HP UEFI Shell User Guide for HP ProLiant DL580 Gen8 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 HP ProLiant DL580 Gen8 servers. It is for the person who installs, administers, and troubleshoots servers and storage systems.
Notices

Confidential computer software. Valid license from HP 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.

The information contained herein is subject to change without notice. The only warranties for HP 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. HP shall not be liable for technical or editorial errors or omissions contained herein.

Acknowledgments

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

Microsoft®, Windows®, Windows® XP, and Windows NT® are U.S. registered trademarks of Microsoft Corporation.

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

UNIX® is a registered trademark of The Open Group.
## Contents

1 Introduction .......................................................... 5  
Commands and capabilities .......................................................... 5  
Accessing the UEFI Shell .......................................................... 5  
   Accessing the HP UEFI System Utilities ........................................ 5  
   Before booting to the UEFI Shell ............................................... 6  
   Selecting the UEFI Mode ......................................................... 7  
   Enabling the Embedded UEFI Shell ........................................... 7  
Configuring the UEFI Shell ....................................................... 8  
   Adding Embedded UEFI Shell to the boot order list .................... 8  
   Changing the UEFI boot order ................................................. 8  
Booting to the UEFI Shell ....................................................... 9  
   Accessing the UEFI Shell from a serial console connection .......... 10  

2 UEFI Shell command reference ......................................... 12  
Command line syntax ......................................................... 12  
   Command line completion ................................................... 12  
Getting started ............................................................... 12  
   Controlling command output ............................................... 13  
      Limiting output to one screen at a time ............................... 13  
      Displaying detailed output .............................................. 14  
      Displaying summary output .......................................... 15  
      Suppressing the confirmation prompt ............................... 15  
Displaying command help ................................................. 16  
   Displaying all command help ............................................. 16  
   Displaying help for a specific command ............................... 16  
   Displaying all help for commands beginning with a specific character 17  
Accessing a file system from the Shell .................................. 19  
UEFI Shell Commands ....................................................... 19  
   alias ................................................................. 19  
   attrib ............................................................. 20  
   cd ................................................................. 21  
   cls ................................................................. 21  
   comp ............................................................. 22  
   cp ................................................................. 23  
   date ............................................................. 24  
   dblk .............................................................. 25  
   devices ......................................................... 26  
   devtree ......................................................... 27  
   dh ............................................................... 28  
   dmem ........................................................... 29  
   drivers ......................................................... 30  
   echo ............................................................ 31  
   edit ............................................................... 32  
   efi compressor ....................................................... 33  
   efidecompress ..................................................... 34  
   exit .............................................................. 34  
   fwupdate ......................................................... 34  
   getmtc .......................................................... 35  
   goto ............................................................. 35  
   help ............................................................. 35  
   ls/dir ........................................................... 36  
   map ............................................................. 39
Contents

Index.........................................................................................................76
Glossary....................................................................................................75

3 Running and editing UEFI Shell scripts.........................................................64
UEFI Shell script commands.............................................................................64
Invoking scripts.............................................................................................64
  Enabling the UEFI Shell Script Auto Start...................................................64
  Manually invoking a Shell script.................................................................64
Editing Shell scripts.......................................................................................64
Sample Shell scripts......................................................................................64

4 Using the UEFI Programming Model............................................................67
Sample Shell scripts......................................................................................68

5 UEFI Shell command status codes...............................................................71

6 Support and other resources.......................................................................72
  Contacting HP.............................................................................................72
    Subscription service....................................................................................72
  Related information.....................................................................................72
  Websites.......................................................................................................72
  Typographic conventions..............................................................................72
  HP Insight Remote Support software..........................................................73
  HP Insight Online.........................................................................................73

7 Documentation feedback..............................................................................74
Glossary........................................................................................................75
Index............................................................................................................76
1 Introduction

The system BIOS in all HP ProLiant DL580 Gen8 servers includes an embedded UEFI Shell in the ROM. Based on the UEFI Shell Specification, the Shell environment provides an API, and command line interfaces (CLI) that allow 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 UEFI Shell is enabled by default.

For more information, see HP UEFI System Utilities User Guide.

Commands and capabilities

The following capabilities are available in the UEFI Shell:

- **Scripting:**
  - nsh files with standard scripting constructs
  - echo, if / else / endif, shift, for/endfor commands
  - A startup.nsh auto start file similar to Autoexec.bat
  - Standard Format Output (-sfo argument) for most commands
    - Comma-separated output that can be parsed 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/output redirection from and to consoles and files

- **Configuration commands:**
  - BIOS configuration (sysconfig)
  - ROM firmware updates (fwupdate)

Accessing the UEFI Shell

You can access the UEFI Shell in one of the following ways:

- **During server POST, press F11 (Boot Menu) in the HP ProLiant POST screen.** For more information, see “Booting to the UEFI Shell” (page 9).
- **Using a serial console connection.** For more information, see “Accessing the UEFI Shell from a serial console connection” (page 10).
- **Using a serial port on the server.** For more information, see the HP UEFI System Utilities User Guide.

Accessing the HP UEFI System Utilities

To access the HP UEFI System Utilities:

1. Reboot the server. The server starts up and the HP ProLiant POST screen appears within a few minutes.
2. Press F9 in the HP ProLiant POST screen.
The **System Utilities** screen appears.

3. To navigate through and modify settings in the menu-driven interface, use the keys defined in the following table.

<table>
<thead>
<tr>
<th>Key</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Up or down arrow</td>
<td>Press to change a selection.</td>
</tr>
<tr>
<td>Enter</td>
<td>Press to select an entry.</td>
</tr>
<tr>
<td>ESC</td>
<td>Press to go back to the previous screen.</td>
</tr>
<tr>
<td>F1</td>
<td>Press to view online help about a selected option.</td>
</tr>
<tr>
<td>F7</td>
<td>Press to load default RBSU configuration settings. You need to reboot the system for changes to take effect. Press Enter to apply defaults. Press ESC if you want to cancel.</td>
</tr>
<tr>
<td>F10</td>
<td>Press to save your changes.</td>
</tr>
</tbody>
</table>

**POST screen keys**

- **F9**: Press during server POST or system reboot to display the System Utilities screen in the iLO 4 Remote Console.
- **F11**: Press during server POST to boot to the One-Time Boot Menu screen.

4. To exit the **System Utilities** screen and restart the server, press **Esc** until the main menu is displayed. Exit the utility by selecting **Exit and Resume Boot** in the main menu.

**Before booting to the UEFI Shell**

Before booting to the UEFI Shell, follow these steps:

1. Ensure that UEFI is selected as the boot mode. By default, the HP ProLiant DL580 boots in Legacy BIOS Boot Mode. See “Selecting the UEFI Mode” (page 7).
2. Ensure the embedded UEFI Shell is enabled. The default setting is enabled. See “Enabling the Embedded UEFI Shell” (page 7).
3. Optionally, add the Embedded UEFI Shell to the boot order list. See “Adding Embedded UEFI Shell to the boot order list” (page 8).
4. Change the Embedded UEFI Shell entry in the UEFI Boot order list. See “Changing the UEFI boot order” (page 8).

Selecting the UEFI Mode

By default, the HP ProLiant DL580 boots in Legacy BIOS Boot Mode.

To change the boot mode to UEFI in the System Utilities, perform the following steps:
1. From the System Utilities screen, select System Configuration→BIOS/Platform Configuration (RBSU)→Boot Options→Boot Mode and press Enter.
2. Select UEFI Mode to configure the system to boot to a UEFI compatible operating system.
3. Press F10 to save your selection.
4. From the System Utilities screen, select Exit and Resume Boot.
5. Reboot the server for the change to take effect.

Figure 1 Boot Options — Boot Mode screen

Enabling the Embedded UEFI Shell

You can enable or disable the Embedded UEFI Shell. The UEFI Shell is a pre-boot command line environment for scripting and running UEFI applications, including UEFI boot loaders. The UEFI Shell also provides CLI-based commands to obtain system information and configure and update the system BIOS. Enabling this option adds the Embedded UEFI Shell to the UEFI boot options. You can only configure this option if Boot Mode is set to UEFI. For more information, see “Selecting the UEFI Mode” (page 7).

To set the Embedded UEFI Shell:
1. From the System Utilities screen, select System Configuration→BIOS/Platform Configuration (RBSU)→System Options→UEFI Shell Options→Embedded UEFI Shell and press Enter.
2. Select one of the following options:
   • Enabled (default)
   • Disabled
3. Press **F10** to save your selection.

For more information, see “Running and editing UEFI Shell scripts” (page 64).

**Configuring the UEFI Shell**

The following sections explain how to add the Embedded UEFI Shell to the boot order list and change the location in the boot order list if needed. You can set the UEFI Shell as the last entry if you want to troubleshoot a failed boot.

**Adding Embedded UEFI Shell to the boot order list**

Adding the Embedded UEFI Shell as an entry in the Boot Order list is only applicable when the Embedded UEFI Shell is enabled and Boot Mode is set as UEFI. Otherwise, this option is grayed out.

**NOTE:** When enabling this option, the Embedded UEFI Shell does not appear in the UEFI Boot Order list until the next system reboot.

To add the Embedded UEFI Shell to the boot order list:
1. From the **System Utilities** screen, select **System Configuration**→**BIOS/Platform Configuration (RBSU)**→**System Options**→**UEFI Shell Options**→**Add Embedded UEFI Shell to Boot Order** and press **Enter**.
2. Select one of the following options:
   - **Enabled** (default)—Adds the embedded UEFI Shell to the boot order.
   - **Disabled**
3. Press **F10** to save your selection.

**Changing the UEFI boot order**

To change the order of the UEFI boot list:
1. From the **System Utilities** screen, select **System Configuration**→**BIOS/Platform Configuration (RBSU)**→**Boot Options**→**UEFI Boot Order** and press **Enter**. The UEFI Boot Order screen appears.
2. Select Enter to open the Boot Order list.
3. Use the arrow keys to navigate within the boot order list.
4. Press the + key (plus) to move an entry higher in the boot list.
5. Press the - key (minus) to move an entry lower in the list.
6. Press **F10** to save your selection.
7. From the System Utilities screen, select **Exit and Resume Boot**.
8. Reboot the server for the change to take effect.
Booting to the UEFI Shell

You can select the UEFI Shell for a one-time boot override whenever necessary. This option does not modify your predefined boot order settings.

To boot to the UEFI Shell:

1. Reboot the server.
2. During the reboot, press F11 Boot Menu in the HP ProLiant POST screen. In a few minutes, the One-Time Boot menu screen appears.
3. Select Embedded UEFI Shell, and press Enter.
4. Press any key to acknowledge that you are physically present. This steps ensure that certain features, such as disabling Secure Boot or managing the Secure Boot certificates using third-party UEFI tools are not restricted.
5. If an admin password is set, enter it at the prompt. For information on setting the admin password, see the HP UEFI System Utilities User Guide.
   The Shell> prompt appears.
6. Enter commands as explained in “UEFI Shell command reference” (page 12).
7. Enter the exit command to exit the Shell.
Figure 3 One-Time Boot menu screen

Figure 4 UEFI Shell Prompt screen

Accessing the UEFI Shell from a serial console connection

**NOTE:** When accessing the Shell from a serial console, English is the only available language for input and output.

To access the UEFI Shell from a serial console connection:
1. Boot the server as explained in “Booting to the UEFI Shell” (page 9).
2. Open a connection in an SSH client application using the server’s IP address.
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 UEFI Shell> prompt appears.
6. Enter commands as explained in “UEFI Shell command reference” (page 12).
7. Enter exit to exit the Shell.

Example 1 (page 11) shows an example of logging into the UEFI Shell from a serial console connection.

**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

Command line syntax

Table 1 (page 12) lists examples of how to interpret command syntax.

