



Hewlett Packard
Enterprise

UEFI Shell User Guide for HPE ProLiant Gen9 and Synergy Servers

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 UEFI-based ProLiant Gen9 servers and Synergy compute modules. It is for the person who installs, administers, and troubleshoots servers and storage systems.

Part Number: 794198-006
Published: December 2016
Edition: 1

© Copyright 2014, 2016 Hewlett Packard Enterprise Development LP

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

Linux® is a U.S. registered trademark of Linus Torvalds.

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

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

UNIX® is a registered trademark of The Open Group.



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

Contents

1 Introduction.....	6
Commands and capabilities.....	6
Accessing the Embedded UEFI Shell.....	6
Accessing the UEFI Shell from a serial console connection.....	7
2 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
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
Switching to a file system from the Shell.....	15
Using common setup and configuration commands.....	16
Updating the firmware using the fwupdate command.....	16
UEFI Shell commands.....	16
ahsdownload.....	17
alias.....	18
attrib.....	19
boot.....	20
cd.....	21
cls.....	22
comp.....	22
compress.....	23
connect.....	24
cp.....	25
date.....	27
dblk.....	28
devices.....	28
devtree.....	29
dh.....	30
disconnect.....	32
dmem.....	32
drivers.....	33
echo.....	34
edit.....	35
eficompress.....	35
efidecompress.....	36
exit.....	36
ftp.....	36
fwupdate.....	37
getmtc.....	38
goto.....	38
help.....	38

ifconfig.....	39
imlview.....	40
load.....	41
ls/dir.....	41
map.....	44
memmap.....	46
mkdir.....	47
mode.....	48
mv.....	49
openinfo.....	49
parse.....	49
partitions.....	50
pause.....	50
pci.....	51
ping.....	52
ramdisk.....	52
reset.....	53
restclient.....	54
rm/del.....	55
secboot.....	56
set.....	57
setsize.....	58
shift.....	59
smbiosview.....	59
stall.....	60
sysconfig.....	60
sysconfig attributes.....	62
sysinfo.....	63
time.....	64
timezone.....	65
touch.....	65
type.....	66
unload.....	66
ver.....	67
vol.....	67
webclient.....	68
3 Running and editing UEFI Shell scripts.....	70
Methods for invoking scripts.....	70
UEFI Shell Script Auto-Start configuration in the System Utilities.....	70
Manually invoking a Shell script.....	70
Exporting and importing settings to files.....	70
Editing Shell scripts.....	70
Sample UEFI Shell scripts.....	70
Application source code script.....	70
Start-up script	72
4 UEFI Programming Model.....	75
5 UEFI Shell command status codes.....	77
6 Support and other resources.....	78
Accessing Hewlett Packard Enterprise Support.....	78
Accessing updates.....	78
Related information.....	78
Websites.....	79
Customer self repair.....	79

Remote support.....	79
Documentation feedback.....	79
Glossary.....	80
Index.....	81

1 Introduction

The system BIOS in all ProLiant Gen9 servers and Synergy compute modules 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 commands and capabilities are available in the UEFI Shell:

- Configuration commands:
 - BIOS configuration (`sysconfig`)
 - ROM, storage, and NIC adapter firmware updates (`fwupdate`)
- Scripting:
 - `nsh` files with standard scripting constructs, including `if`, `else`, `endif`, `shift`, and `for/endif`
 - 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`, `alias`, and `touch`
 - File editing (`edit`) and viewing (`type`)
 - Input and output redirection from and to consoles and files

Accessing the Embedded UEFI Shell

Prerequisites

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.

See "Embedded UEFI Shell options" in the *UEFI System Utilities User Guide for HPE ProLiant Gen9 and Synergy Servers*.

To access the Shell:

Use one of the following methods:

- During server POST, press **F11** (Boot Menu) in the POST screen.
- From the **System Utilities** screen, select **Embedded Applications**→**Embedded UEFI Shell** and press **Enter**. See “Working with Embedded Applications” in the *UEFI System Utilities User Guide for HPE ProLiant Gen9 and Synergy Servers*.
- Using a serial console connection. See [Accessing the UEFI Shell from a serial console connection](#).
- Using a serial port on the server. See “Serial Port Options” in the *UEFI System Utilities User Guide for HPE ProLiant Gen9 and Synergy Servers*.

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.

To access the Shell:

1. Boot the server.
2. Open a connection in an SSH client application using the 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. For more information about the commands used in the UEFI Shell, see [UEFI Shell command reference](#).
7. Enter `exit` to exit the Shell.

Example 1 Logging in from a serial console connection

```
login as: username
@<ip address>'s password: password
User: logged-in to <server path> / <server MAC address>
iLO 4 Standard 1.30 at January 16 2014
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>
```

2 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.
- **PgUp**—Scrolls up a page.
- **PgDown**—Scrolls down a page.

Getting started with the UEFI Shell

The following describes how to 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.

