file (`temp.txt`):

```
ShellCommand, "LS"  
VolumeInfo, "MikesVolume", "400000000", "32000000", "16000000"  
FileInfo, "fs0:/efi/boot/winloader.efi", "45670", "arsh"  
FileInfo, "fs0:/efi/boot/mikesfile.txt", "1250", "a"  
FileInfo, "fs0:/efi/boot/readme.txt", "795", "a"
```

To use the index parameter to parse the `temp.txt` file:

```
fs0:\> parse temp.txt FileInfo 3 -i 3  
795
```

partitions

Syntax

```
partitions[-v]
```

Description



Lists disk partitions in the system.

Option

`-v`

Lists verbose information about all file system partitions.

Example

To display verbose partition information:

```
Shell> partitions -v
```

pause

Syntax

```
pause[-q]
```

Description

Pauses the script file execution.

Usage

This command is available only in scripts. It prints a message to the display, suspends script file execution, and waits for keyboard input. Pressing any key resumes execution, except for `q` or `Q`. If `q` or `Q` is pressed, script processing terminates. Otherwise, execution continues with the next line after the pause command.

Option

`-q`

Hides the display message.

Examples

This script is a sample of the `pause` command:

```
fs0:\> type pause.nsh
#
# Example script for 'pause' command
#
echo pause.nsh begin..
date
time
pause
echo pause.nsh done.
```

To execute the script with echo on:

```
fs0:\> pause.nsh
+pause.nsh> echo pause.nsh begin..
pause.nsh begin..
+pause.nsh> date
06/19/2001
+pause.nsh> time
00:51:45
+pause.nsh> pause
Enter 'q' to quit, any other key to continue:
+pause.nsh> echo pause.nsh done.
pause.nsh done.
```

To execute the script with echo off:

```
fs0:\> echo -off
fs0:\> pause.nsh
pause.nsh begin..
pause.nsh begin..
06/19/2001
00:52:50
Enter 'q' to quit, any other key to continue: q
fs0:\>
```

pci

Syntax `pci[bus dev[func] [-s seg] [-i] [-ec ID]`

Description

Displays a PCI device list or PCI function configuration space information.

Options

bus

Specifies a bus number.

dev

Specifies a device number.

func

Specifies a function number.

-s *seg*

Specifies a segment number.

-i

Displays interpreted information.

-ec *ID*

Displays interpreted information about the extended capability of the specified PCIe.

Usage

This command displays all the PCI devices in the system. The information displayed for the configuration space of a PCI device is based on the specified bus, device, and function addresses. If the function address is not specified, it defaults to 0.

The **-i** option displays verbose information for the specified PCI device. The PCI configuration space for the device is dumped with a detail interpretation.

If no parameters are specified, all PCI devices are dumped with a detailed interpretation.

If the *bus* and *dev* number parameters are specified but the *func* or *seg* parameters are not, *func* or *seg* are set as default value 0.

Examples

To display all PCI devices in the system:

```
fs0:\> pci
```

To display the configuration space of bus 0, device 0, function 0:

```
fs0:\> pci 00 00 00 -i
```

To display the configuration space of segment 0, bus 0, device 0, function 0:

```
fs0:\> pci 00 00 00 -s 0
```

ping

Syntax ping[-n *count*][-l *size*][-s *sourceIP*]*targetIP*

Description

Pings the target host with an IPv4 stack.

Options

-s

Specifies the source adapter as an IPv4 address.

sourceIP

Specifies the IPv4 address of the source machine.

-n *count*

Specifies the number of echo request datagrams to be sent.

-l *size*

Specifies the size of the data buffer in the echo request datagram.

targetIP

Specifies the IPv4 address of the target machine.

Usage

This command uses the ICMPv4 ECHO_REQUEST datagram to elicit an ECHO_REPLY from a host.

Examples

To ping the target host at 192.168.0.1 with 64 bytes of data:

```
FS0:\> ping -l 64 192.168.0.1
```

To ping the target host at 202.120.120.100 by sending 20 echo request datagrams:

```
FS0:\> ping -n 20 202.120.120.100
```

To ping the target host by specifying the source adapter as an IPv4 address:

```
FS0:\> ping -s 202.120.100.12 202.120.100.1
```

ping6

Syntax

```
ping6 [-l size] [-n count] [-s sourceip] targetip
```

Description

Pings a target machine with UEFI IPv6 network stack.

Options

-l *size*

Specifies a buffer size, in bytes. (Default is 16, minimum is 16, and maximum is 32768).

-n *count*

Specifies a request count. (Default is 10, minimum is 1, and maximum is 10000).

-s *sourceip*

Specifies a source IPv6 address.

targetip

Specifies a target IPv6 address.

Usage

When *targetip* is a link local address, *sourceip* must be specified.

Examples

To ping the target host by sending 5 request with 1000 bytes from 2002::1

```
Shell> ping6 -s 2002::1 2002::2 -l 1000 -n 5
```

To ping the target host with 1000 bytes:

```
Shell> ping6 2002::2 -l 1000
```

ramdisk

Syntax ramdisk -c[-s *size*][-v *volumelabel*][-t *type*] [-m *memorytype*]

```
ramdisk -d[fs|all]
```

```
ramdisk -l [-sfo]
```

Description



Creates and deletes RAM disks.

Options

-c

Creates a RAM disk.

-s *size*

Specifies RAM disk size in MB. Valid values are from 4 MB to 512 MB.

-v *volumelabel*

Specifies a volume label name for the RAM disk. Valid values are up to 11 characters, without spaces or the following characters: % ^ * + = [] | : ; \ < > ? /.

-t *type*

Specifies a file system type.

-m *memorytype*

Specifies a memory type.

-d

Deletes one or more RAM disks.

fs

Specifies the file system drive for RAM disk deletion.

all

Deletes all RAM disks.

-l

Lists all RAM disks.

-sfo

Displays information in standard formatted output.

Usage

You can use this command to provision temporary staging locations. It is beneficial in bare metal environments when media is not available for staging or scripting. The command supports up to 10 RAM disks of the following file types and sizes:

- FAT16—(Default) 4 MB minimum size to 512 MB maximum size
- FAT32—512 MB minimum size to 4096 MB maximum size

If you do not specify **-t *type***, a default FAT16 type file system is created.

Valid ***memorytype*** options are *BS_Data* (default) and *Reserved*.

Examples

To create a 512 MB FAT16 RAM disk:

```
Shell> ramdisk -c -s 512
```

To create a 512 MB FAT16 RAM disk with a volume label of `RAMDISK1`:

```
Shell> ramdisk -c -s 512 -v RAMDISK1
```

To create a 1000 MB FAT32 RAM disk:

```
Shell> ramdisk -c -s 1000
```

To list mapping information for all RAM disks:

```
Shell> ramdisk -l
```

To delete RAM disk fs0:

```
Shell> ramdisk -d fs0
```

reconnect

Syntax

```
reconnect devicehandle [driverhandle [childhandle]]  
reconnect -r
```

Description

Reconnects drivers to a device.

Options

devicehandle

Specifies a device handle (a hexadecimal number).

-r

Reconnects drivers to all devices.

driverhandle

Specifies a driver handle (a hexadecimal number). If not specified, all drivers on the specified device are reconnected.

childhandle

Specifies a child handle of a device (a hexadecimal number). If not specified, all drivers all child handles of the *devicehandle* are reconnected.

Usage

This command first disconnects the specified driver from the specified device, and connects the driver to the device recursively.

If the **-r** option is used, all drivers are reconnected to all devices. Any drivers that are bound to any devices are disconnected, and then connected recursively.

For more, information, see the `connect` and `disconnect` commands.

Examples

To reconnect all drivers to all devices:

```
Shell> reconnect -r
```

To reconnect all drivers to device 0x28:

```
fs0:\> reconnect 28
```

To disconnect 0x17 from 0x28, and then reconnect drivers with 0x17 as highest priority to device 0x28:

```
fs0:\> reconnect 28 17
```

To disconnect 0x17 from 0x28, destroying child 0x32, and then reconnect drivers with 0x17 as highest priority to device 0x28:

```
fs0:\> reconnect 28 17 32\
```

reset

Syntax

```
reset [-w | [-s] | -c] [string]
```

Description

Resets the system.

Options

-w

Performs a warm boot.

-s

Performs a shutdown.

-c

Performs a cold boot.

string

Specifies a string to be passed to reset service.

Usage

The default usage of this command performs a cold reset. If *string* is specified, it is passed into the `SystemTable ResetSystem()` function, informing the system of the reason for the system reset.