Table 1 Syntax examples

<table>
<thead>
<tr>
<th>Syntax example</th>
<th>Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>exit</td>
<td>Enter <code>exit</code>.</td>
</tr>
</tbody>
</table>
| `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-format output.  
    **NOTE:** 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 then the compressed output file as the `outfile` parameter. |

Command line completion

The command line completion feature simplifies entering and repeating commands. Table 2 (page 12) describes the command line completion keystrokes.

Table 2 Command line completion keys

<table>
<thead>
<tr>
<th>Key</th>
<th>Effect</th>
</tr>
</thead>
<tbody>
<tr>
<td>up arrow</td>
<td>Scrolls backward through the list of previously entered commands.</td>
</tr>
<tr>
<td>down arrow</td>
<td>Scrolls forward through the list of previously entered commands.</td>
</tr>
<tr>
<td>TAB</td>
<td>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.</td>
</tr>
<tr>
<td>PgUp</td>
<td>Page up.</td>
</tr>
<tr>
<td>PgDown</td>
<td>Page down.</td>
</tr>
</tbody>
</table>

Getting started

This section describes how to use some basic UEFI Shell commands for:

- “Controlling command output” (page 13)
- “Displaying command help” (page 16)
- “Accessing a file system from the Shell” (page 19)
- Using `sysconfig` options.
- Using the `reset` command.
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 provides examples on how to use them.

Limiting output to one screen at a time

Use the `-b` option to limit the display output of certain commands to one screen at a time. Shell commands supporting this option include:

- comp
- devices
- devtree
- dh
- dmem
- help
- memmap

Example 2 (page 14) shows how to limit `devtree` 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 \texttt{-b} to display output one screen at a time

Shell> \texttt{devtree -b}

Ctrl[03] Fv(652280D-28F9-4131-ADC4-F40EBFA45864)
Ctrl[04] Fv(770BF9B6-8AEE-4P4C-85E5-893FC3D2606C)
Ctrl[05] Fv(27A72E80-3118-4C0C-8673-AA5B4EFA9613)
Ctrl[06] MemoryMapped(0xB,0xFFD40000,0xFFD6FFFF)
Ctrl[07] Fv(5A515240-D1F1-4C58-9590-27B1F086827)
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 \texttt{help} command, see Example 7 (page 16).

Example 3 Enabling global page breaks

Use the \texttt{pagebreak} command to enabled global output pagination.

Example

To enable global output page breaks:
\texttt{fs0:/> set -v pagebreak 1}

This command creates a UEFI environmental variable that is retained until a server reboot.

To disable global output page breaks:
\texttt{fs0:/> set -v pagebreak 0}

Displaying detailed output

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

- \texttt{dh}
- \texttt{help}
- \texttt{map}

Example 4 (page 15) shows how to display detailed \texttt{dh} (device handle) output one screen at a time.
Example 4 Using \texttt{-v} to display detailed output

\begin{verbatim}
Shell> dh \texttt{-v} \texttt{-b}
01: LoadedImage
02: Decompress
03: UnknownDevice DevicePath Fv\{6522280D-28F9-4131-ADC4-F40EBFA45864\} UnknownDevice
04: UnknownDevice DevicePath Fv\{770BF9B6-8AF8-4F4C-85E5-893FC3D2606C\} UnknownDevice
05: UnknownDevice DevicePath Fv\{27A72E80-3118-4C0C-8673-AA5B4EFA9613\} UnknownDevice
06: UnknownDevice DevicePath Fv\{29A72E80-7BFE-4101-8459-AB5B3EFA4271\} UnknownDevice
07: UnknownDevice DevicePath Fv\{5A515240-D1F1-4C58-9590-27B1F0E86827\} UnknownDevice
08: UnknownDevice DevicePath Fv\{5E2363B4-3E9E-4203-B873-BB40DF46C8E6\} UnknownDevice
09: UnknownDevice DevicePath Fv\{CDBB7B35-6833-4ED6-9AB2-57D2ACDDFF60\} 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
\end{verbatim}

Press ENTER to continue or 'Q' break:

Displaying summary output

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

- \texttt{ver}

Example 5 (page 15) shows how to use the \texttt{ver} command to display detailed UEFI version information, and then how to use the \texttt{-t} option to display a summary.

Example 5 Using \texttt{-t} to display summary output

\begin{verbatim}
Shell> \texttt{ver}
UEFI Interactive Shell v2.0
Copyright 1982, 2014 Hewlett-Packard Development Company, L.P.
UEFI v2.31 (HP, 0x00010000)
ProLiant System BIOS - P79 (01/14/2014)
Shell> \texttt{ver -t}
UEFI Interactive Shell v2.0
\end{verbatim}

Suppressing the confirmation prompt

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

- \texttt{cp}
- \texttt{rm/del}

Example 6 (page 16) shows how to use this option with the \texttt{rm} command to remove all temp directories without displaying a prompt.
**Example 6 Using `-q` to suppress the confirmation prompt**

```bash
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**

Example 7 (page 16) shows how to display help for all commands one screen at a time:

**Example 7 Displaying all command help**

```
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.
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.
dm mem         - 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          - 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      - Invokes an HP UEFI Shell utility used to update System BIOS firmware.
getmtc        - Gets the MTC from BootServices and displays it.
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 -?**

Example 8 (page 17) shows one way to display help for the `ls` command (one screen at a time).
Example 8 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, then all files will be listed. If -a is not specified, then 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, then 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:
   - 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 35).

Displaying all help for commands beginning with a specific character

Use a wildcard (*) to display help for all commands beginning with a specific character. Example 9 (page 18) shows how to do this for all commands beginning with the character a.
Example 9 Displaying help for commands beginning with a

Shell> help a*
Displays, creates, or deletes UEFI Shell aliases.

```
ALIAS [-d|-v] [alias-name] [command-name]m all files with extension '.inf':
  fs0:
  -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 then 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...]
```

+| - 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 (wild cards 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, then 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, then 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
Accessing a file system from the Shell

Switch from the Shell to a file system before executing commands requiring file input or output. To switch to a file system:
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`” (page 39).
3. Enter one of the `fsx` file systems available, such as `fs0` or `fs1`, and then press Enter.

   The prompt changes to `fsx>`, where `x` is the number of the file system selected.

Files can now be accessed and written to any writable files in the specified file system.

Example 10 (page 19) shows how to access the `fs0` file system from the Shell.

Example 10 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.

UEFI Shell Commands

**NOTE:** All commands require BIOS administrator authority.

For more information, see the *HP UEFI System Utilities User Guide* for information on setting the administrator password.

### Table 3 Shell command listing components

<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Command name and summary line</td>
<td>Command name followed by a brief description of how it’s used</td>
</tr>
<tr>
<td>Syntax</td>
<td>Command syntax, including required and optional parameters</td>
</tr>
<tr>
<td>Options</td>
<td>Description of syntax parameters and variables</td>
</tr>
<tr>
<td>Description</td>
<td>Detailed description of command usage</td>
</tr>
<tr>
<td>Examples</td>
<td>One or more examples of command usage</td>
</tr>
<tr>
<td><strong>NOTE:</strong> User input is indicated in <strong>bold</strong> in command usage examples.</td>
<td></td>
</tr>
<tr>
<td>Output details</td>
<td>Descriptions of the command display fields, when applicable.</td>
</tr>
</tbody>
</table>

**alias**

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

**Syntax**

```
alias [ -d | -v ] [alias-name] [command-name]
```
Description
This command displays, creates, or deletes aliases in the UEFI Shell environment. 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 then 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
Displays or changes the attributes of files or directories.

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

Options
```
[ +a | -a ] Sets or clears the archive attribute.
[ +s | -s ] Sets or clears the system attribute.
[ +h | -h ] Sets or clears the hidden attribute.
[ +r | -r ] Sets or clears the read-only attribute.
file... Specifies the file name. Wild cards are permitted.
directory... Specifies the directory name. Wild cards are permitted.
```
**Description**

This command displays and sets the attributes of files or directories. 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, then 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, then all of the files in the current directory are displayed. If no attribute is specified, then the attributes of the files are displayed.

**Examples**

To display the attributes of a directory:

```plaintext
fs0:~> attrib fs0:\
attrib: D fs0:\
```

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

```plaintext
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:

```plaintext
fs0:~> attrib +s *.efi
```

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

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

**cd**

Displays or changes the current directory.

**Syntax**

`cd [path]`

**Options**

`path` Specifies the relative or absolute directory path.

**Description**

This command changes the current working directory that is used by the UEFI Shell environment. If a file system mapping is specified, then 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, then the current working directory (including file system mapping) is displayed to standard output.

Table 4 (page 22) describes the conventions that are used to refer to the directory, its parent, and the root directory in the UEFI Shell environment.
<table>
<thead>
<tr>
<th>Convention</th>
<th>Refers to the...</th>
</tr>
</thead>
<tbody>
<tr>
<td>.</td>
<td>Current directory.</td>
</tr>
<tr>
<td>..</td>
<td>Parent of the current directory.</td>
</tr>
<tr>
<td>\</td>
<td>Root of the current file system.</td>
</tr>
</tbody>
</table>

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

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

**cls**

Clears the standard output and optionally changes the background color.

**Syntax**

```
cls [color]
```

**Options**

`color` Specifies a new background color from the following options:

0—Black
1—Blue
2—Green
3—Cyan
4—Red
5—Magenta
Description
This command clears the standard output device with an optional background color attribute. If color is not specified, then 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

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

Syntax

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

Options

- -b Displays one screen at a time.
- file1 Specifies the first file name. Directory names or wild cards are not permitted.
- file2 Specifies the second file name. Directory names or wild cards are not permitted.

Description
This command compares the contents of two files in binary mode. It 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
**cp**

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

**Syntax**

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

**Options**

- `-r` Creates a recursive copy.
- `-q` Creates a quiet copy (with no prompt).
- `src src...` Specifies a source file or directory name. Wild cards are permitted.
- `dst` Specifies a destination file or directory name. Wild cards are not permitted. If not specified, then the current working directory is assumed to be the destination. If more than one directory is specified, then the last is always assumed to be the destination.

**Description**

This command copies one or more source files or directories to a destination. If the source is a directory, the `-r` flag must be specified. If `-r` is specified, then the source directory is recursively copied to the destination (which means that all subdirectories are copied). If a destination is not specified, then 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 four 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, then 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:
To copy multiple files to another directory:
```
fs0:\> cp temp.txt readme.txt
  copying fs0:\temp.txt -> fs0:\readme.txt
  - [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   ..
  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**

Displays and sets the current date for the system.

**Syntax**
```
date [mm/dd/ { yy | yyyy }][-sf]
```

**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-format output display.
Description

This command displays or sets the current date for the system. If no parameters are used, it shows the current date. If a valid month, day, and year are provided, then 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> date dblk8/4/98
  Shell> date 08/04/1998

  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

Displays one or more blocks from a block device.

Syntax

dblk [lba] [blocks] [-b]

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, then 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:
\[ \text{fs0:}\> \text{dblk fs0 2} \]

To display 0x5 blocks of fs0, beginning from block 0x12:
\[ \text{fs0:}\> \text{dblk fs0 12 5} \]

**devices**

Displays a list of devices managed by UEFI drivers.

**Syntax**

\[ \text{devices [-b][-lxxx][-sfo]} \]

**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-format output.

**Description**

This command prints a list of devices that are being managed by drivers that follow the UEFI Driver Model.