- [Limiting output to one screen at a time](#)
- [Displaying detailed output](#)
- [Suppressing the confirmation prompt](#)

Limiting output to one screen at a time

The following example shows how to use the `-b` option to limit output to one screen at a time. After displaying the first screen, the Shell prompts for either of the following:

- Press **Enter** to continue to the next screen.
- Press **Q** to quit the display.

Example 2 Using `-b` to display output one screen at a time

```
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:
```

For an example of how to use this option with the `help` command, see [Example 6 \(page 12\)](#).

Enabling or disabling global page breaks

Use the `pagebreak` command to enable or disable global output pagination. This command creates a UEFI environmental variable that is retained until a server reboot.

Examples

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

Use the `-v` option to display detailed (verbose) output of certain commands. Shell commands supporting this option include:

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

The following example shows how to display detailed `dh` (device handle) output one screen at a time.

Example 3 Using `-v` to display detailed output

```
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(CDDB7B35-6833-4ED6-9AB2-57D2ACDDF6F0) 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

Use the `-t` option to display summary (terse) output of certain commands. Shell commands supporting this option include:

- `ver`

The following example shows how to use the `ver` command to display detailed UEFI version information and how to use the `-t` option to display a summary.

Example 4 Using `-t` to display summary output

```
Shell> ver
HP Embedded UEFI Shell v2.1
(C) Copyright 1982 - 2015 Hewlett Packard Enterprise Development L.P.
UEFI v2.40 (HP, 0x00010000)
HP ProLiant System BIOS - P79 (01/14/2014)

Shell> ver -t
HP Embedded UEFI Shell v2.1
```

Suppressing the confirmation prompt

Use the `-q` option to execute certain commands in quiet mode—without a confirmation prompt. Shell commands supporting this option include:

- `cp`
- `rm/del`

The following example shows how to use this option with the `rm` command to remove all `temp` directories without displaying a prompt.

Example 5 Using `-q` to suppress the confirmation prompt

```
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

There are a variety of command options to display detailed and summarized help for one or more commands.

- [Displaying all command help](#)
- [Displaying help for a specific command](#)
- [Displaying all help for commands beginning with a specific character](#)

Displaying all command help

The following example shows how to display help for all commands one screen at a time:

Example 6 Displaying all command help one screen at a time

```
Shell> help -b
ahsdownload - Downloads the Active Heath System (AHS).
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.
fwupdate   - Updates BIOS or device firmware.
Press ENTER to continue or 'Q' break:
```

Displaying help for a specific command

Use any of the following syntax options to display specific command help:

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

The following example shows one way to display help for the `ls` command (one screen at a time).

Example 7 Displaying help for the `ls` command

```
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      - Display 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    - Display 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].
```

For more usage details for the `help` command, see [“help” \(page 38\)](#).

Displaying all help for commands beginning with a specific character

Use a wildcard (*) to display help for all commands beginning with a specific character. The following example shows how to do this for all commands beginning with the character `a`.

Example 8 Displaying help for commands beginning with a

```
Shell> help a*
```

Downloads the Active Health System (AHS).

```
AHSDOWNLOAD [-n <file name> | -u <serial number>]
[-s <start date> -e <end date>]
[-case <case number>] [-name ,contact name>]
[-phone <phone number>] [-email email address>]
[-company <company name>] [-a] [-l] [-q]

-n                -Overrides the default AHS file name.
-u                -Overrides the serial number field in the default AHS
                  file name.
-s                -Specifies a start date.
-e                -Specifies an end date.
-a                -Downloads all data from the AHS log.
-l                -Displays AHS available download date range information.
-q                -Performs the download in quiet mode without user prompts.
-case            -Specifies a support case number.
-name            -Specifies a customer name.
-phone           -Specifies a customer phone number.
-email           -Specifies a customer email address.
-company         -Specifies a customer company name.
-file name       -Specifies a destination file name.
-serial number   -Specifies a system serial number.
-start date      -Specifies a the start date (in YYYY-MM-DD format) on which the data from the AHS
                  log is to be retrieved.
-end date        -Specifies the last date (in YYYY-MM-DD format) on which the data from the AHS
                  log is to be retrieved.
-case number     -Adds a support case number to the customer information file.
-contact name    -Adds a customer name to the customer information file.
-phone number    -Adds a customer phone number to the customer information file.
-email address   -Adds a customer email address to the customer information file.
-company name    -Adds a customer company name to the customer information file.
```

NOTES:

1. The Active Health System monitors and records changes in the server hardware and system configuration. The Active Health System assists in diagnosing problems and delivering rapid resolution when system failures occur.
2. The Active Health System does not collect information about your operations, finances, customer, employee, partners, or data center (for example, IP addresses, host names, user names, and passwords are not collected). By downloading and sending the Active Health System data, you agree to have support personnel use the data for analysis, technical resolution, and quality improvements.
3. This command mounts an AHS partition and downloads AHS log files and bundles into an .ahs package file.
4. If the start date (-s) and end date (-e) are not specified, by default AHS data related to the last 7 days is downloaded.
5. If a start date (-s) without an end date (-e) is specified, AHS uses the current date as and end date to download.
6. If a start date (-s) and end date (-e) are specified, only AHS data in that date range, including start and end dates, is downloaded.
7. If a file name is not specified using the -n option, a default file name is generated. For instance, AHS_XXXXXXXXXX_20140821.ahs.
8. This command does not support downloading an Active Health file (.ahs) to the internal SD card.