Example

To reset the system:

```
Shell> reset
```

restclient


Syntax

```
restclient -uri URI [options]
```

```
restclient -type TYPE [options]
```

```
restclient -t restclient -time
```

Description

 Interacts with the local RESTful API service. The `restcli` command is an internal alias for this command.

Resource selection options

-uri *URI*

Requests a resource from a specific URI.

-type *TYPE*

Returns a resource matching the specific base type, if one exists.

Actions

-m METHOD

Sends the HTTP request using a specified method. Valid values are: GET/POST/PUT/PATCH/DELETE.

-g [PROPERTY...]

Gets the entire resource, or a specific property within the resource. `-m GET` is implied by this option.

-s PROPERTY=VALUE

Specifies a value for a resource property. `-m PATCH` is implied by this option, and request contents are automatically generated.

-t

Lists all resource types.

-time

Displays the RESTful time in ISO 8601 UTC format.

General options

-i FILE

Sends the contents of a specific file in the request contents.

-o FILE

Writes a JSON response body to the specified file using ASCII encoding. Only valid for GETs of the entire resource (`-g` or `-m GET`), ignored otherwise. Use this argument when the resource is larger than 16Kb.

-c

Uses cached content for GET requests when possible. This option improves performance, but might return stale data.

Usage

URIs must start with the root resource, without a leading slash. For example: `rest/v1/Systems`.

`TYPE` is a string in the form of `TypeName.X.Y.Z`, where `X`, `Y`, `Z` are major, minor, and errata type versions, respectively. If some (or all) version specifiers are omitted, wildcard matching is used instead. Additionally, `TYPEs` that specify minor and errata versions also match resources with greater values because those are backward compatible.

`PROPERTY` is case-sensitive, and nested properties can be selected by adding `'/'` delimiters. The `-s` option cannot accept nested properties.

`VALUE` can represent JSON (objects, arrays, null, etc.). Quotes are not necessary for string values, except to preserve white space.

The contents of `FILE` must be valid JSON.

Examples

To get the root resource:

```
Shell> restclient -uri redfish/v1 -g
```

To get the `AdminName` property from the `BIOS` resource:

```
Shell> restclient -uri redfish/v1/Systems/1/Bios -g Attributes/AdminName
```


To get the `iSCSIInitiatorName` property from the `iScsi` resource:

```
Shell> restclient -uri redfish/v1/Systems/1/Bios/iScsi -g iSCSIInitiatorName
```

To get the system ROM version from a `ComputerSystem` resource:

```
Shell> restclient -type ComputerSystem -g Oem/Hpe/Bios/Current
```

To set the `AdminName` property of the `BIOS` resource:

```
Shell> restclient -uri redfish/v1/Systems/1/Bios/Settings  
-s Attributes/AdminName=\"First Last\"
```

To set the `iSCSIInitiatorName` property of the `iScsi` resource:

```
Shell> restclient -uri redfish/v1/Systems/1/Bios/Settings  
-s iSCSIInitiatorName=\"InitiatorName\"
```

To send a `POST` request:

```
Shell> restclient -uri redfish/v1/Systems/1 -m POST -i PostData.json  
To write a response body to a file:
```

```
restclient -uri redfish/v1/Managers/1 -m GET -o Managers1.json
```

rm/del

Syntax

```
rm[-q] file/directory[file/directory...]
```

Description

Deletes one or more files or directories. The `del` command is an internal alias for this command.

Options

-q

Deletes in quiet mode, without displaying a confirmation prompt.

file

Specifies the file name to be deleted. Wildcards are permitted.

directory

Specifies the directory to be deleted. Wildcards are permitted.

Usage

If the target is a directory, this command deletes the directory, including all its subdirectories. This command is not allowed to redirect a file whose parent directory (or the file itself) is being deleted.

Removing a read-only file or directory results in a failure. Removing a directory containing one or more read-only files results in a failure. If an error occurs, `rm` exits immediately, and later files or directories are not removed.

You cannot remove a directory when the current directory is itself or its subdirectory. If the file specified for deletion contains wildcards, you are not prompted for confirmation.

You cannot remove the root directory, or the current directory or its ancestor.

Examples

Attempting to remove multiple directories at one time when directories cannot be found (causing the command to exit):

```
fs0:\> ls test
Directory of: fs0:\test

    06/18/01    01:01p <DIR>          512 .
    06/18/01    01:01p <DIR>           0 ..
    06/19/01    12:59a <DIR>         512 temp1
    06/19/01    12:59a <DIR>         512 temp2
    0 File(s) 0 bytes
    4 Dir(s)
```

```
fs0:\> rm test\temp11 temp2
rm/del: Cannot find 'fs0:\test\temp11' - Not Found
```

To remove multiple directories with wildcards:

```
fs0:\> rm test\temp*
rm/del: Remove subtree 'fs0:\test\temp1' [y/n]? y
removing fs0:\test\temp1\temp1.txt
- [ok]
removing fs0:\test\temp1\boot\nshell.efi
- [ok]
removing fs0:\test\temp1\boot
- [ok]
removing fs0:\test\temp1
- [ok]
rm/del: Remove subtree 'fs0:\test\temp2' [y/n]? y
removing fs0:\test\temp2\temp2.txt
- [ok]
removing fs0:\test\temp2
- [ok]
```

Attempting to remove a directory containing a read-only file, causing an error prompt:

```
fs0:\> attrib +r test\temp1\readme.txt
A R fs0:\test\temp1\readme.txt

fs0:\> rm test\temp1
rm/del: Cannot open 'readme.txt' under 'fs0:\test\temp1' in
writable mode
- [error] - Access Denied
Exit status code: Access Denied
```

secboot

Syntax

```
secboot[-l all][[PK]][[KEK]][[db]][[dbx]][-sfo]
```

```
secboot[-f file]
```

```
secboot -e [PK][[KEK]][[db]][[dbx]][-f file] secboot -r [all | PK | KEK | db |
dbx] [-q]
```

```
secboot -d [all][[PK]][[KEK]][[db]][[dbx]][-i index][-q] secboot -x [all | PK |
KEK | db | dbx] [-i index][-f file]
```

Description



Displays and modifies the Secure Boot databases, keys, and security reports.

Options

-l

Displays Secure Boot databases and keys.

all

Displays or deletes signatures of all Secure Boot variables.

PK

Displays Platform Key (PK) information. This is case sensitive.

KEK

Displays Key Exchange Key (KEK) information. This is case sensitive.

db

Displays Allowed Signatures Database (DB) information.

dbx

Displays Forbidden Signatures Database (DB) information.

-sfo

Displays information in standard formatted output.

-e

Enrolls a DER-format X509 file or a hash of an EFI application or Shell Script in a Secure Boot variable. The Shell script must start with the signature "#!NSH"

-f file

Displays DER-format X509 file information.

-r

Resets all Secure Boot signatures to platform defaults.

-d

Deletes all signatures, or deletes signatures from a specified database.

-i index

Selects a signature (1,2,...) from a specific database.

-q

Displays in quiet mode without confirmation prompts.

-x

Exports certificates or hashes from the Secure Boot databases to destination files.

Examples

To display signatures of all Secure Boot variables:

```
Shell> secboot -l all
```

To display Allowed Signatures Database information:

```
Shell> secboot -l db
```

To display DER-format X509 file information:

```
Shell> secboot -f abc.der
```

To enroll a DER-format X509 file in a Secure Boot variable:

```
Shell> secboot -e db -f abc.der
```

To enroll a hash of an EFI application in the Allowed Signatures Database:

```
Shell> secboot -e db -f bootx64.efi
```

To enroll a hash of an EFI application on the network in the Allowed Signatures Database:

```
Shell> secboot -e db -f http://www.example.com/BOOT/bootx64.efi
```

To enroll a hash of a script in the Allowed Signatures Database:

```
Shell> secboot -e db -f Test.nsh
```

To enroll a hash of a script on a network location in the Allowed Signatures Database:

```
Shell> secboot -e db -f http://www.example.com/TESTS/Test.nsh
```

To Reset all Secure Boot signatures to platform defaults:

```
Shell> secboot -r all
```

To Reset KEK Secure Boot signatures to platform defaults:

```
Shell> secboot -r KEK
```

To delete all Secure Boot signatures:

```
Shell> secboot -d all
```

To delete the Platform Key:

```
Shell> secboot -d PK
```

To clear the Allowed Signatures Database:

```
Shell> secboot -d db
```

To delete the second signature from the Key Exchange Key:

```
Shell> secboot -d KEK -i 2
```

To export the third entry from the KEK Secure Boot database to a file `abc.der`

```
Shell> secboot -x KEK -i 3 -f abc.der
```

To export all entries from all Secure Boot databases(PK/KEK/db/dbx):

```
Shell> secboot -x all
```

To export all entries from the KEK Secure Boot database(PK/KEK/db/dbx):

```
Shell> secboot -x KEK
```

sermode

Syntax

```
sermode[handle[baudrate parity databits stopbits]]
```

Description

Sets serial port attributes.