**Example**

To display all devices compliant with the EFI Driver Model:

\[ \text{Shell> devices} \]

```
   C   T    D
   R   P    A
   L   E    G

== == == == == == ==  =============================================================
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**

Table 5 (page 27) describes the possible output for this command.

<table>
<thead>
<tr>
<th>Column</th>
<th>Displays the...</th>
</tr>
</thead>
<tbody>
<tr>
<td>CTRL</td>
<td>Handle number of the device</td>
</tr>
<tr>
<td>TYPE</td>
<td>Device type. Options are:</td>
</tr>
<tr>
<td></td>
<td>• R—Root controller</td>
</tr>
<tr>
<td></td>
<td>• B—Bus controller</td>
</tr>
<tr>
<td></td>
<td>• D—Device controller</td>
</tr>
</tbody>
</table>
Table 5 Output details—*devices command* (continued)

<table>
<thead>
<tr>
<th>Column</th>
<th>Displays the...</th>
</tr>
</thead>
<tbody>
<tr>
<td>CFG</td>
<td>Configuration Protocol support status:</td>
</tr>
<tr>
<td></td>
<td>• Y—Yes</td>
</tr>
<tr>
<td></td>
<td>• N—No</td>
</tr>
<tr>
<td>DIAG</td>
<td>Diagnostics Protocol support status:</td>
</tr>
<tr>
<td></td>
<td>• Y—Yes</td>
</tr>
<tr>
<td></td>
<td>• N—No</td>
</tr>
<tr>
<td>#P</td>
<td>Number of parent controllers for this device</td>
</tr>
<tr>
<td>#D</td>
<td>Number of this type of devices</td>
</tr>
<tr>
<td>#C</td>
<td>Number of child controllers produced by this device</td>
</tr>
<tr>
<td>Device Name</td>
<td>Name of the device from the Component Name Protocol</td>
</tr>
</tbody>
</table>

devtree

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

**Syntax**

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

**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.

**Description**

This command prints a tree of devices being managed by drivers that follow the UEFI Driver Model. By default, the devices are printed in 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(0x0,0xFFFFD40000,0xFFFFDFFFF)
  Ctrl[07] Fv(SA1515240-D1F1-4C58-9590-27B1F0E86827)
  Ctrl[08] Fv(5E2363B4-3E9F-4203-B873-BB40DF46C8E6)
  Ctrl[09] Fv(CDBB7B35-6833-4ED6-9AB2-57D2ACDDF6F0)extension '.efi':
    Ctrl[15] PciRoot(0x0)/Pci(0x0,0x0)
    Ctrl[16] PciRoot(0x0)/Pci(0x2,0x0) from all files with extension '.inf':
    Ctrl[17] PciRoot(0x0)/Pci(0x2,0x0)/Pci(0x0,0x0)
    Ctrl[18] PciRoot(0x0)/Pci(0x2,0x1)
    Ctrl[19] PciRoot(0x0)/Pci(0x2,0x2)
    Ctrl[1A] PciRoot(0x0)/Pci(0x2,0x3)
    Ctrl[1B] PciRoot(0x0)/Pci(0x3,0x0)
    Ctrl[1C] PciRoot(0x0)/Pci(0x3,0x1)
    Ctrl[1D] PciRoot(0x0)/Pci(0x3,0x2)
```

28 UEFI Shell command reference
dh

Displays the device handles in the UEFI environment.

**Syntax**

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

**Options**

- `-lxxx` Displays device handles in a specific language. For a list of possible code options see the [UEFI Specification](#).

- `handle` Displays the handle for a specific device.

- `-p prot_id` Displays protocol information associated to handles. If not specified, all protocols are displayed.

- `-d` Displays UEFI driver model-related information.

- `-b` Displays one screen at a time.

- `-v` Displays verbose information.

- `-sfo` Displays information in standard-format output. See Table 8 (page 38).

**Description**

This command displays the device handles in the EFI environment. If used with a specific handle number, 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` or `handle` is specified, then 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(0376F3-11D4-BCEA-0080C73C8881)
  7: Image(WinNtBusDriver) DriverBinding
...
```

To display detailed information about handle 0x30:

```
Shell> dh 30
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
Table 6 (page 30) describes the possible output for this command.

Table 6 Output details — dh command

<table>
<thead>
<tr>
<th>Column</th>
<th>Displays the...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Driver Name</td>
<td>Name of driver producing the handle</td>
</tr>
<tr>
<td>Controller Name</td>
<td>Name of controller producing the handle</td>
</tr>
<tr>
<td>Handle number</td>
<td>Integer number of the handle</td>
</tr>
<tr>
<td>Device Path</td>
<td>Device path associated with the handle.</td>
</tr>
<tr>
<td>Protocol Identifiers</td>
<td>Semicolon-delimited list of protocol identifiers or GUIDs</td>
</tr>
<tr>
<td>Free Space</td>
<td>Total number of free bytes in the volume.</td>
</tr>
<tr>
<td>Block Size</td>
<td>Nominal block size by which files are typically grown, in bytes.</td>
</tr>
</tbody>
</table>

```dmem```

Displays the contents of system or device memory.

**Syntax**
```
dmem [-b] [address] [size] [-MMIO]
```
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.

Description

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

Example

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

Shell> dmem 1af3088 16
Memory Address 000000001AF3088 16 Bytes
01AF3088: 49 42 49 20 53 59 53 54-00 00 02 00 18 00 00 00 *IBI SYST........*
01AF3098: FF 9E D7 9B 00 00 00 00 *........*

drivers

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

Syntax

drivers [-lxxx] [-sfo]

Options

- lxxx  Displays drivers in a specific language. For a list of possible code options see the UEFI Specification.
- sfo  Displays in a standard-format output table. See Table 7 (page 32)).

Description

This command displays a list of information for drivers that follow the UEFI Driver Model in the UEFI environment. For a description of what is listed, see Table 7 (page 32).

Example

To display the driver list:

Shell> drivers

<table>
<thead>
<tr>
<th>T</th>
<th>D</th>
<th>V</th>
<th>DRIVER NAME</th>
<th>IMAGE NAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>D</td>
<td></td>
<td>Platform Console Management Driver</td>
<td>ConPlatform</td>
</tr>
<tr>
<td>C</td>
<td>I</td>
<td></td>
<td>Platform Console Management Driver</td>
<td>ConPlatform</td>
</tr>
<tr>
<td>P</td>
<td>F</td>
<td></td>
<td>Console Splitter Driver</td>
<td>ConSplitter</td>
</tr>
<tr>
<td>D</td>
<td>-</td>
<td></td>
<td>Console Splitter Driver</td>
<td>ConSplitter</td>
</tr>
<tr>
<td>B</td>
<td>-</td>
<td></td>
<td>Console Splitter Driver</td>
<td>ConSplitter</td>
</tr>
<tr>
<td>B</td>
<td>-</td>
<td></td>
<td>UGA Console Driver</td>
<td>GraphicsConsole</td>
</tr>
<tr>
<td>B</td>
<td>-</td>
<td></td>
<td>Serial Terminal Driver</td>
<td>Terminal</td>
</tr>
<tr>
<td>B</td>
<td>-</td>
<td></td>
<td>Generic Disk I/O Driver</td>
<td>DiskIo</td>
</tr>
<tr>
<td>B</td>
<td>-</td>
<td></td>
<td>FAT File System Driver</td>
<td>Fat</td>
</tr>
</tbody>
</table>
Output details

Table 7 (page 32) describes possible output for this command.

Table 7 Output details—drivers command

<table>
<thead>
<tr>
<th>Column</th>
<th>Displays the...</th>
</tr>
</thead>
<tbody>
<tr>
<td>DRV</td>
<td>Integer handle of the driver.</td>
</tr>
<tr>
<td>VERSION</td>
<td>Version number of the driver.</td>
</tr>
<tr>
<td>TYPE</td>
<td>Driver type. Possible values are:</td>
</tr>
<tr>
<td></td>
<td>• B—Bus driver</td>
</tr>
<tr>
<td></td>
<td>• D—Device driver</td>
</tr>
<tr>
<td>CFG</td>
<td>Configuration Protocol Support status:</td>
</tr>
<tr>
<td></td>
<td>• Y—Yes</td>
</tr>
<tr>
<td></td>
<td>• N—No</td>
</tr>
<tr>
<td>DIAG</td>
<td>Driver Protocol support status:</td>
</tr>
<tr>
<td></td>
<td>• Y—Yes</td>
</tr>
<tr>
<td></td>
<td>• N—No</td>
</tr>
<tr>
<td>#D</td>
<td>Number of devices this driver is managing.</td>
</tr>
<tr>
<td>#C</td>
<td>Number of child devices this driver has produced.</td>
</tr>
<tr>
<td>DRIVER NAME</td>
<td>Name of the driver from the Component Name Protocol.</td>
</tr>
<tr>
<td>IMAGE NAME</td>
<td>Device path from which the driver was loaded</td>
</tr>
</tbody>
</table>

**echo**

Controls whether or not script commands are displayed as they are read from the script file, and prints the given message to the display.

**Syntax**

echo [ -on | -off ]
echo message

**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.

**Description**

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 prints the specified message to the display.
NOTE: 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**

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

**Syntax**
```
edit [file]
```

**Options**

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

**Description**

This command allows a file to be edited using a full screen editor. The editor supports both UCS-2 and ASCII file types.

**Example**

To edit the shell.log file:
```
fs0:\> edit shell.log
```

**eficompress**

Compresses a file using the EFI Compression Algorithm.

**Syntax**
```
eficompress infile outfile
```

**Options**

`infile`  Specifies the filename for the uncompressed input file

`outfile`  Specifies the filename for the compressed output file
Description
This command compresses a file using the EFI Compression Algorithm and then writes the
compressed form out to a new file.

Example
To compress a file named uncompressed to file named compressed:
fs0:\> eficompress uncompressed compressed

efidecompress
Decompresses a file using the EFI Decompression Algorithm.

Syntax
efidecompress infile outfile

Options
infile Specifies the filename for the compressed input file
outfile Specifies the filename for the decompressed output file

Description
This command decompresses a file using the EFI Decompression Algorithm and then writes the
decompressed form out to a new file.

Example
To decompress a file named compressed to file named uncompressed:
fs0:\> eficompress compressed uncompressed

exit
Exits the UEFI Shell or the current script.

Syntax
exit [/b] [exit-code]

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, then 0 is returned.

Description
This command exits the UEFI Shell or, if /b is specified, the current script.

Example
To exit the UEFI Shell:
fs0:\> exit

fwupdate
Updates the system BIOS firmware.
Syntax
fwupdate firmwarefile [-r]

Options
firmwarefile  Specifies the file name of the firmware image to flash.
-r             Performs a reboot after flashing the firmware update.

Description
This command updates the system BIOS firmware. Optionally, specify an NVRAM reset, a reboot
after the flash, or both. If no options are specified, none are assumed.

Example
To update the system BIOS, reset NVRAM to its default configuration, and execute a reboot:
Shell> fwupdate YourFlashImage.BIN -r

getmtc
Gets the MTC from BootServices and displays it.

Syntax
getmtc
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.

go
go
go

Help to a label in a script.

Syntax
go label

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

Syntax

Options
cmd  Specifies the command for which to display help.
pattern  Specifies the pattern to use for displaying command help.
special  Displays a list of the special characters used in the shell command line.
-usage  Displays usage information for the command. This results in the
         same display as specifying -section:NAME and
         -section:SYNOPSIS .
-v  Displays verbose information.
-section sectionname  Displays the specified section of the help information.
-b  Displays the help output one screen at a time.
Description

This command displays information about one or more shell commands. If no other options are specified, each command is displayed along with a brief description of its function. If \(-v\) is specified, then all help information for the specified commands appears. If \(-s e c t i o n\) is specified, only the help section specified appears (see below). If \(-u s a g e\) is specified, then 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, then only the \(NAME\) section of the page appears.

Example

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

Shell> \texttt{help \ -b}
alias - Displays, creates, or deletes UEFI Shell aliases.
attrib - Displays or changes the attributes of files or directories.
\texttt{cd} - Displays or changes the current directory.
\texttt{cls} - Clears standard output and optionally changes background color.
\texttt{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> \texttt{help ls}
Shell> \texttt{? \ ls}
Shell> \texttt{ls \ -?}

To display the list of commands starting with the character \(p\):

Shell> \texttt{help p*}

\texttt{pause} - Prints a message and suspends for keyboard input

\texttt{ls/dir}

Lists the contents of a directory or file information.

Syntax

\texttt{ls [-r] [-a attrib] [-sfo] [file]}

Options

\texttt{-r} Displays recursively (including subdirectories)

\texttt{-a attrib} Displays only those files with the attributes specified. If no attributes are specified, then all files are listed. If \(-a\) is not specified, then 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

\texttt{-sfo} Displays in standard-format output. For more information, see “Output details—\texttt{ls} command (volume information)” (page 38), and “Output details—\texttt{ls} command (file information)” (page 38).

\texttt{file} Specifies the name of a file or directory. Wild cards are permitted.
Description
This command lists directory contents or file information. If no file name or directory name is
specified, then 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 wild card 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, then all files or directories are displayed, regardless of their attributes. If -a itself is not specified, then 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, 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
To display files with an attribute of `s`:

```bash
fs0:\> ls -as isabus.elf
Directory of: fs0:\

06/18/01 10:59p        28,739  Isabus.elf
1 File(s)       28,739 bytes
0 Dir(s)
```

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

```bash
fs0:\> ls -r -a elf
```

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

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

**Output details**

Table 8 (page 38) describes possible volume (directory) information output from this command. Table 9 (page 38) describes possible file information output from this command.

### Table 8 Output details—`ls` command (volume information)

<table>
<thead>
<tr>
<th>Column</th>
<th>Displays the...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Standard volume label</td>
</tr>
<tr>
<td>Total Size</td>
<td>Total number of bytes in the volume.</td>
</tr>
<tr>
<td>Read Only status</td>
<td>Read-only status as:</td>
</tr>
<tr>
<td></td>
<td>• True</td>
</tr>
<tr>
<td></td>
<td>• False</td>
</tr>
<tr>
<td>Free Space</td>
<td>Total number of free bytes in the volume.</td>
</tr>
<tr>
<td>Block Size</td>
<td>Nominal block size by which files are typically grown, in bytes.</td>
</tr>
</tbody>
</table>

### Table 9 Output details—`ls` command (file information)

<table>
<thead>
<tr>
<th>Column</th>
<th>Displays the...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Complete file name and directory, including the mapped name of the file system.</td>
</tr>
<tr>
<td>Logical Size</td>
<td>Size of the file, in bytes.</td>
</tr>
<tr>
<td>Physical Size</td>
<td>Size of the file in the volume, including any padding, in bytes.</td>
</tr>
<tr>
<td>Attributes</td>
<td>List of file attributes. Possible values are:</td>
</tr>
<tr>
<td></td>
<td>• a—Archive</td>
</tr>
<tr>
<td></td>
<td>• d—Directory</td>
</tr>
<tr>
<td></td>
<td>• h—Hidden</td>
</tr>
<tr>
<td></td>
<td>• r—Read-Only</td>
</tr>
<tr>
<td></td>
<td>• s—System</td>
</tr>
<tr>
<td>File Creation Time</td>
<td>Time when the file was created, in the format: hh:mm:ss</td>
</tr>
<tr>
<td>File Creation Date</td>
<td>Date when the file was created, in the format: dd:mm:yyyy</td>
</tr>
<tr>
<td>File Access Time</td>
<td>Time when the file was accessed, in the format: hh:mm:ss</td>
</tr>
<tr>
<td>File Access Date</td>
<td>Date when the file was accessed, in the format: dd:mm:yyyy</td>
</tr>
</tbody>
</table>
Table 9 Output details—ls command (file information) (continued)

<table>
<thead>
<tr>
<th>Column</th>
<th>Displays the...</th>
</tr>
</thead>
<tbody>
<tr>
<td>File Modification</td>
<td>Date when the file was modified, in the format: dd:mm:yyyy</td>
</tr>
<tr>
<td>Time</td>
<td>Time when the file was modified, in the format: hh:mm:ss</td>
</tr>
</tbody>
</table>

map

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

Syntax

map [-d mappedname]
map [-r | -v | -c | -f | -u | -t type [,type...] | mappedname ] [-sfo]
map [ mappedname | mapping ]

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-format output. For more information, see Table 10 (page 40).
- **-u** Adds mappings for newly-installed devices and removes mappings for uninstalled devices, but does not change the mappings of existing devices. Preserves user-defined mappings.

**handle** Specifies the number of the handle.

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

Description

This command creates a mapping between a user-defined name and a device. 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. It 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 the consistent mappings in the system.

The mapping consists of digits and characters. Other characters are illegal.

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

**Example**

To delete a mapping:

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

**Output details**

Table 10 describes possible output for this command.

**Table 10 Output details—map command**

<table>
<thead>
<tr>
<th>Column Number</th>
<th>Displays the...</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>The name of the table. The name is Mappings.</td>
</tr>
<tr>
<td>2</td>
<td>Mapped name. The mapped device name.</td>
</tr>
<tr>
<td>3</td>
<td>Device Path. The device path that corresponds to the mapped device name.</td>
</tr>
<tr>
<td>4</td>
<td>Consistent Name. The consistent mapped name (if any) that is equivalent to (mappedname). If (mappedname) is already a consistent mapped name, then this column is empty.</td>
</tr>
</tbody>
</table>

**memmap**

Displays the system memory map.

**Syntax**

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

**Options**

\(-b\) Displays one screen at a time.

\(-sfo\) Displays standard-format output in a detailed and a summary table. For more information, see “Output details—memmap command” (page 41)

**Description**

This command displays the memory map that is maintained by the EFI environment, which keeps track of all the physical memory in the system and how it is currently being used.
Example

To display the system memory map:

```bash
fs0:/> memmap
```

<table>
<thead>
<tr>
<th>Type</th>
<th>Start</th>
<th>End</th>
<th># Pages</th>
<th>Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>available</td>
<td>0000000000750000-0000000001841FFF</td>
<td>000000000000010F2</td>
<td>0000000000000009</td>
<td></td>
</tr>
<tr>
<td>LoaderCode</td>
<td>0000000001842000-00000000018A3FFF</td>
<td>0000000000000062</td>
<td>0000000000000009</td>
<td></td>
</tr>
<tr>
<td>available</td>
<td>00000000018A4000-00000000018C1FFF</td>
<td>000000000000001E</td>
<td>0000000000000009</td>
<td></td>
</tr>
<tr>
<td>LoaderData</td>
<td>00000000018C2000-00000000018CAFFF</td>
<td>0000000000000009</td>
<td>0000000000000009</td>
<td></td>
</tr>
<tr>
<td>BS_code</td>
<td>00000000018CB000-0000000001905FFF</td>
<td>000000000000003B</td>
<td>0000000000000009</td>
<td></td>
</tr>
<tr>
<td>BS_data</td>
<td>0000000001906000-00000000019C9FFF</td>
<td>00000000000000C4</td>
<td>0000000000000009</td>
<td></td>
</tr>
<tr>
<td>RT_data</td>
<td>0000000001B2B000-0000000001B2BFFF</td>
<td>0000000000000001</td>
<td>8000000000000009</td>
<td></td>
</tr>
<tr>
<td>reserved</td>
<td>0000000001B2C000-0000000001B4FFFF</td>
<td>0000000000000024</td>
<td>0000000000000009</td>
<td></td>
</tr>
<tr>
<td>available</td>
<td>0000000001B50000-0000000001DFFF</td>
<td>00000000000000200</td>
<td>0000000000000009</td>
<td></td>
</tr>
</tbody>
</table>

| 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

Table 11 (page 41) describes the possible output for this command.

**Table 11 Output details—memmap command**

<table>
<thead>
<tr>
<th>Column</th>
<th>Displays the...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>Type of memory. Options are:</td>
</tr>
<tr>
<td></td>
<td>• Available</td>
</tr>
<tr>
<td></td>
<td>• LoaderCode</td>
</tr>
<tr>
<td></td>
<td>• LoaderData</td>
</tr>
<tr>
<td></td>
<td>• BootServiceCode</td>
</tr>
<tr>
<td></td>
<td>• BootServiceData</td>
</tr>
<tr>
<td></td>
<td>• RuntimeCode</td>
</tr>
<tr>
<td></td>
<td>• RuntimeData</td>
</tr>
<tr>
<td></td>
<td>• Reserved</td>
</tr>
<tr>
<td></td>
<td>• MemoryMappedIO</td>
</tr>
<tr>
<td></td>
<td>• MemoryMappedIOPortSpace</td>
</tr>
<tr>
<td></td>
<td>• UnusableMemory</td>
</tr>
<tr>
<td></td>
<td>• ACPIReclaimMemory</td>
</tr>
<tr>
<td></td>
<td>• ACPIMemoryNVS</td>
</tr>
<tr>
<td></td>
<td>• PalCode</td>
</tr>
<tr>
<td>Start</td>
<td>Starting address</td>
</tr>
<tr>
<td>End</td>
<td>Ending address</td>
</tr>
<tr>
<td># Pages</td>
<td>Number of 4KB pages</td>
</tr>
<tr>
<td>reserved</td>
<td>Reserved memory total size in bytes</td>
</tr>
<tr>
<td>LoaderCode</td>
<td>Loader code total size in bytes</td>
</tr>
<tr>
<td>LoaderData</td>
<td>Loader data total size in bytes</td>
</tr>
<tr>
<td>BS_code</td>
<td>Boot service code total size in bytes</td>
</tr>
</tbody>
</table>
Table 11 Output details—`memmap` command (continued)

<table>
<thead>
<tr>
<th>Column</th>
<th>Displays the...</th>
</tr>
</thead>
<tbody>
<tr>
<td>BS_data</td>
<td>Boot service data total size in bytes</td>
</tr>
<tr>
<td>RT_data</td>
<td>Runtime data total size in bytes</td>
</tr>
<tr>
<td>available</td>
<td>Available memory in bytes</td>
</tr>
<tr>
<td>Total Memory</td>
<td>Total memory size in bytes</td>
</tr>
</tbody>
</table>

**mkdir**

Creates one or more new directories.

**Syntax**

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

**Option**

```
dir   Specifies one or more names for directories. Wild cards are not permitted.
```

**Description**

This command creates one or more new directories. If `dir` includes nested directories, then parent directories are created before child directories. If the directory already exists, then 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 temp2
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**

Displays or changes the console output device mode.

**Syntax**

```
mode [col row]
```
Options

**col**  Specifies the number of columns.

**row**  Specifies the number of rows.

Description

This command is used to change the display mode for the console output device. When used without any parameters, it 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**

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

Syntax

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

Options

**src...**  Specifies a source file or directory name. Wild cards are permitted.

**dst**  Specifies a destination file or directory name Wild cards 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.

Description

This command moves one or more files to a destination within a file system. Moving between file system volumes is not supported. If the destination is an existing directory, then 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
Displays the protocols and agents associated with a handle.

Syntax

```
openinfo Handle [-b]
```

Options

- `-b` Display one screen at a time.
- `Handle` Display open protocol information for specified handle.

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

Syntax

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

Options

- `filename` Specifies a source file name.
- `tablename` Specifies a table name to be parsed.
- `column` Specifies a one-based column index to use to determine which value from a particular record to parse.
- `-i instance` Starts parsing with the `nth` instance of the specified `tablename`, after the specified instance of `ShellCommand`. If not present, then all instances are returned.
- `-s instance` Starts parsing with the `nth` instance of the specified `ShellCommand` table. If not present, then 1 is assumed.

Description

This command enables the parsing of data from a file containing data output from a command having using the `-sfo` parameter. Since 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:
pause

Pauses the script file execution.

Syntax

pause [-q]

Description

This command is available only in scripts. It prints a message to the display, and then 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:

```bash
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:

```bash
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:

```bash
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

Displays a PCI device list or PCI function configuration space information.
Syntax
pci [Bus Dev [Func] [-s Seg] [-i]]

Options
- s Optional segment number Seg specified
mappedname Specifies a mapping name.
- i Information interpreted.
-Bus Bus number
Dev Device number
Func Function number

Description
This command displays all the PCI devices in the system. It can also display the configuration space of a PCI device according to the specified bus (Bus), device (Dev), and function (Func) addresses. If the function address is not specified, it defaults to 0.
The -i option is used to display 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 Device number parameters are specified while the Func or Seg parameters are not, Function or Seg are set as default value 0.

