



Hewlett Packard
Enterprise

HPE ProLiant Gen10 サーバー、ProLiant Gen10 Plus サーバー、および HPE Synergy 用 UEFI シェルユーザーガイド

摘要

このガイドでは、すべての ProLiant Gen10 サーバー、ProLiant Gen10 Plus サーバー、および HPE Synergy コンピュートモジュールのシステム ROM に内蔵されている Unified Extensible Firmware Interface (UEFI) シェルにアクセスして使用方法について詳しく説明します。このガイドは、サーバーおよびストレージシステムのインストール、管理、トラブルシューティングの担当者を対象としています。

ご注意

本書の内容は、将来予告なしに変更されることがあります。Hewlett Packard Enterprise 製品およびサービスに対する保証については、当該製品およびサービスの保証規定書に記載されています。本書のいかなる内容も、新たな保証を追加するものではありません。本書の内容につきましては万全を期しておりますが、本書中の技術的あるいは校正上の誤り、脱落に対して、責任を負いかねますのでご了承ください。

本書で取り扱っているコンピューターソフトウェアは秘密情報であり、その保有、使用、または複製には、Hewlett Packard Enterprise から使用許諾を得る必要があります。FAR 12.211 および 12.212 に従って、商用コンピューターソフトウェア、コンピューターソフトウェアドキュメンテーション、および商用製品の技術データ (Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items) は、ベンダー標準の商用使用許諾のもとで、米国政府に使用許諾が付与されます。

他社の Web サイトへのリンクは、Hewlett Packard Enterprise の Web サイトの外に移動します。Hewlett Packard Enterprise は、Hewlett Packard Enterprise の Web サイト以外の情報を管理する権限を持たず、また責任を負いません。

商標

Microsoft[®]および Windows[®]は、米国および/またはその他の国における Microsoft Corporation の登録商標または商標です。

Intel[®]、インテル、Itanium[®]、Pentium[®]、Intel Inside[®]、および Intel Inside ロゴは、インテルコーポレーションまたはその子会社のアメリカ合衆国およびその他の国における商標または登録商標です。

UEFI[®]は UEFI Forum, Inc.の登録商標です。

Linux[®]は、Linus Torvalds の米国およびその他の国における登録商標です。



目次

UEFI シェルの使用	6
コマンドと機能.....	6
内蔵 UEFI シェルへのアクセス.....	6
シリアルコンソール接続から UEFI シェルへのアクセス.....	7
UEFI シェルコマンドのリファレンス.....	8
カスタム Hewlett Packard Enterprise シェルコマンド.....	8
コマンドライン構文の例.....	8
コマンドラインの補完.....	8
UEFI シェルの概要.....	8
コマンド出力の制御.....	9
画面のスクロールオフを行った出力の表示.....	9
出力を一度に 1 つの画面に制限.....	9
グローバル改ページを有効または無効にする.....	10
詳細出力の表示.....	10
概要出力の表示.....	11
確認プロンプトの抑制.....	11
コマンドヘルプの表示.....	12
すべてのコマンドヘルプを表示する.....	12
特定のコマンドのヘルプの表示.....	12
特定の文字で始まるコマンドのすべてのヘルプの表示.....	13
UEFI 内蔵シェルからのファームウェアのアップデート	15
シェルからファイルシステムへの切り替え.....	15
共通のセットアップコマンドと構成コマンドの使用	15
UEFI シェルコマンド.....	16
acpiview.....	16
ahsdownload.....	18
alias.....	19
attrib.....	20
base64.....	21
boot.....	22
cd.....	23
cls.....	25
comp.....	26
compress.....	27
connect.....	28
cp.....	29
date.....	31
dblk.....	32
devices.....	33
devtree	34
dh.....	35
diags.....	37
disconnect.....	38
dmem.....	39
dmpstore.....	39
drivers.....	41
echo.....	42
edit.....	43



eficompress.....	44
efidecompress.....	44
exit	44
ftp.....	45
fwupdate.....	46
getmtc.....	47
getpcirom.....	47
goto.....	48
hash.....	48
help.....	49
ifconfig.....	50
ifconfig6.....	51
imlview	53
ipmctl.....	53
load.....	54
ls/dir.....	54
map	58
memmap.....	60
mkdir.....	63
mode.....	63
mv.....	64
openinfo.....	65
parse.....	65
partitions.....	66
pause.....	66
pci.....	67
ping.....	68
ping6.....	69
ramdisk.....	70
reconnect.....	71
reset.....	72
restclient.....	72
rm/del.....	74
secboot.....	76
sermode.....	78
set.....	79
setsize.....	80
shift.....	81
smbiosview.....	81
stall.....	82
sysconfig	82
sysconfig 属性.....	86
sysinfo.....	87
tftp.....	88
time.....	89
timezone	90
tlsconfig.....	91
touch.....	93
type.....	93
unload.....	94
ver.....	94
vol.....	95
Webclient.....	96

UEFI シェルスクリプトの実行と編集..... 99



スクリプトの起動方法.....	99
システムユーティリティの UEFI シェルスクリプト自動起動構成.....	99
シェルスクリプトの手動開始.....	99
ファイルへの設定のエクスポートとインポート.....	99
シェルスクリプトの編集.....	99
UEFI シェルスクリプトの例.....	100
アプリケーションソースコードのスクリプト.....	100
起動スクリプト.....	101
UEFI プログラミングモデル.....	105
UEFI シェルコマンドのステータスコード.....	110
Web サイト.....	111
サポートと他のリソース.....	112
Hewlett Packard Enterprise サポートへのアクセス.....	112
アップデートへのアクセス.....	112
カスタマーセルフリペア (CSR)	113
リモートサポート (HPE 通報サービス)	113
保証情報.....	113
規定に関する情報.....	114
ドキュメントに関するご意見、ご指摘.....	114



UEFI シェルの使用

このサーバーのシステム BIOS には ROM に内蔵 UEFI シェルが含まれています。シェル環境には、UEFI シェル仕様に基づいて、スクリプティング、ファイル操作、システム情報の取得を可能にする API と CLI が用意されています。このシェルは他の UEFI アプリケーションも実行します。これらの機能により、UEFI システムユーティリティの機能が強化されます。内蔵 UEFI シェルへのアクセスは、デフォルトで有効になっています。

コマンドと機能

UEFI シェルでは、以下の機能が使用できます。

- ・ 構成コマンド
- ・ スクリプティング
 - `if`、`else`、`endif`、`shift`、および `for/endifor` のような標準のスクリプト作成構文を持つ `nsh` ファイル
 - `echo` コマンド
 - `Autoexec.bat` に似た `startup.nsh` 自動起動ファイル
 - 一部のコマンドの標準形式の出力 (`-sfo`) オプション。これにより、解析コマンドを使用して、カンマ区切り形式の出力を解析できます。
- ・ ファイル操作
 - 任意の FAT16 および FAT32 ファイルの読み取り機能。
 - `md`、`cd`、`cp/copy`、`del`、`dir/ls`、`attrib`、`touch` などの標準ファイル操作コマンド
 - ファイル編集 (`edit`) と表示 (`type`)
 - コンソールおよびファイルとの間の入力および出力のリダイレクト

内蔵 UEFI シェルへのアクセス

注記: UEFI シェルには BIOS 管理者権限が必要です。BIOS 管理者パスワードが設定されている場合、UEFI シェルおよび UEFI シェル起動スクリプトを起動する前にパスワードを入力するようサーバーから要求されます。管理者パスワードの設定については、**sysconfig** または UEFI システムユーティリティユーザーガイドのサーバーセキュリティセクションを参照してください。

手順

1. 内蔵 UEFI シェルが有効になっていることを確認します。デフォルト設定は、有効です。
2. オプション: 内蔵 UEFI シェルを UEFI ブート順序リストに追加する。
3. オプション: UEFI シェルスクリプトの自動実行を有効にします。
4. UEFI ブート順序リストの内蔵 UEFI シェルエントリを変更します。

シェルには、以下の方法でアクセスすることができます。

- ・ 内蔵 UEFI シェルが UEFI ブート順序リストに追加されている場合は、サーバーの POST 中に、POST 画面で **F11**（ブートメニュー）を押します。このオプションを使用できるのは、内蔵 UEFI シェルが UEFI ブート順序リストに追加されている場合のみです。
- ・ システムユーティリティ画面で、**内蔵アプリケーション > 内蔵 UEFI シェル**を選択し、**Enter** キーを押します。
- ・ シリアルコンソール接続を使用する。

シリアルコンソール接続から UEFI シェルへのアクセス

前提条件

端末ソフトウェアが Unicode 文字セット（UTF 8 など）を使用していることが必要です。

注記: シリアルコンソールからシェルにアクセスする場合、入力および出力に使用できる言語は英語だけです。仮想シリアルポート（VSP）では、ブラウザーの選択 GUI モードがサポートされていません。

手順

1. サーバーを起動します。
2. サーバーの iLO IP アドレスを使用して、SSH クライアントアプリケーションで接続を開きます。
3. SSH ポートは **22** のままにします。
4. ログインプロンプトで、ユーザー名とパスワードを入力します。
hpiLO->プロンプトが表示されます。
5. vsp を入力してから、**Enter** キーを押して仮想シリアルポートを開きます。
Shell>プロンプトが表示されます。
6. タスクの完了に必要なコマンドを入力します。
7. exit を入力し、シェルを終了します。

例

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

hpiLO-> vsp

Virtual Serial Port Active: COM2


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

Shell>
```

UEFI シェルコマンドのリファレンス

次の情報では、UEFI シェルコマンドの概要、各コマンドの説明、コマンドの構文、およびコマンドの使用方の例を示します。

カスタム Hewlett Packard Enterprise シェルコマンド

 この記号は、UEFI シェル仕様で提供されるコマンドに Hewlett Packard Enterprise が追加したカスタムコマンドを示します。

コマンドライン構文の例

以下に、コマンドの構文の解釈方法の例を示します。

exit

exit を入力します。

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

現在の日付を表示するには、次のいずれかの操作を行います。

- ・ オプションのパラメーターなしで **date** を入力します。
- ・ 標準形式の出力を指定する場合は、**date** に続けて **-sfo** を入力します。システムに特定の日付を設定するには、**date** に続けて次のいずれかの形式で日付パラメーターを入力します。
 - *mm/dd/yy*
 - *mm/dd/yyyy*

この構文は、月 (*mm*) と日 (*dd*) のパラメーターはオプションですが、2桁の年 (*yy*) または 4桁の年 (*yyyy*) のパラメーターは日付の設定のために必須であることを示しています。

eficompress infile outfile

eficompress に続けて、非圧縮の入力ファイルのファイル名を *infile* パラメーターとして入力し、圧縮された出力ファイルを *outfile* パラメーターとして入力します。

コマンドラインの補完

以下のコマンドライン補完のキーによって、コマンド入力の繰り返しが簡素化されます。

- ・ 上向きの矢印—以前に入力したコマンドのリスト内を古い方に向かって移動します。
- ・ 下向きの矢印—以前に入力したコマンドのリスト内を新しい方に向かって移動します。
- ・ **TAB**—コマンドラインのファイル名を補完します。1つ以上の文字を入力してから、**TAB** キーを押してファイル名を補完します。複数の候補がある場合は、もう一度 **TAB** キーを押してすべての候補を表示します。

UEFI シェルの概要

基本的な UEFI シェルコマンドは、以下の用途で使用できます。

- ・ コマンド出力の制御
- ・ コマンドヘルプの表示
- ・ シェルからファイルシステムへの切り替え
- ・ 共通のセットアップコマンドと構成コマンドの使用

コマンド出力の制御

多くの UEFI シェルコマンドは、出力を画面に表示する方法を制御するためのオプションのパラメーターをサポートしています。以下の各項では、それらのオプションについて説明し、それらのオプションを使用する方法の例を示します。

画面のスクロールオフを行った出力の表示

画面に一度に表示できない量の情報をシェルが出力する場合、上下にスクロールできます。

手順

1. 出力をさらに表示するには、次のいずれかを押します。
 - ・ **PgUp** : ページを上スクロールします。
 - ・ **PgDown** : ページを下スクロールします。

出力を一度に 1 つの画面に制限

手順

1. コマンドの後に `-b` を入力します。
2. 最初の出力画面を確認し、次のいずれかを実行します。
 - ・ 続けて次の画面に進む場合は、**Enter** キーを押します。
 - ・ 表示を終了する場合は、**Q** キーを押します。

例

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

```
Ctrl[19C] PciRoot(0x0)/Pci(0x3,0x1)
Ctrl[19D] PciRoot(0x0)/Pci(0x3,0x2)
Ctrl[19E] PciRoot(0x0)/Pci(0x3,0x3)
Ctrl[19F] PciRoot(0x0)/Pci(0x4,0x0)
Ctrl[1A0] PciRoot(0x0)/Pci(0x4,0x1)
Ctrl[1A1] PciRoot(0x0)/Pci(0x4,0x2)
Ctrl[1A2] PciRoot(0x0)/Pci(0x4,0x3)
Ctrl[1A3] PciRoot(0x0)/Pci(0x4,0x4)
Ctrl[1A4] PciRoot(0x0)/Pci(0x4,0x5)
Ctrl[1A5] PciRoot(0x0)/Pci(0x4,0x6)
Ctrl[1A6] PciRoot(0x0)/Pci(0x4,0x7)
Ctrl[1A7] PciRoot(0x0)/Pci(0x5,0x0)
Ctrl[1A8] PciRoot(0x0)/Pci(0x5,0x1)
Ctrl[1A9] PciRoot(0x0)/Pci(0x5,0x2)
Ctrl[1AA] PciRoot(0x0)/Pci(0x5,0x4)
```

Press ENTER to continue or 'Q' break:

グローバル改ページを有効または無効にする

出力のグローバル改ページを有効または無効にするには、`pagebreak` 環境変数を設定または消去します。このコマンドにより、サーバーの再起動まで保持される UEFI 環境変数が作成されます。

出力のグローバル改ページを有効にするには、次のように入力します。

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

出力のグローバル改ページを無効にするには、次のように入力します。

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

詳細出力の表示

このオプションをサポートするシェルコマンドは、次のとおりです。

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

例

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

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

概要出力の表示

例

```
Shell> ver
Embedded UEFI Shell v2.1
(C) Copyright 1982–2011 Hewlett Packard Enterprise Development LP
UEFI v2.60 (HPE, 0x00012000)
HPE ProLiant システム ROM U31 v1.40 (01/16/2018)

Shell> ver -t
```

確認プロンプトの抑制

このオプションをサポートするシェルコマンドは、次のとおりです。

- ・ cp
- ・ rm/del

例

```
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]
```

コマンドヘルプの表示

すべてのコマンドヘルプを表示する

例

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

特定のコマンドのヘルプの表示

次のいずれかのオプションを使用します。

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

例

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

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



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

NOTES:

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

特定の文字で始まるコマンドのすべてのヘルプの表示

例

```
Shell> help a*
```

Displays, creates, or deletes UEFI Shell aliases.

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

NOTES:

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

EXAMPLES:

- * To display all aliases in the UEFI Shell environment:

```
Shell> alias
```

- * To create an alias in the UEFI Shell environment:

```
Shell> alias shutdown "reset -s"
```

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

Displays or changes the attributes of files or directories.

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

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

NOTES:

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

EXAMPLES:

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

UEFI 内蔵シェルからのファームウェアのアップデート

手順

1. Hewlett Packard Enterprise サポートセンター (<http://www.hpe.com/support/hpesc>) から、ご使用のサーバー用のシステム ROM フラッシュバイナリコンポーネントにアクセスします。
2. USB メディアまたは iLO 仮想メディアにバイナリファイルをコピーします。
3. メディアをサーバーに接続します。
4. UEFI 内蔵シェルへ起動します。
5. USB キーに割り当てられているファイルシステムボリュームを取得するには、`map -r` を入力します。
6. ご使用のサーバー用のシステム ROM フラッシュバイナリコンポーネントを含むファイルシステムへと変更します。fs0:、fs1: など、利用可能な fsx ファイルシステムの 1 つを入力し、**Enter** キーを押します。
7. `cd` コマンドを使用して、現在のディレクトリから、バイナリファイルを含むディレクトリに移動します。
8. `fwupdate -d BIOS -f filename` と入力し、システム ROM をフラッシュします。
9. サーバーを再起動します。アップデートを有効にして、ハードウェアの安定性を維持するには、ファームウェアのアップデート後に再起動する必要があります。

シェルからファイルシステムへの切り替え

ファイル入力またはファイル出力が必要なコマンドを実行する前に、シェルからファイルシステムに切り替えるには、以下の手順に従ってください。

手順

1. HDD、USB、または iLO 仮想 USB を使用して、FAT16 または FAT32 形式のファイルシステムを接続します。
2. `map -r` コマンドを使用して、ファイルシステムのマッピングを更新します。
3. fs0: や fs1: など、利用可能な fsx ファイルシステムの 1 つに入り、**Enter** キーを押します。
プロンプトが fsx> に変わります。x は、選択したファイルシステムの番号です。これで、ファイルがアクセス可能になり、指定したファイルシステムの書き込み可能なファイルに書き込むことができます。

例

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

注記: 出力例では、fs0:\>プロンプトを使用して、シェルからファイルシステムにアクセスする場所を示します。

共通のセットアップコマンドと構成コマンドの使用

内蔵 UEFI シェルに移動する方法を教えてください。

[内蔵 UEFI シェルへのアクセス](#)を参照してください。

サーバーに UEFI ブートオプションがあるかどうかを判断する方法を教えてください。

[sysconfig](#) を参照してください。

タイムゾーンを構成するにはどうすればいいですか？

`timezone` を参照してください。

内蔵 UEFI シェルから一時的な RAM ディスクを作成する方法を教えてください。

`ramdisk` を参照してください。

内蔵 UEFI シェルから、RAM ディスクで選択したファイルにインテグレートドマネジメントログ (IML) をダンプし、ファイルが RAM ディスクにあることを確認する方法を教えてください。

`imlview` を参照してください。

ユーザー定義の名前とデバイスハンドルの間の 1 つまたはすべてのマッピングをリセットする方法を教えてください。

`map` を参照してください。

UEFI ドライバーモデルに準拠したすべてのデバイスを表示する方法を教えてください。

`devtree` を参照してください。

内蔵 UEFI シェルを終了してシステムユーティリティに戻る方法を教えてください。

`exit` を参照してください。

UEFI シェルコマンド

以下では、各コマンド項目の構成要素について説明した後、UEFI シェルコマンドをアルファベット順で示します。

前提条件

構文

コマンドの構文。必須およびオプションのパラメーターを含みます。

説明

コマンドの使用法の簡単な説明。

オプション

構文のパラメーターと変数の説明。

使用法

コマンドの使用法の詳細な説明。

例

コマンドの使用法の 1 つ以上の例。

注記: ユーザー入力は、コマンドの使用例の中で太字で示されます。

出力の詳細

コマンドの表示フィールドの説明 (該当する場合)。

acpiview

構文

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

説明



Advanced Configuration and Power Interface (ACPI) テーブルを表示して逆アセンブルします。

オプション

-s table signature

指定した ACPI テーブルを表示します。RSDP 記述子を除いて、*table signature* は ACPI 仕様で定義した値でなければなりません。RSD PTR ではなく RSDP を使用します。

-a

すべての ACPI 静的テーブルをデコードして表示します。

-p

すべての ACPI 静的テーブルをデコードして表示し、記述子テーブルに AML コードを逆アセンブルします。

-d

テーブルのデコード中に、ACPI テーブルの 16 進数データをダンプします。

-v

テーブルのデコード中に、ACPI テーブルの詳細データを表示します。

-n ID

OEM テーブル ID で指定された記述子テーブルを表示します。

-b

一度に 1 つの画面を表示します。

-o name

-t name

指定された名前のファイルに記述子テーブルのバイナリ ACPI データをダンプします。

使用法

-s、-a、または -p パラメーターのいずれかを含める必要があります。

-p オプションを指定し、他のオプションを指定しない場合、すべての ACPI テーブルがダンプされて解析されます。

-v、-d、または -o が指定されていない場合、詳細表示モードが自動的に有効になります。

-n オプションでは、完全な 8 文字の OEM テーブル ID 文字列が必要です。スペースを含めるには、引用符 (" ") を使用します。

-o オプションには、書き込み可能なファイルシステムが必要です。

例

すべての ACPI テーブルをダンプして解析するには、次のように入力します。

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

DSDT のみをダンプして解析するには、次のように入力します。

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

OEM テーブル ID PCISSDT を持つ SSDT テーブルをダンプして解析するには、次のように入力します。

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



AML を解析せずにすべての ACPI テーブルをダンプするには、次のように入力します。


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

ahsdownload

構文

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

説明

 問題解決のために使用する Active Health System (AHS) ログファイルをダウンロードします。

オプション

-n filename

AHS ファイルの名前を指定します。

-u serialnumber

システムのシリアル番号を指定します。

-s startdate

データを取得する開始日 (YYYY-MM-DD 形式) を指定します。

-e enddate

データを取得する終了日 (YYYY-MM-DD 形式) を指定します。

-case casenumber

サポートケース番号をお客様の情報ファイルに追加します。

-name contactname

連絡先の氏名をお客様の情報ファイルに追加します。

-phone phonenumber

電話番号をお客様の情報ファイルに追加します。

-email emailaddress

メールアドレスをお客様の情報ファイルに追加します。

-a

AHS ログからすべてのデータをダウンロードします。

-l

AHS ダウンロード日付範囲情報を表示します。

-q

ユーザープロンプトを表示せずにクワイエットモードでダウンロードを実行します。

使用法

AHS は、サーバーハードウェアとシステム構成の変化を監視し、記録します。ログファイルは、問題の診断と迅速な解決を支援できます。

AHS は、ユーザーの事業、財務、顧客、社員、パートナー、またはデータセンターに関する情報を収集しません (たとえば、IP アドレス、ホスト名、ユーザー名、パスワードなどは収集されません)。AHS のデータを

ダウンロードし、Hewlett Packard Enterprise に送信することで、分析、技術的な解決、および品質改善のために Hewlett Packard Enterprise がデータを使用することに同意したものと見なされます。

このコマンドは、AHS パーティションをマウントし、AHS ログファイルをダウンロードして、そのファイルを .ahs パッケージファイルにバンドルします。

開始日も終了日も指定しない場合、デフォルトでは直近の 7 日間に関連する AHS データがダウンロードされます。開始日を指定し、終了日は指定しない場合、AHS はダウンロードの終了日として現在の日付を使用します。開始日および終了日を指定した場合は、その日付の範囲内（開始日と終了日を含む）の AHS データのみがダウンロードされます。ファイル名を指定しない場合は、デフォルトのファイル名が生成されます。たとえば、HPE_XXXXXXXXXX_20140821.ahs です。

例

取得可能な AHS ダウンロード日付範囲情報を表示するには、次のように入力します。

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

ログの期間が 7 日間（デフォルトの範囲）の AHS ファイルをデフォルトの場所にダウンロードするには、次のように入力します。