Options

handle

Specifies a device handle for a serial port in hexadecimal format. To retrieve this information, use the `dh` command.

baudrate

Specifies a baud rate. Valid values are 50, 75, 110, 150, 300, 600, 1200, 1800, 2000, 2400, 3600, 4800, 7200, 9600 (default), 19200, 38400, 57600, 115200, 230400, and 460800. All other values will be converted to the next highest setting.

parity

Sets parity bit settings for specified serial port.

- `d`—Default parity
- `n`—No parity
- `e`—Even parity
- `o`—Odd parity
- `m`—Mark parity
- `s`—Space parity

databits

Specifies the number of data bits. Valid values are 4, 7, and 8 (default).

stopbits

Specifies the number of stop bits. Valid values are 0 (0 stop bits - default setting); 1 (1 stop bit); 2 (2 stop bits); and 15 (1.5 stop bits).

Examples

To display the settings for all serial port devices:

```
Shell> sermode
```

To display the settings for the serial port device with the handle `0x6B`:

```
Shell> sermode 6B
```

To configure the serial port settings for handle `0x6B` to 9600bps, even parity, 8 data bits, and 1 stop bit:

```
Shell> sermode 6B 9600 e 8 1
```

set

Syntax

```
set [-v] [sname [value]]
```

```
set [-d sname]
```

Description

Creates, displays, changes, or deletes a UEFI Shell environment variable.

Options

-v

Sets a volatile variable that disappears at the next boot.

-d

Deletes a variable.

sname

Specifies a variable name.

value

Specifies a variable value.

Usage

This command sets the environment variable specified by *sname* to the optional *value* parameters. If used without any parameters, all of the environment variables are displayed. If used with the **-d** option, the environment variable that is specified by *sname* is deleted.

This command does not change the value of the environment variable `lasterror`.

Examples

To add an environment variable:

```
Shell> set DiagnosticPath fs0:\efi\diag;fs1:\efi\diag
```

To display environment variables:

```
Shell> set
* path : .
diagnosticPath : fs0:\efi1.1\diag;fs1:\efi1.1\diag
```

To delete an environment variable:

```
Shell> set -d diagnosticpath
Shell> set
* path : .
```

To change an environment variable:

```
fs0:\> set src efi
fs0:\> set
]* path : .;fs0:\efi\tools;fs0:\efi\boot;fs0:\
src : efi
fs0:\> set src efi1.1
fs0:\> set
* path : .;fs0:\efi\tools;fs0:\efi\boot;fs0:\
src : efi1.1
```

To append an environment variable:

```
Shell> set
* path : .
Shell> set path %path%;fs0:\efi\tools;fs0:\efi\boot;fs0:\
Shell> set
* path : .;fs0:\efi\tools;fs0:\efi\boot;fs0:\
```

To set a volatile variable that disappears at the next boot:

```
Shell> set -v EFI_SOURCE c:\project\EFI1.1
Shell> set
* path : .;fs0:\efi\tools;fs0:\efi\boot;fs0:\
* EFI_SOURCE : c:\project\EFI1.1
```

setsize

Syntax

```
setsize size[-d]file[file...]
```

Description

Adjusts the size of a target file.

Options

size

Specifies the size of the file once it is adjusted.

-d

Deletes a variable.

file

Specifies the file that is adjusted in size.

Usage

When adjusting the size of a file, this command automatically truncates or extends the size of the file based on the passed-in parameters. If the file does not exist, it is created. Setting the size smaller than the actual data contained in the file truncates the data.

Example

To set the size of a file:

```
fs0:\> setsize size file [file...]
```

shift

Syntax

```
shift
```

Description

Shifts the contents of a UEFI Shell script's positional parameters, enabling scripts to process the parameters from left to right.

Usage

This command shifts the contents of a UEFI Shell script's parameters so that %1 is discarded, %2 is copied to %1, %3 is copied to %2, %4 is copied to %3, and so on. This allows UEFI Shell scripts to process script parameters from left to right.

This command does not change the UEFI shell environment variable `lasterror`.

Examples

To execute the script with echo on:

```
fs0:\> shift.nsh welcome EFI world
shift.nsh> echo welcome EFI world
```

```
welcome EFI world
shift
echo EFI world
EFI world
```

To execute the script with echo off:

```
fs0:\> echo -off
shift.nsh> shift.nsh welcome EFI world
welcome EFI world
EFI world
```

smbiosview

Syntax

```
smbiosview[-t SmbiosType][[-h SmbiosHandle]][-s][[-a]
```

Description

Displays SMBIOS information.

Options

-t

Displays all structures of the *SmbiosType*.

SmbiosType

Specifies a SMBIOS structure type. This is a number from 0 to 42. To view supported values and their descriptions, use the `help smbios` command.

-h

Displays structures of the *SmbiosHandle*.

SmbiosHandle

Specifies a unique 16-bit handle of a SMBIOS structure.

-s

Displays a statistics table.

-a

Displays all information.

Example

To display all structures for *SmbiosType* 7 (cache information):

```
fs0:\> smbiosview -t 7
```

stall

Syntax

```
stall time
```

Description

Sets a timed stall of operations, in microseconds, during a script execution.

Option

time

Specifies the number of microseconds for the processor to stall.

Example

To stall the processor for 20 microseconds:

```
Shell> stall 20
```

sysconfig

Syntax `sysconfig -i[all|settingname][-sfo] sysconfig -g[all|settingname]
[settingname][settingname...][-sfo]`

`sysconfig -s[settingname=settingvalue...]`

`sysconfig -s AdminPassword=settingvalue OldAdminPassword=settingvalue`

`sysconfig -s PowerOnPassword=settingvalue OldPowerOnPassword=settingvalue`

`sysconfig -d[get|set][DefaultType][settingname|all] [-sfo] sysconfig -export
filename[-ASCII]`

Description  Configures HPE system BIOS settings.

Options

-b

Displays one screen at a time.

-i

Shows information for the specified settings or all settings, including possible values.

settingname

Specifies a setting name about which to display information. See [sysconfig attributes](#).

all

Displays all information for all settings.

-g

Displays the current values of the selected settings or all settings.

-s

Sets the value of the specified setting.

settingvalue

Specifies a setting value.

AdminPassword

Specifies a new administrator password.

OldAdminPassword

Specifies the administrator password to be reset.

PowerOnPassword

Specifies a new password for powering on the server.

OldPowerOnPassword

Specifies the power-on password to be reset.

-d

Gets (lists) or sets the default value for a specified *DefaultType* or for all default type settings.

DefaultType

Specifies a default type setting to get (list) or set.

-import

Imports all settings from a script file.

-export

Exports all settings to a script file.

filename

Specifies the script file target for importing or exporting.

ASCII

Uses ASCII encoding for the file output.

-sfo

Displays information in standard formatted output.

Usage

To display or set string-type settings, use double quotes for any *settingvalue* that contains spaces or the characters '='. For example: "**sysconfig -s AdminName="Joe Smith"**".

To include a double-quote in a string-type *settingvalue*, enter \.

To include a \ character in a string-type *settingvalue*, enter \\.

To remove a *settingvalue* from a string-type setting, use double-quotes. For example: "**sysconfig -s AdminName=""**".

The `-sfo` option enables easier parsing of the command output using Shell scripts. When you use this option, any semi-colon character in the value string is replaced with the escape sequence '\; '.

The `-d set` option does not support `ProductId`, `SerialNumber`, `RomSelection`, `passwords`, or boot orders.

The `-export`, `-set` and `-import` options do not support SFO. The following `-sfo` formats are supported.

- For `-g (get)`:

Column	Description
1	Table name (<code>SysConfigGet</code>).
2	Setting name.
3	Setting value. Multiple values are separated by a semi-colon.

- For `-i`:

Column	Description
1	Table name (<i>SysConfigGet</i>).
2	Setting name.
3	Current setting value.
4	Possible setting values. Multiple values are separated by a semi-colon.
5	Setting type.