Example
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 -1

reset
Resets the system.

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

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.

Description
This command resets the system. The default is to perform a cold reset. If string is specified, then 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
Deletes one or more files or directories. The `del` command is an internal alias for `rm`.

**Syntax**

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

**Options**

- `-q` Deletes in quiet mode, without displaying a confirmation prompt.
- `file` Specifies the file name to be deleted. Wild cards are permitted.
- `directory` Specifies the directory to be deleted. Wild cards are permitted.

**Description**

This command deletes one or more files or directories. If the target is a directory, it deletes the directory, including all its subdirectories. It 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 wild cards, you are not prompted for confirmation.

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

**Examples**

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

```
fs0:\> ls test
Directory of: fs0:\test
06/18/01 01:01p <DIR>    512 .
06/18/01 01:01p <DIR>    0 ..
06/19/01 12:59a <DIR>    512 temp1
06/19/01 12:59a <DIR>    512 temp2
  0 File(s) 0 bytes
  4 Dir(s)

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

**To remove multiple directories with wild cards:**

```
fs0:\> rm test\*temp*
rm/del: Cannot find 'fs0:\test\temp1' - Not Found
```

```
fs0:\> ls test\temp1
Directory of: fs0:\test\temp1
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 temp11 temp2
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:**

```
rm/del: Remove subtree 'fs0:\test\temp1' [y/n]?
rm/del: Remove subtree 'fs0:\test\temp2' [y/n]?
```
set

Displays, changes, or deletes a UEFI Shell environment variable.

Syntax

set [-v] [sname [value]]
set [-d sname]

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.

Description

This command maintains the UEFI Shell environment variables. It can be used to:

- Display environment variables.
- Create new environment variables.
- Change the value of existing environment variables.
- Delete environment variables.

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

NOTE: 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

setsz

Sets the size of a file.

Syntax
setsz size [-d] file [file...]

Options
size The size of the file once it is adjusted.
-d Deletes a variable.
file The file that is adjusted in size.

Description
This command adjusts the size of a target file. When adjusting the size of a file, it 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:/> setsz size file [file...]

shift

Shifts the contents of a UEFI Shell script’s positional parameters, allowing scripts to process them from left to right.

Syntax
shift

Description
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.

NOTE: This command does not change the UEFI shell environment variable lasterror.
Example
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
Displays SMBIOS information.

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

Options
- `-t` Display all structures of SmbiosType.
- `-h` Display structures of SmbiosHandle.
- `-s` Display statistics table.
- `-a` Display all information.
- `SmbiosType` SMBIOS structure type.
- `SmbiosHandle` SMBIOS structure unique 16–bit handle.

Example
To set the size of a file:
```
fs0:\> setsize size file [file...]
```

stall
Stalls the processor for a specified number of microseconds.

Syntax
```
stall time
```

Option
- `time` Specifies the number of microseconds for the processor to stall.

Description
This command sets a timed stall of operations during a script.

Example
To stall the processor for 20 microseconds:
```
Shell> stall 20
```
sysconfig

Displays or configures HP system BIOS settings.

Syntax

sysconfig -i [ all | settingname ]
sysconfig -g [ all | settingname ] [settingname...]
sysconfig -s [settingname=settingvalue ...]
sysconfig -s AdminPassword=settingvalue OldAdminPassword=settingvalue
sysconfig -s PowerOnPassword=settingvalue OldPowerOnPassword=settingvalue
sysconfig [ -import | export ] filename.txt

Options

-i
  Shows information for the specified settings or all settings, including possible values. An asterisk (*) indicates the current value for that setting.

settingname
  Specifies a setting name about which to display information. See sysconfig attributes (page 52).

all
  Displays all information for all settings.

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

-s
  Sets the value of the specified setting.

settingvalue
  Specifies a setting value. See sysconfig attributes (page 52).

AdminPassword
  Specifies a new administrator password.

OldAdminPassword
  Specifies the administrator password to be reset.

PowerOnPassword
  Specifies a new password for powering on the server.

OldPowerOnPassword
  Specifies the power on password to be reset.

-import
  Imports all settings from a script file.

-export
  Exports all settings to a script file.

filename.txt
  Specifies the script file target for importing or exporting.

Description

This command displays or configures system BIOS settings.

NOTE:

- The BIOS settings in Service Option or Debug Option are not available for executing all in the Shell.

Examples

To view all possible sysconfig attributes existing in system and their current configured values:
Shell> sysconfig -g all
For a description of output values, see sysconfig attributes (page 52).

To show all 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 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"
```

**sysconfig attributes**

You can view all possible sysconfig attribute names, descriptions, current values, possible values, and Enum setting types as shown in the following example.

For more information about each attribute, see *HP UEFI System Utilities User Guide*.