```
fs0:\> ahsdownload
```

日付範囲を指定して AHS ファイルをダウンロードするには、次のように入力します。

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

ログとお客様の情報がすべて含まれる AHS ファイルをデフォルトの場所にダウンロードするには、次のように入力します。

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

AHS ファイル全体をネットワーク上の場所にアップロードするには、次の操作を行います。

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

alias

構文

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

説明

UEFI シェル環境のエイリアスを表示、作成、または削除します。

オプション

-d

エイリアスを削除します。コマンド名を指定してはいけません。

-v

エイリアスを揮発性にします。

alias-name

エイリアス名を指定します。

command-name

元のファイル名またはパスを指定します。

使用法

エイリアスは、既存の UEFI シェルコマンドまたは UEFI アプリケーションの新しい名前を提供します。エイリアスを作成すると、そのエイリアスを使用してコマンドを実行したり、UEFI アプリケーションを起動したりできます。

UEFI シェル環境には、定義済みのエイリアスがいくつかあります。これらのエイリアスは、ファイル処理コマンドについて、MS-DOS および UNIX の場合と同等の名前を提供します。

エイリアスは、`-v` オプションを指定しない限り、シェルの終了後も保持されます。`-v` を指定した場合、エイリアスはシェルの終了後に有効でなくなります。

例

UEFI シェル環境のすべてのエイリアスを表示するには、次のように入力します。

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

UEFI シェル環境でエイリアスを作成するには、次のように入力します。

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

UEFI シェル環境でエイリアスを削除するには、次のように入力します。

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

現在の UEFI 環境で揮発性のエイリアス（行頭に星印*が付いています）を追加するには、次のように入力します。この揮発性のエイリアスは、次の起動時に消えます。

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

attrib

構文

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

説明

ファイルまたはディレクトリの属性を表示、設定、または変更します。

オプション

[+a|-a]

archive 属性を設定またはクリアします。

[+s|-s]

system 属性を設定またはクリアします。

[+h|-h]

hidden 属性を設定またはクリアします。

[+r|-r]

read-only 属性を設定またはクリアします。

file...

ファイル名を指定します。ワイルドカードを使用できます。

directory...

ディレクトリ名を指定します。ワイルドカードを使用できます。

使用法

次の4つの属性タイプがUEFIファイルシステムでサポートされています。

- ・ アーカイブ -A
- ・ システム -S
- ・ 非表示 -H
- ・ 読み取り専用 -R

ファイルがディレクトリの場合は、属性Dも使用できるものとして表示されます。

コマンドラインで指定されたファイルリストのファイルが存在しない場合、attribは、エラーを報告しながら、残りのファイルの処理を続行します。

ファイルまたはディレクトリを指定しない場合は、現在のディレクトリ内のすべてのファイルが表示されます。

属性を指定しない場合は、ファイルの属性が表示されます。

例

ディレクトリの属性を表示するには、次のように入力します。

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

現在のディレクトリ内のすべてのファイルとサブディレクトリの属性を表示するには、次のように入力します。

```
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
```

system属性を拡張子.efiを持つすべてのファイルに追加するには、次のように入力します。

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

read-only属性を拡張子.infを持つすべてのファイルから削除するには、次のように入力します。

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

base64

構文

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

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

説明



base64 エンコードとデコードを提供します。



オプション

value

エンコードまたはデコードする 1 つまたは複数の引数を指定します。

-d

引数をデコードします。新しい行に、それぞれの結果が表示されます。

例

1 つの引数をエンコードするには、次のように入力します。

```
Shell> base64 TestValue
```

1 つの引数をデコードするには、次のように入力します。

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

複数の引数をエンコードするには、次のように入力します。

```
Shell> base64 Value1 Value2
```

複数の引数をデコードするには、次のように入力します。

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

boot

構文

```
boot[-d[-sfo]] [[-a num] | -r]
```

```
[-n num | -all | -pxe | -iscsi | -fcoe | -url | -http]
```

説明



起動するか、UEFI ブートオプションを表示します。

オプション

-d

UEFI ブートオプションを順に表示します。

-n

特定の UEFI ブートオプションを起動します。

-a

特定の UEFI ブートオプションを有効にします。

-i

特定の UEFI ブートオプションを無効にします。

num

起動するオプション番号を指定します。これは、各オプションに対する 4 桁の 16 進値です。

-all

UEFI ブートオプションを順番に起動します。

-pxe

すべての UEFI PXE ブートオプションを順番に起動します。



-iscsi

すべての UEFI PXE ブートオプションを順番に起動します。

-fcoe

UEFI FCoE ブートオプションを順番に起動します。

-url

すべての UEFI URL ブートオプションを順番に起動します。

-http

すべての UEFI HTTP ブートオプションを順番に起動します。

-r

UEFI ブートオプションを更新します。

-sfo

標準形式の出力で情報を表示します。

使用法

このコマンドを使用して、レガシー BIOS ブートオプションの表示や起動を行うことはできません。これを使用して、UEFI ブートターゲット（ダウンロードした OS イメージなど）間の、再起動不要なシームレスな移行を構成できます。

例

すべての UEFI ブートオプションを順に表示するには、次のように入力します。

```
Shell> boot -d
```

ブート順序リスト内のブートオプションを更新し、新しく追加または削除したデバイスを有効にするには、次のように入力します。

```
Shell> boot -r
```

ブートオプションを更新した後、すべての UEFI ブートオプションを順番に表示するには、次のように入力します。

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

UEFI ブートオプションを順番に起動するには、次のように入力します。

```
Shell> boot -all
```

UEFI PXE ブートオプションを順番に起動するには、次のように入力します。

```
Shell> boot -pxe
```

オプション番号 0004 を指定して UEFI ブートオプションを起動するには、次のように入力します。

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

UEFI HTTP ブートオプションを順番に起動するには、次のように入力します。

```
Shell> boot -http
```

cd

構文

```
cd[path]
```

説明

現在のディレクトリを表示または変更します。

オプション

Path

相対的または絶対的なディレクトリパスを指定します。

使用法

ファイルシステムのマッピングを指定すると、現在の作業ディレクトリはそのデバイスに合わせて変更されます。その他の場合は、現在の作業ディレクトリは現在のデバイスに合わせて変更されます。

path が存在しない場合は、現在の作業ディレクトリ（ファイルシステムのマッピングを含む）が標準出力に表示されます。

次の表に、UEFI シェル環境でディレクトリ、その親、およびルートディレクトリの参照に使用される表記規則の説明を示します。

表 1: ディレクトリ名の表記規則

表記規則	説明
.	現在のディレクトリ。
..	現在のディレクトリの親。
\	現在のファイルシステムのルート。

現在の作業ディレクトリは、環境変数 `%cwd%` に保持されます。

例

現在のファイルシステムをマッピング済みの `fs0` ファイルシステムに変更するには、次のように入力します。

```
Shell> fs0:
```

現在のディレクトリをサブディレクトリ `efi` に変更するには、次のように入力します。

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

現在のディレクトリを親ディレクトリ (`fs0:\`) に変更するには、次のように入力します。

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

現在のディレクトリを `fs0:\efi\tools` に変更するには、次のように入力します。

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

現在のディレクトリを現在の `fs` (`fs0`) のルートに変更するには、次のように入力します。

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

注記: `cd` を使用したボリュームの変更は機能しません。例 :

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

まず `fs1:` を入力し、次に目的のディレクトリに対して `cd` を入力します。

ボリューム間を移動し、現在のパスを保持するには、次のように入力します。

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


このシーケンスにより、fs0:\efi\tools 内のすべてのファイルが fs1:\tmp ディレクトリにコピーされます。

cls

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

説明 コンソール出力をクリアし、オプションで背景色と表示色を変更します。

オプション

background

次のオプションから、新しい背景色を指定します。

- 0 - 黒
- 1 - 青
- 2 - 緑
- 3 - シアン
- 4 - 赤
- 5 - マゼンタ
- 6 - 黄
- 7 - ライトグレー

foreground

次のオプションから、新しい表示色を指定します。

- 0 - 黒
- 1 - 青
- 2 - 緑
- 3 - シアン
- 4 - 赤
- 5 - マゼンタ
- 6 - 黄
- 7 - ライトグレー
- 8 - ダークグレー
- 9 - ライトブルー
- 10 - ライトグリーン
- 11 - ライトシアン
- 12 - ライドレッド
- 13 - ライトマゼンタ
- 14 - 黄
- 15 - 白

-sfo

標準形式出力で現在のコンソール色の設定を表示します。



使用法 背景色が指定されていない場合、または背景色と表示色が指定されていない場合、色は変更されません。-sfo フラグが指定されている場合、コンソール出力はクリアされません。コンソールには、現在のコンソールの表示色と背景色の属性設定が表示されます。

例

背景色または表示色を変更せずに標準出力をクリアするには、次のように入力します。

```
fs0:\> cls
```

標準出力をクリアし、背景色をシアンに変更するには、次のように入力します。

```
fs0:\> cls 3
```

 標準出力をクリアし、背景色を黒、表示色を白に変更するには、次のように入力します。

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

comp

構文

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

説明

バイトベースで2つのファイルの内容を比較します。

オプション

-b

一度に1つの画面を表示します。

file1

最初のファイル名を指定します。ディレクトリ名またはワイルドカードは許可されません。

file2

2番目のファイル名を指定します。ディレクトリ名またはワイルドカードは許可されません。

使用法

このコマンドは、2つのファイル間の違いを最大10個表示します。それぞれの違いごとに、違いが始まる場所から最大32バイトがダンプされます。比較するファイルの長さが異なる場合、UEFIシェルは直ちに終了します。

例

長さの異なる2つのファイルを比較するには、次のように入力します。

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

内容が同じ2つのファイルを比較するには、次のように入力します。

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

長さが同じで内容が異なる2つのファイルを比較するには、次のように入力します。

```
fs0:\> comp bios.inf bios2.inf
Compare fs0:\bios.inf to fs0:\bios2.inf
Difference #1:
File1: fs0:\bios.inf
00000000: 5F      *_*
```

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

compress

構文

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

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

説明



MiniZip を使用して、zip ファイルへの圧縮および zip ファイルからの圧縮解除を行って、ネットワークを介して転送するデータを縮小します。

オプション

-z

指定したファイルを zip ファイルに圧縮します。

-u

入力 zip ファイルからファイルを圧縮解除します。

-s

サイレントモードで情報を表示します。出力メッセージはありません。

-ex

ファイルを zip ファイルに追加するか、zip ファイルからファイルを抽出するときに、ファイル名からパスを除外します。

-c1 0-9

圧縮レベルを選択します。0 を指定すると、より高速に圧縮でき、9（デフォルト）を指定すると、圧縮率が上がります。

-o output_file

出力ファイルを指定します。

file...

zip 圧縮する 1 つまたは複数のファイルを指定します。

-p password

zip ファイルのパスワードを指定します。

-od output_directory

zip ファイルからのファイルの抽出先ディレクトリを指定します。

-i input_file

入力ファイルを指定します。

-sfo

標準形式の出力で情報を表示します。



使用法

このコマンドはネットワーク負荷の軽減に役立ちます。これにより、最適なパフォーマンスを維持したまま、複数の圧縮ファイルを転送することができます。 *output_file* 名を指定せずに zip 操作を行うと、拡張子を含まない最初のファイル/ディレクトリの名前が出力 zip ファイルの名前として使用されます。zip 操作でパスワードの暗号化はサポートされていません。パスワード保護された zip ファイルの解読はサポートされていません。zip 操作と解凍操作の両方で、既存のファイルは上書きされます。

例

簡単な zip 操作を実行するには、次のように入力します。

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

zip ファイルを作成し、出力 zip ファイルの名前を指定するには、次のように入力します。

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

より高速の圧縮ロジックで zip ファイルを作成するには、次のように入力します。

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

単純な解凍操作を実行するには、次のように入力します。

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

特定のディレクトリにファイルを解凍するには、次のように入力します。

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

パスワード保護された zip ファイルを特定のディレクトリに解凍するには、次のように入力します。

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

zip ファイルからすべてのファイルをルートディレクトリに解凍するには、次のように入力します。

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

connect

構文

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

説明

ドライバーを特定のデバイスにバインドし、ドライバーを起動します。

オプション

devicehandle

16 進数形式でデバイスハンドルを指定します。

driverhandle

16 進数形式でドライバーハンドルを指定します。

-c

UEFI シェル環境変数で記述されたコンソールデバイスおよび関連するデバイスのみ接続します。

-r

コンソールデバイスに再帰的に接続します。

使用法

devicehandle が指定されない場合は、現在のシステム内のすべてのデバイスハンドルがデフォルトになります。

`driverhandle` が指定されない場合は、一致するすべてのドライバーが、指定されたデバイスにバインドされます。`driverhandle` が指定された場合は、指定された複数のデバイスの接続のなかで優先順位が最も高くなります。`-r` オプションを指定すると、すべてのハンドルが再帰的にスキャンされ、指定されたデバイスに一致するロードされたドライバーまたは内蔵ドライバーがあるかどうかを確認されます。さらに、複数のデバイスハンドルがバインド中に作成された場合、一致するドライバーが指定のデバイスにバインドできるかどうかを確認するために、これらのハンドルもチェックもされます。このプロセスは、デバイスに接続できるドライバーがなくなるまで繰り返されます。

`-r` オプションが指定されていない場合、新しく作成されたデバイスハンドルは、それ以上どのドライバーにもバインドされません。1つのハンドルのみが指定され、そのハンドルに `EFI_DRIVER_BINDING_PROTOCOL` がある場合、そのハンドルはドライバーハンドルとみなされます。それ以外の場合は、デバイスハンドルとみなされます。パラメーターが指定されない場合、コマンドは再帰なしですべてのデバイスに対して適切なドライバーのバインドを試行し、各接続のステータスが表示されます。`connect -r` の使用で、出力のリダイレクトはサポートされません。

例

すべてのドライバーをすべてのデバイスに再帰的に接続するには、次のように入力します。

```
Shell> connect -r
```

すべての接続を表示するには、次のように入力します。

```
Shell> connect
```

0x17 のドライバーを、そのドライバーが管理できるすべてのデバイスに最高の優先順位で接続するには、次のように入力します。

```
Shell> connect 17
```

可能なすべてのドライバーをデバイス 0x19 に接続するには、次のように入力します。

```
Shell> connect 19
```

0x17 のドライバーをデバイス 0x19 に最高の優先順位で接続するには、次のように入力します。

```
Shell> connect 19 17
```

UEFI シェル環境変数で記述されたコンソールデバイスを接続するには、次のように入力します。

```
Shell> connect -c
```

cp

構文

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

説明

1つ以上のソースファイルまたはソースディレクトリをコピー先にコピーします。

オプション

`-r`

再帰コピーを作成します。

`-q`

クワイエットコピー（プロンプトなし）を作成します。

`src src...`

ソースファイルまたはソースディレクトリの名前を指定します。ワイルドカードを使用できます。

dst

コピー先のファイル名またはディレクトリ名を指定します。ワイルドカードは使用できません。指定しない場合は、現在の作業ディレクトリが移動先と見なされます。複数のディレクトリを指定すると、常に最後のディレクトリがコピー先と見なされます。

使用法

ソースがディレクトリの場合は、`-r` フラグを指定する必要があります。`-r` を指定した場合は、ソースディレクトリが再帰的にコピー先にコピーされます(つまり、すべてのサブディレクトリがコピーされます)。コピー先を指定しない場合は、現在の作業ディレクトリがコピー先と見なされます。

ターゲットファイル(ディレクトリではない)がすでに存在する場合は、ファイルを置換するかどうか確認するプロンプトが表示されます。以下の選択項目を使用できます。

- ・ Yes - ファイルを置換します。
- ・ No - ファイルを置換しません。
- ・ All - 後続のすべてのケースで既存のファイルを置換します。
- ・ Cancel - 後続のすべてのケースで既存のファイルを置換しません。

複数のソースファイルまたはソースディレクトリがある場合、コピー先はディレクトリである必要があります。

エラーが発生した場合、コピープロセスは直ちに停止します。

スクリプトで実行する場合、デフォルトは `-q` です。

別のディレクトリにコピーする場合は、ディレクトリがすでに存在している必要があります。

例

現在のディレクトリの内容を表示するには、次のように入力します。