- For `-d (get)`:

Column	Description
1	Table name (<i>SysConfigGet</i>).
2	Setting name.
3	Setting value. Multiple values are separated by a semi-colon.
4	Default setting type.

Examples

To view the current values for all BIOS settings:

```
Shell> sysconfig -g all
```

To show detailed information about the `ProcHyperthreading` setting:

```
Shell> sysconfig -i ProcHyperthreading
```

To get the current value for the `ProcHyperthreading` setting:

```
Shell> sysconfig -g ProcHyperthreading
```

To set the `ProcHyperthreading` setting to disabled:

```
Shell> sysconfig -s ProcHyperthreading=Disabled
```

To list all default configuration settings:

```
Shell> sysconfig -d get
```

To list all system defaults:

```
Shell>sysconfig -d get SystemDefaults all
```

To list default settings for `NicBoot1`:

```
Shell> sysconfig -d get SystemDefaults NicBoot1
```

To set default values for all BIOS settings:

```
Shell> sysconfig -d set SystemDefaults all
```

To set the default setting for `NicBoot1`:

```
Shell> sysconfig -d set SystemDefaults NicBoot1
```

To set 123 as a new administrator password:

```
Shell> sysconfig -s AdminPassword=123 OldAdminPassword=""
```

To remove Joe Smith as a power-on password:

```
Shell> sysconfig -s PowerOnPassword="" OldPowerOnPassword="Joe Smith"
```

To list the UEFI boot order:

```
Shell> sysconfig -g UefiBootOrder
UefiBootOrder=
  0: Embedded UEFI Shell
  8: Rear USB 2
 10: Embedded LOM 1 Port 1
```

To set a new UEFI boot order so that `Embedded LOM 1 Port 1` boots first:

```
Shell> sysconfig -s UefiBootOrder=10,0
The new boot order is:
 10: Embedded LOM 1 Port 1
  0: Embedded UEFI Shell
  8: Rear USB 2
```

To set the default UEFI device priority:

```
Shell> sysconfig -s DefaultUefiDevicePriority=0,1,2,3,4,5,6,7
The new Default UEFI Device Priority is:
0: Floppy Drives
1: Optical Drives
2: USB Mass Storage Drives
3: Embedded Storage Controllers
4: Add-in Storage Controllers
5: Embedded Flexible Network
6: Add-in Network Controllers
7: Embedded UEFI Shell
```

To get the current value for the `EmbeddedUefiShell` setting in standard format output:

```
Shell> sysconfig -g EmbeddedUefiShell -sfo
ShellCommand,"sysconfig"
SysConfigGet,"EmbeddedUefiShell[Embedded UEFI Shell]","Disabled[Disabled]"
```

To show detailed information about the `EmbeddedUefiShell` setting in standard formatted output:

```
Shell> sysconfig -i EmbeddedUefiShell -sfo
ShellCommand,"sysconfig"
Enabled[Enabled];Disabled[Disabled]","Enum"
```

To get the default value for `ServerName` setting in standard formatted output:

```
Shell> sysconfig -d get ServerName -sfo
ShellCommand,"sysconfig"
```

```
SysConfigDefault,"ServerName", "", "SystemDefaults"
```

sysconfig attributes

You can view all current possible `sysconfig` attribute names, descriptions, current and possible values, and `Enum` setting types. The following example shows a portion of possible command output.

NOTE: Output for the `sysconfig` command varies by server model.

```
Setting Name      = EmbeddedSerialPort      [Embedded Serial Port]
Current Value     = Com1Irq4                [COM 1; IRQ4; I/O: 3F8h-3FFh]
Possible Values  = Com1Irq4                [COM 1; IRQ4; I/O: 3F8h-3FFh]
                  Com2Irq3                [COM 2; IRQ3; I/O: 2F8h-2FFh]
                  Disabled                [Disabled]
Setting Type     = Enum

Setting Name      = VirtualSerialPort    [Virtual Serial Port]
Current Value     = Com2Irq3                [COM 2; IRQ3; I/O: 2F8h-2FFh]
Possible Values  = Com1Irq4                [COM 1; IRQ4; I/O: 3F8h-3FFh]
                  Com2Irq3                [COM 2; IRQ3; I/O: 2F8h-2FFh]
                  Disabled                [Disabled]
Setting Type     = Enum

Setting Name      = NicBoot1              [Embedded LOM 1 Port 1]
Current Value     = NetworkBoot           [Network Boot]
Possible Values  = NetworkBoot           [Network Boot]
                  Disabled                [Disabled]
Setting Type     = Enum

Setting Name      = PreBootNetwork        [Pre-Boot Network Interface]
Current Value     = Auto                  [Auto]
Possible Values  = Auto                  [Auto]
                  EmbNicPort1             [Embedded LOM 1 Port 1 : HPE Ethernet 1Gb 4-port xxx Adapter - NIC]
                  EmbNicPort2             [Embedded LOM 1 Port 2 : HPE Ethernet 1Gb 4-port xxx Adapter - NIC]
                  EmbNicPort3             [Embedded LOM 1 Port 3 : HPE Ethernet 1Gb 4-port xxx Adapter - NIC]
                  EmbNicPort4             [Embedded LOM 1 Port 4 : HPE Ethernet 1Gb 4-port xxx Adapter - NIC]
                  FlexLom1Port1           [Embedded FlexibleLOM 1 Port 1 : HPE Ethernet 1Gb 4-port xxx Adapter - NIC]
                  FlexLom1Port2           [Embedded FlexibleLOM 1 Port 2 : HPE Ethernet 1Gb 4-port xxx Adapter - NIC]
                  FlexLom1Port3           [Embedded FlexibleLOM 1 Port 3 : HPE Ethernet 1Gb 4-port xxx Adapter - NIC]
                  FlexLom1Port4           [Embedded FlexibleLOM 1 Port 4 : HPE Ethernet 1Gb 4-port xxx Adapter - NIC]
                  Slot1NicPort1           [Slot 1 Port 1 : HPE Ethernet 1Gb 4-port xxx Adapter - NIC]
                  Slot1NicPort2           [Slot 1 Port 2 : HPE Ethernet 1Gb 4-port xxx Adapter - NIC]
                  Slot1NicPort3           [Slot 1 Port 3 : HPE Ethernet 1Gb 4-port xxx Adapter - NIC]
                  Slot1NicPort4           [Slot 1 Port 4 : HPE Ethernet 1Gb 4-port xxx Adapter - NIC]
Setting Type     = Enum

Setting Name      = Dhcpv4                [DHCPv4]
Current Value     = Enabled               [Enabled]
Possible Values  = Disabled              [Disabled]
                  Enabled                 [Enabled]
Setting Type     = Enum
```

sysinfo

Syntax

```
sysinfo[token][-sfo][-b][-v] [-o <file>]
```

Description



Displays system information, including system name, serial number, product ID, BIOS version, backup BIOS version, power management controller firmware version, boot mode, system memory, processors, iLO IP addresses, and network devices.

Options

token

Specifies what to display. Available tokens are:

- `summary`—A summary of system information
- `all`—All system information
- `cpu`—CPU information
- `mem`—Memory information
- `fw`—Firmware information
- `pci`—PCI device information

-sfo

Displays information in standard formatted output.

-b

Displays information one screen at a time.

-v

Displays information in verbose output.

-o <file>

Redirects output to the specified file.

Examples

To display a brief summary of system information:

```
Shell> sysinfo summary
```

To display memory information:

```
Shell> sysinfo mem
```

To display memory information in standard format output:

```
Shell> sysinfo mem -sfo
```

To display detailed memory information, including information about unpopulated slots:

```
Shell> sysinfo mem -v
```

To store all system information to a file:

```
Shell> sysinfo all -o file.txt
```

To store all system information to a network location:

```
Shell> sysinfo all -o http://www.example.com/file.txt
```

tftp

Syntax

```
tftp [-i interface] [-l port] [-r port] [-c retry count] [-t timeout][-s block size] host remotefilepath [localfilepath]
```

Description

Downloads a file from TFTP server.

Options

-i *interface*

Specifies an adapter name, such as eth0.

-l *port*

Specifies the local port number. Default value is 0, and the port number is automatically assigned.

-r *port*

Specifies the remote port number. Default value is 69.

-c *retry count*

Specifies the number of times to transmit request packets and wait for a response. The default value is 6.

-t *timeout*

Specifies the number of seconds to wait for a response after sending a request packet. Default value is 4.