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 7 days (the default range):
fs0:\> ahsdownload
- * To download an AHS file to the default location with a complete log available:
fs0:\> ahsdownload -a
- * To download an AHS file with a specific date range:
fs0:\> ahsdownload -n ahstestfile.ahs -s 2014-04-02 -e 2014-04-07
- * To download an AHS file to the default location with a complete og and customer information:
fs0:\> ahsdownload -a -case 1234 -name Joe -phone 123-456-7890
-email ahstest@testing.com -company MyCompany

Displays, creates, or deletes UEFI Shell aliases.

```
ALIAS [-d|-v] [alias-name] [command-name]m all files with extension '.inf':
fs0:\> attrib -r *.inf
-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

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:

1. Using HDD, USB, or iLO virtual USB, attach a FAT16 or FAT32 formatted file system.
2. Use the `map -r` command to refresh file system mappings. See [map](#).
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.

The following example shows how to access the `fs0` file system from the Shell.

Example 9 Accessing the `fs0` file system from the Shell

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

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

Using common setup and configuration commands

- 1 **How do I enter the Embedded UEFI Shell?**
See [Accessing the Embedded UEFI Shell](#).
- 2 **How do I update the firmware or system ROM?**
See [fwupdate](#).
- 3 **How do I determine if a server has UEFI boot options?**
See [sysconfig](#).
- 4 **How do I configure the time zone?**
See [timezone](#).
- 5 **How do I create a temporary RAM disk from the Embedded UEFI Shell?**
See [ramdisk](#).
- 6 **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](#).
- 7 **How do I reset one or all mappings between user-defined names and a device handles?**
See [map](#).
- 8 **How do I display all devices that are compliant with the UEFI Driver Model?**
See [devtree](#).
- 9 **How do I exit the Embedded UEFI Shell back to the System Utilities?**
See [exit](#).

Updating the firmware using the `fwupdate` command

1. Access the System ROM Flash Binary component for your server from the Hewlett Packard Enterprise Support Center (<http://www.hpe.com/support/hpesc>). When searching for the component, always select **Cross operating system** to locate the binary file.
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. Enter `fwupdate -d BIOS -f filename` to flash the system ROM.
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.

UEFI Shell commands

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

Prerequisite

All commands require BIOS administrator authority. For information on setting the administrator password, see “Server Security” in the *UEFI System Utilities User Guide for HPE ProLiant Gen9 and Synergy Servers*.

All commands require BIOS administrator authority. For information on setting the administrator password, see [sysconfig](#).

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.

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 Active Health System (AHS) records.

Options

<code>-n <i>filename</i></code>	Specifies a name for the AHS file.
<code>-u <i>serialnumber</i></code>	Specifies a system serial number.
<code>-s <i>startdate</i></code>	Specifies the start date (in YYYY-MM-DD format) for retrieving data.
<code>-e <i>enddate</i></code>	Specifies the end date (in YYYY-MM-DD format) for retrieving data.
<code>-case <i>casenumber</i></code>	Adds a support case number to the customer information file.
<code>-name <i>contactname</i></code>	Adds a contact name to the customer information file.
<code>-phone <i>phonenumber</i></code>	Adds a phone number to the customer information file.
<code>-email <i>emailaddress</i></code>	Adds an email address to the customer information file.
<code>-a</code>	Downloads all data from the AHS log.
<code>-l</code>	Displays AHS download date range information.
<code>-q</code>	Performs the download in quiet mode without user prompts.

Usage

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

The 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 7 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 7 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
```

alias

Syntax

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

Description

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

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 <code>archive</code> attribute.
[+s -s]	Sets or clears the <code>system</code> attribute.
[+h -h]	Sets or clears the <code>hidden</code> attribute.
[+r -r]	Sets or clears the <code>read-only</code> attribute.
<i>file...</i>	Specifies the file name. Wildcards are permitted.
<i>directory...</i>	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
```

boot

Syntax

```
boot [[-d[-sfo]]|[-n num]|-all|-pxe] [r]
```

Description

 Boots or displays UEFI boot options.

Options

- `-d` Displays UEFI boot options in order.
- `-n` Boots 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.
- `-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
```

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

cls

Syntax

```
cls[color]
```

Description

Clears the standard output and changes the background color.

Options

color 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

Usage

If *color* is not specified, the background color does not change.

Examples

To clear the standard output without changing the background 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 the default color:

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

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][-cl 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.
-cl 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.