```
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)
```

同じディレクトリ内のファイルをコピーし、ファイル名を変更するには、次のように入力します。

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

複数のファイルを別のディレクトリにコピーするには、次のように入力します。

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

複数のディレクトリを再帰的に別のディレクトリにコピーするには、次のように入力します。

```
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]
```

上記の操作の結果を確認するには、次のように入力します。

```
fs0:\> ls \test
Directory of: fs0:\test          06/18/01 01:01p <DIR>          512
      06/18/01 01:01p <DIR>          0                ..
      01/28/01 08:21p <DIR>          512                test1
      01/28/01 08:21p <DIR>          512                test2
      01/28/01 08:21p <DIR>          512                boot
      01/28/01 08:23p                29
temp.txt
01/28/01 08:23p                28,739            IsaBus.efi
      2 File(s)                28,828 bytes
      5 Dir(s)
```

Shell>

date

構文

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

説明

システムの現在の日付を表示または設定します。

オプション

mm

設定する日付の月を指定します (1~12)。

dd

設定する日付の日を指定します (1~31)。

yyyy

4桁の年を指定します。

-sfo

標準形式の出力表示を指定します。

使用法

パラメーターを指定しない場合は、現在の日付が表示されます。有効な月、日、および年を指定すると、システムの日付がアップデートされます。ルールは次のとおりです。

- ・ 引数では、数字と/以外の文字はすべて無効です。数字が月/日/年の正しい範囲内がない場合、シェルはエラーを報告します。
- ・ 数字の前後にスペースは使用できません。数字内へのスペースの挿入は無効です。
- ・

年の範囲は 1998 以上です。2 つの数字は年を示します。98 より下の数字は 20xx と見なされ、98 以上の数字は 19xx と見なされます。00 は 2000 を意味します。例：

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

有効な年の範囲は 1998～2099 です。

例

システムの現在の日付を表示するには、次のように入力します。

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

長い年の形式の日付を設定し、それを表示するには、次のように入力します。

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

短い年の形式の日付を設定し、それを表示するには、次のように入力します。

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

dblk

構文

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

説明

ブロックデバイスから 1 つ以上のブロックを表示します。

オプション

-b

一度に 1 つの画面を表示します。

device

ブロックデバイス名。

lba

表示する最初のブロックのインデックス（16 進数）。

blocks

表示するブロック数（16 進数）。デフォルトは 1 です。0x10 より大きい場合は、0x10 のみが表示されません。

例

ブロック 0 から始まる blk0 の 1 つのブロックを表示するには、次のように入力します。

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


ブロック 0x2 から始まる fs0 の 1 つのブロックを表示するには、次のように入力します。

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

ブロック 0x12 から始まる fs0 の 0x5 個のブロックを表示するには、次のように入力します。

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

devices

構文

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

説明

UEFI ドライバーによって管理されるデバイスのリストを表示します。

オプション

-b

一度に 1 つの画面を表示します。

-lxxx

特定の言語でデバイスを表示します。設定可能なコードオプションのリストは、UEFI の仕様を参照してください。

-sfo

標準形式の出力で情報を表示します。

例

EFI ドライバーモデルに準拠するすべてのデバイスを表示するには、次のように入力します。

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

出力の詳細

次の表に、このコマンドで考えられる出力の説明を示します。



表 2: 出力の詳細 - devices コマンド

列	説明
CTRL	デバイスのハンドル番号。
TYPE	デバイスタイプ。オプションは次のとおりです。 <ul style="list-style-type: none">・ R - ルートコントローラー・ B - バスコントローラー・ D - デバイスコントローラー
CFG	構成プロトコルのサポートステータス： <ul style="list-style-type: none">・ Y - はい・ N - いいえ
DIAG	診断プロトコルのサポートステータス： <ul style="list-style-type: none">・ Y - はい・ N - いいえ
#P	このデバイスの親コントローラーの数。
#D	このタイプのデバイスの数。
#C	このデバイスで生成された子コントローラーの数。
Device Name	コンポーネント名プロトコルから取得されたデバイスの名前。

devtree

構文

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

説明

UEFI ドライバーモデルに準拠するデバイスのツリーを表示します。

オプション

-b

一度に 1 つの画面を表示します。

-d

デバイスのパスを使用してデバイスツリーを表示します。

-lxxx

特定の言語でデバイスツリーを表示します。設定可能なコードオプションのリストは、UEFI の仕様を参照してください。

devicehandle

指定したハンドルの下にデバイスツリーを表示します。

使用法

デフォルトでは、コンポーネント名プロトコルから取得されたデバイス名が表示されます。オプション-dを指定した場合は、代わりにデバイスのパスが出力されます。

例

UEFI ドライバーモデルに準拠するすべてのデバイスのツリーを一度に1つの画面に表示するには、次のように入力します。

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

dh

構文

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

説明

UEFI 環境のデバイスハンドルを表示します。

オプション

-lxxx

特定の言語でデバイスハンドルを表示します。設定可能なコードオプションのリストは、UEFI の仕様を参照してください。

handle

特定のデバイスのハンドルを表示します。

-p *prot_id*

ハンドルに関連付けられたプロトコル情報を表示します。指定しない場合は、すべてのプロトコルが表示されます。

-d

UEFI ドライバーモデル関連の情報を表示します。

-b

一度に 1 つの画面を表示します。

-v

冗長情報を表示します。

-sfo

標準形式の出力で情報を表示します。

使用法

ハンドル番号を指定した場合、そのデバイスハンドルに関連付けられているすべてのプロトコルの詳細が表示されます。その他の場合は、-p オプションを使用して、特定のプロトコルが含まれるデバイスハンドルを一覧表示できます。-p *prot_id* も *handle* も指定しない場合は、すべてのハンドルが表示されます。

例

すべてのハンドルを一度に 1 つの画面に表示するには、次のように入力します。

```
Shell> dh -b
```

```
Handle dump
```

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

ハンドル 0x30 に関する詳細情報を表示するには、次のように入力します。

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

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

```
IsaIo
```

```
ROM Size.....: 00000000
ROM Location...: 00000000
ISA Resource List :
  IO : 000003F8-000003FF Attr : 00000000
  INT : 00000004-00000000 Attr : 00000000
```

```
dpath
```

```
PNP Device Path for PnP
HID A0341D0, UID 0x0
```

```

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

```

diskio プロトコルに関連付けられたすべてのハンドルを表示するには、次のように入力します。

```

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))

```

Image プロトコルに関連付けられたすべてのハンドルを表示し、画面がいっぱいになったら改ページするには、次のように入力します。

```

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
...

```

出力の詳細

次の表に、このコマンドで考えられる出力の説明を示します。

表 3: 出力の詳細 - dh コマンド

列	説明
Driver Name	ハンドルを生成するドライバーの名前。
Controller Name	ハンドルを生成するコントローラーの名前。
Handle number	ハンドルの整数の番号。
Device Path	ハンドルに関連付けられたデバイスパス。
Protocol Identifiers	プロトコル識別子または GUID のセミコロン区切りのリスト。

diags

構文

```
diags [-console]
```

説明



内蔵 Diagnostics ツールを起動します。

オプション

-console

内蔵 Diagnostics をテキストモードで起動します。

例

内蔵 Diagnostics を設定するには、次の操作を行います。

```
fs0:\> diags
```

内蔵 Diagnostics をテキストモードで起動するには、次の操作を行います。

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

disconnect

構文

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

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

説明

指定したデバイスから 1 つまたは複数のドライバーを切断します。

オプション

devicehandle

16 進数形式でデバイスハンドルを指定します。

driverhandle

16 進数形式でドライバーハンドルを指定します。指定しない場合、*devicehandle* で指定されたデバイスが切断されます。

childhandle

16 進数形式でデバイスの子ハンドルを指定します。指定しない場合、*devicehandle* で指定されたデバイスのすべての子ハンドルが切断されます。

-r

すべてのデバイスからすべてのドライバーを切断し、コンソールを再接続します。

-nc

コンソールデバイスに再接続しません。

使用法

このコマンドは、出力のリダイレクトをサポートしません。

例

すべてのデバイスからすべてのドライバーを切断し、コンソールデバイスを再接続するには、次のように入力します。

```
Shell> disconnect -r
```

コンソールデバイスを含むすべてのデバイスからすべてのドライバーを切断するには、次のように入力します。

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



デバイス 0x28 からすべてのドライバーを切断するには、次のように入力します。

```
Shell> disconnect 28
```

デバイス 0x28 からドライバー 0x17 を切断するには、次のように入力します。

```
Shell> disconnect 28 17
```

デバイス 0x28 の子 0x32 の制御からドライバー 0x17 を切断するには、次のように入力します。

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

dmem

構文

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

説明

システムまたはデバイスのメモリの内容を表示します。

オプション

-b

一度に 1 つの画面を表示します。

address

特定の開始アドレスからのメモリ内容を表示します (16 進形式)。

size

特定のサイズのメモリ内容を表示します (16 進形式)。

-MMIO

EFI_PCI_ROOT_BRIDGE_IO_PROTOCOL を使用して、メモリにマッピングされた内容を表示します。

使用法

address を指定しない場合は、EFI システムテーブルの内容が表示されます。その他の場合は、*address* から始まるメモリが表示されます。*size* を指定しない場合、表示はデフォルトの 512 バイトになります。**-MMIO** を指定しない場合は、メインシステムメモリが表示されます。その他の場合は、デバイスメモリが EFI_PCI_ROOT_BRIDGE_IO_PROTOCOL を使用して表示されます。

例

1af3088 のメモリ内容を 16 バイトのサイズで表示するには、次のように入力します。

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

```
Memory Address 0000000001AF3088 16 Bytes
```

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

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

```
*.....*
```

dmpstore

構文

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

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

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

説明

すべての UEFI 変数を管理します。

オプション

-b

一度に 1 つの画面を表示します。

-guid *guid*

標準的なテキスト形式で表示する変数の GUID を指定します。これを指定せず、**-all** を指定しない場合、`EFI_GLOBAL_VARIABLE` GUID が想定されます。

-sfo

標準形式の出力で情報を表示します。

-all

`EFI_GLOBAL_VARIABLE` 以外の GUID を持つ変数を含む、すべての変数を表示します。

-d

変数を削除します。

-s

変数をファイルに保存します。

-l

ファイルから変数をロードして設定します。

variable

変数名を指定します。リテラル名、または `EFI_UNICODE_COLLATION2_PROCOOL` の `MetaMatch()` 関数で指定されたパターンを使用できます。

変数の値は、16 進ダンプとして印刷されます。

例

`GUID EFI_GLOBAL_VARIABLE` を持つすべての変数をダンプするには、次のように入力します。

```
Shell> dmpstore
```

GUID または名前とは関係なく、すべての変数をダンプするには、次のように入力します。

```
Shell> dmpstore -all
```

`GUID158DEF5A-F656-419C-B027-7A3192C079D2` を持つ `path` 変数をダンプするには、次のように入力します。

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

GUID または名前に関係なく、すべての変数を `VarDump.txt` という名前のファイルに保存するには、次のように入力します。

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

`GUIDEFI_GLOBAL_VARIABLE` を持つ `BootOrder` 変数を削除するには、次のように入力します。

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


drivers

構文

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

説明

UEFI ドライバーモデルに従うドライバーの情報のリストを表示します。

オプション

-lxxx

特定の言語でドライバーを表示します。設定可能なコードオプションのリストは、UEFI の仕様を参照してください。

-sfo

標準形式の出力表で表示します。

例

ドライバーのリストを表示するには、次のように入力します。

```
Shell> drivers
```

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

出力の詳細

次の表に、このコマンドで考えられる出力の説明を示します。



表 4: 出力の詳細 - drivers コマンド

列	説明
DRV	ドライバーの整数ハンドル。
VERSION	ドライバーのバージョン番号。
TYPE	ドライバーの種類。設定できる値は、次のとおりです。 <ul style="list-style-type: none">・ B - バスドライバー・ D - デバイスドライバー
CFG	構成プロトコルのサポートステータス： <ul style="list-style-type: none">・ Y - はい・ N - いいえ
DIAG	ドライバープロトコルのサポートステータス： <ul style="list-style-type: none">・ Y - はい・ N - いいえ
#D	このドライバーが管理しているデバイスの数。
#C	このドライバーが生成した子デバイスの数。
DRIVER NAME	コンポーネント名プロトコルから取得されたドライバーの名前。
IMAGE NAME	ドライバーのロード元のデバイスパス。

echo

構文

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

```
echo message
```

説明

スクリプトファイルからスクリプトコマンドを読み取るときにスクリプトコマンドを表示するかどうかを制御し、指定したメッセージをディスプレイに出力します。

オプション

-on

スクリプトファイルからコマンドを読み取るときに表示を有効にします。

`-off`

スクリプトファイルからコマンドを読み取る時の表示を無効にします。

message

表示するメッセージを指定します。

使用法

このコマンドの最初の形式では、スクリプトファイルからスクリプトコマンドを読み取る時にスクリプトコマンドを表示するかどうかを制御します。引数を指定しない場合は、現在の `on` または `off` ステータスが表示されます。このコマンドの 2 番目の形式では、指定したメッセージをディスプレイに出力します。

このコマンドは、環境変数 `lasterror` の値を変更しません。

例

Hello World のメッセージ文字列を表示するには、次のように入力します。

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

コマンドのエコーをオンにするには、次のように入力します。

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

HelloWorld.nsh を実行し、スクリプトファイルから行を読み取る時に表示するには、次のように入力します。

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

コマンドのエコーをオフにするには、次のように入力します。

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

現在のエコー設定を表示するには、次のように入力します。

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

edit

構文

```
edit[file]
```

説明

ASCII または UCS-2 ファイルをフルスクリーンモードで編集します。

オプション

file

編集するファイルの名前を指定します。指定しない場合は、デフォルトのファイル名で空のファイルが作成されます。

使用法

このコマンドは、UCS-2 と ASCII の両方のファイルタイプをサポートします。

例

shell.log ファイルを編集するには、次のように入力します。

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

eficompress

構文

```
eficompress infile outfile
```

説明

EFI 圧縮アルゴリズムを使用してファイルを圧縮し、圧縮した形式を新しいファイルに書き出します。

オプション

infile

圧縮されていない入力ファイルのファイル名を指定します。

outfile

圧縮後の出力ファイルのファイル名を指定します。

例

uncompressed という名前のファイルを compressed という名前のファイルに圧縮するには、次のように入力します。

```
fs0:\> efiocompress uncompressed compressed
```

efidecompress

構文

```
efidecompress infile outfile
```

説明

EFI 圧縮解除アルゴリズムを使用してファイルを圧縮解除し、圧縮解除した形式を新しいファイルに書き出します。

オプション

infile

圧縮された入力ファイルのファイル名を指定します。

outfile

圧縮解除後の出力ファイルのファイル名を指定します。

例

compressed という名前のファイルを uncompressed という名前のファイルに圧縮解除するには、次のように入力します。

```
fs0:\> efiocompress compressed uncompressed
```

exit

構文

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

説明

UEFI シェルまたは現在のスクリプトを終了します。

オプション

`/b`

現在の UEFI シェルスクリプトだけを終了することを示します。スクリプト内で使用されていない場合は無視されます。

`exit-code`

UEFI シェルスクリプトを終了する場合は、環境変数 `lasterror` 内に置かれた値を指定します。UEFI シェルのインスタンスを終了する場合は、コーラーに返される値を指定します。指定しない場合は、0 が返されます。

例

UEFI シェルを終了するには、次のように入力します。

```
fs0:\> exit
```

ftp

構文

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

説明



ネットワークファイル転送のための FTP サーバーに接続します。

オプション

`host`

サーバーの IPv4 アドレスまたはホスト名を指定します。

`port`

サーバーの FTP ポートを指定します。

`-b`

サブコマンド用に改ページを有効にします。

使用法

このコマンドは、ネットワークファイル転送 (FTP) 操作のための対話型シェルを起動します。IPv4 アドレスだけがサポートされます。ファイル転送をキャンセルするには、**ESC** キーまたは **Ctrl + C** キーを押します。

- ❗ **重要:** `webclient` または `ftp` を同じネットワークインターフェイス上で実行する場合、そのネットワークインターフェイス上で `ifconfig` を使用する必要はありません。システムユーティリティで構成された **プリブートネットワーク設定** によってこのインターフェイスと IP アドレスの設定が自動的に選択されるためです。

`ftp` と `webclient` で使用するインターフェイスを `ifconfig` で構成した場合、その設定は消去され、代わりに、コマンドの実行時にインターフェイス上でシステムユーティリティの **プリブートネットワーク設定メニュー** が適用されます。

例



指定した IP アドレスの FTP サーバーに接続するには、次のように入力します。

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

ホスト名を使用して FTP サーバーに接続するには、次のように入力します。

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

IP アドレス、ユーザー名 (user)、パスワード (pass) を使用して FTP サーバーに接続するには、次のように入力します。

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

指定した IP アドレスの FTP サーバーに接続し、サブコマンドで改ページを有効にするには、次のように入力します。


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

fwupdate

構文

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

説明

 システム BIOS、NIC、およびストレージカードを含む、ファームウェアコンポーネントをアップデートします。

システムファームウェアのアップデートには、fwupdate コマンドに加えて、システムユーティリティのファームウェアアップデートオプション、FWUPDATE ユーティリティ、および Windows、Linux、VMware オペレーティングシステム用のオンラインフラッシュコンポーネント内のイメージを使用できます。

このコマンドはシステムをチェックし、使用可能なファームウェアリビジョンを選択して提供します (複数存在する場合)。

オプション

—l

ファームウェアアップデート機能を持つデバイスとそのリビジョンの一覧を表示します。

—d device

デバイス名を指定します。

—i image

アップデートするイメージを指定します。

—f file

アップデートするファームウェアイメージファイルを指定します。

—q

すべての確認を省略してクワイエットモードで情報を表示します。

-a

-f で指定されたイメージを使用して該当するすべての NVDIMM をアップデートします。

-sfo

標準形式の出力で情報を表示します。

使用法

一部のデバイスには、複数のファームウェアイメージがあります。たとえば、ネットワークアダプターには、PXE ブートコード用のファームウェアイメージ、および NCSI ファームウェアがある場合があります。このような場合は、個別にイメージをアップデートできます。

例

デバイスとそのファームウェアリビジョンの一覧を表示するには、次のように入力します。

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

システム BIOS ファームウェアをアップデートするには、次のように入力します。

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

ネットワーク上の場所から BIOS ファームウェアをアップロードするには、次の操作を行います。

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

getmtc

構文

