HPE ProLiant m510 および m710x サーバーカートリッジ用UEFIシェルユーザーガイド

摘要
このガイドでは、すべてのUEFIベースProLiant m510およびm710xサーバーカートリッジのシステム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する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。
本書の内容は、将来予告なく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、商業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、および商業用製品の技術データ（Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items）は、ベンダー標準の商業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

Linux®は、Linus Torvalds氏の米国における登録商標です。

Intel®、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。

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。

UNIX®は、The Open Groupの登録商標です。

®はUEFIForum, Inc.の登録商標です。

本製品は、日本国内で使用するための仕様になっており、日本国外で使用される場合は、仕様の変更を必要とすることがあります。本書に掲載されている製品情報には、日本国内で販売されていないものも含まれている場合があります。
目次

1 はじめに.................................................................................................................................5
   コマンドと機能......................................................................................................................5
   [内蔵 UEFI シェル] へのアクセス.....................................................................................5
   シリアルコンソール接続から UEFI シェルへのアクセス.................................................6

2 UEFI シェルコマンドのリファレンス............................................................................7
   カスタム Hewlett Packard Enterprise シェルコマンド..................................................7
   コマンドライン構文の例........................................................................................................7
   コマンドラインの補完............................................................................................................7
   UEFI シェルの概要................................................................................................................7
      コマンド出力の制御..........................................................................................................8
         出力を一度に 1 つの画面に制限.....................................................................................8
         グローバル改ページを有効または無効にする.................................................................9
         詳細出力の表示.................................................................................................................9
         概要出力の表示................................................................................................................9
         確認プロンプトの抑制.......................................................................................................10
      コマンドヘルプの表示......................................................................................................10
         すべてのコマンドヘルプを表示する..............................................................................10
         特定のコマンドのヘルプの表示.....................................................................................10
         特定の文字で始まるコマンドのすべてのヘルプの表示..................................................11
      シェルからファイルシステムへの切り替え.....................................................................13
      共通のセットアップコマンドと構成コマンドの使用.........................................................13
   UEFI シェルコマンド...........................................................................................................14
      alias.................................................................................................................................14
      attrib...............................................................................................................................15
      boot.................................................................................................................................16
      cd......................................................................................................................................17
      cls.....................................................................................................................................18
      comp...............................................................................................................................19
      compress.........................................................................................................................20
      connect..........................................................................................................................21
      cp......................................................................................................................................22
      date...................................................................................................................................24
      dblk...................................................................................................................................25
      devices............................................................................................................................25
      devtree.............................................................................................................................26
      dh......................................................................................................................................27
      disconnect.....................................................................................................................29
      dmem...............................................................................................................................30
      drivers.............................................................................................................................30
      echo..................................................................................................................................31
      edit....................................................................................................................................32
      eficompress.....................................................................................................................33
      efidecompress..................................................................................................................33
      exit.....................................................................................................................................33
      ftp.....................................................................................................................................34
      getmtc.............................................................................................................................35
      goto..................................................................................................................................35
      help..................................................................................................................................35
      ifconfig...........................................................................................................................36
      load...................................................................................................................................37
      ls/dir.................................................................................................................................38
3 UEFI シェルスクリプトの実行と編集.........................................................................................68
   スクリプトの起動方法........................................................................................................68
   [システムユーティリティ]の[UFEI シェルスクリプト自動起動]構成........................................68
   シェルスクリプトの手動開始...............................................................................................68
   ファイルへの設定のエクスポートとインポート.....................................................................68
   シェルスクリプトの編集......................................................................................................68
   UEFI シェルスクリプトの例.................................................................................................68
   アプリケーションソースコードのスクリプト.......................................................................68
   起動スクリプト..................................................................................................................70

4 UEFI プログラミングモデル...............................................................................................73

5 UEFI シェルコマンドのステータスコード..........................................................................75

6 サポートと他のリソース.......................................................................................................76
   Hewlett Packard Enterprise サポートへのアクセス.............................................................76
   アップデートへのアクセス.................................................................................................76
   関連情報.............................................................................................................................77
   Web サイト.........................................................................................................................77
   カスタマーセルフリペア......................................................................................................77
   リモートサポート（HPE 通報サービス）..............................................................................77

用語集.....................................................................................................................................79

索引..........................................................................................................................................80
はじめに
すべての ProLiant m510 および m710x サーバーカートリッジのシステム BIOS には、ROM 内蔵された UEFI シェルが含まれます。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 シェルでは以下のコマンドと機能を使用できます。

- 構成コマンド:
  - BIOS 構成（sysconfig）
  - ROM、ストレージ
- スクリプティング:
  - if、else、endif、shift、および for/endfor 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、alias、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
前提条件
1. 内蔵 UEFI シェルが有効になっていることを確認する。デフォルトの設定は有効です。
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する。
4. [UEFI ブート順序] リストの [内蔵 UEFI シェル] エントリーを変更する。
詳細は、『HPE ProLiant m510 および m710x サーバーカートリッジ用 UEFI システムユーティリティユーザーガイド』の「内蔵 UEFI シェルオプション」を参照してください。

シェルにアクセスするには、以下の手順に従ってください。
次のいずれかの方法を使用します。
- サーバーの POST 処理中に、ProLiant POST 画面で [F11] キー ([ブートメニューコ) を押す。
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。詳しくは、『HPE ProLiant m510およびm710xサーバーカートリッジ用UEFIシステムユーティリティユーザーガイド』の「内蔵UEFIシェルの起動」を参照してください。

シナリオコンソール接続を使用する。シナリオコンソール接続からUEFIシェルへのアクセスを参照してください。

サーバーのシリアルポートを使用する。詳しくは、『HPE ProLiant m510およびm710xサーバーカートリッジ用UEFIシステムユーティリティユーザーガイド』の「シリアルポートオプション」を参照してください。

シナリオ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（UTF8など）を使用していることが必要です。

注記：シナリオ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。

シェルにアクセスするには、以下の手順に従ってください。
1. サーバーを起動します。
2. サーバーの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。UEFIシェルで使用するコマンドについて詳しくは、UEFIシェルコマンドのリファレンスを参照してください。
7. exitを入力し、シェルを終了します。

例1 シナリオコンソール接続からログインする

```
login as: username

@<ip address>'s password: password

User: logged-in to <server path> / <server MAC address>
iLO 4 Standard 2.40 at July 16 2016
Server Name:
Server Power: On

hpiLO-> vsp

Virtual Serial Port Active: COM2

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

Shell>
2 UEFI 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法の例を示します。

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。
• PgUp — ページを上にスクロールします。
• PgDown — ページを下にスクロールします。

UEFI シェルの概要

以下では、基本的な UEFI シェルコマンドを使用して、次の操作を行う方法について説明します。

• 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。以下の各項目では、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。

- 出力を一度に1つの画面に制限
- 詳細出力の表示
- 確認プロンプトの抑制

出力を一度に1つの画面に制限
次の例は、-bオプションを使用して、一度に1つの画面に出力を制限する方法を示しています。最初の画面が表示された後、以下のいずれかのシェルプロンプトが表示されます。

- 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 2 -b を使用して一度に1つの画面に出力を表示する

Shell> devtree -b
Ctrl[03] Fv(6522280D-28F9-4131-ADC4-F40EBFA45864)
Ctrl[04] Fv(770BF9B6-8AFE-4F4C-85E5-893FC3D2606C)
Ctrl[05] Fv(27A72E80-3118-4CO5-8673-AA5B4EFA9613)
Ctrl[06] MemoryMap(0xB,0xFFD40000,0xFFD6FFFF)
Ctrl[07] Fv(5A515240-D1F1-4C58-9590-27B1F0E86827)
Ctrl[08] Fv(5E2363B4-3E9E-4203-B873-BB40DF46C8E6)
Ctrl[09] Fv(5E2363B4-3E9E-4203-B873-BB40DF46C8E6)
Ctrl[0F] 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:

help コマンドでこのオプションを使用する方法の例は、例4 (10ページ) を参照してください。
グローバル改ページを有効または無効にする
pagebreak コマンドを使用して、出力のグローバル改ページを有効または無効にします。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。

例
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
-v オプションを使用して、特定のコマンドの詳細（verbose）出力を表示します。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

次の例に、詳細な dh (device handle：デバイスハンドル) 出力を一度に1つずつ画面に表示する方法を示します。

例 3 -v を使用して詳細出力を表示する