-s *block size*

Specifies the TFTP `blksize` option as defined in RFC 2348. Valid range is between 8 and 65464, and the default value is 512.

host

Specifies a TFTP server IPv4 address.

remotefilepath

Specifies a TFTP server path of a file to download.

localfilepath

Specifies a local destination file path.

Usage

When an optional *localfilepath* is provided, the downloaded file is stored locally using the provided file path. When a local file path is not specified, the file is stored in the current directory using the file server's name.

Before using this command, the network interface intended to be used to retrieve the file must be configured. You can use the `ifconfig` command to do so.

When a network interface is defined with the `-i` option, only that interface is used to retrieve the remote file. Otherwise, all network interfaces are tried in the order they are discovered during the boot phase.

Example

To get the file `dir1/file1.dat` from the TFTP server at 192.168.1.1, and store it as `file2.dat` in the current directory:

```
fs0:\> tftp 192.168.1.1 dir1/file1.dat file2.dat
```

time

Syntax

```
time[hh:mm[:ss]][-tz tz][-d dl]
```

Description

Displays or sets the current time for the system.

Options

hh

Sets a new hour value (0–23).

mm

Sets a new minute value (0–59).

ss

Sets a new second value (0–59). If not specified, zero is used.

-tztz

Sets a time zone adjustment, in minutes, offset from GMT. Valid values range from -1440 and 1440 or be set at 2047. If the value is not present or set to 2047, time is interpreted as local time.

-d dl

Sets a daylight savings time value. Valid values are:

- 0—Time is not affected by daylight savings time.
- 1—Time is affected by daylight savings time but has not been adjusted.
- 3—Time is affected by daylight savings time and has been adjusted.

If no value follows `-d`, the current daylight savings time is displayed.

Usage

If no parameters are specified, this command shows the current time. If valid hours, minutes, and seconds are provided, the system's time is updated.

Except for numeric characters and the `:` (colon) character, all other characters in the argument are invalid. The Shell reports an error if the number is in the wrong hour/minute/second range. Spaces before or after the numeric character are not allowed. Spaces inserted into the number are not allowed either.

If the `seconds` parameter is not specified, seconds are set to zero by default.

Examples

To display the current system time:

```
fs0:\> time
16:51:03 (GMT+08:00)
```

To set the current system time:

```
fs0:\> time 9:51:30
fs0:\> time
09:51:31 (GMT+08:00)
```

To set the system time, and display the daylight savings time setting:

```
fs0:\> time 9:51:30
fs0:\> time -d
09:51:31 (GMT+08:00) DST: Not Affected
```

timezone

Syntax

```
timezone [-s:hh:mm] [-l-b-f]
```

Description

Displays or sets time zone information.

Options

- s**
Set time zone associated with hh:mm offset from UTC.
- l**
Display list of all time zones.
- b**
Displays one screen at a time.
- f**
Displays full information for specified time zone.

Usage

If no parameters are specified, this command shows the current time zone. If a valid `hh:mm` parameter is specified, the system's time zone information is updated.

Examples

To display all available time zones:

```
fs0:\> timezone -l
```

To set the time zone:

```
fs0:\> timezone -s 7:00
```

To display detailed information for the current time zone:

```
fs0:\> timezone -f
```

tlsconfig

Syntax

```
tlsconfig [-sfo]
tlsconfig [-f file] [-sfo]
tlsconfig -r [-q]
tlsconfig -e [-f file]
tlsconfig -d [-i index] [-q]
tlsconfig -x [-i index] [-f file] [-t DER | PEM] [-sfo]
tlsconfig -k [none | peer] [-o hostname | none]
tlsconfig -c [list | cipher-list]
tlsconfig -v [auto | 1.0 | 1.1 | 1.2]
```

Description



Displays and modifies TLS connection settings and certificates.

Options

- sfo**
Displays information in standard format output.

- f *file***
Displays X509 information from a PEM or DER-formatted file.
- r**
Resets the TLS configuration to platform defaults and deletes the certificate.
- q**
Suppresses the confirmation prompt.
- e**
Enrolls a X509 file into the TLS certificates database.
- d**
Deletes a TLS certificate.
- i**
Selects a TLS certificate.
- x**
Exports a TLS certificate.
- t**
Specifies the file format of exported certificate.
- k**
When `none` is specified, certificate validation for every TLS connection is disabled. When `peer` is specified, certificate validation for every TLS connection is enabled.
- o**
When `none` is specified, host name checking for certificate validation on every TLS connection is disabled. When `hostname` is specified, host name checking for certificate validation on every TLS connection is enabled.
- c**
Specifies or shows which ciphers are allowed for TLS connections.
- v**
Specifies which TLS protocol version to use. The `auto` setting negotiates the highest version supported by both TLS server and the client.

Usage

This command affects pre-boot features that use HTTPS functionality, such as HTTPS boot, and UEFI Shell commands, such as `webclient`.

Examples

To display all TLS settings and the certificates:

```
Shell> tlsconfig
```

To display detailed information from a X509 file:

```
Shell> tlsconfig -f abc.der
```

To display detailed information from a X509 file located on an http server:

```
Shell> tlsconfig -f http://www.example.com/CERT/abc.der
```

To enroll a X509 file in the TLS certificate database from a file:

```
Shell> tlsconfig -e -f abc.der
```

To enroll in the TLS certificate database from a file on network:

```
Shell> tlsconfig -e -f http://www.example.com/CERT/abc.der
```

To delete the first TLS certificate without a confirmation prompt:

```
Shell> tlsconfig -d -i 1 -q
```

To display the TLS certificates in PEM format:

```
Shell> tlsconfig -x
```

To export the third TLS certificate in DER format to a file named `abc.crt`:

```
Shell> tlsconfig -x -i 3 -f abc.crt -t DER
```

To export the third TLS certificate in DER format to a file `abc.crt`:

```
Shell> tlsconfig -x -i 3 -f http://www.example.com/CERT/abc.crt -t DER
```

To specify TLS cipher algorithms:

```
Shell> tlsconfig -c AES128-SHA256:AES256-SHA256
```

To show TLS supported ciphers:

```
Shell> tlsconfig -c list
```

To enable certificate validation on a TLS connection:

```
Shell> tlsconfig -k PEER -o HOSTNAME
```

To specify that TLS v1.2 protocol version is used:

```
Shell> tlsconfig -v 1.2
```

touch

Syntax

```
touch[-r]file[file...]
```

Description

Updates the time and date on a file to the current time and date.

Options

-r

Makes the update recursive into subdirectories.

file

Specifies the name or pattern of the file or directory to be updated. Multiple files can be updated at once.

Usage

When multiple files are specified, the system processes the files one by one and errors are ignored.

This command cannot change the time and date of read-only files and directories.

Example

To update the time and date on a file:

```
fs0:\> touch myfile.txt
```

type

Syntax

```
type file[file...]
```

Description

Sends the contents of a file to the standard output device.

Option

file

Specifies a file name to display.

Usage

If no options are specified, this command attempts to detect the file type. If the command fails, UCS-2 is presumed.

Examples

To display a file in format:

```
fs0:\> type pause.nsh
#
# Example script for 'pause' command
#
echo pause.nsh begin..
\date
time
pause
echo pause.nsh done.
```

To display multiple files:

```
fs0:\> type test.*
How to Install?
time
stall 3000000
time
```

unload

Syntax

```
unload[-n] [-v|verbose] handle
```

Description

Unloads a UEFI driver image from memory.

Options

-n

Skips all prompts during unloading so that the output can be used in a script file.

-v

Displays verbose image information before unloading.

verbose

Dumps verbose image information before unloading.

handle

Specifies the handle of the driver to unload in hexadecimal format.

Usage

Only drivers that support unloading can be unloaded.

Examples

To find the handle for the UEFI driver image to unload:

```
Shell> dh -b
```

To unload the UEFI driver image with handle 27:

```
Shell> unload 27
```

ver

Syntax

```
ver[-s|-t]
```

Description

Displays version information for the UEFI Shell and the underlying UEFI firmware.

Options

-s

Displays only the UEFI Shell version.

-t

Displays summary (terse) content.

Usage

This command retrieves information from the UEFI System Table or the Shell image.

Examples

To display only the UEFI Shell version:

```
fs0:\> ver -s 2.1
```

To display all information about the UEFI Shell firmware version:

```
fs0:\> ver
```

vol

Syntax

```
vol[fs][-n volumelabel]
```

```
vol[fs][-d]
```

Description

Displays volume information for a file system.

Options

fs

Specifies the name of the file system to display.

-n volumelabel

Specifies a name for the volume label. The following characters cannot be used: % ^ * + = [] | : ; " < > ? / . No spaces are allowed in the volume label.