```
getmtc
```

説明

ブートサービスから MTC (モノトニックカウンター) 値を取得し、それを表示します。

使用法

このコマンドは、現在のモノトニックカウンター値を表示します。下位 32 ビットは、このコマンドを実行するたびに増分されます。システムをリセットするたびに、上位 32 ビットが増分され、下位 32 ビットが 0 にリセットされます。

getpcirom

構文

```
getpcirom
```

説明

PCI ROM イメージを取得します。

使用法

このコマンドは、ドライバーをダウンロードしてストレージドライバイメージを抽出し、ハッシュを DB に登録できます。

オプション

```
GETPCIROM [-i <index> | -d <device path>]
[-o <filename>]
[-l]
```

- ・ -l - サポートされているすべてのデバイスを一覧表示します。
- ・ -i - リストからインデックスを指定します。
- ・ -d - デバイスのパスを指定します。
- ・ -o - 出力ファイル名を指定します。

注記

-d または -i コマンドは選択されたデバイスであり、サポートされているすべてのデバイスはコマンド (-l) によって一覧表示されます。

例

サポートされているすべてのデバイスを一覧表示するには、次のように入力します。

```
fs0:\> getpcirom -l
```

GetPciRom.bin にインデックス番号と出力イメージを指定して、指定のデバイスから PCI オプション ROM イメージを取得するには、次のように入力します。

```
fs0:\> getpcirom -i 1
```

GetPciRom.bin にデバイスパス文字列と出力イメージを指定して、指定のデバイスから PCI オプション ROM イメージを取得するには、次のように入力します。

```
fs0:\> getpcirom -d PciRoot(0x0)/Pci(0x1C,0x0)/Pci(0x0,0x2)/
Offset(0x14600,0x309FF)
```

指定したファイルにインデックス番号と出力イメージを指定して、指定のデバイスから PCI オプション ROM イメージを取得するには、次の手順を実行します。

```
fs0:\> getpcirom -i 1 -o abc.rom
```

goto

構文

```
goto label
```

説明

スクリプト内のラベルに移動します。

hash

構文

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

説明



1 つまたは複数のファイルのハッシュを計算します。

オプション



option

ハッシュオプションを指定します。有効な値は次のとおりです。

- ・ `-md5` - MD5 ダイジェスト
- ・ `-sha1` - SHA1 (160 ビット) ダイジェスト
- ・ `-sha256` - SHA256 ダイジェスト
- ・ `-sha384` - SHA384 ダイジェスト
- ・ `-sha512` - SHA512 ダイジェスト

file...

ハッシュする 1 つまたは複数のファイルを指定します。

例

ファイルの SHA256 メッセージダイジェストを計算するには、次のように入力します。

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

help

構文

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

説明

UEFI シェルに組み込まれているコマンドのリストを表示します。

オプション

cmd

ヘルプを表示するコマンドを指定します。

pattern

コマンドヘルプの表示に使用するパターンを指定します。

special

シェルコマンドラインで使用される特殊文字のリストを表示します。

`-usage`

コマンドの使用法に関する情報を表示します。この表示は、`-section:NAME` および `-section:SYNOPSIS` を指定する場合と同じになります。

`-v`

冗長情報を表示します。

`-section sectionname`

ヘルプ情報の指定したセクションを表示します。

`-b`

ヘルプ出力を一度に 1 つの画面に表示します。

使用法

オプションを指定しない場合は、各コマンドとその機能の簡単な説明が表示されます。`-v` を指定した場合は、指定したコマンドのすべてのヘルプ情報が表示されます。`-section` を指定した場合は、指定したヘルプセクションだけが表示されます (例を参照)。`-usage` を指定した場合は、コマンド、簡単な説明、および使用法が表示されます。

シェルまたはシェルコマンドの実行可能ファイルがあるディレクトリにある UCS-2 テキストファイルから、ヘルプテキストが収集されます。ファイルには、`command-name.man` という名前が付いています。ここで、`command-name` はシェルコマンドの名前です。ファイルは、例に示すように、MAN ページ形式のサブセットに従います。

オプションを指定しない場合は、ページの `NAME` セクションだけが表示されます。

例

UEFI シェルのコマンドのリストを表示し、1つの画面の後で改ページするには、次のように入力します。

```
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.
```

シェルコマンド `ls` に関するヘルプ情報を表示するには、以下の構文オプションのいずれかを使用します。

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

文字 `p` で始まるコマンドのリストを表示するには、次のように入力します。

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

ifconfig

構文

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

説明

UEFI IPv4 ネットワークスタックのデフォルト IP アドレスを変更します。

オプション

`-r`

インターフェイス構成を更新して、DHCP ポリシーを設定します。

`-l`

構成を一覧表示します。

`-s`

構成を設定します。

`name`

アダプター名を指定します。たとえば、`eth0` などです。

`dhcp`

すべてまたは特定のインターフェイスの IPv4 アドレスを DHCP4 が動的に要求することを指定します。

`static IPaddress`

静的 IPv4 アドレスを、それぞれが 0~255 の範囲のピリオドで区切られた 4 つの整数値で指定します。

`mask`

サブネットマスクを、それぞれが 0~255 の範囲のピリオドで区切られた 4 つの整数値で指定します。

gateway

デフォルトゲートウェイを、それぞれが 0~255 の範囲のピリオドで区切られた 4 つの整数値で指定します。

dns IP

DNS サーバーアドレスを指定します。

permanent

構成が（1 回のみでなく）permanent になるように指定します。

使用法

- ❗ **重要:** 同じインターフェイス上で `webclient` または `ftp` を実行する予定の場合、ネットワークインターフェイス上で `ifconfig` コマンドを使用する必要はありません。システムユーティリティ内で構成されている **プリブートネットワーク設定** により、インターフェイスおよび IP アドレス設定が自動的に選択されます。

`ftp` および `webclient` によって使用されるインターフェイスが `ifconfig` で構成されると、その設定は消去され、代わりにシステムユーティリティの **プリブートネットワーク設定** メニューがコマンド実行時にインターフェイス上で適用されます。

このコマンドを使用して、シェルからのプリブートネットワークアクセス用のホストネットワークを構成できます。 `-c` オプションは、すべてまたは特定のインターフェイスの構成をクリアします。そのため、関連するインターフェイスのネットワークスタックのデフォルトが DHCP に戻ります。

例

`eth0` インターフェイスの構成を一覧表示するには、次のように入力します。

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

DHCP4 を使用して `eth0` インターフェイスの IPv4 アドレス構成を動的に要求するには、次のように入力します。

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

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

`eth0` インターフェイスの DNS サーバーアドレスを構成するには、次のように入力します。

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

ifconfig6

構文

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

```
ifconfig6 -s name [dad num] [auto | [man [id mac] [host IPv6 gwIPv6] [dns IPv6]]]
```

説明

ネットワークインターフェイスの IPv6 構成を表示または変更します。

オプション

name

`eth0` などのアダプター名を指定します。

-r name

指定されたインターフェイスをすべて再構成し、自動ポリシーを設定します。指定されたインターフェイスがすでに自動的に設定されている場合、IPv6 構成が更新されます。

-l name

指定したインターフェイスの構成を一覧表示します。

-s name dad num

指定したインターフェイスの重複アドレス検出送信数を設定します。

-s name auto

指定したインターフェイスの自動ポリシーを設定します。

-s name man id mac

指定したインターフェイスの代替インターフェイス ID を設定します。インターフェイスは、手動のポリシーで制御する必要があります。

-s name man host IPv6 gw IPv6

指定したインターフェイスの静的ホスト IP とゲートウェイアドレスを設定します。インターフェイスは、手動のポリシーで制御する必要があります。

-s name man dns IPv6

指定したインターフェイスの DNS サーバー IP アドレスを設定します。インターフェイスは、手動のポリシーで制御する必要があります。

例

インターフェイス eth0 の構成を一覧表示するには、次のように入力します。

```
Shell> ifconfig6 -l eth0
```

自動構成を使用してインターフェイス eth0 の IPv6 アドレス構成を動的に要求するには、次のように入力します。

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

自動ポリシーで eth0 の DAD 送信数を設定するには、次のように入力します。

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

手動のポリシーで eth0 の代替インターフェイス ID を設定するには、次のように入力します。

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

インターフェイス eth0 に対して静的 IPv6 アドレス構成を使用して、この構成をネットワークの再ロードまで続くように指定するには、次のように入力します。

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

DER 形式で 3 番目の TLS 証明書を abc.crt という名前のファイルにエクスポートするには、次のように入力します。

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

TLS 暗号アルゴリズムを指定するには、次のように入力します。

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

TLS でサポートされている暗号を表示するには、次のように入力します。

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

TLS 接続の証明書検証を有効にするには、次のように入力します。

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

TLS v1.2 プロトコルバージョンを使用することを指定するには、次のように入力します。

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

imlview

構文

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

説明



インテグレートドマネジメントログ (IML) を表示し、エクスポートします。

オプション

-sfo

標準形式の出力で情報を表示します。

-b

IML ログを一度に 1 つのページに表示します。

-c

IML ログのエントリーをクリアします。

-export filename

指定したファイルに IML ログをエクスポートします。

start_id

この指定された番号よりも大きい ID を持つレコードを出力します。

end_id

この指定された番号よりも小さい ID を持つレコードを出力します。

使用法

IML は、サーバーで発生した履歴イベントの記録です。IML のエントリーが問題の診断や発生する可能性がある問題の特定に役立つ可能性があります。

例

現在のファイルシステムをマッピング済みの FS0 ファイルシステムに変更するには、次のように入力します。

```
Shell> fs0:
```

現在の IML ログを表示するには、次のように入力します。

```
FS0:\> imlview
```

output.txt という名前のファイルに IML ログをエクスポートするには、次のように入力します。

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

ipmctl

構文

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



説明

インテル Optane DC パーシステントメモリーモジュールを管理するには、`ipmctl` コマンドを使用します。

`-h`

コマンドのヘルプを表示します。

使用法

このコマンドの使用方法について詳しくは、<https://github.com/intel/ipmctl/tree/master/Documentation/ipmctl> にあるインテルのドキュメントを参照してください。

load

構文

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

説明

UEFI ドライバーをメモリにロードします。

オプション

`-nc`

ドライバーをロードしますが、ドライバーを接続しません。

file

ロードする UEFI ドライバーを含むイメージファイルを指定します。ワイルドカードを使用できます。

使用法

このコマンドを使用して一度に複数のファイルをロードできます。また、ファイルを指定するときにワイルドカードを使用できます。`-nc` が指定されていない場合、システムはドライバーを適切なデバイスに接続しようとします。また、以前にロードされたドライバーを対応するデバイスに接続することもできます。

例

`Isabus.efi` ファイルに含まれるドライバーをロードするには、次のように入力します。

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

`Isabus.efi` ファイルおよび `IsaSerial.efi` ファイルに含まれているドライバーをロードするには、次のように入力します。

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

ファイル名に `Isa` のある複数のファイルに含まれているドライバーをロードするには、次のように入力します。

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

接続せずに `Isabus.efi` ファイルに含まれているドライバーをロードするには、次のように入力します。

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

ls/dir

構文

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

説明

ディレクトリまたはファイル情報の内容を一覧表示します。dir コマンドは、このコマンドの内部エイリアスです。

オプション

-r

再帰的に表示します（サブディレクトリを含みます）。

-a attrib

指定した属性を持つファイルだけを表示します。属性を指定しない場合は、すべてのファイルが一覧表示されます。-a を指定しない場合は、システムファイルと隠しファイル以外のファイルがすべて一覧表示されます。属性 (*attrib*) には、以下の1つ以上を使用できます。

- ・ a - アーカイブ
- ・ s - システム
- ・ h - 非表示
- ・ r - 読み取り専用
- ・ d - ディレクトリ

-sfo

標準形式の出力で表示します。

file

ファイルまたはディレクトリの名前を指定します。ワイルドカードを使用できます。

使用法

ファイル名またはディレクトリ名を指定しない場合は、現在の作業ディレクトリが想定されます。ディレクトリの内容は、以下がすべて真の場合に表示されます。

- ・ オプション **-r** が指定されていない。
- ・ *file* パラメーターでワイルドカード文字が指定されていない。
- ・ 指定された *file* が既存のディレクトリを表している。その他の場合はすべて、コマンドは次のように機能します。
 - 指定した名前と一致するすべてのファイルまたはディレクトリが表示されます。
 - **-r** オプションは、再帰検索を実行するかどうかを指定します。
 - オプションの **-a attrib** は、指定した属性を持つファイルだけを表示します。複数の属性を指定した場合は、それらすべての属性を持つファイルだけが表示されます。-a の後に何も続かない場合は、属性に関係なく、すべてのファイルまたはディレクトリが表示されます。-a 自体を指定しない場合は、システムファイルと隠しファイルを除くすべてのファイルが表示されます。

例

隠し属性またはシステム属性をファイルに追加してそのファイルを隠すには、次のように入力します。

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

h または s 属性を持つファイルまたはディレクトリを除き、すべてのファイルとディレクトリを表示するには、次のように入力します。

```
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)
```

現在のディレクトリ内のすべての属性を持つファイルを表示するには、次のように入力します。

```
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)
```

現在のディレクトリ内の読み取り専用属性を持つファイルを表示するには、次のように入力します。

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

s の属性を持つファイルを表示するには、次のように入力します。

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

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

fs0:\efi ディレクトリ内にあるすべてを再帰的に表示するには、次のように入力します。

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


現在のディレクトリ内の指定した種類のファイルを再帰的に検索するには、次のように入力します。

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

出力の詳細

次の表に、このコマンドから出力される可能性のあるボリューム（ディレクトリ）とファイルの情報の説明を示します。

表 5: 出力の詳細 - ls コマンド（ボリューム情報）

列	説明
Name	標準のボリュームラベル。
Total Size	ボリューム内の合計バイト数。
Read Only status	読み取り専用ステータス： <ul style="list-style-type: none">・ True・ False
Free Space	ボリューム内の空きバイトの合計数。
Block Size	ファイルが通常大きくなる際の名目ブロックサイズ（バイト単位）。

表 6: 出力の詳細 - ls コマンド（ファイル情報）

列	説明
Name	完全なファイル名とディレクトリ。ファイルシステムのマッピングされた名前を含みます。
Logical Size	ファイルのサイズ（バイト単位）。
Physical Size	パディングを含む、ボリューム内のファイルのサイズ（バイト単位）。
Attributes	ファイル属性のリスト。利用可能な値は次のとおりです。 <ul style="list-style-type: none">・ a - アーカイブ・ d - ディレクトリ・ h - 非表示・ r - 読み取り専用・ s - システム
File Creation Time	ファイルを作成した時刻（hh:mm:ss 形式）。

表は続く

列	説明
File Creation Date	ファイルを作成した日付 (dd:mm:yyyy 形式)。
File Access Time	ファイルにアクセスした時刻 (hh:mm:ss 形式)。
File Access Date	ファイルにアクセスした日付 (dd:mm:yyyy 形式)。
File Modification Time	ファイルを変更した時刻 (hh:mm:ss 形式)。
File Modification Date	ファイルを変更した日付 (dd:mm:yyyy 形式)。

map

構文

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

説明

ユーザー定義名とデバイスハンドル間のマッピングを表示または構成します。

オプション

-d

マッピングを削除します。

mappedname

マッピング名を指定します。

-r

マッピングをリセットします。

-v

すべてのマッピングに関する冗長情報を表示します。

-c

一貫性のあるマッピングを表示します。

-f

通常のマッピングを表示します。

-t

デバイスタイプに応じてフィルタリングされたデバイスマッピングを表示します。サポートされているタイプは次のとおりです。

- ・ fp-ディスク
- ・ hd-ハードディスク
- ・ cd-CD-ROM

2つのタイプの間にはカンマを入れてタイプを結合することができます。タイプの間にはスペースを使用することはできません。

-sfo

標準形式の出力で表示します。

-u

新しく取り付けられたデバイスのマッピングを追加したり、取り外したデバイスのマッピングを削除したりしますが、既存のデバイスのマッピングは変更しません。ユーザー定義のマッピングは保持します。

handle

ハンドルの番号を指定します。

mapping

デバイスに割り当てる、新しくマッピングされる名前を指定します。マッピングは、コロン (:) で終了する必要があります。

使用法

このコマンドの最も一般的な使用法は、ファイルシステムのプロトコルをサポートするデバイスのマッピング名を作成することです。このマッピングを作成すると、その名前はすべてのファイル操作コマンドで使用できます。

UEFI シェル環境では、認識されたファイルシステムをサポートするすべてのデバイスのデフォルトマッピングを作成します。

このコマンドを使用して、追加のマッピングを作成することができます。または、`-d` オプションを指定して既存のマッピングを削除することができます。パラメーターなしでコマンドを使用すると、現在のすべてのマッピングが一覧表示されます。`-v` オプションを使用すると、マッピングと各デバイスに関する追加情報が表示されます。

`-r` オプションは、システムのすべてのデフォルトマッピングをリセットします。これは、システム構成が前回の起動以降変更された場合に便利です。

`-u` オプションは、新しく取り付けられたデバイスのマッピングを追加したり、取り外したデバイスのマッピングを削除したりしますが、既存のデバイスのマッピングは変更しません。ユーザー定義のマッピングも保持されます。元のマッピング名が前回特定のデバイスパスで使用された場合は、そのデバイスパスを持つデバイスでそのマッピング名が使用されるように、マッピングの履歴が保存されます。現在のデバイスが変更されない場合は、現在のディレクトリも保持されます。

システム内の各デバイスには、一貫性のあるマッピングがあります。ハードウェア構成が変更されていない場合、デバイスの一貫性のあるマッピングは変更されません。複数のマシンのハードウェア構成が同じ場合、デバイスの一貫性のあるマッピングは同じです。システム内の一貫性のあるすべてのマッピングを一覧表示するには、`-c` オプションを使用します。

マッピングは、数字と文字で構成されます。その他の文字は使用できません。

このコマンドは、マッピングを削除または表示する際、ワイルドカードをサポートします。ただし、マッピングを割り当てる場合は、ワイルドカードは禁止されています。

例

マッピングを削除するには、次のように入力します。

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

出力の詳細

次の表に、このコマンドで考えられる出力の説明を示します。

表 7: 出力の詳細-map コマンド

列番号	説明
1	テーブルの名前。名前は mappings です。
2	マッピングされた名前。マッピングされたデバイス名。
3	デバイスパス。マッピングされたデバイス名に対応するデバイスパス。
4	一貫性のある名前。 <i>mappedname</i> と同等の、マッピングされた一貫性のある名前（存在する場合）。 <i>mappedname</i> がすでにマッピングされた一貫性のある名前である場合、この列は空です。

memmap

構文

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

説明

システムのメモリマップを表示します。

オプション

-b

一度に 1 つの画面を表示します。

-sfo

詳細な表と概要の表に、標準形式の出力を表示します。

使用法

メモリマップは、システム内のすべての物理メモリおよびそれらが現在どのように使用されているかを追跡します。

例

システムのメモリマップを表示するには、次のように入力します。