```
Shell> sysconfig -i all
```

<table>
<thead>
<tr>
<th>Setting Name</th>
<th>Description</th>
<th>Current Value</th>
<th>Possible Values</th>
<th>Setting Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>EmbeddedSerialPort</td>
<td>[Embedded Serial Port]</td>
<td>Com1Irq4 (COM 1; IRQ4; I/O: 3F8h-3FFh)</td>
<td>Com1Irq4 (COM 1; IRQ4; I/O: 3F8h-3FFh), Com2Irq3 (COM 2; IRQ3; I/O: 2F8h-2FFh)</td>
<td>Enum</td>
</tr>
<tr>
<td>VirtualSerialPort</td>
<td>[Virtual Serial Port]</td>
<td>Com1Irq4 (COM 1; IRQ4; I/O: 3F8h-3FFh)</td>
<td>Com1Irq4 (COM 1; IRQ4; I/O: 3F8h-3FFh), Com2Irq3 (COM 2; IRQ3; I/O: 2F8h-2FFh)</td>
<td>Enum</td>
</tr>
<tr>
<td>NicBoot1</td>
<td>[Embedded FlexibleLOM 1 Port 1]</td>
<td>NetworkBoot (Network Boot)</td>
<td>NetworkBoot (Network Boot), Disabled (Disabled)</td>
<td>Enum</td>
</tr>
<tr>
<td>NicBoot2</td>
<td>[Embedded FlexibleLOM 1 Port 2]</td>
<td>Disabled (Disabled)</td>
<td>Disabled (Disabled)</td>
<td>Enum</td>
</tr>
<tr>
<td>NicBoot3</td>
<td>[Embedded FlexibleLOM 1 Port 3]</td>
<td>NetworkBoot (Network Boot)</td>
<td>NetworkBoot (Network Boot), Disabled (Disabled)</td>
<td>Enum</td>
</tr>
<tr>
<td>NicBoot4</td>
<td>[Embedded FlexibleLOM 1 Port 4]</td>
<td>Disabled (Disabled)</td>
<td>Disabled (Disabled)</td>
<td>Enum</td>
</tr>
<tr>
<td>AdvancedMemProtection</td>
<td>[Advanced Memory Protection]</td>
<td>AdvancedEcc (Advanced ECC Support)</td>
<td>AdvancedEcc (Advanced ECC Support), OnlineSpareAdvancedEcc (Online Spare with Advanced ECC Support), MirroredAdvancedEcc (Mirrored Memory with Advanced ECC Support), LockstepDddc (Lockstep Mode with DDDC Support)</td>
<td>Enum</td>
</tr>
<tr>
<td>UsbControl</td>
<td>[USB Control]</td>
<td>UsbEnabled (USB Enabled)</td>
<td>UsbEnabled (USB Enabled), ExternalUsbDisabled (External USB Ports Disabled)</td>
<td>Enum</td>
</tr>
<tr>
<td>UsbBoot</td>
<td>[USB Boot Support]</td>
<td>Enabled (Enabled)</td>
<td>Enabled (Enabled), Disabled (Disabled)</td>
<td>Enum</td>
</tr>
<tr>
<td>RemovableFlashBootSeq</td>
<td>[Removable Flash Media Boot Sequence]</td>
<td>ExternalKeysFirst (External DriveKeys First)</td>
<td>ExternalKeysFirst (External DriveKeys First), InternalSdCardFirst (Internal SD Card First), ExternalSdCardFirst (External DriveKeys First)</td>
<td>Enum</td>
</tr>
<tr>
<td>ProcNoExecute</td>
<td>[No-Execute Protection]</td>
<td>Enabled (Enabled)</td>
<td>Enabled (Enabled), Disabled (Disabled)</td>
<td>Enum</td>
</tr>
<tr>
<td>ProcVirtualization</td>
<td>[Virtualization Technology]</td>
<td>Enabled (Enabled)</td>
<td>Enabled (Enabled)</td>
<td>Enum</td>
</tr>
</tbody>
</table>
### Setting Name: ProcHyperthreading
- **Current Value:** Enabled
- **Possible Values:** Enabled, Disabled
- **Setting Type:** Enum

### Setting Name: ProcTurbo
- **Current Value:** Enabled
- **Possible Values:** Enabled, Disabled
- **Setting Type:** Enum

### Setting Name: IntelProcVtd
- **Current Value:** Enabled
- **Possible Values:** Enabled, Disabled
- **Setting Type:** Enum

### Setting Name: EmbeddedUefiShell
- **Current Value:** Enabled
- **Possible Values:** Enabled, Disabled
- **Setting Type:** Enum

### Setting Name: UefiShellBootOrder
- **Current Value:** Disabled
- **Possible Values:** Enabled, Disabled
- **Setting Type:** Enum

### Setting Name: UefiShellStartup
- **Current Value:** Disabled
- **Possible Values:** Enabled, Disabled
- **Setting Type:** Enum

### Setting Name: PowerProfile
- **Current Value:** BalancedPowerPerf
- **Possible Values:** BalancedPowerPerf, MinPower, MaxPerf, Custom
- **Setting Type:** Enum

### Setting Name: PowerRegulator
- **Current Value:** DynamicPowerSavings
- **Possible Values:** DynamicPowerSavings, StaticLowPower, StaticHighPerf, OsControl
- **Setting Type:** Enum

### Setting Name: RedundantPowerSupply
- **Current Value:** BalancedMode
- **Possible Values:** BalancedMode, HighEfficiencyAuto, HighEfficiencyOddStandby, HighEfficiencyEvenStandby
- **Setting Type:** Enum

### Setting Name: IntelQpiPowerManagement
- **Current Value:** Enabled
- **Possible Values:** Enabled, Disabled
- **Setting Type:** Enum

### Setting Name: IntelQpiFreq
- **Current Value:** Auto
- **Possible Values:** Auto, MinQpiSpeed
- **Setting Type:** Enum

### Setting Name: QpiBandwidthOpt
- **Current Value:** Balanced
- **Possible Values:** Balanced, OptimizedForI0
- **Setting Type:** Enum

### Setting Name: MinProcIdlePower
- **Current Value:** C6
- **Possible Values:** C6, C3, C1E, NoCStates
- **Setting Type:** Enum

### Setting Name: MinProcIdlePkgState
- **Current Value:** C6NonRetention
- **Possible Values:** C6Retention
- **Setting Type:** Enum
C6 Non Retention
No State

Setting Type = Enum
Setting Name = Energy Perf Bias
Current Value = Balanced Perf
Possible Values = Balanced Perf, Max Perf, Balanced Power

Setting Type = Enum
Setting Name = Max Mem Bus Freq MHz
Current Value = Auto
Possible Values = Auto, 1600 MHz, 1333 MHz, 1066 MHz

Setting Type = Enum
Setting Name = Channel Interleaving
Current Value = Enabled
Possible Values = Enabled, Disabled

Setting Type = Enum
Setting Name = Max PCIe Speed
Current Value = Max Supported
Possible Values = Max Supported, PCIe Gen 1

Setting Type = Enum
Setting Name = Dynamic Power Response
Current Value = Fast
Possible Values = Fast, Slow

Setting Type = Enum
Setting Name = Collaborative Power Control
Current Value = Enabled
Possible Values = Enabled, Disabled

Setting Type = Enum
Setting Name = ACPI SLIT
Current Value = Enabled
Possible Values = Enabled, Disabled

Setting Type = Enum
Setting Name = Embedded SAS Enable
Current Value = Enabled
Possible Values = Enabled, Disabled

Setting Type = Enum
Setting Name = Flex LOM 1 Enable
Current Value = Enabled
Possible Values = Enabled, Disabled

Setting Type = Enum
Setting Name = ASR Status
Current Value = Enabled
Possible Values = Enabled, Disabled

Setting Type = Enum
Setting Name = ASR Timeout Minutes
Current Value = 10
Possible Values = 10, 15, 20, 30, 5

Setting Type = Enum
Setting Name = Thermal Shutdown
Current Value = Enabled
Possible Values = Enabled, Disabled

Setting Type = Enum
Setting Name = Wake On Lan
Current Value = Enabled
Possible Values = Enabled, Disabled

Setting Type = Enum
Setting Name = POST F1 Prompt
Current Value = Delayed 20 Sec
Possible Values = Delayed 20 Sec, Delayed 2 Sec
<table>
<thead>
<tr>
<th>Setting Name</th>
<th>Current Value</th>
<th>Possible Values</th>
<th>Setting Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Delayed20Sec</td>
<td></td>
<td>[Delayed 20 seconds]</td>
<td>Enum</td>
</tr>
<tr>
<td>PowerButton</td>
<td>Enabled</td>
<td>[Enabled]</td>
<td>Enum</td>
</tr>
<tr>
<td>PowerOnDelay</td>
<td>None</td>
<td>[No Delay]</td>
<td>Enum</td>
</tr>
<tr>
<td>IntelligentProvisioning</td>
<td>Enabled</td>
<td>[Enabled]</td>
<td>Enum</td>
</tr>
<tr>
<td>TpmState</td>
<td>NotPresent</td>
<td>[Not Present]</td>
<td>Enum</td>
</tr>
<tr>
<td>TpmOperation</td>
<td>Disable</td>
<td>[Disable]</td>
<td>Enum</td>
</tr>
<tr>
<td>SerialConsolePort</td>
<td>Auto</td>
<td>[Auto]</td>
<td>Enum</td>
</tr>
<tr>
<td>SerialConsoleEmulation</td>
<td>Vt100Plus</td>
<td>[VT100+]</td>
<td>Enum</td>
</tr>
<tr>
<td>SerialConsoleBaudRate</td>
<td>115200</td>
<td>[115200]</td>
<td>Enum</td>
</tr>
<tr>
<td>EmsConsole</td>
<td>Disabled</td>
<td>[Disabled]</td>
<td>Enum</td>
</tr>
<tr>
<td>RomSelection</td>
<td>CurrentRom</td>
<td>[Use Current ROM]</td>
<td>Enum</td>
</tr>
<tr>
<td>NmiDebugButton</td>
<td>Enabled</td>
<td>[Enabled]</td>
<td>Enum</td>
</tr>
<tr>
<td>Setting Name</td>
<td>Type</td>
<td>Possible Values</td>
<td>Current Value</td>
</tr>
<tr>
<td>----------------------</td>
<td>--------</td>
<td>--------------------------</td>
<td>---------------</td>
</tr>
<tr>
<td>VirtualInstallDisk</td>
<td>Enum</td>
<td>[Disabled]</td>
<td>[Enabled]</td>
</tr>
<tr>
<td>PciBusPadding</td>
<td>Enum</td>
<td>[Disabled]</td>
<td>[Enabled]</td>
</tr>
<tr>
<td>PowerOnLogo</td>
<td>Enum</td>
<td>[Disabled]</td>
<td>[Enabled]</td>
</tr>
<tr>
<td>F11BootMenu</td>
<td>Enum</td>
<td>[Disabled]</td>
<td>[Enabled]</td>
</tr>
<tr>
<td>Sriov</td>
<td>Enum</td>
<td>[Disabled]</td>
<td>[Enabled]</td>
</tr>
<tr>
<td>ConsistentDevNaming</td>
<td>Enum</td>
<td>[Disabled]</td>
<td>LomsOnly</td>
</tr>
<tr>
<td>NetworkBootRetry</td>
<td>Enum</td>
<td>[Disabled]</td>
<td>[Enabled]</td>
</tr>
<tr>
<td>BootOrderPolicy</td>
<td>Enum</td>
<td>[Disabled]</td>
<td>RetryIndefinitely</td>
</tr>
<tr>
<td>VideoOptions</td>
<td>Enum</td>
<td>[Disabled]</td>
<td>BothVideoEnabled</td>
</tr>
<tr>
<td>UefiPxeBoot</td>
<td>Enum</td>
<td>[Disabled]</td>
<td>Auto</td>
</tr>
<tr>
<td>PwrSupplyReqOverride</td>
<td>Enum</td>
<td>[Disabled]</td>
<td>Default</td>
</tr>
<tr>
<td>HwPrefetcher</td>
<td>Enum</td>
<td>[Disabled]</td>
<td>[Enabled]</td>
</tr>
<tr>
<td>AdjSecPrefetch</td>
<td>Enum</td>
<td>[Disabled]</td>
<td>[Enabled]</td>
</tr>
</tbody>
</table>

56 UEFI Shell command reference
Current Value = Disabled   [Disabled]
Possible Values = Enabled     [Enabled]
Setting Type = Enum

Setting Name = AcpiRtcSupport [ACPI RTC Support]
Current Value = Disabled     [Disabled]
Possible Values = Enabled     [Enabled]
Setting Type = Enum

Setting Name = ServerName [Server Name]
Current Value =
Setting Type = String

Setting Name = ServerPrimaryOs [Server Primary OS]
Current Value =
Setting Type = String

Setting Name = ServerOtherInfo [Server Other Information]
Current Value =
Setting Type = String

Setting Name = AdminName [Administrator Name]
Current Value =
Setting Type = String

Setting Name = AdminPhone [Administrator Phone Number]
Current Value =
Setting Type = String

Setting Name = AdminEmail [Administrator E-mail Address]
Current Value =
Setting Type = String

Setting Name = AdminOtherInfo [Administrator Other Information]
Current Value =
Setting Type = String

Setting Name = ServiceName [Service Contact Name]
Current Value =
Setting Type = String

Setting Name = ServicePhone [Service Contact Phone Number]
Current Value =
Setting Type = String

Setting Name = ServiceEmail [Service Contact E-mail Address]
Current Value =
Setting Type = String

Setting Name = ServiceOtherInfo [Service Contact Other Information]
Current Value =
Setting Type = String

Setting Name = CustomPostMessage [Custom POST Message]
Current Value =
Setting Type = String

Setting Name = ProcCoreDisable [Processor Core Disable]
Current Value = 0
Setting Type = Numeric

Setting Name = ProductId [Product ID]
Current Value =
Setting Type = String

Setting Name = ServerAssetTag [Server Asset Tag]
Current Value =
Setting Type = String

Setting Name = SerialNumber [Serial Number]
Current Value =
Setting Type = String

Setting Name = AssetTagProtection [Asset Tag Protection]
Current Value = Unlocked [Unlocked]
Possible Values = Locked [Locked]
Setting Type = Enum

Setting Name = SecureBoot [Secure Boot]
Current Value = Disabled [Disabled]
Possible Values = Enabled [Enabled]
Setting Type = Enum

Setting Name = TimeZone [Time Zone]
Current Value = Utc0 [UTC-00:00, Greenwich Mean Time, Dublin, London]
Possible Values = UtcM12 [UTC-12:00, International Date Line West]
UtcM11 [UTC-11:00, Midway Island, Samoa]
UtcM10 [UTC-10:00, Hawaii]
UtcM9 [UTC-09:00, Alaska]
time

Displays or sets the current time for the system.

Syntax

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

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 can be between -1440 and 1440 or 2047. If 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, then the current daylight savings time is displayed.

Description

This command displays or sets the current time for the system. If no parameters are used, it shows the current time. If valid hours, minutes, and seconds are provided, then the system's time is updated.
NOTE:

- 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.
- The seconds parameter is optional. If not specified, seconds are set to zero by default.

Examples

To display the current system time:

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

To set the current system time:

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

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

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

timezone

Displays or sets time zone information.

Syntax

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

Options

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

Description

This command displays or sets the current timezone for the system. If no parameters are used, it shows the current timezone. If a valid hh:mm parameter is provided, then the system’s timezone information in updated.

Examples

To display all available timezones:

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

To set the timezone:

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

To display detailed information for the current timezone:

```bash
fs0:\> timezone -f
```
**touch**

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

**Syntax**

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

**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.

**Description**

This command updates the time and date on the file specified by the `file` parameter to the current time and date.

If multiple files are specified, the system continues processing. It 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**

Sends the contents of a file to the standard output device.

**Syntax**

type file[file...]

**Option**

- `file` Specifies a file name to display.

**Description**

This command sends the contents of a file to the standard output device. If no options are used, then the command attempts to detect the file type. If it fails, then UCS-2 is presumed.

**Examples**

To display a file in format:

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

To display multiple files:

```
fs0:\> type test.*
How to Install?
time
```
ver

Displays version information for the UEFI Shell and the underlying UEFI firmware.

Syntax

ver [ -s | -t ]

Options

- s Displays only the UEFI Shell version.
- t Displays summary ( terse ) content.

Description

This command displays the version information for the UEFI firmware or the version information for the UEFI Shell itself. The information is retrieved through 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
UEFI Interactive Shell v2.0
Copyright 1982, 2014 Hewlett-Packard Development Company, L.P.
UEFI v2.31 (HP, 0x00010000)
ProLiant System BIOS - P79 (01/14/2014)

vol

Displays volume information for a file system.

Syntax

vol [ fs ] [- n volumelabel]
vol [ fs ] [- d ]

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.

Description

This command displays the volume information for a file system. If fs is not specified, the current file system is assumed. If - n is specified, then 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
3 Running and editing UEFI Shell scripts

The following information describes the scripting functions provided by the UEFI Shell.

UEFI Shell script commands

Examples

To export all settings to a file:
```
fs0:\> Sysconfig –export filename.txt
```
To import all settings to a file:
```
fs0:\> Sysconfig –import filename.txt
```

Invoking scripts

You can invoke UEFI Shell scripts using either of these two methods:

- “Enabling the UEFI Shell Script Auto Start” (page 64)
- “Manually invoking a Shell script” (page 64)

Enabling the UEFI Shell Script Auto Start

You can enable or disable automatic execution of the default UEFI Shell script during shell start up. When enabled, the shell looks for the `startup.nsh` file in any of the FAT16 or FAT32 file systems available. HP recommends having only one `startup.nsh` file on one file system.

To enable the UEFI Shell script auto start:
1. Access the System Utilities menu. For more information, see “Accessing the HP UEFI System Utilities” (page 5).
2. From the System Utilities screen, select System Configuration → BIOS/Platform Configuration (RBSU) → System Options → UEFI Shell Options → UEFI Shell Script Auto-Start and press Enter.
3. Press F10 to save your selection.
4. Reboot the server for the change to take effect.

Manually invoking a Shell script

To manually invoke 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.

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 Shell scripts

Examples

The following sample scripts show how to capture a configuration (export) and replicate it (import) from `startup.nsh` mounted on an iLO virtual media.

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 in (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 "%cwd%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 in (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
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 Table 12 (page 67). 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 12 UEFI Application APIs

<table>
<thead>
<tr>
<th>Function Type</th>
<th>Function Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EFI_SHELL_EXECUTE</td>
<td>Execute</td>
<td>Causes the shell to parse and execute the command line.</td>
</tr>
<tr>
<td>EFI_SHELL_GET_ENV</td>
<td>GetEnv</td>
<td>Gets the environment variable.</td>
</tr>
<tr>
<td>EFI_SHELL_SET_ENV</td>
<td>SetEnv</td>
<td>Changes a specific environment variable.</td>
</tr>
<tr>
<td>EFI_SHELL_GET_ALIAS</td>
<td>GetAlias</td>
<td>Retrieves the alias for a specific shell command.</td>
</tr>
<tr>
<td>EFI_SHELL_SET_ALIAS</td>
<td>SetAlias</td>
<td>Adds or removes the alias for a specific shell command.</td>
</tr>
<tr>
<td>EFI_SHELL_GET_HELP_TEXT</td>
<td>GetHelpText</td>
<td>Return help information about a specific command.</td>
</tr>
<tr>
<td>EFI_SHELL_GET_DEVICE_PATH_FROM_MAP</td>
<td>GetDevicePathFromMap</td>
<td>Returns the device path that corresponds to a mapping.</td>
</tr>
<tr>
<td>EFI_SHELL_GET_MAP_FROM_DEVICE_PATH</td>
<td>GetMapFromDevicePath</td>
<td>Returns the mapping that corresponds to a particular device path.</td>
</tr>
<tr>
<td>EFI_SHELL_GET_DEVICE_PATH_FROM_FILE_PATH</td>
<td>GetDevicePathFromFilePath</td>
<td>Converts a file path to a device path, where all mappings have been replaced with the corresponding device paths.</td>
</tr>
<tr>
<td>EFI_SHELL_GET_FILE_PATH_FROM_DEVICE_PATH</td>
<td>GetFilePathFromDevicePath</td>
<td>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.</td>
</tr>
<tr>
<td>EFI_SHELL_SET_MAP</td>
<td>SetMap</td>
<td>It creates/updates/deletes a mapping between a device and a device path.</td>
</tr>
<tr>
<td>EFI_SHELL_GET_CUR_DIR</td>
<td>GetCurDir</td>
<td>Returns the current directory on a device.</td>
</tr>
<tr>
<td>EFI_SHELL_SET_CUR_DIR</td>
<td>SetCurDir</td>
<td>Changes the current directory on a device.</td>
</tr>
<tr>
<td>EFI_SHELL_OPEN_FILE_LIST</td>
<td>OpenFileList</td>
<td>Opens the files that match the path pattern specified.</td>
</tr>
<tr>
<td>EFI_SHELL_FREE_FILE_LIST</td>
<td>FreeFileList</td>
<td>Frees the file list that created by OpenFileList().</td>
</tr>
<tr>
<td>EFI_SHELL_REMOVE_DUP_IN_FILE_LIST</td>
<td>RemoveDupInFileList</td>
<td>Deletes the duplicate files in the given file list.</td>
</tr>
<tr>
<td>EFI_SHELL_BATCH_IS_ACTIVE</td>
<td>BatchIsActive</td>
<td>Returns whether any script files are currently being processed.</td>
</tr>
<tr>
<td>EFI_SHELL_IS_ROOT_SHELL</td>
<td>IsRootShell</td>
<td>Judges whether the active Shell is the root shell.</td>
</tr>
<tr>
<td>EFI_SHELL_ENABLE_PAGE_BREAK</td>
<td>EnablePageBreak</td>
<td>Enables the page break output mode.</td>
</tr>
<tr>
<td>EFI_SHELL_DISABLE_PAGE_BREAK</td>
<td>DisablePageBreak</td>
<td>Disables the page break output mode.</td>
</tr>
<tr>
<td>Function Type</td>
<td>Function Name</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>---------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>EFI_SHELL_GET_PAGE_BREAK</td>
<td>GetPageBreak</td>
<td>Gets the enable status of the page break output mode.</td>
</tr>
<tr>
<td>EFI_SHELL_GET_DEVICE_NAME</td>
<td>GetDeviceName</td>
<td>Gets the name of the device specified by the device handle.</td>
</tr>
<tr>
<td>EFI_SHELL_GET_FILE_INFO</td>
<td>GetFileInfo</td>
<td>Return information about a specific file handle.</td>
</tr>
<tr>
<td>EFI_SHELL_SET_FILE_INFO</td>
<td>SetFileInfo</td>
<td>Change information about a specific file handle.</td>
</tr>
<tr>
<td>EFI_SHELL_OPEN_FILE_BY_NAME</td>
<td>OpenFileByName</td>
<td>Given a file name, open a file and return a file handle.</td>
</tr>
<tr>
<td>EFI_SHELL_CLOSE_FILE</td>
<td>CloseFile</td>
<td>Close an open file.</td>
</tr>
<tr>
<td>EFI_SHELL_CREATE_FILE</td>
<td>CreateFile</td>
<td>Create a new file.</td>
</tr>
<tr>
<td>EFI_SHELL_READ_FILE</td>
<td>ReadFile</td>
<td>Read data from a file.</td>
</tr>
<tr>
<td>EFI_SHELL_WRITE_FILE</td>
<td>WriteFile</td>
<td>Write data to a file.</td>
</tr>
<tr>
<td>EFI_SHELL_DELETE_FILE</td>
<td>DeleteFile</td>
<td>Delete a file.</td>
</tr>
<tr>
<td>EFI_SHELL_DELETE_FILE_BY_NAME</td>
<td>DeleteFileByName</td>
<td>Delete a file by name.</td>
</tr>
<tr>
<td>EFI_SHELL_GET_FILE_POSITION</td>
<td>GetFilePosition</td>
<td>Return the current read/write position within a file.</td>
</tr>
<tr>
<td>EFI_SHELL_SET_FILE_POSITION</td>
<td>SetFilePosition</td>
<td>Change the current read/write position within a file.</td>
</tr>
<tr>
<td>EFI_SHELL_FLUSH_FILE</td>
<td>FlushFile</td>
<td>Write all buffered data to a file.</td>
</tr>
<tr>
<td>EFI_SHELL_FIND_FILES</td>
<td>FindFiles</td>
<td>Return all files that match a pattern in a file list.</td>
</tr>
<tr>
<td>EFI_SHELL_FIND_FILES_IN_DIR</td>
<td>FindFilesInDir</td>
<td>Return all files in a specified directory in a file list.</td>
</tr>
<tr>
<td>EFI_SHELL_GET_FILE_SIZE</td>
<td>GetFileSize</td>
<td>Return the size of a file.</td>
</tr>
<tr>
<td>EFI_SHELL_OPEN_ROOT</td>
<td>OpenRoot</td>
<td>Return the root directory of a file system.</td>
</tr>
<tr>
<td>EFI_SHELL_OPEN_ROOT_BY_HANDLE</td>
<td>OpenRootByHandle</td>
<td>Return the root directory of a file system on a particular handle.</td>
</tr>
<tr>
<td>EFI_EVENT</td>
<td>ExecutionBreak</td>
<td>Event signaled by the UEFI Shell when the user presses CTRL-C to indicate that the current UEFI Shell command execution should be interrupted.</td>
</tr>
<tr>
<td>UINT32</td>
<td>MajorVersion</td>
<td>The major version of the shell environment.</td>
</tr>
<tr>
<td>UINT32</td>
<td>MinorVersion</td>
<td>The minor version of the shell environment.</td>
</tr>
</tbody>
</table>

This sample script shows some protocol functions.

**Sample Shell scripts**

**Examples**

The following sample scripts how to capture a configuration (export) and replicate it (import) from startup.nsh mounted on an iLO virtual media.
Example

/** @file
 *  This is an example Shell Application. Note that there are
 *  other ways to add a command to the shell. This example
demonstrates just one.

 *  Copyright (c) 2013 Hewlett-Packard Development Company, L.P.

 *  This software contains information confidential and proprietary to
Hewlett-Packard Company. It shall not be reproduced in whole or in part,
or transferred to other documents, or disclosed to third parties, or used
for any purpose other than that for which it was obtained without the prior
written consent of Hewlett-Packard Company.
**/

#include <Protocol/EFIShell.h>
#include <Library/UefiLib.h>
#include <Library/UefiBootServicesTableLib.h>

/**
 *  Implements simple shell HelloWorld Application that prints out
 *  the string "Hello World" to the screen and then display UEFI
 *  Shell revision and registered environment variables in system.
 *  This example demonstrates how a new UEFI Shell Application can
 *  be created to run at UEFI Shell Prompt.

 *  @param ImageHandle Handle to the Image
 *  @param SystemTable Pointer to the System Table

 *  @retval EFI_SUCCESS In all cases

 **/
 EFI_STATUS
 EFIAPI
 UefiMain (  
 IN EFI_HANDLE ImageHandle,  
 IN EFI_SYSTEM_TABLE *SystemTable
 ) {  
  EFI_SHELL_PROTOCOL *UefiShellProtocol;  
  CONST CHAR16 *ConstEnvNameList;  
  CONST CHAR16 *Value;  
  EFI_STATUS Status;
  Print (L"Hello World!\n");
  //
  // Locate UEFI Shell Protocol
  //
  Status = gBS->OpenProtocol (  
  ImageHandle,  
  &gEfiShellProtocolGuid,  
  (VOID **) &UefiShellProtocol,  
  NULL,  
  EFI_OPEN_PROTOCOL_GET_PROTOCOL  
  );
  if (EFI_ERROR(Status)) {  
    // Search for the shell protocol  
    Status = gBS->LocateProtocol (  
    &gEfiShellProtocolGuid,  
    NULL,  
    (VOID **) &UefiShellProtocol  
    );
    if (EFI_ERROR(Status)) {  
      UefiShellProtocol = NULL;
      return EFI_SUCCESS;
    }
  }
  if (UefiShellProtocol != NULL) {  
    // Sample to read UEFI Shell Major and Minor Version Variables  
    //
    Print (L"UEFI Shell Revision: %d.%d\n", UefiShellProtocol->MajorVersion, UefiShellProtocol->MinorVersion);
    //
    // Sample to get list of all environment variables  
    //
    ConstEnvNameList = UefiShellProtocol->GetEnv (NULL);
    if (ConstEnvNameList == NULL) {
      return EFI_SUCCESS;
    }
    Print (L"\n");
  }
Print (L"ENVIRONMENT VARIABLES:\n");
Print (L"\n");

