



# Shell Command Reference Manual

*November 2007*

*Revision 1.1*

INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH INTEL® PRODUCTS. NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT. EXCEPT AS PROVIDED IN INTEL'S TERMS AND CONDITIONS OF SALE FOR SUCH PRODUCTS, INTEL ASSUMES NO LIABILITY WHATSOEVER, AND INTEL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO SALE AND/OR USE OF INTEL PRODUCTS INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT. Intel products are not intended for use in medical, life saving, or life sustaining applications.

Intel may make changes to specifications and product descriptions at any time, without notice.

Designers must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined." Intel reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them.

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

\*Other names and brands may be claimed as the property of others.

Copyright © 2007, Intel Corporation. All rights reserved.

# Contents

---

1	Introduction .....	1
1.1	Overview .....	1
1.2	Related Information.....	1
1.3	Terms.....	2
2	Command Descriptions .....	5
2.1	Overview .....	5
2.1.1	Command Summary .....	5
2.1.2	Explanation of Command Description Layout .....	7
2.1.3	Supported EFI Protocols in the Shell .....	7
2.2	Commands from Default Build Shell .....	10
2.2.1	alias .....	10
2.2.2	attrib.....	11
2.2.3	cd .....	13
2.2.4	cls.....	15
2.2.5	connect .....	16
2.2.6	cp .....	19
2.2.7	date .....	21
2.2.8	del .....	22
2.2.9	dh.....	25
2.2.10	dir .....	27
2.2.11	disconnect.....	30
2.2.12	drivers.....	31
2.2.13	drvcfg.....	33
2.2.14	drvdiag.....	36
2.2.15	echo.....	39
2.2.16	exit .....	40
2.2.17	help .....	41
2.2.18	load .....	42
2.2.19	ls.....	43
2.2.20	map .....	46
2.2.21	mkdir .....	50
2.2.22	mv .....	51
2.2.23	reconnect.....	53
2.2.24	reset .....	54
2.2.25	rm .....	55
2.2.26	set .....	58
2.2.27	time .....	59
2.2.28	touch.....	61
2.2.29	type .....	62
2.2.30	unload .....	64
2.2.31	ver .....	65
2.2.32	vol .....	68
2.3	Additional Commands from Full Build Shell.....	70
2.3.1	comp.....	70
2.3.2	dblk .....	71
2.3.3	devices .....	74

2.3.4	devtree.....	75
2.3.5	dmem .....	76
2.3.6	dmpstore .....	79
2.3.7	edit .....	80
2.3.8	eficompress.....	81
2.3.9	efidecompress .....	82
2.3.10	err .....	83
2.3.11	guid .....	85
2.3.12	hexedit .....	86
2.3.13	ipconfig .....	87
2.3.14	loadpcirom .....	88
2.3.15	mem .....	89
2.3.16	memmap .....	91
2.3.17	mm.....	92
2.3.18	mode .....	95
2.3.19	openinfo .....	97
2.3.20	pci .....	98
2.3.21	sermode .....	101
2.3.22	smbiosview .....	103
2.3.23	telnetmgmt .....	105
2.3.24	timezone.....	106

## Tables

Table 1	Commands from Default Build Shell.....	5
Table 2	Additional Commands from Full Build Shell .....	6
Table 3	EFI Shell Protocol Information Table .....	7
Table 4	Conventions for Directory Names .....	14
Table 5	Default Values for the Type Parameter .....	34
Table 6	Display Format of devices .....	74
Table 7	Open Protocol Information Layout .....	97

## *Revision History*

---

Revision Number	Description	Revision Date
1.0	Initial release.	January 2007
1.1	Add -u option to map.	November 2007

§

# 1 Introduction

---

## 1.1 Overview

The EFI Shell environment provides a rich set of commands that extend and enhance the EFI Shell capability. These commands can be used directly from a command prompt.

This document describes all the shell commands a user can execute directly in EFI Shell command prompt.

Shell commands can be categorized as two sets. One is built into the Default Build Shell and the other is only built into the Full Build Shell. Both of Shell Builds can be obtained from the EDK release package, It can be downloaded from <http://edk.tianocore.org>.

The Default Build Shell location:

`Edk\Other\Maintained\Application\[UEFI_PREFIX]Shell\bin\[PROCESSOR]\Shell.efi`

The Full Build Shell location:

`Edk\Other\Maintained\Application\[UEFI_PREFIX]Shell\bin\[PROCESSOR]\Shell_Full.efi`

**UEFI\_PREFIX** is "Uefi" if the image is UEFI build or "" if the image is EFI build, and **PROCESSOR** could be **ia32**, **x64** or **ipf**.

This document describes both command categories.

## 1.2 Related Information

The following publications and sources of information may be useful or are referred to by this document:

*Extensible Firmware Interface Specification*, Version 1.10, Intel, 2001, <http://developer.intel.com/technology/efi>.