```
Shell> dh -v -b
```

```
01: LoadedImage
02: Decompress
03: UnknownDevice DevicePath Fv(6522280D-28F9-4131-AD4F-40EBFA45864) UnknownDevice
04: UnknownDevice DevicePath Fv(77DB9BE6-8AFE-4F4C-85E5-893FC3D2606C) UnknownDevice
05: UnknownDevice DevicePath Fv(27A72E80-3118-4C0C-8673-AA5B4EFA9613) UnknownDevice
06: UnknownDevice DevicePath Fv(29A72E80-7BBF-4101-8459-AB5B3EFA4271) UnknownDevice
07: UnknownDevice DevicePath Fv(5A515240-D1F1-4C58-9590-27B1F0EB827) UnknownDevice
08: UnknownDevice DevicePath Fv(5E2363B4-3E9E-4203-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
-t オプションを使用して、特定のコマンドの概要（terse）出力を表示します。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。

- ver
確認プロンプトの抑制
- `q` オプションを使用して、特定のコマンドをクワイエットモードで、つまり確認プロンプト
なしで実行します。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`

コマンドヘルプの表示
1つ以上のコマンドの詳細なヘルプおよび要約したヘルプを表示するさまざまなコマンドオプションがあります。

- すべてのコマンドヘルプを表示する
- 特定のコマンドのヘルプの表示
- 特定の文字で始まるコマンドのすべてのヘルプの表示

すべてのコマンドヘルプを表示する
次の例に、すべてのコマンドのヘルプを一度に1つの画面に表示する方法を示します。

例4 すべてのコマンドヘルプを一度に1つの画面に表示する

```bash
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.
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.
devtree Displays the list of devices managed by UEFI drivers.
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.
```

特定のコマンドのヘルプの表示
次のいずれかの構文オプションを使用して、特定のコマンドヘルプを表示します。

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

次の例に、`ls` コマンドのヘルプを表示する1つの方法を示します（一度に1つの画面）。

例 5 ls コマンドのヘルプを表示する

Shell> help ls -b

Lists a directory's contents or file information.

LS [-r] [-a[attrib]] [-sfo] [file] format:
  -r - Displays recursively (including subdirectories)
  -a - Display only those files with the attributes of type attrib. If no
       attributes are listed, all files will be listed. If -a is not
       specified, all non-system and non-hidden files will be listed.
  -sfo - Display information in Standard-Format Output.
  attrib - File attribute list:
       a - Archive
       s - System
       h - Hidden
       r - Read-only
       d - Directory
  file - Name of file or directory (wildcards are permitted)

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

help コマンドの使用法について詳しくは、「help」(35 ページ)を参照してください。

特定の文字で始まるコマンドのすべてのヘルプの表示

ワイルドカード（*）を使用して、特定の文字で始まるすべてのコマンドのヘルプを表示します。次の例に、文字 a で始まるすべてのコマンドでこれを行う方法を示します。
Displays, creates, or deletes UEFI Shell aliases.

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

```
fs0:\> attrib -r *.inf
-d   - Delete an alias. command-name must not be specified.
-v   - Make the alias volatile.
alias-name - Alias name
command-name - Original command's name or path.
```

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

EXAMPLES:
* To display all aliases in the UEFI Shell environment:
  Shell> alias

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

* To delete an alias in the UEFI Shell environment:
  Shell> alias -d shutdown

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

Displays or changes the attributes of files or directories.

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

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>+a</td>
<td>-a</td>
</tr>
<tr>
<td>+s</td>
<td>-s</td>
</tr>
<tr>
<td>+h</td>
<td>h</td>
</tr>
<tr>
<td>+r</td>
<td>r</td>
</tr>
<tr>
<td>file</td>
<td>File name (wildcards are permitted)</td>
</tr>
<tr>
<td>directory</td>
<td>Directory name (wildcards are permitted)</td>
</tr>
</tbody>
</table>

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

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。
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。mapを参照してください。
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。
次の例に、シェルから fs0 ファイルシステムにアクセスする方法を示します。

例 7 シェルから fs0 ファイルシステムにアクセスする

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

注記: このガイドの出力例では、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
1 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。
2 サーバーに UEFI ブートオプションがあるかどうかを確認するにはどうすれば良いですか。
sysconfigを参照してください。
3 タイムゾーンを構成するにはどうすれば良いですか。
4 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。
5 ユーザー定義名とデバイスハンドル間の1つまたはすべてのマッピングをリセットするにはどうすれば良いですか。
mapを参照してください。
6 UEFI 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どうすれば良いですか。
devtreeを参照してください。
7 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
すべてのコマンドで、BIOS 管理者権限が必要です。管理者パスワードの設定については、「HPE ProLiant m510 および m710x サーバーカートリッジ用 UEFI システムユーティリティユーザーガイド」の「管理者パスワードの設定」を参照してください。
すべてのコマンドで、BIOS 管理者権限が必要です。管理者パスワードの設定については、sysconfigを参照してください。

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    コマンドの使用方法の簡単な説明
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：ユーザーアイテルは、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）

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。

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。
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

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

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、次回の起動時に消えます。

Shell> alias -v 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けるします。
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。
```bash
fs0:/> attrib 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。
```bash
fs0:/> attrib *
```
```bash
\attrib: AS fs0:\serial.\efi
attrib: DA fs0:\test\l
attrib: A HR fs0:\bios.inf
attrib: A fs0:\VerboseHelp.txt
attrib: AS fs0:\\isaBus.\efi
```
```bash
system属性を拡張子 .efi を持つすべてのファイルに有効にするには、次のように入力します。
```
```bash
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。
```bash
fs0:/> attrib -r *.inf
```
```bash
\attrib: A H fs0:\bios.inf
```

boot

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

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。
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。
- 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 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
```

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

<table>
<thead>
<tr>
<th>表記規則</th>
<th>説明</th>
</tr>
</thead>
<tbody>
<tr>
<td>.</td>
<td>現在のディレクトリ。</td>
</tr>
<tr>
<td>..</td>
<td>現在のディレクトリの親。</td>
</tr>
<tr>
<td>\</td>
<td>現在のファイルシステムのルート。</td>
</tr>
</tbody>
</table>
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。

\texttt{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。

\texttt{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。

\texttt{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。

\texttt{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。

\texttt{fs0:\efi\tools/> cd \}

\texttt{fs0:/>}

注記: cd を使用したボリュームの変更は機能しません。次に例を示します。

\texttt{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。

\texttt{fs0:\/> cd \efi\tools}

\texttt{fs0:\efi\tools/> fs1:}

\texttt{fs1:\/> cd tmp}

\texttt{fs1:\tmp/> cp fs0:*.* .}

これにより、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。

\texttt{cls}

構文
\texttt{cls[color]}

説明
標準出力をクリアし、背景色を変更します。

オプション

\texttt{color} 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
使用法
color を指定しない場合、背景色は変更されません。

例
背景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。
\texttt{fs0:}\texttt{> 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。
\texttt{fs0:}\texttt{> cls 3}
標準出力をクリアし、背景をデフォルトの色に変更するには、次のように入力します。
\texttt{fs0:}\texttt{> cls 0}

comp

構文
\texttt{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
- \texttt{-b} 一度に 1 つの画面を表示します。
- \texttt{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。
- \texttt{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。
\texttt{fs0:}\texttt{> 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。
\texttt{fs0:}\texttt{> 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。
\texttt{fs0:}\texttt{> 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
 Difference #2:
File1: fs0:\bios.inf
File2: fs0:\bios2.inf
[difference(s) encountered]

compress

構文
compress[-z][-s][-ex][-cl 0-9][-o output_file][file...][-sfo]
compress[-u][-s][-ex][-p password][-od output_directory][-i input_file][-sfo]

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 zipファイルにファイルを追加したり、zipファイルからファイルを抽出したりするときに、ファイル名からパスを除外します。
-cl 0-9 圧縮レベルを選択します。0:圧縮時間が短縮されます。9:(デフォルト)圧縮率が高まります。
-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ファイルからファイルを抽出する際の抽出元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す。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ハンドルまたは内蔵ドライバがあるかどうかが確認されます。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や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や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やもバインドされていません。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、そのハンドルが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をバインドを試行し、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コピー先と見なされます。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 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)
```
**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）。
- **yy** 2桁の年を指定します。
- **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、システムの日付が更新されます。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。次に例を示します。
  