-d

Specifies an empty volume label.

Usage

If *fs* is not specified, the current file system is assumed. If *-n* is specified, the volume label for *fs* is set to the *volume_label* parameter. The maximum length for *volume_label* is 11 characters.

Examples

To display the volume of the current file system:

```
fs0:\> vol
Volume has no label (rw)
1,457,664 bytes total disk space
1,149,440 bytes available on disk
512 bytes in each allocation unit
```

To change the label of *fs0*:

```
shell> vol fs0 -n help_test
Volume HELP_TEST (rw)
1,457,664 bytes total disk space
1,149,440 bytes available on disk
512 bytes in each allocation unit
```

To delete the volume label of *fs0*:

```
fs0:\> vol fs0 -d
Volume has no label (rw)
1,457,664 bytes total disk space
220,160 bytes available on disk
512 bytes in each allocation unit
```

Webclient

Syntax

```
webclient -g URL [-u user [-x passwd]][-o file][[-m][-noproxy] webclient -p
URL [-u user [-x passwd]][-i file][[-noproxy] webclient -l
```

Description



Transfers files from or to HTTP and FTP locations, and mounts ISO file systems.

Options

-g

Retrieves a document.

-p

Uploads a document.

URL

Specifies the hypertext address of a document.

-o

Redirects output to a specified file.

-i

Redirects input from a specified file.

file

Specifies a file name.

-m

Downloads and mounts an ISO file as a file system.

-l

Displays information on the network settings used.

-u user

Specifies a user name.

-x passwd

Specifies a password.

-noproxy

Does not use a proxy for the requested operation.

Usage

! **IMPORTANT:** You do not need to use `ifconfig` on a network interface if you plan to run `webclient` or `ftp` over the same interface because these interface and IP address settings are automatically selected by the **Pre-Boot Network Settings** configured in the System Utilities.

If the interface used by `ftp` and `webclient` happens to be configured by `ifconfig`, that setting is erased and, instead, the System Utilities **Pre-Boot Network Settings** menu is applied on the interface when the commands are run.

This command enables scriptable network transfers. A key benefit of using this command is that you can specify a URL with an HTTP address to retrieve a document, output it to a file at that address, and download a file or mount an ISO file. Press **ESC** or **Ctrl-C** to cancel a file transfer. For FTP URLs, authentication by supplying a user name and password in cleartext in the URL is supported. The maximum size of a file that can be saved to a FAT32 partition is 4 GB. The URL for downloads or uploads can be either HTTP or FTP. HTTP URL can be specified using either an IPv4/IPv6 address or host name. FTP URL can be specified using either an IPv4 address or host name. FTP over IPv6 is not supported. If neither `-m` nor `-o` are specified during the download operation, output is redirected to a new file on the current file system with the same name as the remote file. If `-i` is not specified during the upload operation, input is redirected from a file with the same name on the current file system. Make sure that you have free system memory (RAM) equivalent to the file size being transferred. If proxy is set, all download or upload operations are attempted via proxy. If proxy is not set or `-noproxy` is specified, all download or upload operations are attempted without proxy. When `-l` is specified, the settings used by the web client for network operations are displayed. You can change these settings in the Embedded UEFI Shell using `sysconfig`, or in the System Utilities.

Examples

To download an ISO file and mount an ISO file system:

```
fs0:\> webclient -g http://192.168.1.20/filename.iso -m
```

or

```
fs0:\> webclient -g http://[1234:0001:0002:0003:0004:0005:0006:0007]/  
filename.iso -m
```

or

```
fs0:\> webclient -g http://[1234::1]/filename.iso -m
```

or

```
fs0:\> webclient -g http://www.example.com/filename.iso -m
```

or

```
fs0:\> webclient -g ftp://192.168.1.20/filename.iso
```

To download a file from an HTTP server and save it to the current file system:

```
fs0:\> webclient -g http://192.168.1.20/file.html -o file.html
```

To download a file from an FTP server with the user name `user` and the password `pass`:

```
ftp://192.168.1.20/file.html -u user -x pass -o file.html
```

```
fs0:\> webclient -g
```

To upload a file from an FTP server with the user name `user` and the password `pass`:

```
ftp://192.168.1.20/file.html -u user -x pass -o file.html
```

```
fs0:\> webclient -p
```

To download a file from an external HTTP server and save it to the current file system when proxy is set:

```
webclient -g http://www.hpe.com/file.html
```

To download a file from an internal HTTP server and save it to the current file system when proxy is set:

```
-g http://192.168.1.20/file.html -noproxy
```

To display the network settings:

```
fs0:\> webclient -l
```


Running and editing UEFI Shell scripts

The following information describes how to run and edit the scripting functions in the UEFI Shell. A sample application that displays "Hello World" is also provided.

Methods for invoking scripts

You can invoke UEFI Shell scripts using either of these two methods:

- Using the UEFI Shell Script Auto-Start configuration in the System Utilities
- Manually invoking a Shell script

UEFI Shell Script Auto-Start configuration in the System Utilities

The startup script enables you to create a RAM disk, download files from the network, collect data, upload results back to network, and then boot to the OS without rebooting the system. You can store the script file on local media, or access it from a network location.

By default, **UEFI Shell Script Auto-Start** is enabled in the System Utilities and is configured so that the Shell looks for the `startup.nsh` file in any FAT16 or FAT32 file systems available. You can modify these settings so that the Shell looks for the startup script in a specific file system on attached media, or in a specific network location. When configured for a network location, you can specify the URL in HTTP or FTP format of the `startup.nsh` file location.

Manually invoking a Shell script

Procedure

1. Navigate to the location of the `.nsh` script file.
2. Type the name of the script you want to run.

Alternatively, you could enter the absolute path of the script to run, without navigating to the script location.

Exporting and importing settings to files

Procedure

1. To export all settings to a file:

```
fs0:\> sysconfig -export filename.txt
```

2. To import all settings to a file:

```
fs0:\> sysconfig -import filename.txt
```

Editing Shell scripts

You can edit script files offline or in the Shell using the `edit` command. You can also use the `type` command to output the script to the screen.

Sample UEFI Shell scripts

The following examples show a sample application source code script and a startup script.

Application source code script

The following sample source code shows how to implement the UEFI Shell application to print "Hello World" on the screen, and display the UEFI Shell version and environment variables. This sample script loops through all the `FS*` : file systems (`FS0`, `FS1`, `FS2`), looking for a specific input file (in this case, `sysconfig_backup.txt`).

Export example:

```
fs0:\>
```

```
@echo -off
cls
set -v myfs 0
if exist FS0:\* then
FS0:
echo "FS0:\ Found!"
goto FSFOUND
endif
echo "FS0:\ not found in system"
echo "Going to search first available file system from FS1, FS2,..., FS100"
pause
for %a run (1 100)
set -v myfs %a
if exist FS%myfs%:\* then
FS%myfs%:
echo "FS%myfs%:\ Found!"
goto FSFOUND
endif
endifor
## No valid FS found in system, so exit now
echo "No valid File System (FS0, FS1,..., FS100) found in system"
goto END
:FSFOUND
if exist sysconfig_backup.txt then
echo =====
echo "%cwd%sysconfig_backup.txt already exists! Continuing the execution of the"
echo "script will remove existing sysconfig_backup.txt file and create a new"
echo "latest system configuration sysconfig_backup.txt file."
echo =====
pause
rm sysconfig_backup.txt
endif
echo "Saving latest system configuration in sysconfig_backup.txt file."
sysconfig -export sysconfig_backup.txt
:END
set -d myfs
```

Import example:

```
fs0:\>
```

```
@echo -off
cls
set -v myfs 0
```

```

if exist FS0:\sysconfig_backup.txt then
FS0:
echo "FS0:\sysconfig_backup.txt Found!"
goto FSFOUND
endif
echo =====
echo "FS0:\sysconfig_backup.txt not found in system"
echo "Going to search sysconfig_backup.txt from all available file system"
echo "from FS1, FS2,..., FS100"
echo =====
pause
for %a run (1 100)
set -v myfs %a
if exist FS%myfs%:\sysconfig_backup.txt then
FS%myfs%:
echo "FS%myfs%:\sysconfig_backup.txt Found!"
goto FSFOUND
endif
endfor
## No valid sysconfig_backup.txt found in system, so exit now
echo "No valid sysconfig_backup.txt found from File System (FS0, FS1,..., FS100)"
goto END
:FSFOUND
if exist sysconfig_backup.txt then
echo =====
echo "%cwd%\sysconfig_backup.txt Found! Continuing the execution of the script"
echo "will cause system to override previous configuration and use the"
echo "configuration settings stored in sysconfig_backup.txt file."
echo =====
pause
sysconfig -import sysconfig_backup.txt
endif
:END
set -d myfs

```

Start-up script

The following is a sample configuration script that the Embedded UEFI Shell can run from a specified network location. You can use this script to create a RAM disk and then find the `FS` file system of the RAM disk to use for redirecting file output.

```

@echo -off

#
# Setup the environment variables. All of them are created as volatile.
#

#
# The volume label for the RAMDISK.
#
set -v VolumeLabel MYRAMDISK

#
# Variable to store the file system index that will be looped
# to determine the FS<x> number for the RAMDISK that is created.
#
set -v FsIndex 0

#
# Variable to store the output string of the ramdisk -c command.