<code>-od <i>output_directory</i></code>	Specifies a directory where files from the zip file are to be extracted.
<code>-i <i>input_file</i></code>	Specifies an input file.
<code>-sfo</code>	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

<i>devicehandle</i>	Specifies a device handle in hexadecimal format.
<i>driverhandle</i>	Specifies a driver handle in hexadecimal format.
<code>-c</code>	Connects only the console devices described in UEFI Shell environment variables and related devices.
<code>-r</code>	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

<code>-r</code>	Creates a recursive copy.
<code>-q</code>	Creates a quiet copy (with no prompt).
<code>src src...</code>	Specifies a source file or directory name. Wildcards are permitted.
<code>dst</code>	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).

yy Specifies a two-digit year date.

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

- | | |
|--------------------------------|--|
| <code>-lxxx</code> | Displays device handles in a specific language. For a list of possible code options, see the UEFI specification. |
| <code><i>handle</i></code> | Displays the handle for a specific device. |
| <code><i>-p prot_id</i></code> | Displays protocol information associated to handles. If not specified, all protocols are displayed. |
| <code>-d</code> | Displays UEFI driver model-related information. |
| <code>-b</code> | Displays one screen at a time. |
| <code>-v</code> | Displays verbose information. |
| <code>-sfo</code> | Displays information in standard formatted output. See Table 5 (page 43) . |

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 `-pprot_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
Driver Name	Name of driver producing the handle.
Controller Name	Name of controller producing the handle.
Handle number	Integer number of the handle.
Device Path	Device path associated with the handle.
Protocol Identifiers	Semicolon-delimited list of protocol identifiers or GUIDs.

disconnect

Syntax

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

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.

Usage

This command does not support output redirection.

Examples

To disconnect all drivers from all devices:

```
Shell> disconnect -r
```

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 0000000001AF3088 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 *.....*
```

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. See [Table 4 \(page 34\)](#).

Usage

For a description of what is listed, see [Table 4 \(page 34\)](#).

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
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[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.
`-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
```

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.

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

<i>cmd</i>	Specifies the command for which to display help.
<i>pattern</i>	Specifies the pattern to use for displaying command help.
<i>special</i>	Displays a list of the special characters used in the shell command line.
<code>-usage</code>	Displays usage information for the command. This results in the same display as specifying <code>-section:NAME</code> and <code>-section:SYNOPSIS</code> .
<code>-v</code>	Displays verbose information.
<code>-section <i>sectionname</i></code>	Displays the specified section of the help information.
<code>-b</code>	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 below). 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 below.

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[-c [name]][-l[name]]
ifconfig[-s name dhcp|[static IPaddress mask gateway]|permanent]
```

Description

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

Options

<code>-c</code>	Clears the configuration.
-----------------	---------------------------

<code>-l</code>	Lists the configuration.
<code>-s</code>	Sets the configuration.
<code>name</code>	Specifies an adapter name. For example, <code>eth0</code> .
<code>dhcp</code>	Specifies that DHCP4 is to dynamically request IPv4 addresses for all or a specific interface.
<code>static IPaddress</code>	Specifies a static IPv4 address in four integer values, each between 0 and 255, separated by periods.
<code>mask</code>	Specifies a subnet mask in four integer values, each between 0 and 255, separated by periods.
<code>gateway</code>	Specifies a default gateway in four integer values, each between 0 and 255, separated by periods.
<code>permanent</code>	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 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.

You can use this command to configure host networks for pre-boot 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. If `permanent` is not specified, the configuration is one-time only. If `permanent` is specified, the configuration survives a network stack reload.

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
```

To use the static IPv4 address configuration for the `eth0` interface, and to set this configuration to survive a network reload:


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

imlview

Syntax

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

Description

 Displays and exports the Integrated Management Log (IML).

Options

<code>-sfo</code>	Displays information in standard formatted output.
<code>-b</code>	Displays the IML log one page at a time.
<code>-c</code>	Clears IML log entries.

`-export filename` Exports the IML log to a specified file.

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
```

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 non-system and non-hidden 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. See [Output details—ls command \(volume information\)](#), and [Output details—ls command \(file information\)](#).
- 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 of 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` flag determines whether a recursive search is performed.
- The option flag `-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)          61,788 bytes
    4 Dir(s)
```

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)          29 bytes
    0 Dir(s)
```

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)          28,739 bytes
    0 Dir(s)
```

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.
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.

Table 6 Output details—ls command (file information) (continued)

Column	Description
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"> • a—Archive • d—Directory • h—Hidden • r—Read-Only • s—System
File Creation Time	Time when the file was created, in the format hh:mm:ss.
File Creation Date	Date when the file was created, in the format dd:mm:yyyy.
File Access Time	Time when the file was accessed, in the format hh:mm:ss.
File Access Date	Date when the file was accessed, in the format dd:mm:yyyy.
File Modification Time	Time when the file was modified, in the format hh:mm:ss.
File Modification Date	Date when the file was modified, in the format dd:mm:yyyy.