  Shell> date 8/4/97

  Shell> date 8/04/2097
  Shell> date 08/04/1998

  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。
**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のみが表示されます。

**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。

```bash
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。

```bash
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。

```bash
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。

```bash
Shell> devices
```

<table>
<thead>
<tr>
<th>C</th>
<th>T</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>Y</td>
<td>C</td>
</tr>
<tr>
<td>R</td>
<td>P</td>
<td>F</td>
</tr>
<tr>
<td>L</td>
<td>E</td>
<td>G</td>
</tr>
<tr>
<td>#P</td>
<td>#D</td>
<td>#C</td>
</tr>
</tbody>
</table>

Device Name

---

20 R - - - 1 13 VenHw(58C518B1-76F3-11D4-BCEA-0080C73C8881)

3D D - - 3 - Primary Console Input Device
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

<table>
<thead>
<tr>
<th>列</th>
<th>説明</th>
</tr>
</thead>
<tbody>
<tr>
<td>CTRL</td>
<td>デバイスのハンドル番号。</td>
</tr>
<tr>
<td>TYPE</td>
<td>デバイスタイプ。オプションは次のとおりです。</td>
</tr>
<tr>
<td></td>
<td>• R - ルートコントローラー</td>
</tr>
<tr>
<td></td>
<td>• B - バスコントローラー</td>
</tr>
<tr>
<td></td>
<td>• D - デバイスコントローラー</td>
</tr>
<tr>
<td>CFG</td>
<td>構成プロトコルのサポートステータス：</td>
</tr>
<tr>
<td></td>
<td>• Y - はい</td>
</tr>
<tr>
<td></td>
<td>• N - いいえ</td>
</tr>
<tr>
<td>DIAG</td>
<td>診断プロトコルのサポートステータス：</td>
</tr>
<tr>
<td></td>
<td>• Y - はい</td>
</tr>
<tr>
<td></td>
<td>• N - いいえ</td>
</tr>
<tr>
<td>#P</td>
<td>このデバイスの親コントローラーの数。</td>
</tr>
<tr>
<td>#D</td>
<td>このタイプのデバイスの数。</td>
</tr>
<tr>
<td>#C</td>
<td>このデバイスで生成された子コントローラーの数。</td>
</tr>
<tr>
<td>Device Name</td>
<td>コ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。</td>
</tr>
</tbody>
</table>

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。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。
- `-d` 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。
- `-lxxx` 指定したハンドルの下にデバイスツリーを表示します。

devtree
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: \nCtrl[04] Fv(770BF9B6-8AFE-4F4C-85E5-893FC3D2606C) \nCtrl[05] Fv(27A72E80-3118-4C0C-8673-AA5B4EFA9613)-directories in the current \nCtrl[06] MemoryMapped(0xB,0xFFD40000,0xFFD6FFFF) \nCtrl[07] Fv(5A515240-D1F1-4C56-9590-27B1F0E86827) \nCtrl[08] Fv(5E2363B4-3E9E-4203-B873-BA40DF46C8E6) \nCtrl[09] Fv(CDBB7B35-6833-4ED6-9AB2-57D2AC0DF6F0)extension '.efi': \nCtrl[0F] PciRoot(0x0).efi \nCtrl[195] PciRoot(0x0)/Pci(0x0,0x0) \nCtrl[196] PciRoot(0x0)/Pci(0x2,0x0)from all files with extension '.inf': \nCtrl[197] PciRoot(0x0)/Pci(0x2,0x0)/Pci(0x0,0x0) \nCtrl[198] PciRoot(0x0)/Pci(0x2,0x1) \nCtrl[199] PciRoot(0x0)/Pci(0x2,0x2) \nCtrl[19A] PciRoot(0x0)/Pci(0x2,0x3) \nCtrl[19B] PciRoot(0x0)/Pci(0x2,0x4) \nCtrl[19C] PciRoot(0x0)/Pci(0x2,0x5) \nCtrl[19D] PciRoot(0x0)/Pci(0x2,0x6) \nCtrl[19E] PciRoot(0x0)/Pci(0x2,0x7) \nCtrl[19F] PciRoot(0x0)/Pci(0x4,0x0) \nCtrl[1A0] PciRoot(0x0)/Pci(0x4,0x1) \nCtrl[1A1] PciRoot(0x0)/Pci(0x4,0x2) \nCtrl[1A2] PciRoot(0x0)/Pci(0x4,0x3) \nCtrl[1A3] PciRoot(0x0)/Pci(0x4,0x4) \nCtrl[1A4] PciRoot(0x0)/Pci(0x4,0x5) \nCtrl[1A5] PciRoot(0x0)/Pci(0x4,0x6) \nCtrl[1A6] PciRoot(0x0)/Pci(0x4,0x7) \nCtrl[1A7] PciRoot(0x0)/Pci(0x5,0x0) \nCtrl[1A8] PciRoot(0x0)/Pci(0x5,0x1) \nCtrl[1A9] PciRoot(0x0)/Pci(0x5,0x2) \nCtrl[1AA] PciRoot(0x0)/Pci(0x5,0x4) \nCtrl[1AB] PciRoot(0x0)/Pci(0x6,0x7) \nPress 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
-1xxx 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。
標準形式の出力で情報を表示します。表 5 (40 ページ) を参照してください。

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。-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(.i/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
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。

<table>
<thead>
<tr>
<th>列</th>
<th>説明</th>
</tr>
</thead>
<tbody>
<tr>
<td>Driver Name</td>
<td>ハンドルを生成するドライバーの名前。</td>
</tr>
<tr>
<td>Controller Name</td>
<td>ハンドルを生成するコントローラーの名前。</td>
</tr>
<tr>
<td>Handle number</td>
<td>ハンドルの整数の番号。</td>
</tr>
<tr>
<td>Device Path</td>
<td>ハンドルに関連付けられたデバイスパス。</td>
</tr>
<tr>
<td>Protocol Identifiers</td>
<td>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。</td>
</tr>
</tbody>
</table>

disconnect

構文
disconnect devicehandle [driverhandle][childhandle][\-r]

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ます。

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するには、次のように入力します。
Shell> disconnect -r
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 パイトになります。-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 パ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 *......*

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。表 4 (31 ページ) を参照してください。

使用法
一覧表示の内容の説明については、表 4 (31 ページ) を参照してください。

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。
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

<table>
<thead>
<tr>
<th>列</th>
<th>説明</th>
</tr>
</thead>
<tbody>
<tr>
<td>DRV</td>
<td>ドライバーの整数ハンドル。</td>
</tr>
<tr>
<td>VERSION</td>
<td>ドライバーのバージョン番号。</td>
</tr>
<tr>
<td>TYPE</td>
<td>ドライバーの種類。設定できる値は、次のとおりです。</td>
</tr>
<tr>
<td></td>
<td>• B - バスドライバー</td>
</tr>
<tr>
<td></td>
<td>• D - デバイスドライバー</td>
</tr>
<tr>
<td>CFG</td>
<td>構成プロトコルのサポートステータス：</td>
</tr>
<tr>
<td></td>
<td>• Y - はい</td>
</tr>
<tr>
<td></td>
<td>• N - いいえ</td>
</tr>
<tr>
<td>DIAG</td>
<td>ドライバー・プロトコルのサポートステータス：</td>
</tr>
<tr>
<td></td>
<td>• Y - はい</td>
</tr>
<tr>
<td></td>
<td>• N - いいえ</td>
</tr>
<tr>
<td>#D</td>
<td>このドライバーが管理しているデバイスの数。</td>
</tr>
<tr>
<td>#C</td>
<td>このドライバーが生成した子デバイスの数。</td>
</tr>
<tr>
<td>DRIVER NAME</td>
<td>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。</td>
</tr>
<tr>
<td>IMAGE NAME</td>
<td>ドライバーのロード元のデバイスパス。</td>
</tr>
</tbody>
</table>

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の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とき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ときに表示するには、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 壓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 壓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:\> eficompress uncompressed compressed

efidecompress

構文
efidecompress infile outfile

説明
EFI 壓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 壓縮された入力ファイルのファイル名を指定します。
outfile 壓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:\> efidecompress 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、コラー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。

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

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。

goto

構文
goto label

説明
スクリプト内のラベルに移動します。

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

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>cmd</td>
<td>ヘルプを表示するコマンドを指定します。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。</td>
</tr>
<tr>
<td>pattern</td>
<td>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。</td>
</tr>
<tr>
<td>special</td>
<td>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。</td>
</tr>
<tr>
<td>-usage</td>
<td>冗長情報を表示します。</td>
</tr>
<tr>
<td>-v</td>
<td>ヘルプ情報の指定したセクションを表示します。</td>
</tr>
<tr>
<td>-section sectionname</td>
<td>ヘルプ出力を一度に1つの画面に表示します。</td>
</tr>
<tr>
<td>-b</td>
<td></td>
</tr>
</tbody>
</table>
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（下を参照）。
-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の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。command-name が、シェルコマンドの名前です。ファイルは、以下に示すように、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[-c [name]][-l[name]]
ifconfig[-s name dhcp][[static IPaddress mask gateway]|permanent]

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
-c 構成をクリアします。
-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 DHCP4 が動的にすべてまたは特定のインタフェースの IPv4 アドレスを要求するように指定します。
static IPaddress 静的 IPv4 アドレスを、それぞれが 0 〜 255 の範囲のビ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 の範囲のビリオドで区切られた 4 つの整数値で指定します。
デフォルトゲートウェイを、それぞれが 0〜255 の範囲のビリオドで区切られた 4 つの整数値で指定します。
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

① 重要： webclient または ftp を同じネットワークインテルフェイス上で実行する場合、そのネットワークインテルフェイス上で 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フェースと 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 で使用するインタフェースを 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フェー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フェースの構成をクリアします。そのため、関連するインタフェー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。permanent を指定しない場合、構成は 1 回限りです。permanent を指定した場合、構成はネットワークスタックが再ロードされても維持されます。

例

Eth0 インタフェースの構成を一覧表示するには、次のように入力します。