```
fs0:\> memmap
```

Type	Start	End	# Pages	Attributes
Available	0000000000000000-000000000008DFFF	000000000000008E	000000000000008E	000000000000000F
Reserved	000000000008E000-000000000008FFFF	0000000000000002	0000000000000002	000000000000000F
Available	0000000000090000-000000000009FFFF	0000000000000010	0000000000000010	000000000000000F
Available	0000000000100000-0000000072CFFFFFF	000000000072C00	000000000072C00	000000000000000F
BS_Data	0000000072D00000-0000000072DFFFFFF	000000000000100	000000000000100	000000000000000F
Available	0000000072E00000-00000000837F0FFF	0000000000109F1	0000000000109F1	000000000000000F
LoaderCode	00000000837F1000-0000000084030FFF	000000000000840	000000000000840	000000000000000F
Available	0000000084031000-00000000847BDFFF	00000000000078D	00000000000078D	000000000000000F
BS_Data	00000000847BE000-0000000084809FFF	00000000000004C	00000000000004C	000000000000000F
Available	000000008480A000-0000000084827FFF	00000000000001E	00000000000001E	000000000000000F
BS_Data	0000000084828000-00000000850B7FFF	000000000000890	000000000000890	000000000000000F
Available	00000000850B8000-00000000850B9FFF	000000000000002	000000000000002	000000000000000F
BS_Data	00000000850BA000-00000000850CCFFF	000000000000013	000000000000013	000000000000000F

```

LoaderData 00000000850CD000-00000000850D5FFF 0000000000000009 000000000000000F
BS_Data   00000000850D6000-00000000861DFFFF 000000000000110A 000000000000000F
BS_Code   00000000861E0000-00000000861F9FFF 000000000000001A 000000000000000F
RT_Data   00000000861FA000-00000000861FEFFF 0000000000000005 800000000000000F
BS_Code   00000000861FF000-0000000086958FFF 0000000000000075A 000000000000000F
BS_Data   0000000086959000-000000008695CFFF 0000000000000004 000000000000000F
BS_Code   000000008695D000-0000000086E41FFF 000000000000004E5 000000000000000F
BS_Data   0000000086E42000-000000008C842FFF 000000000000005A01 000000000000000F
BS_Code   000000008C843000-000000008CDE1FFF 0000000000000059F 000000000000000F
RT_Data   000000008CDE2000-000000008CDE6FFF 0000000000000005 800000000000000F
Reserved  000000008CDE7000-000000008D006FFF 00000000000000220 000000000000000F
RT_Data   000000008D007000-000000008D00FFFF 0000000000000009 800000000000000F
Reserved  000000008D010000-000000008E00FFFF 0000000000000100 000000000000000F
RT_Data   000000008E010000-000000008E020FFF 0000000000000011 800000000000000F
BS_Data   000000008E021000-000000008E021FFF 0000000000000001 000000000000000F
RT_Data   000000008E022000-000000008E742FFF 00000000000000721 800000000000000F
BS_Data   000000008E743000-0000000096742FFF 0000000000000800 000000000000000F
Available 0000000096743000-0000000096748FFF 0000000000000006 000000000000000F
LoaderData 0000000096749000-0000000096B42FFF 000000000000003FA 000000000000000F
Available 0000000096B43000-0000000096F12FFF 000000000000003D0 000000000000000F
LoaderCode 0000000096F13000-0000000096F42FFF 00000000000000030 000000000000000F
Available 0000000096F43000-0000000096F43FFF 0000000000000001 000000000000000F
BS_Code   0000000096F44000-0000000097F42FFF 0000000000000FFF 000000000000000F
RT_Data   0000000097F43000-0000000098342FFF 0000000000000400 800000000000000F
RT_Code   0000000098343000-0000000098B42FFF 0000000000000800 800000000000000F
Reserved  0000000098B43000-0000000098C42FFF 0000000000000100 000000000000000F
ACPI_NVS  0000000098C43000-000000009B042FFF 0000000000000240 000000000000000F
ACPI_Recl 000000009B043000-000000009B242FFF 0000000000000200 000000000000000F
BS_Data   000000009B243000-000000009B244FFF 0000000000000002 000000000000000F
RT_Data   000000009B245000-000000009B2C5FFF 0000000000000081 800000000000000F
BS_Data   000000009B2C6000-00000000A2C63FFF 0000000000000799E 000000000000000F
BS_Code   00000000A2C64000-00000000A2C64FFF 0000000000000001 000000000000000F
BS_Data   00000000A2C65000-00000000A2CFBFFF 0000000000000097 000000000000000F
Available 00000000A2CFC000-00000000AF7FFFFF 0000000000000CB04 000000000000000F
Available 0000000100000000-000000103FFFFEFFF 0000000000F3FFFF 000000000000000F
Reserved  00000000000F0000-00000000000FFFFF 0000000000000010 0000000000001001
Reserved  00000000AF800000-00000000BFFFFFFF 0000000000010800 0000000000000008
MMIO      00000000C0000000-00000000CFFFFFFF 0000000000010000 8000000000000001
MMIO      00000000FE000000-00000000FE00FFFF 0000000000000010 8000000000000001
Reserved  000000103FFFFF000-000000103FFFFEFFF 0000000000000001 0000000000000000

```

```

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

```

```

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

```



出力の詳細

次の表に、このコマンドで考えられる出力の説明を示します。

表 8: 出力の詳細 - memmap コマンド

列	説明
Type	メモリのタイプ。オプションは次のとおりです。 <ul style="list-style-type: none">• Available• LoaderCode• LoaderData• BootServiceCode• BootServiceData• RuntimeCode• RuntimeData• Reserved• MemoryMappedIO• MemoryMappedIOPortSpace• UnusableMemory• ACPIReclaimMemory• ACPIMemoryNVS• PalCode
Start	開始アドレス。
End	終了アドレス。
# Pages	4 KB ページの数。
reserved	予約済みメモリの合計サイズ (バイト単位)。
LoaderCode	ローダーコードの合計サイズ (バイト単位)。
LoaderData	ローダーデータの合計サイズ (バイト単位)。
BS_code	ブートサービスコードの合計サイズ (バイト単位)。
BS_data	ブートサービスデータの合計サイズ (バイト単位)。
RT_data	ランタイムデータの合計サイズ (バイト単位)。

表は続く

列	説明
available	使用可能なメモリ (バイト単位)。
Total Memory	合計メモリサイズ (バイト単位)。

mkdir

構文

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

説明

1つ以上の新しいディレクトリを作成します。

オプション

dir

ディレクトリの1つ以上の名前を指定します。ワイルドカードは使用できません。

使用法

dir に入れ子のディレクトリが含まれる場合は、子ディレクトリの前に親ディレクトリが作成されます。ディレクトリがすでに存在する場合、コマンドはエラーで終了します。

例

新しいディレクトリを作成し、その内容を表示するには、次のように入力します。

```
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)
```

複数のディレクトリを作成して表示するには、次のように入力します。

```
fs0:\> mkdir temp1 temp 2
fs0:\> ls
Directory of: fs0:\

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

mode

構文

```
mode[col row]
```

説明

コンソール出力デバイスのモードを表示または変更します。

オプション

col

列の数を指定します。

row

行の数を指定します。

使用法

このコマンドをパラメーターなしで入力すると、標準出力デバイスで現在サポートされているモードのリストが表示されます。次に *row* および *col* パラメーターを使用してこのコマンドを実行し、標準出力デバイスの行と列の数を変更することができます。

注記: *mode* コマンドを使用して、現在選択されている表示モードを変更するたびに、表示はクリアされます。

例

標準出力で使用可能なすべてのモードと現在選択されているモード (*で示されます) を表示するには、次のように入力します。

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

現在のモード設定を 80x50 テキストモード表示に変更するには、次のように入力します。

```
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

構文

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

説明

1つ以上のファイルをファイルシステム内の移動先に移動します。

オプション

src...

ソースファイルまたはソースディレクトリの名前を指定します。ワイルドカードを使用できます。

dst

ワイルドカードが使用可能な移動先のファイル名またはディレクトリ名を指定します。指定しない場合は、現在の作業ディレクトリが移動先と見なされます。コマンドラインに複数の引数がある場合は、常に最後の引数が移動先と見なされます。

使用法

このコマンドは、ファイルシステムボリューム間の移動をサポートしません。移動先が既存のディレクトリである場合、ソースはそのディレクトリ内に移動します。そうでない場合、ソースはディレクトリの名前が変更され、移動先に移動します。移動先を指定しない場合は、現在のディレクトリが移動先であると想定されます。

読み取り専用のファイルまたはディレクトリを移動しようとする、エラーになります。読み取り専用ファイルを含むディレクトリの移動が可能です。ディレクトリをそれ自体またはそのサブディレクトリの中に移動することはできません。現在の作業ディレクトリが移動対象のディレクトリである場合、または移動対象のディレクトリのサブディレクトリである場合は、ディレクトリを移動できません。

エラーが発生した場合、残りのファイルまたはディレクトリは引き続き移動します。

例

ファイルの名前を変更するには、次のように入力します。

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

openinfo

構文

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

説明

ハンドルに関連付けられたプロトコルとエージェントを表示します。

オプション

-b

一度に1つの画面を表示します。

handle

指定したハンドルのオープンプロトコル情報を表示します。

parse

構文

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

説明

標準形式の出力に出力された指定レコードから値を取得します。

オプション

filename

ソースファイル名を指定します。

tablename

解析するテーブル名を指定します。

column

特定のレコードからどの値を解析するかを決定する、1ベースの列インデックスを指定します。



-i instance

ShellCommand の指定したインスタンスの後、指定した *tablename* の *n* 番目のインスタンスから解析を開始します。存在しない場合は、すべてのインスタンスが返されます。

-s instance

指定した ShellCommand テーブルの *n* 番目のインスタンスから解析を開始します。存在しない場合は、1 が想定されます。

使用法

このコマンドにより、`-sfo` パラメーターを使用したコマンドからのデータ出力を含むファイルのデータの解析が可能になります。標準形式出力にはよく知られている解析手段があるので、このコマンドは、スクリプトがそのような構築出力ファイルを消費して、UEFI シェル用に記述されたスクリプトのロジックでその取得データを使用するようにする簡単な手段として使用するためのものです。

例

以下のデータは、一時ファイル (`temp.txt`) に含まれています。

```
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"
```

インデックスパラメーターを使用して `temp.txt` ファイルを解析するには、次のように入力します。

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

partitions

構文

```
partitions [-v]
```

説明



システム内のディスクパーティションを一覧表示します。

オプション

`-v`

すべてのファイルシステムパーティションに関する詳細情報を一覧表示します。

例

詳細なパーティション情報を表示するには、次のように入力します。

```
Shell> partitions -v
```

pause

構文

```
pause [-q]
```

説明

スクリプトファイルの実行を一時停止します。

使用法



このコマンドは、スクリプトでのみ使用できます。このコマンドは、ディスプレイにメッセージを出力し、スクリプトファイルの実行を一時停止して、キーボード入力を待ちます。**q** または **Q** キーを除く任意のキーを押すと、実行が再開されます。**q** または **Q** キーを押すと、スクリプトの処理が終了します。終了しない場合は、`pause` コマンド後の次の行から実行が継続されます。

オプション

-q

ディスプレイのメッセージを非表示にします。

例

このスクリプトは、`pause` コマンドの例です。

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

エコーをオンにしてスクリプトを実行するには、次のように入力します。

```
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.
```

エコーをオフにしてスクリプトを実行するには、次のように入力します。

```
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

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

説明

PCI デバイスリストまたは PCI 関数構成スペース情報を表示します。

オプション

bus

バス番号を指定します。

dev

デバイス番号を指定します。

func

関数番号を指定します。

-s seg

セグメント番号を指定します。

-i

解釈された情報を表示します。

-ec ID

指定した PCIe の拡張機能に関して解釈された情報を表示します。

使用法

このコマンドは、システム内のすべての PCI デバイスを表示します。PCI デバイスの構成スペースに表示される情報は、指定したバス、デバイス、および関数の各アドレスに基づいています。関数アドレスを指定しない場合は、デフォルトで 0 になります。

-i オプションは、指定した PCI デバイスの詳細情報を表示します。デバイスの PCI 構成スペースは、詳細な解釈とともにダンプされます。

パラメーターを指定しない場合は、すべての PCI デバイスが詳細な解釈とともにダンプされます。

番号パラメーター **bus** と **dev** を指定し、パラメーター **func** と **seg** をいずれも指定しない場合は、**func** または **seg** がデフォルト値の 0 に設定されます。

例

システム内のすべての PCI デバイスを表示するには、次のように入力します。

```
fs0:\> pci
```

バス 0、デバイス 0、関数 0 の構成スペースを表示するには、次のように入力します。

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

セグメント 0、バス 0、デバイス 0、関数 0 の構成スペースを表示するには、次のように入力します。

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

ping

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

説明

IPv4 スタックを持つターゲットホストの ping を実行します。

オプション

-s

ソースアダプターが、IPv4 アドレスであると指定します。

sourceIP

ソースマシンの IPv4 アドレスを指定します。

-n count

送信するエコー要求データグラムの数を指定します。

-l size

エコー要求データグラムのデータバッファのサイズを指定します。

targetIP

ターゲットマシンの IPv4 アドレスを指定します。

使用法

このコマンドは ICMPv4 ECHO_REQUEST データグラムを使用して、ホストから ECHO_REPLY を取り出します。

例

192.168.0.1 にある 64 バイトのデータを持つターゲットホストに ping を実行するには、次のように入力します。

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

20 個のエコー要求データグラムを送信することによって、202.120.120.100 にあるターゲットホストに ping を実行するには、次のように入力します。

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

ソースアダプターを IPv4 アドレスとして指定することによってターゲットホストに ping を実行するには、次のように入力します。

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

ping6

構文

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

説明

UEFI IPv6 ネットワークスタックでターゲットマシンの ping を実行します。

オプション

-l size

バッファサイズをバイト単位で指定します（デフォルトは 16、最小値は 16、最大値は 32768）。

-n count

要求回数を指定します（デフォルトは 10、最小値は 1、最大値は 10000）。

-s sourceip

ソース IPv6 アドレスを指定します。

targetip

ターゲット IPv6 アドレスを指定します。

使用法

targetip がリンクローカルアドレスである場合、*sourceip* を指定する必要があります。

例

2002::1 から 1000 バイトの要求を 5 回送信することで、ターゲットホストに ping を実行するには、次のように入力します。

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

1000 バイトでターゲットホストに ping を実行するには、次のように入力します。

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

ramdisk

構文 `ramdisk -c[-s size][-v volumelabel][-t type] [-m memorytype]`

`ramdisk -d[fs|all]`

`ramdisk -l [-sfo]`

説明



RAM ディスクの作成と削除を行います。

オプション

-c

RAM ディスクを作成します。

-s *size*

RAM ディスクのサイズを MB 単位で指定します。有効な値は、4~512 MB です。

-v *volumelabel*

RAM ディスクのボリュームラベル名を指定します。有効な値は、スペースと次の文字を含まない最大 11 の文字です。% ^ * + = [] | : ; \ < > ? /

-t *type*

ファイルシステムの種類を指定します。

-m *memorytype*

メモリタイプを指定します。

-d

1 つ以上の RAM ディスクを削除します。

fs

RAM ディスクを削除するためのファイルシステムのドライブを指定します。

all

すべての RAM ディスクを削除します。

-l

すべての RAM ディスクを一覧表示します。

-sfo

標準形式の出力で情報を表示します。

使用法

このコマンドを使用して、一時的なステージング場所をプロビジョニングできます。このコマンドは、ベアメタル環境でステージングまたはスクリプティング用のメディアを使用できない場合に役立ちます。以下のファイルタイプとサイズの最大 10 の RAM ディスクがサポートされます。



- ・ FAT16 - (デフォルト) 最小サイズ 4 MB~最大サイズ 512 MB
- ・ FAT32 - 最小サイズ 512 MB~最大サイズ 4096 MB

-t *type* を指定しない場合、デフォルトの FAT16 タイプのファイルシステムが作成されます。
有効な *memorytype* オプションは *BS_Data* (デフォルト) で、予約済みです。

例

512 MB の FAT16 RAM ディスクを作成するには、次のように入力します。

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

RAMDISK1 のボリュームラベルで、512 MB の FAT16 RAM ディスクを作成するには、次のように入力します。

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

1000 MB の FAT32 RAM ディスクを作成するには、次のように入力します。

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

すべての RAM ディスクのマッピング情報のリストを表示するには、次のように入力します。

```
Shell> ramdisk -l
```

RAM ディスク *fs0* を削除するには、次のように入力します。

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

reconnect

構文

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

説明

ドライバーをデバイスに再接続します。

オプション

devicehandle

デバイスハンドルを指定します (16 進数)。

-r

ドライバーをすべてのデバイスに再接続します。

driverhandle

ドライバーハンドルを指定します (16 進数)。指定しない場合、指定されたデバイス上のすべてのドライバーが再接続されます。

childhandle

デバイスの子ハンドルを指定します (16 進数)。指定しない場合、*devicehandle* のすべてのドライバーのすべての子ハンドルが再接続されます。

使用法

このコマンドは、まず指定したデバイスから指定したドライバーを切断し、次にドライバーを再帰的にデバイスに接続します。

`-r` オプションを使用すると、すべてのドライバーがすべてのデバイスに再接続されます。デバイスにバインドされているすべてのドライバーは切断され、次に再帰的に接続されます。

詳細は、`connect` コマンドと `disconnect` コマンドを参照してください。

例

すべてのドライバーをすべてのデバイスに再接続するには、次のように入力します。

```
Shell> reconnect -r
```

すべてのドライバーをデバイス 0x28 に再接続するには、次のように入力します。

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

0x28 から 0x17 を切断し、0x17 を持つドライバーを最高の優先順位としてデバイス 0x28 に再接続するには、次のように入力します。

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

0x28 から 0x17 を切断し、子の 0x32 を破棄してから、0x17 を持つドライバーを最高の優先順位としてデバイス 0x28 に再接続するには、次のように入力します。

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

reset

構文

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

説明

システムをリセットします。

オプション

`-w`

ウォームブートを実行します。

`-s`

シャットダウンを実行します。

`-c`

コールドブートを実行します。

string

サービスをリセットするために渡される文字列を指定します。

使用法

このコマンドのデフォルトの使用法では、コールドリセットが実行されます。*string* を指定すると、その文字列が `SystemTable ResetSystem()` 関数に渡され、システムリセットの理由がシステムに通知されます。

例

システムをリセットするには、次のように入力します。


```
Shell> reset
```

restclient

構文


```
restclient -uri URI [options]
restclient -type TYPE [options]
restclient -t restclient -time
```

説明

 ローカルの RESTful API サービスと対話します。restcli コマンドは、このコマンドの内部エイリアスです。

リソースの選択オプション

-uri URI

特定の URI のリソースを要求します。

-type TYPE

特定のベースタイプに一致するリソースを返します (存在する場合)。

アクション

-m METHOD

指定されたメソッドを使用して HTTP 要求を送信します。次の値が有効です。POST、GET、PUT、PATCH、または DELETE

-g [PROPERTY...]