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. See [Table 7 \(page 45\)](#).

<code>-u</code>	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.
<code>handle</code>	Specifies the number of the handle.
<code>mapping</code>	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. See [Output details—memmap command](#).

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 000000000750000-0000000001841FFF 0000000000010F2 0000000000000009
LoaderCode 0000000001842000-00000000018A3FFF 000000000000062 0000000000000009
available 00000000018A4000-00000000018C1FFF 000000000000001E 0000000000000009
LoaderData 00000000018C2000-00000000018CAFFF 0000000000000009 0000000000000009
BS_code    00000000018CB000-0000000001905FFF 000000000000003B 0000000000000009
BS_data    0000000001906000-00000000019C9FFF 00000000000000C4 0000000000000009
...
RT_data    0000000001B2B000-0000000001B2BFFF 0000000000000001 8000000000000009
BS_data    0000000001B2C000-0000000001B4FFFF 0000000000000024 0000000000000009
reserved   0000000001B50000-0000000001D4FFFF 0000000000000200 0000000000000009

reserved :      512 Pages (2,097,152)
LoaderCode:     98 Pages (401,408)
LoaderData:     32 Pages (131,072)
BS_code :       335 Pages (1,372,160)
BS_data :       267 Pages (1,093,632)
RT_data :        19 Pages (77,824)
available :    4,369 Pages (17,895,424)
Total Memory: 20 MB (20,971,520) 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

Table 8 Output details—memmap command (continued)

Column	Description
	<ul style="list-style-type: none"> • 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

<i>filename</i>	Specifies a source file name.
<i>tablename</i>	Specifies a table name to be parsed.
<i>column</i>	Specifies a one-based column index for determining which value from a particular record to parse.
<code>-i instance</code>	Starts parsing with the <i>n</i> th instance of the specified <i>tablename</i> , after the specified instance of <code>ShellCommand</code> . If not present, all instances are returned.
<code>-s instance</code>	Starts parsing with the <i>n</i> th instance of the specified <code>ShellCommand</code> 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]]
```

Description

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

Options

<i>bus</i>	Bus number
<i>dev</i>	Device number
<i>func</i>	Function number
-s seg	Specifies a segment number.

`-i` Information interpreted.

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
```

ping

Syntax

```
ping[-s sourceIP][-n count][-l size]targetIP
```

Description

Pings the target host with an IPv4 stack.

Options

`-s` Specifies that the source adapter is 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 a 20 byte echo request datagram:

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


ramdisk

Syntax

```
ramdisk -c[-s size][v volumelabel][-t type]  
ramdisk -d[fs|all]
```

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

Description

 Creates and deletes RAM disks.

Options

<code>-c</code>	Creates a RAM disk.
<code>-s <i>size</i></code>	Specifies RAM disk size in MB. Valid values are from 4 MB to 512 MB.
<code>-v <i>volumelabel</i></code>	Specifies a volume label name for the RAM disk. Valid values are up to 11 characters, without spaces or the following characters: % ^ * + = [] : ; \ < > ? /.
<code>-t <i>type</i></code>	Specifies a file system type.
<code>-d</code>	Deletes one or more RAM disks.
<code><i>fs</i></code>	Specifies the file system drive for RAM disk deletion.
<code>all</code>	Deletes all RAM disks.
<code>-l</code>	Lists all RAM disks.
<code>-sfo</code>	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 2000 MB maximum size

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

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
```

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
```

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 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.

General options

- i *FILE* Sends the contents of a specific file in the request contents.
- 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 `Type Name.X.Y.Z`, where *X*, *Y*, *Z* are major, minor, and errata type versions, respectively. If some (or all) version specifiers are omitted, then 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.

Example

To get the root resource:

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

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

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

To get the `BIOS` version from a `ComputerSystem` resource:

```
Shell> restclient -type ComputerSystem -g Bios/Current/VersionString
```

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

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

To send a `POST` request:

```
Shell> restclient -uri rest/v1/Systems/1 -m POST -i PostData.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

<code>-q</code>	Deletes in quiet mode, without displaying a confirmation prompt.
<code>file</code>	Specifies the file name to be deleted. Wildcards are permitted.
<code>directory</code>	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\temp1 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] [-q]
secboot [-d all] | [PK] | [KEK] | [db] | [dbx] [-i index] [-q]

```

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.

<code>dbx</code>	Displays Forbidden Signatures Database (DB) information.
<code>-sfo</code>	Displays information in standard formatted output.
<code>-e</code>	Enrolls a DER-format X509 file or a hash of an EFI application in a Secure Boot variable.
<code>-f file</code>	Displays DER-format X509 file information.
<code>-r</code>	Re-initializes all Secure Boot signatures to platform defaults.
<code>-d</code>	Deletes all signatures, or deletes signatures from a specified database.
<code>-i index</code>	Selects a signature (1,2,...) from a specific database.
<code>-q</code>	Displays in quiet mode without confirmation prompts.