```
fs0:\>ifconfig -l eth0
``` DHCP4 を使用して eth0 インタフェースの 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。

```
fs0:\>ifconfig -s eth0 dhcp
``` eth0 インタフェースに対して静的 IPv4 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設定するには、次のように入力します。

```
fs0:\>ifconfig -s eth0 static 192.168.0.5 255.255.255.0 192.168.0.1 permanent
```

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。

t 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。出力の詳細 -ls コマンド（ボリューム情報）および出力の詳細 -ls コマンド（ファイル情報）を参照してください。
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 フラグは、再帰検索を実行するかどうかを指定します。
- オプションのフラグ -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 の後に何も続く場合は、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
```

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
```

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:\

<table>
<thead>
<tr>
<th>Data</th>
<th>Date/Time</th>
<th>Size</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>for.nsh</td>
<td>06/18/01 09:32p</td>
<td>153</td>
<td></td>
</tr>
<tr>
<td>efi</td>
<td>06/18/01 01:02p &lt;DIR&gt;</td>
<td>512</td>
<td></td>
</tr>
<tr>
<td>test1</td>
<td>06/18/01 01:02p &lt;DIR&gt;</td>
<td>512</td>
<td></td>
</tr>
<tr>
<td>test2</td>
<td>06/18/01 01:02p &lt;DIR&gt;</td>
<td>512</td>
<td></td>
</tr>
<tr>
<td>temp.txt</td>
<td>06/18/01 08:04p</td>
<td>29</td>
<td></td>
</tr>
<tr>
<td>test</td>
<td>06/18/01 08:05p &lt;DIR&gt;</td>
<td>512</td>
<td></td>
</tr>
<tr>
<td>readme.txt</td>
<td>01/28/01 08:24p r</td>
<td>29</td>
<td></td>
</tr>
</tbody>
</table>

```
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
```

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:\

<table>
<thead>
<tr>
<th>Data</th>
<th>Date/Time</th>
<th>Size</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>for.nsh</td>
<td>06/18/01 09:32p</td>
<td>153</td>
<td></td>
</tr>
<tr>
<td>efi</td>
<td>06/18/01 01:02p &lt;DIR&gt;</td>
<td>512</td>
<td></td>
</tr>
<tr>
<td>test1</td>
<td>06/18/01 01:02p &lt;DIR&gt;</td>
<td>512</td>
<td></td>
</tr>
<tr>
<td>test2</td>
<td>06/18/01 01:02p &lt;DIR&gt;</td>
<td>512</td>
<td></td>
</tr>
<tr>
<td>IsaBus.efi</td>
<td>06/18/01 10:59p</td>
<td>28,739</td>
<td></td>
</tr>
<tr>
<td>IsaSerial.efi</td>
<td>06/18/01 10:59p</td>
<td>32,838</td>
<td></td>
</tr>
<tr>
<td>temp.txt</td>
<td>06/18/01 08:04p</td>
<td>29</td>
<td></td>
</tr>
<tr>
<td>test</td>
<td>06/18/01 08:05p &lt;DIR&gt;</td>
<td>512</td>
<td></td>
</tr>
<tr>
<td>readme.txt</td>
<td>01/28/01 08:24p r</td>
<td>29</td>
<td></td>
</tr>
</tbody>
</table>

```
5 File(s) 61,788 bytes
4 Dir(s)
```
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）

<table>
<thead>
<tr>
<th>列</th>
<th>説明</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>標準のボリュームラベル。</td>
</tr>
<tr>
<td>Total Size</td>
<td>ボリューム内の合計バイト数。</td>
</tr>
<tr>
<td>Read Only status</td>
<td>読み取り専用ステータス：</td>
</tr>
<tr>
<td></td>
<td>• True</td>
</tr>
<tr>
<td></td>
<td>• False</td>
</tr>
<tr>
<td>Free Space</td>
<td>ボリューム内の空きバイトの合計数。</td>
</tr>
<tr>
<td>Block Size</td>
<td>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）。</td>
</tr>
</tbody>
</table>

### 表 6 出力の詳細 - ls コマンド（ファイル情報）

<table>
<thead>
<tr>
<th>列</th>
<th>説明</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>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</td>
</tr>
<tr>
<td>Logical Size</td>
<td>ファイルのサイズ（バイト単位）。</td>
</tr>
<tr>
<td>Physical Size</td>
<td>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）。</td>
</tr>
<tr>
<td>属性</td>
<td>ファイル属性のリスト。設定できる値は、次のとおりです。</td>
</tr>
<tr>
<td></td>
<td>• a - アーカイブ</td>
</tr>
<tr>
<td></td>
<td>• d - ディレクトリ</td>
</tr>
<tr>
<td></td>
<td>• h - 非表示</td>
</tr>
<tr>
<td></td>
<td>• r - 読み取り専用</td>
</tr>
<tr>
<td></td>
<td>• s - システム</td>
</tr>
<tr>
<td>File Creation Time</td>
<td>ファイルを作成した時刻（hh:mm:ss 形式）。</td>
</tr>
<tr>
<td>File Creation Date</td>
<td>ファイルを作成した日付（dd:mm:yyyy 形式）。</td>
</tr>
<tr>
<td>File Access Time</td>
<td>ファイルにアクセスした時刻（hh:mm:ss 形式）。</td>
</tr>
<tr>
<td>列</td>
<td>説明</td>
</tr>
<tr>
<td>------------------------</td>
<td>----------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| 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

```bash
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-ディスケット
  - hd-ハードディスク
  - cd-CD-ROM
  2つのタイプの間に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。表7(42ページ)を参照してください。
- **-u**
  新しく取り付け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

マッピングの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。

- t 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前に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。ユーザーディレクトリ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。

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。

<table>
<thead>
<tr>
<th>列番号</th>
<th>説明</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>テーブルの名前。名前はmappingsです。</td>
</tr>
<tr>
<td>2</td>
<td>マッピングされた名前。マッピングされたデバイス名。</td>
</tr>
<tr>
<td>3</td>
<td>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。</td>
</tr>
<tr>
<td>4</td>
<td>一貫性のある名前。mappedname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）。mappedname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。</td>
</tr>
</tbody>
</table>

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。出力の詳細 - memmap コマンドを参照してください。