リソース全体を取得するか、またはリソース内の特定のプロパティを取得します。このオプションで **-m GET** が暗黙指定されます。

-s PROPERTY=VALUE

リソースプロパティの値を指定します。このオプションで **-m PATCH** が暗黙指定され、要求内容が自動的に生成されます。

-t

すべてのリソースタイプを一覧表示します。

-time

ISO 8601 UTC 形式で、RESTful 時間を表示します。

一般的なオプション

-i FILE

要求内容にある特定ファイルの内容を送信します。

-o FILE

ASCII エンコーディングを使用して、JSON 応答本体を指定したファイルに書き込みます。リソース全体の GET (**-g** または **-m GET**) のみ有効であり、それ以外の場合は無視されます。リソースが 16 Kb を超える場合は、この引数を使用します。

-c

使用可能な場合、キャッシュされた内容を GET 要求に使用します。このオプションにより、パフォーマンスは向上しますが、古いデータが返される可能性があります。

使用法

URI は、先頭にスラッシュを付けずにルートリソースから始める必要があります。たとえば、rest/v1/Systems です。

TYPE は、TypeName.X.Y.Z という形式の文字列です。ここで X、Y、Z は、それぞれメジャー、マイナー、および Errata のタイプのバージョンです。一部 (またはすべて) のバージョン指定子が省略されると、ワイ

ルドカードによるマッチングが代わりに使用されます。また、マイナーと Errata のバージョンを指定する TYPE は、下位互換性があるため、値がより大きいリソースにも一致します。

PROPERTY では大文字と小文字が区別され、区切り記号「/」を追加することでネストされたプロパティを選択できます。-s オプションでは、ネストされたプロパティを指定できません。

VALUE は、JSON (オブジェクト、アレイ、null など) を表すことができます。空白を保持する場合を除いて、文字列値に引用符は必要ありません。

FILE の内容は、有効な JSON である必要があります。

例

ルートリソースを取得するには、次のように入力します。

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

BIOS リソースから AdminName プロパティを取得するには、次のように入力します。

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

iScsi リソースから iSCSIInitiatorName プロパティを取得するには、次のように入力します。

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

ComputerSystem リソースからシステム ROM のバージョンを取得するには、次のように入力します。

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

BIOS リソースの AdminName プロパティを設定するには、次のように入力します。

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

iScsi リソースの iSCSIInitiatorName プロパティを設定するには、次のように入力します。

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

POST 要求を送信するには、次のように入力します。

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

応答本体をファイルに書き込むには、次のように入力します。

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

rm/del

構文

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

説明

1 つ以上のファイルまたはディレクトリを削除します。del コマンドは、このコマンドの内部エイリアスです。

オプション

-q

クワイエットモードで、確認プロンプトを表示せずに削除します。

file

削除するファイル名を指定します。ワイルドカードを使用できます。

directory

削除するディレクトリを指定します。ワイルドカードを使用できます。

使用法

ターゲットがディレクトリの場合、すべてのサブディレクトリを含む、そのディレクトリを削除します。このコマンドは、親ディレクトリ（またはファイル自体）が削除対象であるファイルをリダイレクトすることはできません。

読み取り専用のファイルまたはディレクトリの削除は失敗します。1つ以上の読み取り専用ファイルを含むディレクトリの削除は失敗します。エラーが発生した場合、`rm` は直ちに終了し、以降のファイルまたはディレクトリは削除されません。

現在のディレクトリが削除対象のディレクトリである場合、または削除対象のディレクトリのサブディレクトリである場合は、ディレクトリを削除できません。削除対象に指定したファイルにワイルドカードが含まれる場合、確認のプロンプトは表示されません。

ルートディレクトリ、現在のディレクトリ、またはその上位ディレクトリを削除することはできません。

例

次のように入力すると、ディレクトリが見つからない場合に複数のディレクトリを一度に削除しようとすることとなります（コマンドが終了する原因となります）。

```
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
```

複数のディレクトリをワイルドカードを使用して削除するには、次のように入力します。

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

次のように入力すると、読み取り専用ファイルを含むディレクトリを削除しようとするようになります。エラープロンプトの原因になります。

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

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

secboot

構文

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

説明



セキュアブートデータベース、キー、およびセキュリティレポートを表示および修正します。

オプション

-l

セキュアブートデータベースとキーを表示します。

all

すべてのセキュアブート変数の署名を表示または削除します。

PK

プラットフォームキー (PK) の情報を表示します。これは大文字と小文字が区別されます。

KEK

キー変換キー (KEK) の情報を表示します。これは大文字と小文字が区別されます。

db

許可済み署名データベース (DB) の情報を表示します。

dbx

禁止された署名データベース (DB) の情報を表示します。

-sfo

標準形式の出力で情報を表示します。

-e

DER 形式の X509 ファイル、EFI アプリケーションのハッシュ、またはシェルスクリプトをセキュアブート変数に登録します。シェルスクリプトは署名 "#!NSH" で始める必要があります。



-f file

DER 形式の X509 ファイル情報を表示します。

-r

すべてのセキュアブート署名をプラットフォームのデフォルトにリセットします。

-d

すべての署名を削除するか、指定されたデータベースの署名を削除します。

-i index

特定のデータベースから署名 (1、2、...) を選択します。

-q

確認のためのプロンプトを表示せずにクワイエットモードで表示します。

-x

セキュアブートデータベースから宛先ファイルに証明書またはハッシュをエクスポートします。

例

すべてのセキュアブート変数の署名を表示するには、次のように入力します。

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

許可済み署名データベース情報を表示するには、次のように入力します。

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

DER 形式の X509 ファイル情報を表示するには、次のように入力します。

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

セキュアブート変数に DER 形式の X509 ファイルを登録するには、次のように入力します。

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

許可済み署名データベースに EFI アプリケーションのハッシュを登録するには、次のように入力します。

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

許可済み署名データベースにネットワーク上の EFI アプリケーションのハッシュを登録するには、次のように入力します。

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

許可済み署名データベースにスクリプトのハッシュを登録するには、次のように入力します。

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

ネットワーク上の場所にあるスクリプトのハッシュを許可済み署名データベースに登録するには、次のように入力します。

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

すべてのセキュアブート署名をプラットフォームのデフォルトにリセットするには、次のように入力します。

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

KEK セキュアブート署名をプラットフォームのデフォルトにリセットするには、次のように入力します。

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

すべてのセキュアブート署名を削除するには、次のように入力します。

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

プラットフォームキーを削除するには、次のように入力します。

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

許可済み署名データベースをクリアするには、次のように入力します。

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

キー交換キーから 2 番目の署名を削除するには、次のように入力します。

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

KEK セキュアブートデータベースから 3 番目のエントリーをファイル `abc.der` にエクスポートするには、次のように入力します。

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

すべてのセキュアブートデータベース (PK/KEK/db/dbx) からすべてのエントリーをエクスポートするには、次のように入力します。

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

KEK セキュアブートデータベース (PK/KEK/db/dbx) からすべてのエントリーをエクスポートするには、次のように入力します。

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

sermode

構文

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

説明

シリアルポートの属性を設定します。

オプション

handle

16 進数形式でシリアルポートのデバイスハンドルを指定します。この情報を取得するには、`dh` コマンドを使用します。

baudrate

ボーレートを指定します。有効な値は、50、75、110、150、300、600、1200、1800、2000、2400、3600、4800、7200、9600 (デフォルト)、19200、38400、57600、115200、230400、460800 です。その他のすべての値は、次に最も高い設定に変換されます。

parity

指定されたシリアルポートのパリティビット設定を設定します。

- ・ `d` - デフォルトパリティ
- ・ `n` - パリティなし
- ・ `e` - 偶数パリティ
- ・ `o` - 奇数パリティ
- ・ `m` - マークパリティ
- ・ `s` - スペースパリティ

databits

データビット数を指定します。有効な値は、4、7、8（デフォルト）です。

stopbits

ストップビット数を指定します。有効な値は、0（0ストップビット - デフォルト設定）、1（1ストップビット）、2（2ストップビット）、15（1.5ストップビット）です。

例

すべてのシリアルポートデバイスに関する設定を表示するには、次のように入力します。

```
Shell> sermode
```

ハンドル 0x6B のシリアルポートデバイスの設定を表示するには、次のように入力します。

```
Shell> sermode 6B
```

ハンドル 0x6B のシリアルポート設定を、9600 bps、偶数パリティ、8 データビット、1ストップビットに構成するには、次のように入力します。

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

set

構文

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

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

説明

UEFI シェル環境変数を作成、表示、変更、または削除します。

オプション

-v

次回の起動時に消える揮発性の変数を設定します。

-d

変数を削除します。

sname

変数名を指定します。

value

変数の値を指定します。

使用法

このコマンドは、*sname* で指定した環境変数をオプションの *value* パラメーターに設定します。パラメーターなしで使用すると、すべての環境変数が表示されます。**-d** オプションを指定して実行すると、*sname* で指定した環境変数が削除されます。

このコマンドは、環境変数 `lasterror` の値を変更しません。

例

環境変数を追加するには、次のように入力します。

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

環境変数を表示するには、次のように入力します。

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

環境変数を削除するには、次のように入力します。

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

環境変数を変更するには、次のように入力します。

```
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
```

環境変数を付加するには、次のように入力します。

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

次の起動時に消える揮発性の変数を設定するには、次のように入力します。

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

setsize

構文

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

説明

ターゲットファイルのサイズを調整します。

オプション

size

調整後のファイルのサイズを指定します。

-d

変数を削除します。

file

サイズを調整するファイルを指定します。

使用法

このコマンドは、ファイルのサイズを調整する際に、渡されたパラメーターに基づいてそのファイルのサイズを自動的に切り捨てるかまたは拡張します。ファイルが存在しない場合は作成されます。ファイルに含まれる実際のデータより小さいサイズを設定すると、そのデータは切り捨てられます。

例

ファイルのサイズを設定するには、次のように入力します。

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

shift

構文

```
shift
```

説明

UEFI シェルスクリプトの位置パラメーターの内容をシフトして、スクリプトがパラメーターの内容を左から右に処理できるようにします。

使用法

このコマンドは、UEFI シェルスクリプトのパラメーターの内容をシフトして、%1 を破棄、%2 を%1 にコピー、%3 を%2 にコピー、%4 を%3 にコピー、のようにします。これにより、UEFI シェルスクリプトは、左から右にスクリプトのパラメーターを処理できます。

このコマンドは、UEFI シェル環境変数 `lasterror` を変更しません。

例

エコーをオンにしてスクリプトを実行するには、次のように入力します。

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

エコーをオフにしてスクリプトを実行するには、次のように入力します。

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

smbiosview

構文

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

説明

SMBIOS 情報を表示します。

オプション

-t

SmbiosType のすべての構造を表示します。

SmbiosType

SMBIOS 構造のタイプを指定します。これは 0~42 の数字です。サポートされている値とその説明を表示するには、`help smbios` コマンドを使用します。

-h

SmbiosHandle の構造を表示します。

SmbiosHandle

SMBIOS 構造の一意の 16 ビットハンドルを指定します。

-s

統計表を表示します。

-a

すべての情報を表示します。

例

SmbiosType 7 (キャッシュ情報) のすべての構造を表示するには、次のように入力します。

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

stall

構文

```
stall time
```

説明

スクリプトの実行中の操作を停止する予定の時間をマイクロ秒単位で設定します。

オプション

time

プロセッサを停止するマイクロ秒の秒数を指定します。

例

20 マイクロ秒の間プロセッサを停止するには、次のように入力します。

```
Shell> stall 20
```

sysconfig

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

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

`sysconfig -s AdminPassword=settingvalue OldAdminPassword=settingvalue sysconfig -s PowerOnPassword=settingvalue OldPowerOnPassword=settingvalue sysconfig -d[get|set][DefaultType][settingname|all] [-sfo] sysconfig -export filename[-ASCII]`

説明 HPE システム BIOS の設定を構成します。

オプション

-b

一度に1つの画面を表示します。

-i

設定可能な値を含め、指定した設定またはすべての設定の情報を表示します。

settingname

情報を表示する設定の名前を指定します。「**sysconfig 属性**」を参照してください。

all

すべての設定のすべての情報を表示します。

-g

選択した設定またはすべての設定の現在の値を表示します。

-s

指定した設定の値を設定します。

settingvalue

設定値を指定します。

AdminPassword

新しい管理者パスワードを指定します。

OldAdminPassword

リセットする管理者パスワードを指定します。

PowerOnPassword

サーバーの電源をオンにするための新しいパスワードを指定します。

OldPowerOnPassword

リセットする電源投入時パスワードを指定します。

-d

指定された *DefaultType*、またはすべてのデフォルトのタイプの設定のデフォルト値を取得（リスト）または設定します。

DefaultType

取得（一覧表示）または設定するデフォルトタイプ設定を指定します。

-import

スクリプトファイルからすべての設定をインポートします。

-export

スクリプトファイルにすべての設定をエクスポートします。

filename

インポートまたはエクスポート対象のスクリプトファイルを指定します。

ASCII

ファイル出力に ASCII エンコーディングを使用します。

-sfo

標準形式の出力で情報を表示します。

使用法

文字列タイプの設定を表示または設定するには、`settingvalue` にスペースまたは文字 '=' が含まれる場合は、二重引用符を使用します。たとえば、"`sysconfig -s AdminName="Joe Smith"`"です。

文字列タイプ `settingvalue` に二重引用符を含めるには、\を入力します。

文字列タイプ `settingvalue` に\文字を含めるには、\\を入力します。

文字列タイプ設定から `settingvalue` を削除するには、二重引用符を使用します。たとえば、"`sysconfig -s AdminName=""`"です。

-sfo オプションを使用すると、シェルスクリプトを使用してコマンド出力を簡単に解析することができます。このオプションを使用するときは、値文字列のセミコロン文字がエスケープシーケンス '\;' に置き換えられます。

-d set オプションは、ProductId、SerialNumber、RomSelection、パスワード、またはブート順序をサポートしていません。

-export、-set、および-import オプションは、SFO をサポートしません。次の -sfo 形式がサポートされています。

- ・ -g (get) の場合：-g および-i オプションは* (ワイルドカード文字) をサポートしており、可能な構成文字列に一致します。たとえば、`sysconfig -g uefi*` は、文字列 `uefi` で始まる現在の設定値を表示します。結果には、`UefiShellBootOrder`、`UefiSerialDebugLevel`、`UefiBootOrder` などが含まれます。

`sysconfig -i uefi*startup*` は、文字列 `uefi` で始まる設定に関する情報を表示し、かつ文字列 `startup` を含みます。結果には、`UefiShellStartup`、`UefiShellStartupLocation`、`UefiShellStartupUrl` などが含まれます。

列	説明
1	テーブル名 (SysConfigGet)。
2	設定名。
3	設定値。複数の値はセミコロンで区切られます。

- ・ -i の場合：

列	説明
1	テーブル名 (SysConfigGet)。
2	設定名。
3	現在の設定値。
4	設定可能な値。複数の値はセミコロンで区切られます。
5	設定タイプ

- ・ -d (get) の場合：

列	説明
1	テーブル名 (SysConfigGet)。
2	設定名。
3	設定値。複数の値はセミコロンで区切られます。
4	デフォルトの設定タイプ。

例

すべての BIOS 設定の現在値を表示する :

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

uefi で始まる設定に関する情報を表示するには、次のように入力します。

```
Shell> sysconfig -i uefi*
```

ProcHyperthreading 設定の詳細情報を表示するには、次のように入力します。

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

ProcHyperthreading 設定の現在値を取得する :

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

ProcHyperthreading 設定を無効に設定するには、次のように入力します。

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

すべてのデフォルトの構成設定をリストするには、次のように入力します。

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

すべてのシステムデフォルトをリストするには、次のように入力します。

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

NicBoot1 のデフォルト設定を一覧表示する:

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

すべての BIOS 設定のデフォルト値を設定する :

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

NicBoot1 のデフォルト設定を設定する :

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

新しい管理者パスワードに 123 を設定する :

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

電源オンのパスワードとして Joe Smith を削除する :

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

UEFI ブート順序を一覧表示するには、次のように入力します。

```
Shell> sysconfig -i UefiBootOrder
```

```
UefiBootOrder=
```

```
0: 内蔵 UEFI シェル
```

```
8: Rear USB 2
```

```
10: Embedded LOM 1 Port 1
```

Embedded LOM 1 Port 1 が最初にブートするように新しい UEFI ブート順序を設定する :

```
Shell> sysconfig -s UefiBootOrder=10,0
```

新しいブート順序は次のとおりです。

```
10: Embedded LOM 1 Port 1
0: 内蔵 UEFI シェル
8: Rear USB 2
```

デフォルトの UEFI デバイス優先順位を設定するには、次のように入力します。

```
Shell> sysconfig -s DefaultUefiDevicePriority=0,1,2,3,4,5,6,7
```

新しいデフォルトの UEFI デバイス優先順位は次のとおりです。

```
0: フロッピードライブ
1: オプティカルドライブ
2: USB 大容量記憶デバイス
3: 内蔵ストレージコントローラー
4: アドインストレージコントローラー
5: 内蔵/フレキシブルネットワーク
6: アドインネットワークコントローラー
7: 内蔵 UEFI シェル
```

EmbeddedUefiShell 設定の現在値を標準出力形式で取得する :

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

標準形式の出力で EmbeddedUefiShell 設定に関する詳細情報を表示するには、次のように入力します。

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

標準形式の出力で ServerName 設定のデフォルト値を取得するには、次のように入力します。

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

sysconfig 属性

考えられるすべての sysconfig 属性の名前、説明、現在の値、設定可能な値、および Enum 設定タイプを表示することができます。以下にコマンド出力の一例を示しています。

注記: sysconfig コマンドの出力は、サーバーモデルによって異なります。

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

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

Setting Name      = NicBoot1              [Embedded LOM 1 Port 1]
```

```

Current Value = NetworkBoot [Network Boot]
Possible Values = NetworkBoot [Network Boot]
                 Disabled [Disabled]
Setting Type = Enum

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

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

```

sysinfo

構文

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

説明



システム情報を表示します。これには、システム名、シリアル番号、製品 ID、BIOS バージョン、バックアップ BIOS バージョン、パワーマネジメントコントローラーのファームウェアバージョン、ブートモード、システムメモリ、プロセッサ、iLO IP アドレス、ネットワークデバイスが含まれます。

オプション

token

表示する内容を指定します。使用可能なトークンは、次のとおりです。

- ・ *summary* - システム情報の概要
- ・ *all* - すべてのシステム情報
- ・ *cpu* - CPU 情報
- ・ *mem* - メモリ情報
- ・ *fw* - ファームウェア情報
- ・ *pci* - PCI デバイス情報

-sfo

標準形式の出力で情報を表示します。

-b

一度に 1 つの画面に情報を表示します。

-v

詳細出力で情報を表示します。



-o <file>

指定したファイルに出力をリダイレクトします。

例

システム情報の概要を表示するには、次のように入力します。