```

```

# Successful creation of RAMDISK will give the following output:
# "RAM disk 'FSx:' created successfully." where x=0,1,2,...
#
set -v RamDiskStr 0

#
# Size of the RAMDISK in MegaBytes (MB).
#
set -v RamDiskSize 512

#
# Server URL hosting the OS loader and images.
# Can be HTTP or FTP. Names or IP addresses are allowed.
# Ensure DNS service is available and configured (see pre-requisites)
# when server names are used.
#
set -v Url http://192.168.1.1

#
# Files to be downloaded
#
set -v DownloadFile1 efilinux.efi
set -v DownloadFile2 deploy.kernel
set -v DownloadFile3 deploy.ramdisk

#
# Step 1. Create RAMDISK to store the downloaded OS programs.
#
echo "Creating a RAM Disk to save downloaded files..."
ramdisk -c -s %RamDiskSize% -v %VolumeLabel% -t F32 >v RamDiskStr
if %lasterror% ne 0x0 then
    echo "Cannot create a RAMDISK of size %RamDiskSize%."
    goto EXITSCRIPT
endif
echo "RAM Disk with Volume Label %VolumeLabel% created successfully."

#
# Step 2. Check each word in the output (RamDiskStr) and see if it matches
# the FSx: pattern. The newly created RAMDISK will be FS1: or higher.
# Here the check goes upto FS3: (the inner for loop), but a larger limit
# may be used in case many other file systems already exist before
# the creation of this RAMDISK. The FS for the RAMDISK is found when the
# FsIndex matches the FS<x> in RamDiskStr. Change the working directory
# to FS<FsIndex>:, so all downloads get saved there.
#
# FS0: is ignored. In the worst case, when no other usable
# file system is present, FS0: will map to the file system
# that this script is executing from.
#
#
for %a in %RamDiskStr%
    for %b run (1 10)
        set -v FsIndex %b
        if 'FS%FsIndex%:' == %a then
            FS%FsIndex%:
            goto RDFOUND
        endif
    endfor
endfor

#
# The following message appears if the newly created RAMDISK cannot be found.

```

```

#
echo "RAMDISK with Volume Label %VolumeLabel% not found!"
goto EXITSCRIPT

#
# The following message appears if the RAMDISK FS<x> has been found and you are in the
# RAMDISK's root folder.
#
:RDFOUND
echo "RAMDISK with Volume Label %VolumeLabel% found at FS%FsIndex%:."

#
# Step 3: Download the required files into the RAMDISK.
#
echo "Downloading %Url%/deploy/%DownloadFile1% (File 1 of 3...)"
webclient -g %Url%/deploy/%DownloadFile1% -o %DownloadFile1%
if %lasterror% ne 0x0 then
    goto EXITSCRIPT
endif

echo "Downloading %Url%/deploy/%DownloadFile2% (File 2 of 3...)"
webclient -g %Url%/deploy/%DownloadFile2% -o %DownloadFile2%
if %lasterror% ne 0x0 then
    goto EXITSCRIPT
endif

echo "Downloading %Url%/deploy/%DownloadFile3% (File 3 of 3...)"
webclient -g %Url%/deploy/%DownloadFile3% -o %DownloadFile3%
if %lasterror% ne 0x0 then
    goto EXITSCRIPT
endif

#
# Step4: Launch the boot loader.
#
echo "Starting the OS..."
%DownloadFile1% -f %DownloadFile2% initrd=%DownloadFile3%

#
# You reach here only if the downloads and booting failed.
#
:EXITSCRIPT
echo "Exiting Script."

```

The sample script does the following:

Procedure

1. Creates a temporary RAM disk for saving the downloaded boot loader, the OS kernel, file system and any configuration files required for the boot loader and kernel to initialize themselves and proceed with the installation over the network.
2. Determines the FS<x> ID for the newly-created RAM disk
3. Sets the working directory to the root of the RAM disk (for example FS1:\).
4. Downloads the required files to launch the OS: the boot loader, the OS kernel and an in-memory file system for the OS kernel.
5. Does one of the following:
 - a. If download of all the required files fails, performs cleanup and exits the startup script.

- b.** If the download is successful, launches the boot loader, and passes to the boot loader as command line arguments the path to the OS kernel file, its in-memory file system, and any arguments to the OS kernel (that the boot loader must pass to the kernel upon launching it).

The role of the UEFI Shell and the pre-boot script ends here, and the OS now is capable of proceeding with the deployment on its own, with the help of OS-specific deployment scripts embedded in its in-memory file system.

UEFI Programming Model

The UEFI Shell provides a programming API. You can use it to write your own UEFI applications for calling some of the Shell programmatic APIs or protocols. For more information, see the UEFI Shell Specification and EDK2. The UEFI Shell provides a programming API as listed in the following table. EFI_SHELL_PROTOCOL provides Shell services to UEFI applications. It provides UEFI Shell applications access to the low-level Shell functions, including files, pipes, environment variables, the current working directory, mappings, help text, aliases, and launching Shell applications and scripts.

Table 9: UEFI Application APIs

Function Type	Function Name	Description
EFI_SHELL_EXECUTE	Execute	Causes the Shell to parse and execute the command line.
EFI_SHELL_GET_ENV	GetEnv	Gets the environment variable.
EFI_SHELL_SET_ENV	SetEnv	Changes a specific environment variable.
EFI_SHELL_GET_ALIAS	GetAlias	Retrieves the alias for a specific Shell command.
EFI_SHELL_SET_ALIAS	SetAlias	Adds or removes the alias for a specific Shell command.
EFI_SHELL_GET_HELP_TEXT	GetHelpText	Returns help information about a specific command.
EFI_SHELL_GET_DEVICE_PATH_FROM_MAP	GetDevicePathFromMap	Returns the device path that corresponds to a mapping.
EFI_SHELL_GET_MAP_FROM_DEVICE_PATH	GetMapFromDevicePath	Returns the mapping that corresponds to a particular device path.

Table Continued

Function Type	Function Name	Description
EFI_SHELL_GET_DEVICE_PATH_FROM_FILE_PATH	GetDevicePathFromFilePath	Converts a file path to a device path, where all mappings have been replaced with the corresponding device paths.
EFI_SHELL_GET_FILE_PATH_FROM_DEVICE_PATH	GetFilePathFromDevicePath	Converts a device path to a file path, where the portion of the device path corresponding to one of the mappings is replaced with that mapping.
EFI_SHELL_SET_MAP	SetMap	Creates, updates, or deletes a mapping between a device and a device path.
EFI_SHELL_GET_CUR_DIR	GetCurDir	Returns the current directory on a device.
EFI_SHELL_SET_CUR_DIR	SetCurDir	Changes the current directory on a device.
EFI_SHELL_OPEN_FILE_LIST	OpenFileList	Opens the files that match the path pattern specified.
EFI_SHELL_FREE_FILE_LIST	FreeFileList	Frees the file list created by OpenFileList().
EFI_SHELL_REMOVE_DUP_IN_FILE_LIST	RemoveDupInFileList	Deletes the duplicate files in the given file list.