42 UEFI シェルコマンドのリファレンス
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

<table>
<thead>
<tr>
<th>Type</th>
<th>Start</th>
<th>End</th>
<th># Pages</th>
<th>Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>available</td>
<td>00000000000750000-0000000001841FFF</td>
<td></td>
<td>00000000000010F2</td>
<td>0000000000000009</td>
</tr>
<tr>
<td>LoaderCode</td>
<td>0000000001842000-00000000018A3FFF</td>
<td></td>
<td>0000000000000062</td>
<td>0000000000000009</td>
</tr>
<tr>
<td>available</td>
<td>00000000018A4000-00000000018C1FFFF</td>
<td></td>
<td>000000000000001E</td>
<td>0000000000000009</td>
</tr>
<tr>
<td>LoaderData</td>
<td>00000000018C2000-00000000018CAFFF</td>
<td></td>
<td>00000000000000B9</td>
<td>0000000000000009</td>
</tr>
<tr>
<td>BS_code</td>
<td>00000000018CB000-0000000001905FFF</td>
<td></td>
<td>00000000000003B</td>
<td>0000000000000009</td>
</tr>
<tr>
<td>BS_data</td>
<td>0000000001906000-00000000019C9FFF</td>
<td></td>
<td>0000000000000C4</td>
<td>0000000000000009</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RT_data</td>
<td>0000000001B2B000-0000000001B2BFFF</td>
<td></td>
<td>000000000000001</td>
<td>8000000000000009</td>
</tr>
<tr>
<td>BS_data</td>
<td>0000000001B2C000-0000000001B4FFFF</td>
<td></td>
<td>000000000000024</td>
<td>0000000000000009</td>
</tr>
<tr>
<td>reserved</td>
<td>0000000001B50000-0000000001D4FFFF</td>
<td></td>
<td>000000000000200</td>
<td>0000000000000009</td>
</tr>
</tbody>
</table>

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

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

<table>
<thead>
<tr>
<th>列</th>
<th>説明</th>
</tr>
</thead>
<tbody>
<tr>
<td>種類</td>
<td>メモリのタイプ。オプションは次のとおりです。</td>
</tr>
<tr>
<td></td>
<td>• Available</td>
</tr>
<tr>
<td></td>
<td>• LoaderCode</td>
</tr>
<tr>
<td></td>
<td>• LoaderData</td>
</tr>
<tr>
<td></td>
<td>• BootServiceCode</td>
</tr>
<tr>
<td></td>
<td>• BootServiceData</td>
</tr>
<tr>
<td></td>
<td>• RuntimeCode</td>
</tr>
<tr>
<td></td>
<td>• RuntimeData</td>
</tr>
<tr>
<td></td>
<td>• Reserved</td>
</tr>
<tr>
<td></td>
<td>• MemoryMappedIO</td>
</tr>
<tr>
<td></td>
<td>• MemoryMappedIOPortSpace</td>
</tr>
<tr>
<td></td>
<td>• UsableMemory</td>
</tr>
<tr>
<td></td>
<td>• ACPIReclaimMemory</td>
</tr>
<tr>
<td></td>
<td>• ACPIMemoryNVS</td>
</tr>
<tr>
<td></td>
<td>• PalCode</td>
</tr>
<tr>
<td>Start</td>
<td>開始アドレス。</td>
</tr>
<tr>
<td>End</td>
<td>終了アドレス。</td>
</tr>
</tbody>
</table>
### 表 8 出力の詳細 - memmap コマンド (続き)

<table>
<thead>
<tr>
<th>列</th>
<th>説明</th>
</tr>
</thead>
<tbody>
<tr>
<td># Pages</td>
<td>4 KB ページの数。</td>
</tr>
<tr>
<td>reserved</td>
<td>予約済みメモリの合計サイズ（バイト単位）。</td>
</tr>
<tr>
<td>LoaderCode</td>
<td>ローダーコードの合計サイズ（バイト単位）。</td>
</tr>
<tr>
<td>LoaderData</td>
<td>ローダーデータの合計サイズ（バイト単位）。</td>
</tr>
<tr>
<td>BS_code</td>
<td>ブートサービスコードの合計サイズ（バイト単位）。</td>
</tr>
<tr>
<td>BS_data</td>
<td>ブートサービスデータの合計サイズ（バイト単位）。</td>
</tr>
<tr>
<td>RT_data</td>
<td>ランタイムデータの合計サイズ（バイト単位）。</td>
</tr>
<tr>
<td>available</td>
<td>使用可能なメモリ（バイト単位）。</td>
</tr>
<tr>
<td>Total Memory</td>
<td>合計メモリサイズ（バイト単位）。</td>
</tr>
</tbody>
</table>

### 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`に入れる親ディレクトリが含まれる場合は、親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。

```bash
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。

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

Directory of: fs0:\

```
06/18/01  08:05p <DIR>      512  test
```
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 80×50 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と、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。

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

openinfo

構文

```bash
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

```bash
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。
使用法
このコマンドにより、-s 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。

```shell
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。

```shell
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
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。

UEFI シェルコマンド  47
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。

```bash
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。

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

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。

```bash
fs0:\> echo -off
fs0:\> pause.nsh
```

pci

構文
pci [bus dev[func][ -s seg][ -i]]

説明
PCI デバイスリストまたは PCI 関数成スペース情報を表示します。

オプション
- bus バス番号
- dev デバイス番号
- func 関数番号
-s seg セグメント番号を指定します。
-i 情報の解釈。

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:>
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:>

ping

構文
ping[-s sourceIP][-n count][-l size]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部分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、ホストからの 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 -1 64 192.168.0.1
20 バイト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
ramdisk

構文
ramdisk -c[-s size][v volumelabel][-t type]
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で指定します。有効な値は、4MB〜512MB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または%、^、*、+、=、[ ]、|、;、\、<、>、?、/の文字を除く最大11文字です。
-t type ファイルシステムの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4MB〜最大サイズ512MB
- FAT32 -最小サイズ512MB〜最大サイズ2000MB
-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。

例
512MBのFAT16RAM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、512MBのFAT16RAM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
1000MBのFAT32RAM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

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

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。有効な値は GET/POST/PUT/PATCH/DELETE です。
-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。
```
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。
-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 rest/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 rest/v1/Systems/1/Bios -g AdminName
ComputerSystem リソースから BIOS 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 Bios/Current/VersionString
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 rest/v1/Systems/1/Bios/Settings -s AdminName="First Last"
POST 要求を送信するには、次のように入力します。
Shell> restclient -uri rest/v1/Systems/1 -m POST -i PostData.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。
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。

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うることになります（コマンドが終了する原因になります）。