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 hash of an EFI application in the Allowed Signatures Database:

```
Shell> secboot -e db -f boot64.efi
```

To re-initialize all Secure Boot signatures to platform defaults:

```
Shell> secboot -r
```

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
```

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

<code>-t</code>	Displays all structures of the <i>SmbiosType</i> .
<i>SmbiosType</i>	Specifies a SMBIOS structure type. This is a number from 0 to 42. To view supported values and their descriptions, use the <code>help smbios</code> command.
<code>-h</code>	Displays structures of the <i>SmbiosHandle</i> .

<i>SmbiosHandle</i>	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] [-b]  
sysconfig -g[all|settingname] [settingname...] [-sfo] [-b]  
sysconfig -s[settingname=settingvalue...]  
sysconfig -s AdminPassword=settingvalue OldAdminPassword=settingvalue  
sysconfig -s PowerOnPassword=settingvalue  
OldPowerOnPassword=settingvalue [-b]  
sysconfig -d[get|set] [DefaultType] [settingname|all] [-sfo] [-b]  
sysconfig -import filename.txt [ASCII]  
sysconfig -export filename.txt
```

Description

 Displays or 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.
<i>settingname</i>	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.

<code>-s</code>	Sets the value of the specified setting.
<code>settingvalue</code>	Specifies a setting value.
<code>AdminPassword</code>	Specifies a new administrator password.
<code>OldAdminPassword</code>	Specifies the administrator password to be reset.
<code>PowerOnPassword</code>	Specifies a new password for powering on the server.
<code>OldPowerOnPassword</code>	Specifies the power-on password to be reset.
<code>-d</code>	Gets (lists) or sets the default value for a specified <i>DefaultType</i> or for all default type settings.
<code>DefaultType</code>	Specifies a default type setting to get (list) or set.
<code>-import</code>	Imports all settings from a script file.
<code>-export</code>	Exports all settings to a script file.
<code>filename</code>	Specifies the script file target for importing or exporting.
<code>ASCII</code>	Uses ASCII encoding for the file output.
<code>-sfo</code>	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 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 `-export`, `-set` and `-import` options do not support SFO.

The \ character is allowed in string-type settings.

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 type settings:

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

To list all default settings:

```
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 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.

For more information about each attribute, see *UEFI System Utilities User Guide for HPE ProLiant Gen9 and Synergy Servers*.

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     = FlexLom1               [Embedded : FlexLOM 1]
Possible Values   = FlexLom1               [Embedded : FlexLOM 1]
                  PciSlot1                 [PCIe Slot 1]
                  PciSlot2                 [PCIe Slot 2]
                  PciSlot3                 [PCIe Slot 3]
                  PciSlot4                 [PCIe Slot 4]
                  PciSlot5                 [PCIe Slot 5]
                  PciSlot6                 [PCIe Slot 6]
                  PciSlot7                 [PCIe Slot 7]
                  PciSlot8                 [PCIe Slot 8]
                  PciSlot9                 [PCIe Slot 9]
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]
```

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.

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
```

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.

-tz tz 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
```

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.

Example

To display only the UEFI Shell version:

```
fs0:\> ver -s  
2.0
```

To display all information about the UEFI Shell firmware version:

```
fs0:\> ver  
HP Embedded UEFI Shell v2.1  
(C) Copyright 1982 – 2015 Hewlett Packard Enterprise Development L.P.  
UEFI v2.40 (HP, 0x00010000)  
HP ProLiant System BIOS - P79 (01/14/2014)
```

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 *volumelabel* parameter. The maximum length for *volumelabel* 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][-o file][[-m][[-l]]]
```

Description

 Downloads files from HTTP or FTP, and mounts an ISO file system.

Options

- g Retrieves a document specified by the URL.
- URL* Specifies the hypertext address of the document to retrieve. This can be either an IPv4 address or host name.
- o Redirects output to a specified file.
- file* Specifies the file name to which output is redirected.
- m Downloads and mounts an ISO file as a file system.
- l Displays information on the network settings used.

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 clear text in the URL is supported. The maximum size of a file that can be saved to a FAT32 partition is 4 GB. You must use an IPv4 address.

Examples

To download an ISO file from an HTTP or FTP server and mount an ISO file system:

```
fs0:\> webclient -g http://192.168.1.20/filename.iso -m
```

or

```
fs0:\> webclient -g http://www.hpe.com/filename.iso -m
```

or

```
fs0:\> webclient -g ftp://192.168.1.20/filename.iso -m
```

To download a file from an FTP or HTTP server and save it to the current file system:

```
fs0:\> webclient -g http://192.168.1.20/file.html -o file.html
```

or

```
fs0:\> webclient -g ftp://192.168.1.20/file.html -o file.html
```

or

```
fs0:\> webclient -g ftp://ftp.hpe.com/file.html -o file.html
```

To download a file from an FTP server with the user-name `user` and a password of `pass`:

```
fs0:\> webclient -g ftp://user:pass@192.168.1.20/file.html -o file.html
```

To display the network settings:

```
fs0:\> webclient -l
```

3 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:

- [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.nshfile` location. For more information, see "UEFI Shell Script Auto-Start" in the *UEFI System Utilities User Guide for HPE ProLiant Gen9 and Synergy Servers*.

Manually invoking a Shell script

1. Navigate to the location of the `.nsh` script file.
2. Double-click the file, or right-click it, and then select **Open**.

Exporting and importing settings to files

To export all settings to a file:

```
fs0:\> sysconfig -export filename.txt
```

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
endfor
## 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:

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:

- If download of all the required files fails, performs cleanup and exits the startup script.
- 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.

4 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.
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.
EFI_SHELL_BATCH_IS_ACTIVE	BatchIsActive	Displays whether any script files are being processed.

Table 9 UEFI Application APIs (continued)

Function Type	Function Name	Description
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 new file.
EFI_SHELL_READ_FILE	ReadFile	Reads data from a file.
EFI_SHELL_WRITE_FILE	WriteFile	Writes data to a file.
EFI_SHELL_DELETE_FILE	DeleteFile	Deletes a file.
EFI_SHELL_DELETE_FILE_BY_NAME	DeleteFileByName	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.
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.

5 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.

6 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/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, go to either of the following:
 - Hewlett Packard Enterprise Support Center **Get connected with updates** page:
<http://www.hpe.com/support/e-updates>
 - Software Depot website:
<http://www.hpe.com/support/softwaredepot>
- 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:
<http://www.hpe.com/support/AccessToSupportMaterials>

- ① **IMPORTANT:** Access to some updates might require product entitlement when accessed through the Hewlett Packard Enterprise Support Center. You must have an HP Passport set up with relevant entitlements.
-

Related information

The latest documentation for the UEFI System Utilities and Embedded Shell is available at: <http://www.hpe.com/info/ProLiantUEFI/docs>. Available documents include:

- *UEFI System Utilities User Guide for HPE ProLiant Gen9 and Synergy Servers*
- *UEFI Shell Quick Reference Card for HPE ProLiant Gen9 Servers*

- *UEFI System Utilities and Shell Release Notes for HPE ProLiant Gen9 and Synergy Servers*
- *UEFI Deployment Guide for HPE ProLiant Gen9 and Synergy Servers*

UEFI System Utilities and Shell Command Mobile Help for HPE ProLiant Gen9 and Synergy Servers is available by scanning the QR code located at the bottom of the System Utilities screen, or at <http://www.hpe.com/qref/ProLiantUEFI/Help>.

Websites

Website	Link
UEFI Specification	http://www.uefi.org/specifications
UEFI Learning Resources	http://www.uefi.org/learning_center
UEFI EDK2 project on SourceForge (download specifications and code)	http://sourceforge.net/apps/mediawiki/tianocore
Hewlett Packard Enterprise Information Library	http://www.hpe.com/info/enterprise/docs
Hewlett Packard Enterprise Support Center	http://www.hpe.com/support/hpesc
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

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.

For more information and device support details, go to the following website:

<http://www.hpe.com/info/insightremotesupport/docs>

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.

Glossary

ASR	Automatic Server Recovery
EFI	Extensible Firmware Interface
EMS	Emergency Management Services
iLO	Integrated Lights-Out
IML	Integrated Management Log
MTC	Monotonic Counter
PCIe	Peripheral Component Interconnect Express
POST	Power-On Self-Test
PXE	Pre-Boot Execution Environment
QPI	Intel's QuickPath Interconnect
RIS	Remote Installation Services
RTC	Real-Time Clock
SMBIOS	System Management BIOS
TPM	Trusted Platform Module
UEFI	Unified Extensible Firmware Interface
VSP	Virtual Serial Port

Index

A

- accessing
 - updates, 78
- accessing a file system, 15
- accessing the UEFI Shell, 6
- admin password, 60
- aliases, 18
- application APIs, 75
- attributes, 19

B

- background color, 22
- binding drivers, 24
- boot, 20

C

- changing the directory, 21
- command line completion, 8
- command line help, 12
- command line syntax examples, 8
- commands and capabilities, 6
- common tasks, 16
- comparing files, 22
- compressing files, 23