Table Continued

Function Type	Function Name	Description
EFI_SHELL_BATCH_IS_ACTIVE	BatchIsActive	Displays whether any script files are being processed.
EFI_SHELL_IS_ROOT_SHELL	IsRootShell	Verifies whether the active Shell is the root Shell.
EFI_SHELL_ENABLE_PAGE_BREAK	EnablePageBreak	Enables the page break output mode.
EFI_SHELL_DISABLE_PAGE_BREAK	DisablePageBreak	Disables the page break output mode.
EFI_SHELL_GET_PAGE_BREAK	GetPageBreak	Gets the enable status of the page break output mode.
EFI_SHELL_GET_DEVICE_NAME	GetDeviceName	Gets the name of the device specified by the device handle.
EFI_SHELL_GET_FILE_INFO	GetFileInfo	Displays information about a specific file handle.
EFI_SHELL_SET_FILE_INFO	SetFileInfo	Changes information about a specific file handle.
EFI_SHELL_OPEN_FILE_BY_NAME	OpenFileByName	Opens a file specified by name and returns a file handle.
EFI_SHELL_CLOSE_FILE	CloseFile	Closes an open file.
EFI_SHELL_CREATE_FILE	CreateFile	Creates a file.
EFI_SHELL_READ_FILE	ReadFile	Reads data from a file.

Table Continued

Function Type	Function Name	Description
EFI_SHELL_WRITE_FILE	WriteFile	Writes data to a file.
EFI_SHELL_DELETE_FILE	DeleteFile	Deletes a file.
EFI_SHELL_DELETE_FILE_BY_NAME	DeleteFileName	Deletes a file by name.
EFI_SHELL_GET_FILE_POSITION	GetFilePosition	Displays the current read/write position within a file.
EFI_SHELL_SET_FILE_POSITION	SetFilePosition	Changes the current read/write position within a file.
EFI_SHELL_FLUSH_FILE	FlushFile	Writes all buffered data to a file.
EFI_SHELL_FIND_FILES	FindFiles	Displays all files that match a pattern in a file list.
EFI_SHELL_FIND_FILES_IN_DIR	FindFilesInDir	Displays all files in a specified directory in a file list.
EFI_SHELL_GET_FILE_SIZE	GetFileSize	Displays the size of a file.
EFI_SHELL_OPEN_ROOT	OpenRoot	Displays the root directory of a file system.
EFI_SHELL_OPEN_ROOT_BY_HANDLE	OpenRootByHandle	Displays the root directory of a file system on a particular handle.

Table Continued

Function Type	Function Name	Description
EFI_EVENT	ExecutionBreak	An event signaled by the UEFI Shell when the user presses CTRL-C to indicate that the current UEFI Shell command execution should be interrupted.
UINT32	MajorVersion	The major version of the Shell environment.
UINT32	MinorVersion	The minor version of the Shell environment.

UEFI Shell command status codes

The following table lists the possible status codes displayed by the UEFI Shell when you issue a command. Codes vary by command.

Table 10: UEFI Shell command status codes

Status code	Description
SHELL_SUCCESS	The action is completed as requested.
SHELL_NOT_FOUND	The target file or set of files cannot be found.
SHELL_SECURITY_VIOLATION	The function cannot be performed due to a security violation. When Secure Boot is enabled, any UEFI application that is not digitally signed using one of the embedded Secure Boot certificates cannot run and returns a SECURITY_VIOLATION status code instead.
SHELL_INVALID_PARAMETER	One of the passed-in parameters is formatted incorrectly or its value is out of bounds.
SHELL_OUT_OF_RESOURCES	A request to set a variable in a non-volatile fashion cannot be completed. The resulting non-volatile request is converted into a volatile request.
SHELL_WRITE_PROTECTED	The media on which the action takes place is write-protected.
SHELL_DEVICE_ERROR	There is a hardware error preventing the completion of this command.

Websites

General websites

Hewlett Packard Enterprise Information Library

www.hpe.com/info/EIL

Single Point of Connectivity Knowledge (SPOCK) Storage compatibility matrix

www.hpe.com/storage/spock

Storage white papers and analyst reports

www.hpe.com/storage/whitepapers

UEFI Specification

www.uefi.org/specifications

UEFI Learning Resources

www.uefi.org/learning_center

RESTful API Tool

<http://www.hpe.com/info/redfish>

Contact Hewlett Packard Enterprise Worldwide

<http://www.hpe.com/assistance>

Subscription Service/Support Alerts

<http://www.hpe.com/support/e-updates>

Software Depot

<http://www.hpe.com/support/softwaredepot>

Customer Self Repair

<http://www.hpe.com/support/selfrepair>

Insight Remote Support

<http://www.hpe.com/info/insightremotesupport/docs>

For additional websites, see [Support and other resources](#).

Support and other resources

Accessing Hewlett Packard Enterprise Support

- For live assistance, go to the Contact Hewlett Packard Enterprise Worldwide website:
<http://www.hpe.com/info/assistance>
- To access documentation and support services, go to the Hewlett Packard Enterprise Support Center website:
<http://www.hpe.com/support/hpesc>

Information to collect

- Technical support registration number (if applicable)
- Product name, model or version, and serial number
- Operating system name and version
- Firmware version
- Error messages
- Product-specific reports and logs
- Add-on products or components
- Third-party products or components

Accessing updates

- Some software products provide a mechanism for accessing software updates through the product interface. Review your product documentation to identify the recommended software update method.
- To download product updates:
Hewlett Packard Enterprise Support Center
www.hpe.com/support/hpesc
Hewlett Packard Enterprise Support Center: Software downloads
www.hpe.com/support/downloads
Software Depot
www.hpe.com/support/softwaredepot
- To subscribe to eNewsletters and alerts:
www.hpe.com/support/e-updates
- To view and update your entitlements, and to link your contracts and warranties with your profile, go to the Hewlett Packard Enterprise Support Center **More Information on Access to Support Materials** page:
www.hpe.com/support/AccessToSupportMaterials

! **IMPORTANT:** Access to some updates might require product entitlement when accessed through the Hewlett Packard Enterprise Support Center. You must have an HPE Passport set up with relevant entitlements.

Customer self repair

Hewlett Packard Enterprise customer self repair (CSR) programs allow you to repair your product. If a CSR part needs to be replaced, it will be shipped directly to you so that you can install it at your convenience. Some parts do not qualify for CSR. Your Hewlett Packard Enterprise authorized service provider will determine whether a repair can be accomplished by CSR.

For more information about CSR, contact your local service provider or go to the CSR website:

<http://www.hpe.com/support/selfrepair>

Remote support

Remote support is available with supported devices as part of your warranty or contractual support agreement. It provides intelligent event diagnosis, and automatic, secure submission of hardware event notifications to Hewlett Packard Enterprise, which will initiate a fast and accurate resolution based on your product's service level. Hewlett Packard Enterprise strongly recommends that you register your device for remote support.

If your product includes additional remote support details, use search to locate that information.

Remote support and Proactive Care information

HPE Get Connected

www.hpe.com/services/getconnected

HPE Proactive Care services

www.hpe.com/services/proactivecare

HPE Proactive Care service: Supported products list

www.hpe.com/services/proactivecaresupportedproducts

HPE Proactive Care advanced service: Supported products list

www.hpe.com/services/proactivecareadvancedsupportedproducts

Proactive Care customer information

Proactive Care central

www.hpe.com/services/proactivecarecentral

Proactive Care service activation

www.hpe.com/services/proactivecarecentralgetstarted

Warranty information

To view the warranty information for your product, see the links provided below:

HPE ProLiant and IA-32 Servers and Options

www.hpe.com/support/ProLiantServers-Warranties

HPE Enterprise and Cloudline Servers

www.hpe.com/support/EnterpriseServers-Warranties

HPE Storage Products

www.hpe.com/support/Storage-Warranties

HPE Networking Products

www.hpe.com/support/Networking-Warranties

Regulatory information

To view the regulatory information for your product, view the *Safety and Compliance Information for Server, Storage, Power, Networking, and Rack Products*, available at the Hewlett Packard Enterprise Support Center:

www.hpe.com/support/Safety-Compliance-EnterpriseProducts

Additional regulatory information

Hewlett Packard Enterprise is committed to providing our customers with information about the chemical substances in our products as needed to comply with legal requirements such as REACH (Regulation EC No 1907/2006 of the European Parliament and the Council). A chemical information report for this product can be found at:

www.hpe.com/info/reach

For Hewlett Packard Enterprise product environmental and safety information and compliance data, including RoHS and REACH, see:

www.hpe.com/info/ecodata

For Hewlett Packard Enterprise environmental information, including company programs, product recycling, and energy efficiency, see:

www.hpe.com/info/environment

Documentation feedback

Hewlett Packard Enterprise is committed to providing documentation that meets your needs. To help us improve the documentation, send any errors, suggestions, or comments to Documentation Feedback (docsfeedback@hpe.com). When submitting your feedback, include the document title, part number, edition, and publication date located on the front cover of the document. For online help content, include the product name, product version, help edition, and publication date located on the legal notices page.