```
Shell> sysinfo summary
```

メモリ情報を表示するには、次のように入力します。

```
Shell> sysinfo mem
```

標準形式の出力でメモリ情報を表示するには、次のように入力します。

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

メモリの詳細情報（未装着のスロットに関する情報を含む）を表示するには、次のように入力します。

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

ファイルにすべてのシステム情報を保存するには、次のように入力します。

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

ネットワーク上の場所にすべてのシステム情報を保存するには、次のように入力します。

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

tftp

構文

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

説明

TFTP サーバーからファイルをダウンロードします。

オプション

-i *interface*

eth0 などのアダプター名を指定します。

-l *port*

ローカルポート番号を指定します。デフォルト値は 0 で、ポート番号が自動的に割り当てられます。

-r *port*

リモートポート番号を指定します。デフォルト値は 69 です。

-c *retry count*

要求パケットを送信し、応答を待機する回数を指定します。デフォルト値は 6 です。

-t *timeout*

要求パケットを送信した後、応答を待機する秒数を指定します。デフォルト値は 4 です。

-s *block size*

RFC 2348 で定義されているように TFTP `blksize` オプションを指定します。有効な範囲は 8~65464、デフォルト値は 512 です。

host

TFTP サーバーの IPv4 アドレスを指定します。

remotefilepath

ダウンロードするファイルの TFTP サーバーのパスを指定します。

localfilepath

ローカルの宛先ファイルパスを指定します。

使用法

オプションの *localfilepath* が指定されている場合、ダウンロードしたファイルは、指定されたファイルパスを使用してローカルに保存されます。ローカルファイルパスが指定されていない場合、ファイルは、ファイルサーバーの名前を使用して、現在のディレクトリに格納されます。

このコマンドを使用する前に、ファイルの取得に使用する予定のネットワークインターフェイスを構成しなければなりません。これには `ifconfig` コマンドを使用することができます。

ネットワークインターフェイスが `-i` オプションで定義されている場合は、リモートファイルの取得にそのインターフェイスのみが使用されます。それ以外の場合は、すべてのネットワークインターフェイスがブートフェーズ中に検出される順序で試されます。

例

192.168.1.1 にある TFTP サーバーからファイル `dir1/file1.dat` を取得し、それを現在のディレクトリに `file2.dat` として格納するには、次のように入力します。

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

time

構文

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

説明

システムの現在の時刻を表示または設定します。

オプション

hh

新しい時の値を設定します (0~23)。

mm

新しい分の値を設定します (0~59)。

ss

新しい秒の値を設定します (0~59)。指定しないと、ゼロが使用されます。

`-tz tz`

GMT を基準にして分単位でタイムゾーンの調整を設定します。有効な値は、-1440~1440 の間、または 2047 です。値を設定しない場合、または 2047 に設定した場合は、現地時間として解釈されます。

`-d dl`

夏時間の値を設定します。次の値が有効です。

- ・ 0 - 時刻は夏時間の影響を受けません。
- ・ 1 - 時刻は夏時間の影響を受けませんが、調整されていません。
- ・ 3 - 時刻は夏時間の影響を受け、調整されています。

`-d` の後に値が続かない場合は、現在の夏時間が表示されます。

使用法

パラメーターを指定しない場合は、現在の時刻が表示されます。有効な時間、分、および秒を指定すると、システムの時刻がアップデートされます。

引数では、数字と：（コロン）文字以外の文字はすべて無効です。数字が時間/分/秒の正しい範囲内がない場合、シェルはエラーを報告します。数字の前後にスペースは使用できません。数字内へのスペースの挿入も許可されません。

seconds パラメーターを指定しない場合、秒はデフォルトでゼロに設定されます。

例

現在のシステム時刻を表示するには、次のように入力します。

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

現在のシステム時刻を設定するには、次のように入力します。

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

システムの時刻を設定し、夏時間の設定を表示するには、次のように入力します。

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

timezone

構文

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

説明

タイムゾーンの情報を表示または設定します。

オプション

-s

UTC を基準にして、hh:mm に関連付けられたタイムゾーンを設定します。

-l

すべてのタイムゾーンのリストを表示します。

-b

一度に 1 つの画面を表示します。

-f

指定したタイムゾーンの完全な情報を表示します。

使用法

パラメーターを指定しない場合は、現在のタイムゾーンが表示されます。有効な hh:mm パラメーターを指定すると、システムのタイムゾーン情報がアップデートされます。

例

使用可能なすべてのタイムゾーンを表示するには、次のように入力します。

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

タイムゾーンを設定するには、次のように入力します。

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

現在のタイムゾーンの詳細情報を表示するには、次のように入力します。

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

tlsconfig

構文

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

説明



TLS 接続の設定と証明書を表示および修正します。

オプション

-sfo

標準形式の出力で情報を表示します。

-f file

PEM または DER 形式のファイルからの X509 情報を表示します。

-r

TLS 構成をプラットフォームのデフォルトにリセットし、証明書を削除します。

-q

確認プロンプトの表示を停止します。

-e

X509 ファイルを TLS 証明書データベースに登録します。

-d

TLS 証明書を削除します。

-i

TLS 証明書を選択します。

-x

TLS 証明書をエクスポートします。

-t

エクスポートされた証明書のファイル形式を指定します。

-k

none が指定されている場合、すべての TLS 接続の証明書検証は無効です。peer が指定されている場合、すべての TLS 接続の証明書検証は有効です。



-o

`none` が指定されている場合、すべての TLS 接続の証明書検証について、ホスト名チェックは無効です。
`hostname` が指定されている場合、すべての TLS 接続の証明書検証について、ホスト名チェックは有効です。

-c

TLS 接続に許可されている暗号を指定または表示します。

-v

使用する TLS プロトコルバージョンを指定します。`auto` を選択すると、TLS サーバーとクライアントの両方でサポートされる最新バージョンがネゴシエートされます。

使用法

このコマンドは、HTTPS 機能 (HTTPS ブートなど) および UEFI シェルコマンド (`webclient` など) を使用するプリブート機能に影響を及ぼします。

例

すべての TLS 設定と証明書を表示します。

```
Shell> tlsconfig
```

X509 ファイルからの詳細な情報を表示するには、次のように入力します。

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

http サーバー上にある X509 ファイルからの詳細な情報を表示するには、次のように入力します。

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

X509 ファイルを TLS 証明書データベースに登録するには、次のように入力します。

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

ネットワーク上のファイルから TLS 証明書データベースに登録するには、次のように入力します。

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

確認プロンプトなしで最初の TLS 証明書を削除するには、次のように入力します。

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

PEM 形式で TLS 証明書を表示するには、次のように入力します。

```
Shell> tlsconfig -x
```

DER 形式で 3 番目の TLS 証明書を `abc.crt` という名前のファイルにエクスポートするには、次のように入力します。

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

DER 形式で 3 番目の TLS 証明書を `abc.crt` という名前のファイルにエクスポートするには、次のように入力します。

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

TLS 暗号アルゴリズムを指定するには、次のように入力します。

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

TLS でサポートされている暗号を表示するには、次のように入力します。

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

TLS 接続の証明書検証を有効にするには、次のように入力します。

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

TLS v1.2 プロトコルバージョンを使用することを指定するには、次のように入力します。

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

touch

構文

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

説明

ファイルの時刻と日付を現在の時刻と日付にアップデートします。

オプション

-r

サブディレクトリ内を再帰的にアップデートします。

file

アップデートするファイルまたはディレクトリの名前またはパターンを指定します。一度に複数のファイルをアップデートできます。

使用法

複数のファイルを指定した場合、システムはファイルを1つずつ処理し、エラーは無視されます。

このコマンドは、読み取り専用のファイルおよびディレクトリの時刻と日付は変更できません。

例

ファイルの時刻と日付をアップデートするには、次のように入力します。

```
fs0:\> touch myfile.txt
```

type

構文

```
type file[file...]
```

説明

ファイルの内容を標準出力デバイスに送信します。

オプション

file

表示するファイル名を指定します。

使用法

オプションを指定しない場合、このコマンドはファイルタイプの検出を試みます。検出できない場合は、UCS-2 と推定されます。

例

ファイルの形式を表示するには、次のように入力します。

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

```
pause
echo pause.nsh done.
```

複数のファイルを表示するには、次のように入力します。

```
fs0:\> type test.*
How to Install?
time
stall 3000000
time
```

unload

構文

```
unload[-n] [-v|verbose] handle
```

説明

メモリから UEFI ドライバーイメージをアンロードします。

オプション

-n

アンロード中は、スクリプトファイルで出力を使用できるように、すべてのプロンプトをスキップします。

-v

アンロードの前の詳細なイメージ情報を表示します。

verbose

アンロード前の詳細なイメージ情報をダンプします。

handle

アンロードするドライバーのハンドルを 16 進形式で指定します。

使用法

アンロードをサポートしているドライバーのみアンロードできます。

例

アンロードする UEFI ドライバーイメージのハンドルを検出するには、次のように入力します。

```
Shell> dh -b
```

ハンドル 27 の UEFI ドライバーイメージをアンロードするには、次のように入力します。

```
Shell> unload 27
```

ver

構文

```
ver[-s|-t]
```

説明

UEFI シェルと基になる UEFI ファームウェアのバージョン情報を表示します。

オプション

-s

UEFI シェルバージョンのみを表示します。

-t

要約された (terse) 内容を表示します。

使用法

このコマンドは、UEFI システムテーブルまたはシェルイメージから情報を取得します。

例

UEFI シェルバージョンのみを表示するには、次のように入力します。

```
fs0:\> ver -s 2.1
```

UEFI シェルファームウェアバージョンのすべての情報を表示するには、次のように入力します。

```
fs0:\> ver
```

vol

構文

```
vol[fs][-n volumelabel]
```

```
vol[fs][-d]
```

説明

ファイルシステムのボリューム情報を表示します。

オプション

fs

表示するファイルシステムの名前を指定します。

-n volumelabel

ボリュームラベルの名前を指定します。次の文字は使用できません。% ^ * + = [] | : ; " < > ? /。ボリュームラベルでスペースは使用できません。

-d

空のボリュームラベルを指定します。

使用法

fs を指定しない場合は、現在のファイルシステムが想定されます。-n を指定すると、*fs* のボリュームラベルが *volumelabel* パラメーターに設定されます。*volumelabel* は、最長 11 文字です。

例

現在のファイルシステムのボリュームを表示するには、次のように入力します。

```
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
```

fs0 のラベルを変更するには、次のように入力します。

```
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
```

fs0 のボリュームラベルを削除するには、次のように入力します。

```
fs0:\> vol fs0 -d
Volume has no label (rw)
1,457,664 bytes total disk space
220,160 bytes available on disk
512 bytes in each allocation unit
```

Webclient

構文

```
webclient -g URL [-u user [-x passwd]][-o file][[-m][-noproxy] webclient -p URL
[-u user [-x passwd]][-i file][-noproxy] webclient -l
```

説明



HTTP と FTP の場所との間でファイルを転送し、ISO ファイルシステムをマウントします。

オプション

-g

ドキュメントを取得します。

-p

ドキュメントをアップロードします。

URL

ドキュメントのハイパーテキストアドレスを指定します。

-o

指定したファイルに出力をリダイレクトします。

-i

指定したファイルからの入力をリダイレクトします。

file

ファイル名を指定します。

-m

ISO ファイルをダウンロードして、ファイルシステムとしてマウントします。

-l

使用されているネットワーク設定情報を表示します。

-u *user*

ユーザー名を指定します。

-x *passwd*

パスワードを指定します。

-noproxy

要求された操作にプロキシを使用しません。

使用法

- ① **重要:** `webclient` または `ftp` を同じネットワークインターフェイス上で実行する場合、そのネットワークインターフェイス上で `ifconfig` を使用する必要はありません。システムユーティリティで構成された **Pre-Boot Network Settings** によってこのインターフェイスと IP アドレスの設定が自動的に選択されるためです。

`ftp` と `webclient` で使用するインターフェイスを `ifconfig` で構成した場合、その設定は消去され、代わりに、コマンドの実行時にインターフェイス上でシステムユーティリティの**プリブートネットワーク設定メニュー**が適用されます。

このコマンドにより、ネットワーク転送をスクリプト化できます。このコマンドを使用する主な利点は、HTTP アドレスを使った URL を指定し、そのアドレスのドキュメントの取得、そのアドレスのファイルへのドキュメントの出力、ファイルのダウンロード、ISO ファイルのマウントができることです。ファイル転送をキャンセルするには、**ESC** キーまたは **Ctrl + C** キーを押します。FTP URL では、ユーザー名とパスワードをクリアテキストで URL に指定することで認証を行うことができます。FAT32 パーティションに保存できるファイルの最大サイズは 4 GB です。アップロードまたはダウンロードするための URL には、HTTP または FTP のいずれかを指定できます。HTTP URL は、IPv4/IPv6 アドレスまたはホスト名のいずれかを使用して指定することができます。FTP URL は、IPv4 アドレスまたはホスト名のいずれかを使用して指定することができます。FTP over IPv6 はサポートされていません。ダウンロード操作中に `-m` も `-o` も指定されていない場合、現在のファイルシステム上の、リモートファイルと同じ名前の新しいファイルに出力がリダイレクトされます。アップロード操作中に `-i` が指定されていない場合、現在のファイルシステム上の同じ名前のファイルから入力のリダイレクトされます。転送されているファイルサイズに相当するシステムの空きメモリ (RAM) があることを確認してください。プロキシを設定すると、すべてのダウンロード操作またはアップロード操作がプロキシ経由で試行されます。プロキシが設定されていない場合、または `-noproxy` が指定されている場合、すべてのダウンロード操作またはアップロード操作はプロキシなしで試行されます。`-l` が指定されている場合、Web クライアントがネットワーク操作に使用する設定が表示されます。これらの設定は、`sysconfig` を使用して内蔵 UEFI シェルで変更することも、システムユーティリティで変更することもできます。

例

ISO ファイルをダウンロードして、ISO ファイルシステムをマウントするには、次のように入力します。

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

または

```
fs0:\> webclient -g http://[1234:0001:0002:0003:0004:0005:0006:0007]/  
filename.iso -m
```

または

```
fs0:\> webclient -g http://[1234::1]/filename.iso -m
```

または

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

または

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

HTTP サーバーからファイルをダウンロードして現在のファイルシステムに保存するには、次のように入力します。

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

ユーザー名 `user` とパスワード `pass` を使用して FTP サーバーからファイルをダウンロードするには、次のように入力します。

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

ユーザー名 `user` とパスワード `pass` を使用して FTP サーバーにファイルをアップロードするには、次のように入力します。

```
ftp://192.168.1.20/file.html -u user -x pass -o file.html  
fs0:\> webclient -p
```

プロキシが設定されている場合に、外部 HTTP サーバーからファイルをダウンロードして現在のファイルシステムに保存するには、次のように入力します。

```
webclient -g http://www.hpe.com/file.html
```

プロキシが設定されている場合に、内部 HTTP サーバーからファイルをダウンロードして現在のファイルシステムに保存するには、次のように入力します。

```
-g http://192.168.1.20/file.html -noproxy
```

ネットワーク設定を表示するには、次のように入力します。

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

UEFI シェルスクリプトの実行と編集

ここでは、UEFI シェルでスクリプティング関数を実行および編集する方法について説明します。「Hello World」を表示するサンプルアプリケーションも提供されています。

スクリプトの起動方法

UEFI シェルスクリプトは、次の 2 つの方法のいずれかを使用して起動することができます。

- ・ システムユーティリティの UEFI Shell Script Auto-Start 構成の使用
- ・ シェルスクリプトの手動開始

システムユーティリティの UEFI シェルスクリプト自動起動構成

起動スクリプトを使用すると、RAM ディスクの作成、ネットワークからのファイルのダウンロード、データの収集、ネットワークへの結果のアップロードを行った後、システムを再起動せずに OS を起動できます。スクリプトファイルはローカルメディアに保存できます。また、ネットワーク上の場所からスクリプトファイルにアクセスすることもできます。

デフォルトでは、システムユーティリティで **UEFI シェルスクリプトの自動起動** が有効になっており、利用可能な任意の FAT16 または FAT32 ファイルシステムでシェルが `startup.nsh` ファイルを探すように構成されています。接続されているメディア上の特定のファイルシステムまたはネットワーク上の特定の場所でシェルが起動スクリプトを探すようにこれらの設定を変更できます。ネットワーク上の場所を探すように構成した場合、`startup.nsh` ファイルの場所として HTTP または FTP 形式の URL を指定できます。

シェルスクリプトの手動開始

手順

1. `.nsh` スクリプトファイルの場所に移動します。
2. 実行するスクリプト名を入力します。
または、スクリプトの場所に移動せずに、実行するスクリプトの絶対パスを入力します。

ファイルへの設定のエクスポートとインポート

手順

1. すべての設定をファイルにエクスポートするには、次のように入力します。

```
fs0:\> sysconfig -export filename.txt
```
2. すべての設定をファイルにインポートするには、次のように入力します。

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

シェルスクリプトの編集

スクリプトファイルは、`edit` コマンドを使用して、オフラインで、またはシェルで編集できます。`type` コマンドを使用して、スクリプトを画面に出力することもできます。

UEFI シェルスクリプトの例

以下の例では、アプリケーションソースコードのスクリプトサンプルと起動スクリプトを示します。

アプリケーションソースコードのスクリプト

次のサンプルソースコードは、画面に「Hello World」と出力し、UEFI シェルバージョンと環境変数を表示する UEFI シェルアプリケーションを実装する方法を示しています。このスクリプト例では、すべての FS*：ファイルシステム（FS0、FS1、FS2）をループして、特定の入力ファイル（この場合、sysconfig_backup.txt）を検索します。

エクスポート例：

```
fs0:\>
```

```
@echo -off
cls
set -v myfs 0
if exist FS0:\* then
FS0:
echo "FS0:\ Found!"
goto FSFOUND
endif
echo "FS0:\ not found in system"
echo "Going to search first available file system from FS1, FS2,..., FS100"
pause
for %a run (1 100)
set -v myfs %a
if exist FS%myfs%:\* then
FS%myfs%:
echo "FS%myfs%:\ Found!"
goto FSFOUND
endif
endfor
## No valid FS found in system, so exit now
echo "No valid File System (FS0, FS1,..., FS100) found in system"
goto END
:FSFOUND
if exist sysconfig_backup.txt then
echo =====
echo "%cwd%sysconfig_backup.txt already exists! Continuing the execution of the"
echo "script will remove existing sysconfig_backup.txt file and create a new"
echo "latest system configuration sysconfig_backup.txt file."
echo =====
pause
rm sysconfig_backup.txt
endif
echo "Saving latest system configuration in sysconfig_backup.txt file."
sysconfig -export sysconfig_backup.txt
:END
set -d myfs
```

インポート例：

```
fs0:\>
```

```
@echo -off
cls
```

```

set -v myfs 0
if exist FS0:\sysconfig_backup.txt then
FS0:
echo "FS0:\sysconfig_backup.txt Found!"
goto FSFOUND
endif
echo =====
echo "FS0:\sysconfig_backup.txt not found in system"
echo "Going to search sysconfig_backup.txt from all available file system"
echo "from FS1, FS2, ..., FS100"
echo =====
pause
for %a run (1 100)
set -v myfs %a
if exist FS%myfs%:\sysconfig_backup.txt then
FS%myfs%:
echo "FS%myfs%:\sysconfig_backup.txt Found!"
goto FSFOUND
endif
endfor
## No valid sysconfig_backup.txt found in system, so exit now
echo "No valid sysconfig_backup.txt found from File System (FS0, FS1, ..., FS100)"
goto END
:FSFOUND
if exist sysconfig_backup.txt then
echo =====
echo "%cwd%\sysconfig_backup.txt Found! Continuing the execution of the script"
echo "will cause system to override previous configuration and use the"
echo "configuration settings stored in sysconfig_backup.txt file."
echo =====
pause
sysconfig -import sysconfig_backup.txt
endif
:END
set -d myfs

```

起動スクリプト

指定されたネットワーク上の場所から内蔵 UEFI シェルが実行できる構成スクリプトの例を以下に示します。このスクリプトを使用して RAM ディスクを作成してから、ファイル出力のリダイレクトに使用する RAM ディスクの FS ファイルシステムを検索できます。

```

@echo -off

#
# Setup the environment variables. All of them are created as volatile.
#

#
# The volume label for the RAMDISK.
#
set -v VolumeLabel MYRAMDISK

#
# Variable to store the file system index that will be looped
# to determine the FS<x> number for the RAMDISK that is created.
#
set -v FsIndex 0

```

```

#
# Variable to store the output string of the ramdisk -c command.
# Successful creation of RAMDISK will give the following output:
# "RAM disk 'FSx:' created successfully." where x=0,1,2,...
#
set -v RamDiskStr 0

#
# Size of the RAMDISK in MegaBytes (MB).
#
set -v RamDiskSize 512

#
# Server URL hosting the OS loader and images.
# Can be HTTP or FTP. Names or IP addresses are allowed.
# Ensure DNS service is available and configured (see pre-requisites)
# when server names are used.
#
set -v Url http://192.168.1.1

#
# Files to be downloaded
#
set -v DownloadFile1 efilinux.efi
set -v DownloadFile2 deploy.kernel
set -v DownloadFile3 deploy.ramdisk

#
# Step 1. Create RAMDISK to store the downloaded OS programs.
#
echo "Creating a RAM Disk to save downloaded files..."
ramdisk -c -s %RamDiskSize% -v %VolumeLabel% -t F32 >v RamDiskStr
if %lasterror% ne 0x0 then
    echo "Cannot create a RAMDISK of size %RamDiskSize%."
    goto EXITSCRIPT
endif
echo "RAM Disk with Volume Label %VolumeLabel% created successfully."