- configuring system BIOS, 60
- confirmation prompt, 11
- console output mode, 48
- contacting Hewlett Packard Enterprise, 78
- controlling command output, 9
- copying files and directories, 25
- creating and deleting RAM disks, 52
- customer self repair, 79

D

- date, 27
- daylight savings time, 64
- decompressing files, 23
- del alias, 55
- device blocks, 28
- device handles, 30
- device tree, 29
- devices, 28
- dir alias, 41
- directories
 - changing, 21
 - creating, 47
 - deleting, 55
 - listing, 41
- disconnecting device drivers, 32
- displaying command help, 12, 38
 - for a specific command, 12
 - for all commands, 12
 - for commands beginning with a specific character, 13
- displaying Secure Boot databases, keys, and security reports, 56
- document

- related information, 78
- documentation
 - providing feedback on, 79
- downloading files from HTTP or FTP, 68
- drivers, 33
 - binding, 24
 - loading, 41
 - unloading, 66

E

- echo, 34
- editing files, 35
- editing Shell scripts, 70
- EFI decompress, 36
- Embedded UEFI Shell
 - accessing , 6
 - before booting , 6
 - booting to , 6
 - prerequisites, 6
- exiting the Shell or a script, 36

F

- file and directory attributes , 19
- file system, 15
- files
 - comparing, 22
 - compressing and decompressing, 23
 - copying, 25
 - decompressing EFI, 36
 - deleting, 55
 - listing, 41
 - manipulating, 6
 - sending contents to an output device, 66
 - setting size, 58
 - transferring via FTP, 36
 - updating date and time, 65
 - volume information, 67
- firmware
 - displaying UEFI version, 67
 - updating, 37
- FTP transfer, 36, 68

G

- getting started, 8
- global page breaks, 10
- goto, 38

H

- handle protocols and agents, 49
- help, 12, 38
- Hewlett Packard Enterprise Shell commands, 8
- HTTP
 - downloading files from, 68

I

- ifconfig, 39
- imlview, 40

Integrated Management Log, 40

introduction, 6

invoking scripts, 70

ISO

mounting a file system, 68

L

listing file or directory information, 41

load, 41

M

manually invoking a Shell script, 70

map, 44

mapping names and device handles, 44

memory

displaying, 32

map, 46

MiniZip, 23

mkdir, 47

mode

console output, 48

modifying the IPv4 network stack address, 39

monotonic counter, 38

mounting an ISO file system, 68

moving files, 49

MTC, 38

O

openinfo, 49

options

boot, 20

output

controlling, 9

detailed, 10

one screen at a time, 9

summary, 11

output details

devices command, 29

dh command, 32

drivers command, 34

ls command file information, 44

ls command volume information, 43

memmap command, 47

sysconfig command, 62

P

parsing, 49

partitions, 50

passwords, 60

pausing, 50

PCI information, 51

ping, 52

power on password, 60

pre-boot network configuration, 39

R

RAM disks, 52

related documentation, 78

remote support, 79

resetting the system, 53

RESTful API, 54

S

sample Shell scripts, 70

script commands, 70

displaying, 34

exiting, 36

finding a label, 38

pausing, 50

shifting, 59

stalling execution, 60

scripting, 6

Secure Boot

displaying databases, keys, and security reports, 56

serial console connection, 7

setting file size, 58

Shell script manual invocation, 70

Shell scripts

editing, 70

manually invoking, 70

samples, 70

SMBIOS information, 59

stalling script execution, 60

support

Hewlett Packard Enterprise, 78

suppressing the confirmation prompt, 11

sysconfig

attributes, 62

displaying and configuring, 60

system date, 27

system information, 63

system reset, 53, 54

system time, 64

T

time zone, 65

touch, 65, 66

U

UEFI Application APIs, 75

UEFI Programming Model, 75

UEFI Shell command reference, 8

UEFI Shell command status

status code meanings, 77

UEFI Shell command status codes

status codes, 77

UEFI Shell commands, 16

ahsdownload, 17

alias, 18

attrib, 19

boot, 20

cd, 21

cls, 22

comp, 22

compress, 23

connect, 24

cp, 25

date, 27

- dblk, 28
- devices, 28
- devtree, 29
- dh, 30
- disconnect, 32
- dmem, 32
- drivers, 33
- echo, 34
- edit, 35
- eficompress, 35
- efidecompress, 36
- exit, 36
- ftp, 36
- fwupdate, 37
- getmtc, 38
- goto, 38
- help, 38
- ifconfig, 39
- imlview, 40
- load, 41
- ls, 41
- map, 44
- memmap, 46
- mkdir, 47
- mode, 48
- mv, 49
- openinfo, 49
- parse, 49
- partitions, 50
- pause, 50
- pci, 51
- ping, 52
- ramdisk, 52
- reset, 53
- restclient, 54
- rm/del, 55
- secboot, 56
- set, 57
- setsize, 58
- shift, 59
- smbiosview, 59
- stall, 60
- sysconfig, 60
- sysinfo, 63
- time, 64
- timezone, 65
- touch, 65
- type, 66
- unload, 66
- ver, 67
- vol, 67
- webclient, 68
- UEFI Shell script commands, 70
- unloading a driver image, 66
- updates
 - accessing, 78
- updating firmware components, 37
- URL
 - retrieving files from, 68

V

- variables
 - environment, 57
- version information, 67
- volume, 67

W

- webclient, 68
- websites, 79
 - customer self repair, 79