// Sample to get and display all environment variables
while (*ConstEnvNameList != CHAR_NULL){
    Value = UefiShellProtocol->GetEnv (ConstEnvNameList);
    Print (L"%s = %s\n", ConstEnvNameList, Value);
    ConstEnvNameList += StrLen (ConstEnvNameList)+1;
}

return EFI_SUCCESS;
Table 13 (page 71) lists the possible status codes displayed by the UEFI Shell when you issue a command. Codes vary depending on the command.

**Table 13 UEFI Shell command status codes**

<table>
<thead>
<tr>
<th>Code</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>SHELL_SUCCESS</td>
<td>The action is completed as requested.</td>
</tr>
<tr>
<td>SHELL_NOT_FOUND</td>
<td>The target file or set of files cannot be found.</td>
</tr>
<tr>
<td>SHELL_SECURITY_VIOLATION</td>
<td>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.</td>
</tr>
<tr>
<td>SHELL_INVALID_PARAMETER</td>
<td>One of the passed in parameters is incorrectly formatted or its value is out of bounds.</td>
</tr>
<tr>
<td>SHELL_OUT_OF_RESOURCES</td>
<td>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.</td>
</tr>
<tr>
<td>SHELL_WRITE_PROTECTED</td>
<td>The media on which the action is to take place is write-protected.</td>
</tr>
<tr>
<td>SHELL_DEVICE_ERROR</td>
<td>There is a hardware error preventing the completion of this command.</td>
</tr>
</tbody>
</table>
6 Support and other resources

Contacting HP

For worldwide technical support information, see the HP Support Center:
http://www.hp.com/go/hpsc

Before contacting HP, collect the following information:
- Product model names and numbers
- Technical support registration number (if applicable)
- Product serial numbers
- Error messages
- Operating system type and revision level
- Detailed questions

Subscription service

Receive, by email, support alerts announcing product support communications, driver updates, software releases, firmware updates, and customer-replaceable component information by signing up at http://www.hp.com/go/myadvisory.

To change options for support alerts you already receive, click the Sign in link on the right.

Related information

The following documents provide related information:
- HP UEFI System Utilities User Guide
- HP UEFI Release Notes

You can find these documents at the following website:
http://www.hp.com/go/ProLiantUEFI/docs

Websites

- 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

Typographic conventions

<table>
<thead>
<tr>
<th>Table 14 Document conventions</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Convention</th>
<th>Element</th>
</tr>
</thead>
<tbody>
<tr>
<td>Blue text: Table 14 (page 72)</td>
<td>- Cross-reference links and e-mail addresses</td>
</tr>
<tr>
<td></td>
<td>- A cross reference to the glossary definition of the term in blue text</td>
</tr>
<tr>
<td>Blue, bold, underlined text</td>
<td>email addresses</td>
</tr>
<tr>
<td>Blue, underlined text: <a href="http://www.hp.com">http://www.hp.com</a></td>
<td>Website addresses</td>
</tr>
</tbody>
</table>
Table 14 Document conventions (continued)

<table>
<thead>
<tr>
<th>Convention</th>
<th>Element</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bold text</td>
<td>• Keys that are pressed</td>
</tr>
<tr>
<td></td>
<td>• Text typed into a GUI element, such as a box</td>
</tr>
<tr>
<td></td>
<td>• GUI elements that are clicked or selected, such as menu and list items, buttons, tabs, and check boxes</td>
</tr>
<tr>
<td>Italic text</td>
<td>Text emphasis</td>
</tr>
<tr>
<td>Monospace text</td>
<td>• File and directory names</td>
</tr>
<tr>
<td></td>
<td>• System output</td>
</tr>
<tr>
<td></td>
<td>• Code</td>
</tr>
<tr>
<td></td>
<td>• Commands, their arguments, and argument values</td>
</tr>
<tr>
<td>Monospace, italic text</td>
<td>• Code variables</td>
</tr>
<tr>
<td></td>
<td>• Command variables</td>
</tr>
<tr>
<td>Monospace, bold text</td>
<td>Emphasized monospace text</td>
</tr>
</tbody>
</table>

NOTE: Provides additional information.

HP Insight Remote Support software

HP strongly recommends that you register your device for remote support to enable enhanced delivery of your HP Warranty, HP Care Pack Service, or HP contractual support agreement. HP Insight Remote Support supplements your monitoring continuously to ensure maximum system availability by providing intelligent event diagnosis, and automatic, secure submission of hardware event notifications to HP, which will initiate a fast and accurate resolution, based on your product’s service level. Notifications may be sent to your authorized HP Channel Partner for onsite service, if configured and available in your country.

For more information, see HP Insight Remote Support and Insight Online Setup Guide for ProLiant Gen8 Servers and BladeSystem c-Class Enclosures on the HP website (http://www.hp.com/go/enterprise/docs). HP Insight Remote Support is available as part of HP Warranty, HP Care Pack Service, or HP contractual support agreement.

HP Insight Online

HP Insight Online is a capability of the HP Support Center portal. Combined with HP Insight Remote central connect or HP Insight Online direct connect, it automatically aggregates device health, asset, and support information with contract and warranty information, and then secures it in a single, personalized dashboard that is viewable from anywhere at any time. The dashboard organizes your IT and service data to help you understand and respond to that information more quickly. With specific authorization from you, an authorized HP Channel Partner can also view your IT environment remotely by using HP Insight Online.

For more information, see the following documents on the HP website (http://www.hp.com/go/insightremotesupport/docs):

- HP Insight Online User’s Guide
- HP Insight Remote Support and Insight Online Setup Guide for ProLiant Servers and BladeSystem c-Class Enclosures
7 Documentation feedback

HP 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@hp.com). Include the document title and part number, version number, or the URL when submitting your feedback.
<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACR</td>
<td>Array Configuration Replication Utility</td>
</tr>
<tr>
<td>ASR</td>
<td>Automatic Server Recovery</td>
</tr>
<tr>
<td>BIOS</td>
<td>Basic Input/Output System</td>
</tr>
<tr>
<td>CLI</td>
<td>Command Line Interface</td>
</tr>
<tr>
<td>CNA</td>
<td>Converged Network Adapter</td>
</tr>
<tr>
<td>CONREP</td>
<td>Configuration Replication utility</td>
</tr>
<tr>
<td>ECC</td>
<td>Error Checking and Correcting</td>
</tr>
<tr>
<td>ECP</td>
<td>Extended Capabilities Port Mode</td>
</tr>
<tr>
<td>EMS</td>
<td>Emergency Management Services</td>
</tr>
<tr>
<td>EPP</td>
<td>Enhanced Parallel Port Mode</td>
</tr>
<tr>
<td>IDE</td>
<td>Integrated Device Electronics</td>
</tr>
<tr>
<td>iLO</td>
<td>Integrated Lights-Out</td>
</tr>
<tr>
<td>IMD</td>
<td>Integrated Management Display</td>
</tr>
<tr>
<td>IOMMU</td>
<td>I/O Memory Management Unit</td>
</tr>
<tr>
<td>IPL</td>
<td>Initial Program Load</td>
</tr>
<tr>
<td>IRQ</td>
<td>interrupt Request</td>
</tr>
<tr>
<td>LPT</td>
<td>Local Port</td>
</tr>
<tr>
<td>MEMBIST</td>
<td>Memory Built-in Self Test</td>
</tr>
<tr>
<td>MPS</td>
<td>Multi-Processor Specification</td>
</tr>
<tr>
<td>NIC</td>
<td>Network Interface Controller</td>
</tr>
<tr>
<td>NMI</td>
<td>Non-Maskable Interrupt</td>
</tr>
<tr>
<td>NUMA</td>
<td>Non-Uniform Memory Architecture</td>
</tr>
<tr>
<td>NVRAM</td>
<td>Non-Volatile Memory</td>
</tr>
<tr>
<td>ORCA</td>
<td>Option ROM Configuration for Arrays</td>
</tr>
<tr>
<td>PCC</td>
<td>Processor Clocking Control</td>
</tr>
<tr>
<td>PCI</td>
<td>Peripheral Component Interface</td>
</tr>
<tr>
<td>PCI-X</td>
<td>Peripheral Component Interconnect Extended</td>
</tr>
<tr>
<td>PCIe</td>
<td>Peripheral Component Interconnect Express</td>
</tr>
<tr>
<td>POST</td>
<td>Power-On Self-Test</td>
</tr>
<tr>
<td>PXE</td>
<td>Pre-Boot Execution Environment</td>
</tr>
<tr>
<td>RAID</td>
<td>Redundant Array of Inexpensive (or Independent) Disks</td>
</tr>
<tr>
<td>RBSU</td>
<td>ROM-Based Setup Utility</td>
</tr>
<tr>
<td>ROM</td>
<td>Read-Only Memory</td>
</tr>
<tr>
<td>RTC</td>
<td>Real-Time Clock</td>
</tr>
<tr>
<td>SPP</td>
<td>Standard Parallel Port Mode</td>
</tr>
<tr>
<td>TPM</td>
<td>Trusted Platform Module</td>
</tr>
<tr>
<td>UEFI</td>
<td>Unified Extensible Firmware Interface</td>
</tr>
<tr>
<td>VGA</td>
<td>Video Graphics Array</td>
</tr>
<tr>
<td>VSP</td>
<td>Virtual Serial Port</td>
</tr>
</tbody>
</table>
Index

A
accessing a file system, 19
accessing the UEFI Shell, 5
accessing the UEFI shell from a serial console connection, 10
application APIs, 67

B
boot order list, 8
booting to the UEFI Shell, 9

C
changing the UEFI boot order, 8
command line completion, 12
key, 12
command line help, 16
command line syntax, 12
commands and capabilities, 5
confirmation prompt, 15
contacting HP, 72
controlling command output, 13
conventions
document, 72

displaying command help, 16
for all command help, 16
for commands beginning with a specific character, 17
for the ls command, 17
displaying help for a specific command, 16
document
conventions, 72
related information, 72
documentation
HP website, 72
providing feedback on, 74

E
editing Shell scripts, 64
Embedded UEFI Shell
adding to boot order list, 8
enabling, 7

F
File
Accessing the fs0 file system from the shell, 19
file system, 19

G
getting started, 12
global page breaks, 14

H
help, 16
obtaining, 72
HP
subscription service, 72
technical support, 72
HP Insight Online, 73
HP ProLiant POST screen, 5
I
invoking scripts, 64
M
manually invoking a Shell script, 64
N
nsh files, 5
O
One-Time Boot menu, 9
P
passwords, 51
R
related documentation, 72
S
sample Shell scripts, 64, 68
script commands, 64
serial console connection, 10
Shell command reference
command syntax examples, 12
Shell script auto start
startup.nsh, 64
Shell script manual invocation, 64
Shell scripts
editing, 64
manually invoking, 64
samples, 64, 68
subscription service, HP, 72
suppressing the confirmation prompt, 15
sysconfig attributes, 52
System Utilities
accessing, 5
T
technical support, 72
HP, 72
typographic conventions, 72
U
UEFI Application APIs, 67
UEFI boot order, 8
UEFI mode, 7
UEFI Programming Model, 67
UEFI Shell command reference
command line syntax, 12
UEFI Shell command status
Status code meanings, 71
UEFI Shell script commands, 64

W
web sites
HP subscription service, 72
websites, 72
product manuals, 72

UEFI Shell prompt, 9
UEFI Shell Script Auto Start, 64

UEFI Shell command status codes
status codes, 71
UEFI Shell commands, 19
alias, 19
attrib, 20
cd, 21
cls, 22
comp, 23
cp, 24
date, 25
dblk, 26
devices, 27
devtree, 28
dh, 29
dh details, 30
dmem, 30
drivers, 31
eco, 32
edit, 33
eficompress, 33
efidecompress, 34
exit, 34
fwupdate, 34
getmtc, 35
goto, 35
help, 35
ls file information, 39
ls volume information, 38
ls/dir, 36
map, 39
map details, 40
memmap, 40
memmap details, 42
mkdir, 42
mode, 42
mv, 43
openinfo, 44
parse, 44
pause, 45
pci, 45
reset, 46
rm/del, 47
set, 48
setsize, 49
shift, 49
smbiosview, 50
stall, 50
sysconfig, 51
sysconfig attributes, 52
time, 59
timezone, 60
touch, 61
type, 61
ver, 62
vol, 62
UEFI Shell command status codes, 71