#
# Step 2. Check each word in the output (RamDiskStr) and see if it matches
# the FSx: pattern. The newly created RAMDISK will be FS1: or higher.
# Here the check goes upto FS3: (the inner for loop), but a larger limit
# may be used in case many other file systems already exist before
# the creation of this RAMDISK. The FS for the RAMDISK is found when the
# FsIndex matches the FS<x> in RamDiskStr. Change the working directory
# to FS<FsIndex>:, so all downloads get saved there.
#
# FS0: is ignored. In the worst case, when no other usable
# file system is present, FS0: will map to the file system
# that this script is executing from.
#
#
for %a in %RamDiskStr%
    for %b run (1 10)
        set -v FsIndex %b
        if 'FS%FsIndex%:' == %a then
            FS%FsIndex%:
            goto RDFOUND
        endif
    endif

```

```

    endfor
endfor

#
# The following message appears if the newly created RAMDISK cannot be found.
#
echo "RAMDISK with Volume Label %VolumeLabel% not found!"
goto EXITSCRIPT

#
# The following message appears if the RAMDISK FS<x> has been found and you are in the
# RAMDISK's root folder.
#
:RDFOUND
echo "RAMDISK with Volume Label %VolumeLabel% found at FS%FsIndex%:."

#
# Step 3: Download the required files into the RAMDISK.
#
echo "Downloading %Url%/deploy/%DownloadFile1% (File 1 of 3...)"
webclient -g %Url%/deploy/%DownloadFile1% -o %DownloadFile1%
if %lasterror% ne 0x0 then
    goto EXITSCRIPT
endif

echo "Downloading %Url%/deploy/%DownloadFile2% (File 2 of 3...)"
webclient -g %Url%/deploy/%DownloadFile2% -o %DownloadFile2%
if %lasterror% ne 0x0 then
    goto EXITSCRIPT
endif

echo "Downloading %Url%/deploy/%DownloadFile3% (File 3 of 3...)"
webclient -g %Url%/deploy/%DownloadFile3% -o %DownloadFile3%
if %lasterror% ne 0x0 then
    goto EXITSCRIPT
endif

#
# Step4: Launch the boot loader.
#
echo "Starting the OS..."
%DownloadFile1% -f %DownloadFile2% initrd=%DownloadFile3%

#
# You reach here only if the downloads and booting failed.
#
:EXITSCRIPT
echo "Exiting Script."

```

このサンプルスクリプトは、以下の操作を行います。

手順

1. ダウンロードしたブートローダー、OS カーネル、ファイルシステムのほか、ブートローダーとカーネルの初期化に必要な構成ファイルを保管する一時 RAM ディスクを作成し、ネットワーク経由のインストールに進みます。
2. 新しく作成した RAM ディスクの FS<x> ID を判別します。

3. 作業ディレクトリを RAM ディスクのルートに設定します (例 : FS1:\)。
 4. OS の起動に必要なファイル (ブートローダー、OS カーネル、OS カーネルのメモリ上のファイルシステム) をダウンロードします。
 5. 次のいずれかを実行します。
 - a. 必要なすべてのファイルのダウンロードに失敗した場合、クリーンアップを実行し、起動スクリプトを終了します。
 - b. ダウンロードが成功した場合、ブートローダーを起動し、OS カーネルファイル、そのメモリ上のファイルシステム、および OS カーネルへのすべての引数 (ブートローダーがカーネルの起動時にカーネルに渡す必要がある引数) をコマンドライン引数としてブートローダーに渡します。
- UEFI シェルとプリブートスクリプトの役割はここで終了です。これで、OS は、メモリ上のファイルシステムに内蔵されている、OS 固有の展開スクリプトを使用して、自身で展開を行うことができます。

UEFI プログラミングモデル

UEFI シェルには、プログラミング API が用意されています。これを使用して、シェルのプログラム API またはプロトコルを呼び出すための独自の UEFI アプリケーションを記述することができます。詳しくは、UEFI Shell Specification および EDK2 を参照してください。UEFI シェルには、以下の表に示すプログラミング API が用意されています。EFI_SHELL_PROTOCOL は、UEFI アプリケーションにシェルサービスを提供します。これは、ファイル、パイプ、環境変数、現在の作業ディレクトリ、マッピング、ヘルプテキスト、エイリアス、起動シェルアプリケーションおよびスクリプトなど、低レベルのシェル関数への UEFI シェルアプリケーションアクセスを提供します。

表 9: UEFI アプリケーション API

関数タイプ	関数名	説明
EFI_SHELL_EXECUTE	Execute	シェルがコマンドラインを解析して実行します。
EFI_SHELL_GET_ENV	GetEnv	環境変数を取得します。
EFI_SHELL_SET_ENV	SetEnv	特定の環境変数を変更します。
EFI_SHELL_GET_ALIAS	GetAlias	特定のシェルコマンドのエイリアスを取得します。
EFI_SHELL_SET_ALIAS	SetAlias	特定のシェルコマンドのエイリアスを追加または削除します。
EFI_SHELL_GET_HELP_TEXT	GetHelpText	特定のコマンドに関するヘルプ情報を返します。
EFI_SHELL_GET_DEVICE_PATH_FROM_MAP	GetDevicePathFromMap	マッピングに対応するデバイスパスを返します。
EFI_SHELL_GET_MAP_FROM_DEVICE_PATH	GetMapFromDevicePath	特定のデバイスパスに対応するマッピングを返します。

表は続く



関数タイプ	関数名	説明
EFI_SHELL_GET_DEVICE_PATH_FROM_FILE_PATH	GetDevicePathFromFilePath	ファイルパスをデバイスパスに変換します。すべてのマッピングが、対応するデバイスパスに置き換えられます。
EFI_SHELL_GET_FILE_PATH_FROM_DEVICE_PATH	GetFilePathFromDevicePath	デバイスパスをファイルパスに変換します。デバイスパスのうちいずれかのマッピングに対応する部分が、そのマッピングに置き換えられます。
EFI_SHELL_SET_MAP	SetMap	デバイスとデバイスパス間のマッピングを作成、アップデート、または削除します。
EFI_SHELL_GET_CUR_DIR	GetCurDir	デバイスの現在のディレクトリを返します。
EFI_SHELL_SET_CUR_DIR	SetCurDir	デバイスの現在のディレクトリを変更します。
EFI_SHELL_OPEN_FILE_LIST	OpenFileList	指定したパスパターンに一致するファイルを開きます。
EFI_SHELL_FREE_FILE_LIST	FreeFileList	OpenFileList()で作成したファイルリストを解放します。
EFI_SHELL_REMOVE_DUP_IN_FILE_LIST	RemoveDupInFileList	指定されたファイルリスト内の重複ファイルを削除します。

表は続く

関数タイプ	関数名	説明
EFI_SHELL_BATCH_IS_ACTIVE	BatchIsActive	処理されているスクリプトファイルがあるかどうかを表示します。
EFI_SHELL_IS_ROOT_SHELL	IsRootShell	アクティブなシェルがルートシェルであるかどうかを確認します。
EFI_SHELL_ENABLE_PAGE_BREAK	EnablePageBreak	改ページ出力モードを有効にします。
EFI_SHELL_DISABLE_PAGE_BREAK	DisablePageBreak	改ページ出力モードを無効にします。
EFI_SHELL_GET_PAGE_BREAK	GetPageBreak	改ページ出力モードの有効化ステータスを取得します。
EFI_SHELL_GET_DEVICE_NAME	GetDeviceName	デバイスハンドルで指定されたデバイスの名前を取得します。
EFI_SHELL_GET_FILE_INFO	GetFileInfo	特定のファイルハンドルに関する情報を表示します。
EFI_SHELL_SET_FILE_INFO	SetFileInfo	特定のファイルハンドルに関する情報を変更します。
EFI_SHELL_OPEN_FILE_BY_NAME	OpenFileByName	名前に基づいてファイルを開き、ファイルハンドルを返します。
EFI_SHELL_CLOSE_FILE	CloseFile	開いているファイルを閉じます。
EFI_SHELL_CREATE_FILE	CreateFile	ファイルを作成します。

表は続く



関数タイプ	関数名	説明
EFI_SHELL_READ_FILE	ReadFile	データをファイルから読み取ります。
EFI_SHELL_WRITE_FILE	WriteFile	データをファイルに書き込みます。
EFI_SHELL_DELETE_FILE	DeleteFile	ファイルを削除します。
EFI_SHELL_DELETE_FILE_BY_NAME	DeleteFileName	ファイルを名前に基づいて削除します。
EFI_SHELL_GET_FILE_POSITION	GetFilePosition	ファイル内の現在の読み取り/書き込み位置を表示します。
EFI_SHELL_SET_FILE_POSITION	SetFilePosition	ファイル内の現在の読み取り/書き込み位置を変更します。
EFI_SHELL_FLUSH_FILE	FlushFile	すべてのバッファデータをファイルに書き込みます。
EFI_SHELL_FIND_FILES	FindFiles	ファイルリスト内のパターンに一致するすべてのファイルを表示します。
EFI_SHELL_FIND_FILES_IN_DIR	FindFilesInDir	ファイルリストで指定されたディレクトリ内のすべてのファイルを表示します。
EFI_SHELL_GET_FILE_SIZE	GetFileSize	ファイルのサイズを表示します。
EFI_SHELL_OPEN_ROOT	OpenRoot	ファイルシステムのルートディレクトリを表示します。

表は続く

関数タイプ	関数名	説明
EFI_SHELL_OPEN_ROOT_BY_HANDLE	OpenRootByHandle	特定のハンドル上のファイルシステムのルートディレクトリを表示します。
EFI_EVENT	ExecutionBreak	ユーザーが CTRL + C キーを押して現在の UEFI シェルコマンドの実行を中断するよう指示したときに、UEFI シェルから通知されるイベント。
UINT32	MajorVersion	シェル環境のメジャーバージョン。
UINT32	MinorVersion	シェル環境のマイナーバージョン。



UEFI シェルコマンドのステータスコード

以下の表に、コマンドを発行したときに UEFI シェルによって表示される可能性のあるステータスコードを示します。コードは、コマンドによって異なります。

表 10: UEFI シェルコマンドのステータスコード

ステータスコード	説明
SHELL_SUCCESS	操作が要求どおりに完了しました。
SHELL_NOT_FOUND	ターゲットのファイルまたはファイルセットが見つかりません。
SHELL_SECURITY_VIOLATION	セキュリティ違反のため、関数を実行できません。セキュアブートを有効にすると、いずれかの内蔵セキュアブート証明書を使用してデジタル署名されていない UEFI アプリケーションは実行できず、代わりに SECURITY_VIOLATION ステータスコードが返されます。
SHELL_INVALID_PARAMETER	渡されたいずれかのパラメーターの形式が正しくないか、その値が範囲外です。
SHELL_OUT_OF_RESOURCES	非揮発方式で変数を設定する要求を完了できません。結果として得られる非揮発性要求は、揮発性要求に変換されます。
SHELL_WRITE_PROTECTED	操作対象のメディアは書き込み禁止です。
SHELL_DEVICE_ERROR	ハードウェアエラーが発生したため、このコマンドを完了できません。



Web サイト

全般的な Web サイト

Hewlett Packard Enterprise Information Library

<http://www.hpe.com/info/EIL>

Single Point of Connectivity Knowledge (SPOCK) ストレージ互換性マトリックス RESTful インターフェイスツールのドキュメント

<http://www.hpe.com/storage/spock>

ストレージのホワイトペーパーおよびアナリストレポート

<http://www.hpe.com/storage/whitepapers>

UEFI の仕様

<http://www.uefi.org/specifications>

UEFI の学習資料

http://www.uefi.org/learning_center

RESTful API ツール

<http://www.hpe.com/info/redfish-ja>

Hewlett Packard Enterprise Worldwide の連絡先

<http://www.hpe.com/assistance>

サブスクリプションサービス/サポートのアラート

<http://www.hpe.com/support/e-updates-ja>

Software Depot

<http://www.hpe.com/support/softwaredepot>

カスタマーセルフリペア

<http://www.hpe.com/support/selfrepair>

Insight Remote Support

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

上記以外の Web サイトについては、[サポートと他のリソース](#)を参照してください。

サポートと他のリソース

Hewlett Packard Enterprise サポートへのアクセス

- ・ ライブアシスタンスについては、Contact Hewlett Packard Enterprise Worldwide の Web サイトにアクセスします。

<http://www.hpe.com/info/assistance>

- ・ ドキュメントとサポートサービスにアクセスするには、Hewlett Packard Enterprise サポートセンターの Web サイトにアクセスします。

<http://www.hpe.com/support/hpesc>

ご用意いただく情報

- ・ テクニカルサポートの登録番号（該当する場合）
- ・ 製品名、モデルまたはバージョン、シリアル番号
- ・ オペレーティングシステム名およびバージョン
- ・ ファームウェアバージョン
- ・ エラーメッセージ
- ・ 製品固有のレポートおよびログ
- ・ アドオン製品またはコンポーネント
- ・ 他社製品またはコンポーネント

アップデートへのアクセス

- ・ 一部のソフトウェア製品では、その製品のインターフェイスを介してソフトウェアアップデートにアクセスするためのメカニズムが提供されます。ご使用の製品のドキュメントで、ソフトウェアの推奨されるソフトウェアアップデート方法を確認してください。
- ・ 製品のアップデートをダウンロードするには、以下のいずれかにアクセスします。

Hewlett Packard Enterprise サポートセンター

<http://www.hpe.com/support/hpesc>

Hewlett Packard Enterprise サポートセンター：ソフトウェアのダウンロード

<http://www.hpe.com/support/downloads>

Software Depot

<http://www.hpe.com/support/softwaredepot>

- ・ eNewsletters およびアラートをサブスクライブするには、以下にアクセスします。

<http://www.hpe.com/support/e-updates-ja>

- ・ お客様の資格を表示、アップデート、または契約や保証をお客様のプロファイルにリンクするには、Hewlett Packard Enterprise サポートセンターの **More Information on Access to Support Materials** ページに移動します。

- ❶ **重要:** 一部のアップデートにアクセスするには、Hewlett Packard Enterprise サポートセンターからアクセスするときに製品資格が必要になる場合があります。関連する資格を使って HPE パスポートをセットアップしておく必要があります。

カスタマーセルフリペア (CSR)

Hewlett Packard Enterprise カスタマーセルフリペア (CSR) プログラムでは、ご使用の製品をお客様ご自身で修理することができます。CSR 部品を交換する必要がある場合、お客様のご都合のよいときに交換できるよう直接配送されます。一部の部品は CSR の対象になりません。Hewlett Packard Enterprise もしくはその正規保守代理店が、CSR によって修理可能かどうかを判断します。

リモートサポート (HPE 通報サービス)

リモートサポートは、保証またはサポート契約の一部としてサポートデバイスでご利用いただけます。リモートサポートは、インテリジェントなイベント診断を提供し、ハードウェアイベントを Hewlett Packard Enterprise に安全な方法で自動通知します。これにより、ご使用の製品のサービスレベルに基づいて、迅速かつ正確な解決が行われます。ご使用のデバイスをリモートサポートに登録することを強くおすすめします。

ご使用の製品にリモートサポートの追加詳細情報が含まれる場合は、検索を使用してその情報を見つけてください。

リモートサポートおよびプロアクティブケア情報

HPE 通報サービス

<http://www.hpe.com/jp/hpalert>

HPE プロアクティブケアサービス

<http://www.hpe.com/services/proactivecare-ja>

HPE プロアクティブケアサービス : サポートされている製品のリスト

<http://www.hpe.com/services/proactivecaresupportedproducts>

HPE プロアクティブケアアドバンスドサービス : サポートされている製品のリスト

<http://www.hpe.com/services/proactivecareadvancedsupportedproducts>

保証情報

ご使用の製品の保証に関する情報を表示するには、以下のリンクを参照してください。

HPE ProLiant と IA-32 サーバーおよびオプション

<http://www.hpe.com/support/ProLiantServers-Warranties>

HPE Enterprise および Cloudline サーバー

<http://www.hpe.com/support/EnterpriseServers-Warranties>

HPE ストレージ製品

<http://www.hpe.com/support/Storage-Warranties>

HPE ネットワーク製品

<http://www.hpe.com/support/Networking-Warranties>

規定に関する情報

安全、環境、および規定に関する情報については、Hewlett Packard Enterprise サポートセンターからサーバー、ストレージ、電源、ネットワーク、およびラック製品の安全と準拠に関する情報を参照してください。

<http://www.hpe.com/support/Safety-Compliance-EnterpriseProducts>

規定に関する追加情報

Hewlett Packard Enterprise は、REACH（欧州議会と欧州理事会の規則 EC No 1907/2006）のような法的な要求事項に準拠する必要に応じて、弊社製品の含有化学物質に関する情報をお客様に提供することに全力で取り組んでいます。この製品の含有化学物質情報レポートは、次を参照してください。

<http://www.hpe.com/info/reach>

RoHS、REACH を含む Hewlett Packard Enterprise 製品の環境と安全に関する情報と準拠のデータについては、次を参照してください。

<http://www.hpe.com/info/ecodata>

社内プログラム、製品のリサイクル、エネルギー効率などの Hewlett Packard Enterprise の環境に関する情報については、次を参照してください。

<http://www.hpe.com/info/environment>

ドキュメントに関するご意見、ご指摘

Hewlett Packard Enterprise では、お客様により良いドキュメントを提供するように努めています。ドキュメントを改善するために役立てさせていただきますので、何らかの誤り、提案、コメントなどがございましたら、ドキュメントフィードバック担当 (docsfeedback@hpe.com) へお寄せください。このメールには、ドキュメントのタイトル、部品番号、版数、およびドキュメントの表紙に記載されている刊行日をご記載ください。オンラインヘルプの内容に関するフィードバックの場合は、製品名、製品のバージョン、ヘルプの版数、およびご利用規約ページに記載されている刊行日もお知らせください。