```
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うすることになります。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
```
secboot

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

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 X 509 ファイルまたは EFI アプリケーションのハッシュをセキュアブート変数に登録します。
-f file DER 形式の X 509 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再初期化します。
-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。

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
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。

54 UEFI シェルコマンドのリファレンス
secboot -e db –f boot64.efi
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再初期化するには、次のように入力します。

secboot -r
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。

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。

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。

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。

secboot -d KEK -i 2

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 バラメーターに設定します。バラメーターとして使用すると、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 ef1.1
fs0:\> set
]* path : .;fs0:\efi\tools;fs0:\efi\boot;fs0:\
src : ef1.1
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:\
次回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\EF1.1
Shell> set 
]* path : .;fs0:\efi\tools;fs0:\efi\boot;fs0:\
* EFI_SOURCE : c:\project\EF1.1

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 * プロセッサー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ーを停止するには、次のように入力します。

Shell> stall 20

sysconfig

構文

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

説明

HPE システムの BIOS 設定を表示または構成します。

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 * 設定の値を指定します。

* 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タイプ設定用のデフォルト値を取得（一覧表示）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 エンコードを使用します。
-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ときに、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。例: "sysconfig -s AdminName ="Joe Smith"
文字列タイプの設定から settingvalue を削除するには、二重引用符を使用します。例："sysconfig -s AdminName="
-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容易に解析できます。このオプションを使用すると、値文字列内のすべてのセミコロン文字がエスケープシーケンス ';\';' で置き換えられます。
-export、-set、および-import オプションは SFO をサポートしません。
'文字は、文字列タイプの設定で使用できます。

例
すべての BIOS 設定の現在の値を表示するには、次のように入力します。
Shell> sysconfig -g all
ProcHyperthreading 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 ProcHyperthreading
ProcHyperthreading 設定の現在の値を取得するには、次のように入力します。
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タイプの設定を一覧表示するには、次のように入力します。
Shell> sysconfig -d get
すべてのデフォルトタイプの設定のリストを表示するには、次のように入力します。
Shell> sysconfig -d get SystemDefaults all
NicBoot1 のデフォルト設定のリストを表示するには、次のように入力します。
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には、次のように入力します。
Shell> sysconfig -d set SystemDefaults all
NicBoot1 のデフォルト設定を設定するには、次のように入力します。
Shell> sysconfig -d set SystemDefaults NicBoot1
新しい管理者パスワードとして 123 を設定するには、次のように入力します。
Shell> sysconfig -s AdminPassword=123 OldAdminPassword=""
電源投入時パスワードの Joe Smith を削除するには、次のように入力します。
Shell> sysconfig -s PowerOnPassword="" OldPowerOnPassword="Joe Smith"
UEFI ブート順序のリストを表示するには、次のように入力します。

Shell> sysconfig -g UefiBootOrder

UefiBootOrder=
  0: Embedded UEFI Shell
  8: Rear USB 2
  10: Embedded LOM 1 Port 1

Embedded LOM 1 Port 1 を 1 番目に起動するように新しい UEFI ブート順序を設定するには、次のように入力します。

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: Embedded UEFI Shell
  8: Rear USB 2

標準形式の出力で EmbeddedUefiShell 設定の現在の値を取得するには、次のように入力します。

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現在の sysconfig 属性の名前、説明、現在の値と設定可能な値、および Enum 設定タイプを表示できます。次の例では、考えられるコマンド出力の一部を示します。

各属性について詳しくは、『HPE ProLiant m510 および m710x サーバーカートリッジ用 UEFI システムユーティリティユーザーガイド』を参照してください。

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

<table>
<thead>
<tr>
<th>Setting Name</th>
<th>Current Value</th>
<th>Possible Values</th>
<th>Setting Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>VirtualSerialPort</td>
<td>Com2Irq3</td>
<td>Com2Irq3</td>
<td>Enum</td>
</tr>
<tr>
<td>NetworkBoot</td>
<td>NetworkBoot</td>
<td>NetworkBoot</td>
<td>Enum</td>
</tr>
<tr>
<td>ServerName</td>
<td>NetworkBoot</td>
<td>NetworkBoot</td>
<td>Enum</td>
</tr>
<tr>
<td>EmbeddedLOM1Port1</td>
<td>EmbeddedLOM1Port1</td>
<td>EmbeddedLOM1Port1</td>
<td>Enum</td>
</tr>
</tbody>
</table>

UEFI シェルコマンドのリファレンス
Setting Name = PreBootNetwork [Pre-Boot Network Interface]
Current Value = FlexLom1 [Embedded : FlexLOM 1]
Possible Values = FlexLom1 [Embedded : FlexLOM 1]
PciSlot1 [PCIe Slot 1]
PciSlot2 [PCIe Slot 2]
PciSlot3 [PCIe Slot 3]
PciSlot4 [PCIe Slot 4]
PciSlot5 [PCIe Slot 5]
PciSlot6 [PCIe Slot 6]
PciSlot7 [PCIe Slot 7]
PciSlot8 [PCIe Slot 8]
PciSlot9 [PCIe Slot 9]
Setting Type = Enum

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

sysinfo

構文
sysinfo[token][-sfo][-b][-v]

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、プロセッサー、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。

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。
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 -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

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すが、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/秒の正しい範囲内にない場合、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。

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
9:51:31 (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更新されます。

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
```

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を変更できません。

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
2.0

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。
-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:

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

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:

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][-o file][[-m][-l]

説明
HTTP または FTP からファイルをダウンロードし、ISO ファイルシステムをマウントします。

オプション
-g URL で指定したドキュメントを取得します。
-URL 取得する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。これには、IPv4 アドレスまたはホスト名を指定できます。
-o 指定したファイルに出力をリダイレクトします。
-file 出力のリダイレクト先の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。

使用法
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、それに設定は消去され、代わりに、コマンドの実行時にインターネット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
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。IPv4アドレスを使用する必要があります。

例
HTTPまたはFTPサーバーからISOファイルをダウンロードし、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:\/\/www.hpe.com/filename.iso -m
```
または
```
fs0:\> webclient -g ftp:\/\/192.168.1.20/filename.iso -m
```
FTPまたは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
```
または
```
fs0:\> webclient -g ftp:\/\/192.168.1.20/file.html -o file.html
```
または
```
fs0:\> webclient -g ftp:\/\/ftp.hpe.com/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。
```
fs0:\> webclient -g ftp:\/\/user:pass@192.168.1.20/file.html -o file.html
```
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
```
3 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」を表示するシ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 シェルスクリプト自動起動] 構成
- 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 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。詳しくは、『HPE ProLiant m510 および m710x サーバーカーティッジ用 UEFI システムユーティリティユーザーガイド』の「UEFI シェルスクリプト自動起動」を参照してください。

シェルスクリプトの手動開始
1. .nsh スクリプトファイルの場所に移動します。
2. ファイルをダブルクリックまたは右クリックしてから、「開く」を選択します。

ファイルへの設定のエクスポートとインポート
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
```
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：
@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."
syconfig -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 "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 ===========================================================================
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
techo "Cannot create a RAMDISK of size %RamDiskSize%."
go to 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 FS<x> pattern. The newly created RAMDISK will be FS1: or higher.
Here the check goes up to FS3: (the inner for loop), but a larger limit may be used in case many other file systems already exist before the creation of this RAMDISK. The FS for the RAMDISK is found when the FsIndex matches the FS<x> in RamDiskStr. Change the working directory to FS<FsIndex>:, so all downloads get saved there.
FS0: is ignored. In the worst case, when no other usable file system is present, FS0: will map to the file system that this script is executing from.

for %a in %RamDiskStr%
  for %b run (1 10)
    set -v FsIndex %b
    if 'FS%FsIndex%:' == %a then
      FS%FsIndex%:
      goto RDFOUND
    endif
  endfor
endfor

The following message appears if the newly created RAMDISK cannot be found.
echo "RAMDISK with Volume Label %VolumeLabel% not found!"
goto EXITSCRIPT

The following message appears if the RAMDISK FS<x> has been found and you are in the RAMDISK's root folder.
:RDFOUND
echo "RAMDISK with Volume Label %VolumeLabel% found at FS%FsIndex%:."
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<id> 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。
   - 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。
   - 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。
4 UEFI プログラミングモデル

UEFI シェルには、プログラム 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
ミング 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
ができる。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

<table>
<thead>
<tr>
<th>関数タイプ</th>
<th>関数名</th>
<th>説明</th>
</tr>
</thead>
<tbody>
<tr>
<td>EFI_SHELL_EXECUTE</td>
<td>Execute</td>
<td>シェルがコマンドラインを解析して実行します。</td>
</tr>
<tr>
<td>EFI_SHELL_GET_ENV</td>
<td>GetEnv</td>
<td>環境変数を取得します。</td>
</tr>
<tr>
<td>EFI_SHELL_SET_ENV</td>
<td>SetEnv</td>
<td>特定の環境変数を変更します。</td>
</tr>
<tr>
<td>EFI_SHELL_GET_ALIAS</td>
<td>GetAlias</td>
<td>特定のシェルコマンドのエイリアスを取得します。</td>
</tr>
<tr>
<td>EFI_SHELL_SET_ALIAS</td>
<td>SetAlias</td>
<td>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。</td>
</tr>
<tr>
<td>EFI_SHELL_GET_HELP_TEXT</td>
<td>GetHelpText</td>
<td>特定のコマンドに関するヘルプ情報を返します。</td>
</tr>
<tr>
<td>EFI_SHELL_GET_DEVICE_PATH_FROM_MAP</td>
<td>GetDevicePathFromMap</td>
<td>マッピングに対応するデバイスパスを返します。</td>
</tr>
<tr>
<td>EFI_SHELL_MAP_FROM_DEVICE_PATH</td>
<td>GetMapFromDevicePath</td>
<td>特定のデバイスパスに対応するマッピングを返します。</td>
</tr>
<tr>
<td>EFI_SHELL_GET_FILE_PATH_FROM_FILE_PATH</td>
<td>GetDevicePathFromFilePath</td>
<td>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。</td>
</tr>
<tr>
<td>EFI_SHELL_GET_FILE_PATH_FROM_DEVICE_PATH</td>
<td>GetFilePathFromDevicePath</td>
<td>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。</td>
</tr>
<tr>
<td>EFI_SHELL_SET_MAP</td>
<td>SetMap</td>
<td>デバイスとデバイスパス間のマッピングを作成、更新、または削除します。</td>
</tr>
<tr>
<td>EFI_SHELL_GET_CUR_DIR</td>
<td>GetCurDir</td>
<td>デバイスの現在のディレクトリを返します。</td>
</tr>
<tr>
<td>EFI_SHELL_SET_CUR_DIR</td>
<td>SetCurDir</td>
<td>デバイスの現在のディレクトリを変更します。</td>
</tr>
<tr>
<td>EFI_SHELL_OPEN_FILE_LIST</td>
<td>OpenFileList</td>
<td>指定したパスパターンに一致するファイルを開きます。</td>
</tr>
<tr>
<td>EFI_SHELL_FREE_FILE_LIST</td>
<td>FreeFileList</td>
<td>OpenFileList() で作成したファイルリストを解放します。</td>
</tr>
<tr>
<td>EFI_SHELL_REMOVE_DUP_IN_FILE_LIST</td>
<td>RemoveDupInFileList</td>
<td>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。</td>
</tr>
<tr>
<td>EFI_SHELL_BATCH_IS_ACTIVE</td>
<td>BatchIsActive</td>
<td>处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。</td>
</tr>
</tbody>
</table>
### 表 9 UEFI アプリケーション API (続き)

<table>
<thead>
<tr>
<th>関数タイプ</th>
<th>関数名</th>
<th>説明</th>
</tr>
</thead>
<tbody>
<tr>
<td>EFI_SHELL_IS_ROOT_SHELL</td>
<td>IsRootShell</td>
<td>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。</td>
</tr>
<tr>
<td>EFI_SHELL_ENABLE_PAGE_BREAK</td>
<td>EnablePageBreak</td>
<td>改ページ出力モードを有効にします。</td>
</tr>
<tr>
<td>EFI_SHELL_DISABLE_PAGE_BREAK</td>
<td>DisablePageBreak</td>
<td>改ページ出力モードを無効にします。</td>
</tr>
<tr>
<td>EFI_SHELL_GET_PAGE_BREAK</td>
<td>GetPageBreak</td>
<td>改ページ出力モードの有効化ステータスを取得します。</td>
</tr>
<tr>
<td>EFI_SHELL_GET_DEVICE_NAME</td>
<td>GetDeviceName</td>
<td>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。</td>
</tr>
<tr>
<td>EFI_SHELL_GET_FILE_INFO</td>
<td>GetFileInfo</td>
<td>特定のファイルハンドルに関する情報の取得です。</td>
</tr>
<tr>
<td>EFI_SHELL_SET_FILE_INFO</td>
<td>SetFileInfo</td>
<td>特定のファイルハンドルに関する情報を変更します。</td>
</tr>
<tr>
<td>EFI_SHELL_OPEN_FILE_BY_NAME</td>
<td>OpenFileByName</td>
<td>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。</td>
</tr>
<tr>
<td>EFI_SHELL_CLOSE_FILE</td>
<td>CloseFile</td>
<td>開いているファイルを閉じます。</td>
</tr>
<tr>
<td>EFI_SHELL_CREATE_FILE</td>
<td>CreateFile</td>
<td>新しいファイルを作成します。</td>
</tr>
<tr>
<td>EFI_SHELL_READ_FILE</td>
<td>ReadFile</td>
<td>データをファイルから読み取ります。</td>
</tr>
<tr>
<td>EFI_SHELL_WRITE_FILE</td>
<td>WriteFile</td>
<td>データをファイルに書き込みます。</td>
</tr>
<tr>
<td>EFI_SHELL_DELETE_FILE</td>
<td>DeleteFile</td>
<td>ファイルを削除します。</td>
</tr>
<tr>
<td>EFI_SHELL_DELETE_FILE_BY_NAME</td>
<td>DeleteFileByName</td>
<td>ファイル名に基づいて削除します。</td>
</tr>
<tr>
<td>EFI_SHELL_GET_FILE_POSITION</td>
<td>GetFilePosition</td>
<td>ファイルの現在の読み取り/書き込み位置を表示します。</td>
</tr>
<tr>
<td>EFI_SHELL_SET_FILE_POSITION</td>
<td>SetFilePosition</td>
<td>ファイルの現在の読み取り/書き込み位置を変更します。</td>
</tr>
<tr>
<td>EFI_SHELL_FLUSH_FILE</td>
<td>FlushFile</td>
<td>すべてのバッファーデータをファイルに書き込みます。</td>
</tr>
<tr>
<td>EFI_SHELL_FIND_FILES</td>
<td>FindFiles</td>
<td>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。</td>
</tr>
<tr>
<td>EFI_SHELL_FIND_FILES_IN_DIR</td>
<td>FindFilesInDir</td>
<td>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。</td>
</tr>
<tr>
<td>EFI_SHELL_GET_FILE_SIZE</td>
<td>GetFileSize</td>
<td>ファイルのサイズを表示します。</td>
</tr>
<tr>
<td>EFI_SHELL_OPEN_ROOT</td>
<td>OpenRoot</td>
<td>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。</td>
</tr>
<tr>
<td>EFI_SHELL_OPEN_ROOT_BY_HANDLE</td>
<td>OpenRootByHandle</td>
<td>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。</td>
</tr>
<tr>
<td>EFI_EVENT</td>
<td>ExecutionBreak</td>
<td>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。</td>
</tr>
<tr>
<td>UINT32</td>
<td>MajorVersion</td>
<td>シェル環境のメジャーバージョン。</td>
</tr>
<tr>
<td>UINT32</td>
<td>MinorVersion</td>
<td>シェル環境のマイナーバージョン。</td>
</tr>
</tbody>
</table>
5 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

<table>
<thead>
<tr>
<th>ステータスコード</th>
<th>説明</th>
</tr>
</thead>
<tbody>
<tr>
<td>SHELL_SUCCESS</td>
<td>操作が要求どおりに完了しました。</td>
</tr>
<tr>
<td>SHELL_NOT_FOUND</td>
<td>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。</td>
</tr>
<tr>
<td>SHELL_SECURITY_VIOLATION</td>
<td>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。</td>
</tr>
<tr>
<td>SHELL_INVALID_PARAMETER</td>
<td>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。</td>
</tr>
<tr>
<td>SHELL_OUT_OF_RESOURCES</td>
<td>非揮発方式で変数を設定する要求を完了できませんでした。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。</td>
</tr>
<tr>
<td>SHELL_WRITE_PROTECTED</td>
<td>操作対象のメディアは書き込み禁止です。</td>
</tr>
<tr>
<td>SHELL_DEVICE_ERROR</td>
<td>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でした。</td>
</tr>
</tbody>
</table>
6 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を受けるには、Web サイト「Contact Hewlett Packard Enterprise Worldwide」に移動します。
  http://www.hpe.com/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 サイトに移動します。
  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
- 増設した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。製品のドキュメントを確認し、推奨されるソフトウェアアップデートの方法を特定します。
- 製品のアップデートをダウンロードするには、以下のいずれかに移動します。
  ◦ Hewlett Packard Enterprise サポートセンターの [メールニュース配信登録] ページ：
    http://www.hpe.com/support/e-updates-ja
  ◦ Software Depot の Web サイト：
    http://www.hpe.com/support/softwaredepot

1 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の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いてください。
関連情報
- 『HPE ProLiant m510 および m710x サーバーカートリッジ用 UEFI システムユーティリティユーザーガイド』
- 『HPE ProLiant m510 および m710x サーバーカートリッジ用 UEFI シェルクイックリファレンスカード』
- 『HPE ProLiant m510 および m710x サーバーカートリッジ用 UEFI システムユーティリティおよびシェルリリースノート』
- 『HPE ProLiant m510 および m710x サーバーカートリッジ用 UEFI 展開ガイド』
- 『HPE ProLiant m510 および m710x サーバーカートリッジ用 UEFI 設定クイックリファレンスガイド』

『HPE ProLiant m510 および m710x サーバーカートリッジ用 UEFI システムユーティリティおよびシェルコマンドモバイルヘルプ』は、システムユーティリティ画面の下部にある QR コードをスキャンするか、または http://www.hpe.com/qref/ProLiantUEFI/Help で入手できます。

Web サイト

<table>
<thead>
<tr>
<th>Web サイト</th>
<th>リンク</th>
</tr>
</thead>
<tbody>
<tr>
<td>UEFI Specification.マルチプロセッサー仕様</td>
<td><a href="http://www.uefi.org/specifications">http://www.uefi.org/specifications</a></td>
</tr>
<tr>
<td>UEFI の学習資料</td>
<td><a href="http://www.uefi.org/learning_center">http://www.uefi.org/learning_center</a></td>
</tr>
<tr>
<td>SourceForge の UEFI EDK2 プロジェクト（仕様およびコードのダウンロード）</td>
<td><a href="http://sourceforge.net/apps/mediawiki/tianocore">http://sourceforge.net/apps/mediawiki/tianocore</a></td>
</tr>
<tr>
<td>Hewlett Packard Enterprise Information Library</td>
<td><a href="http://www.hpe.com/info/enterprise/docs">http://www.hpe.com/info/enterprise/docs</a></td>
</tr>
<tr>
<td>Hewlett Packard Enterprise サポートセンター</td>
<td><a href="http://www.hpe.com/support/hpesc">http://www.hpe.com/support/hpesc</a></td>
</tr>
<tr>
<td>Contact Hewlett Packard Enterprise Worldwide</td>
<td><a href="http://www.hpe.com/assistance">http://www.hpe.com/assistance</a></td>
</tr>
<tr>
<td>サブスクリプションサービス/サポートのアラート</td>
<td><a href="http://www.hpe.com/support/e-updates-ja">http://www.hpe.com/support/e-updates-ja</a></td>
</tr>
<tr>
<td>Software Depot</td>
<td><a href="http://www.hpe.com/support/softwaredepot">http://www.hpe.com/support/softwaredepot</a></td>
</tr>
<tr>
<td>カスタマーサービスフリーペア</td>
<td><a href="http://www.hpe.com/support/selfrepair">http://www.hpe.com/support/selfrepair</a></td>
</tr>
<tr>
<td>Insight Remote Support</td>
<td><a href="http://www.hpe.com/info/insightremotesupport/docs">http://www.hpe.com/info/insightremotesupport/docs</a></td>
</tr>
</tbody>
</table>

カスタマーサービスフリーペア

Hewlett Packard Enterprise カスタマーサービスフリーペア（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。
デバイスサポートについて詳しくは、次のWebサイトを参照してください。

http://www.hpe.com/info/insightremotesupport/docs
### 用語集

<table>
<thead>
<tr>
<th>ASR</th>
<th>Automatic Server Recovery。自動サーバー復旧</th>
</tr>
</thead>
<tbody>
<tr>
<td>EFI</td>
<td>Extensible Firmware Interface</td>
</tr>
<tr>
<td>EMS</td>
<td>Emergency Management Services</td>
</tr>
<tr>
<td>iLO</td>
<td>Integrated Lights-Out</td>
</tr>
<tr>
<td>PCIe</td>
<td>PCI Express（Peripheral Component Interconnect Express）</td>
</tr>
<tr>
<td>POST</td>
<td>Power-On Self-Test。電源投入時セルフテスト</td>
</tr>
<tr>
<td>PXE</td>
<td>Pre-Boot Execution Environment</td>
</tr>
<tr>
<td>QPI</td>
<td>Intel's QuickPath Interconnect。インテルの QuickPath インターコネクト</td>
</tr>
<tr>
<td>RIS</td>
<td>Remote Installation Services。リモートインストールサービス</td>
</tr>
<tr>
<td>RTC</td>
<td>Real-Time Clock。リアルタイムクロック</td>
</tr>
<tr>
<td>SMBIOS</td>
<td>System Management BIOS。システム管理 BIOS</td>
</tr>
<tr>
<td>TPM</td>
<td>Trusted Platform Module</td>
</tr>
<tr>
<td>UEFI</td>
<td>Unified Extensible Firmware Interface</td>
</tr>
<tr>
<td>VSP</td>
<td>Virtual Serial Port。仮想シリアルポート</td>
</tr>
</tbody>
</table>

索引

D
del エイリアス, 52
dir エイリアス, 38

E
EFI 壓縮解除, 33

F
FTP 転送, 34, 66

G
goto, 35

H
Hewlett Packard Enterprise シェルコマンド, 7
Hewlett Packard Enterprise への問い合わせ, 76
HTTP
ファイルのダウンロード元, 66
HTTP または FTP からのファイルのダウンロード, 66

I
ifconfig, 36
IPv4 ネットワークスタックアドレスの変更, 36
ISO
ファイルシステムのマウント, 66
ISO ファイルシステムのマウント, 66

M
MiniZip, 20
mkdir, 44
MTC, 35

O
openinfo, 46

P
partitions, 47
PCI 情報, 48
ping, 49

R
RAM ディスク, 50
RAM ディスクの作成と削除, 50
RESTful API, 51

S
SMBIOS 情報, 57
sysconfig
属性, 60
表示と構成, 58

T
touch, 63–64

U
UEFI アプリケーション API, 73

UEFI シェルコマンド, 14
alias, 14
attrib, 15
boot, 16
cd, 17
cls, 18
comp, 19
compress, 20
connect, 21
cp, 22
date, 24
dblk, 25
devices, 25
devtree, 26
dh, 27
disconnect, 29
dmem, 30
drivers, 30
echo, 31
edit, 32
eficompress, 33
efidecompress, 33
exit, 33
ftp, 34
getmtc, 35
goto, 35
help, 35
ifconfig, 36
load, 37
ls, 38
map, 41
memmap, 42
mkdir, 44
mode, 45
mv, 45
openinfo, 46
parse, 46
partitions, 47
pause, 47
pci, 48
ping, 49
ramdisk, 50
reset, 51
restclient, 51
rm/del, 52
secboot, 54
set, 55
setsize, 56
shift, 56
smbiosview, 57
stall, 57
sysconfig, 58
sysinfo, 61
time, 62
timezone, 63
touch, 63
type, 64
コマンドラインの補完, 7
コマンドラインヘルプ, 10
コンソール出力モード, 45

サポート
Hewlett Packard Enterprise, 76

シェルスクリプト
手動開始, 68
編集, 68
例, 68
シェルスクリプトの手動開始, 68
シェルスクリプトの編集, 68
シェルスクリプトの例, 68
シェルまたはスクリプトの終了, 33
システム BIOS の構成, 58
システム時刻, 62
システム情報, 61
システムのリセット, 51
システム日付, 24
システムリセット, 51
出力
一度に 1 つの画面, 8
概要, 9
詳細, 9
制御, 8
出力の詳細
devices コマンド, 26
dh コマンド, 29
drivers コマンド, 31
ls コマンドのファイル情報, 41
ls コマンドのボリューム情報, 40
memmap コマンド, 44
sysconfig コマンド, 60
シリアルコンソール接続, 6

スクリプティング, 5
スクリプトコマンド, 68
一時停止, 47
実行の停止, 57
シフト, 56
終了, 33
表示, 31
ラベルの検索, 35
スクリプトの起動, 68
スクリプトの実行の停止, 57

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の
表示, 54
セキュアブートデータベース、キー、およびセキュリ
ティレポートの表示, 54

属性, 15
タイムゾーン, 63
ディレクトリ
一覧表示, 38
削除, 52
作成, 44
変更, 17
ディレクトリの変更, 17
デバイス, 25
デバイスツリー, 26
デバイスドライバーの切断, 29
デバイスのブロック, 25
デバイスハンドル, 27
電源投入時パスワード, 58

ドキュメント
関連情報, 77
ドライバー, 30
アンロード, 64
バインド, 21
ロード, 37
ドライバーイメージのアンロード, 64
ドライバーのバインド, 21

内蔵 UEFI シェル
アクセス, 5
前提条件, 5
の起動, 5
を起動する前に, 5
夏時間, 62
名前とデバイスハンドルのマッピング, 41

バージョン情報, 65
背景色, 18
はじめに, 5
パスワード, 58
ハンドルのプロトコルとエージェント, 46

日付, 24

ファームウェア
UEFI バージョンの表示, 65
ファイル
EFI の圧縮解除, 33
FTP 経由の転送, 34
圧縮と圧縮解除, 20
一覧表示, 38
コピー, 22
サイズの設定, 56
削除, 52
出力デバイスへの内容の送信, 64
操作, 5