*Unified Extensible Firmware Interface Specification*, Version 2.0, Unified EFI, Inc, 2006, <http://www.uefi.org>.

*Intel® Platform Innovation Framework for EFI Specifications*, Intel, 2006, <http://www.intel.com/technology/framework/>.

*EFI Shell User's Guide*, Version 1.0\_to7thRvw, Intel, 2005, <http://efi-shell.tianocore.org>.

## 1.3 Terms

The following terms are used throughout this document to describe varying aspects of input localization:

**Component**

An executable image. Components defined in this specification support one of the defined module types.

**EFI**

Generic term that refers to one of the versions of the EFI specification: EFI 1.02, EFI 1.10, or UEFI 2.0.

**EFI 1.10 Specification**

Intel Corporation published the Extensible Firmware Interface Specification. Intel donated the EFI specification to the Unified EFI Forum, and the UEFI now owns future updates of the EFI specification. See the current UEFI Specification.

**Foundation**

The set of code and interfaces that glue implementations of EFI together.

**Framework**

Intel® Platform Innovation Framework for EFI consists of the Foundation, plus other modular components that characterize the portability surface for modular components designed to work on any implementation of the Tiano architecture.

**GUID**

Globally Unique Identifier. A 128-bit value used to name entities uniquely. without the help of a centralized authority, an individual can generate a unique GUID. This allows the generation of names that will never conflict, even among multiple, unrelated parties.

**Protocol**

An API named by a GUID as defined by the EFI specification.

**UEFI Application**

An application that follows the UEFI specification. The only difference between a UEFI application and a UEFI driver is that an application is unloaded from memory when it exits regardless of return status, while a driver that returns a successful return status is not unloaded when its entry point exits.

**UEFI Driver**

A driver that follows the UEFI specification.



**UEFI Specification Version 2.0**

The first EFI specification released by the Unified EFI Forum. This specification builds on the EFI 1.10 specification and transfers ownership of the EFI specification from Intel to a non-profit, industry trade organization.

**UEFI Specification Version 2.1**

Current version of the UEFI specification released by the Unified EFI Forum.

**Unified EFI Forum**

A non-profit collaborative trade organization formed to promote and manage the UEFI standard. For more information, see [www.uefi.org](http://www.uefi.org).

§

## 2 *Command Descriptions*

---

### 2.1 Overview

#### 2.1.1 Command Summary

The tables below list all Shell commands.

**Table 1 Commands from Default Build Shell**

Command	Description
<a href="#">alias</a>	Displays, creates, or deletes aliases in the EFI Shell.
<a href="#">attrib</a>	Displays or changes the attributes of files or directories
<a href="#">cd</a>	Displays or changes the current directory
<a href="#">cls</a>	Clears the standard output and optionally changes the background color
<a href="#">connect</a>	Binds a driver to a specific device and starts the driver
<a href="#">cp</a>	Copies one or more source files or directories to a destination
<a href="#">date</a>	Displays and sets the current date for the system
<a href="#">del</a>	Deletes one or more files or directories
<a href="#">dh</a>	Displays the device handles in the EFI environment
<a href="#">dir</a>	Lists directory contents or file information
<a href="#">disconnect</a>	Disconnects one or more drivers from the specified devices
<a href="#">drivers</a>	Displays a list of information for drivers
<a href="#">drvcfg</a>	Invokes the Driver Configuration Protocol
<a href="#">drvdiag</a>	Invokes the Driver Diagnostics Protocol
<a href="#">echo</a>	Displays messages or turns command echoing on or off
<a href="#">exit</a>	Exits the EFI Shell environment
<a href="#">help</a>	Displays the list of commands that are built into the EFI Shell
<a href="#">load</a>	Loads an EFI driver into memory
<a href="#">ls</a>	Lists a directory's contents or file information
<a href="#">map</a>	Defines a mapping between a user-defined name and a device handle
<a href="#">mkdir</a>	Creates one or more new directories
<a href="#">mv</a>	Moves one or more files to a destination within a file system
<a href="#">reconnect</a>	Reconnects drivers to the specific device
<a href="#">reset</a>	Resets the system
<a href="#">rm</a>	Deletes one or more files or directories

<a href="#">set</a>	Used to maintain the environment variables
<a href="#">time</a>	Displays or sets the current time for the system
<a href="#">touch</a>	Updates the time and date on a file to the current time and date
<a href="#">type</a>	Sends the contents of a file to the standard output device
<a href="#">unload</a>	Unloads a driver image that was already loaded
<a href="#">ver</a>	Displays the version information for this EFI firmware
<a href="#">vol</a>	Displays the volume information for the file system

**Table 2 Additional Commands from Full Build Shell**

Command	Description
<a href="#">comp</a>	Compares the contents of two files on a byte for byte basis
<a href="#">dblk</a>	Displays the contents of one or more blocks from a block device
<a href="#">devices</a>	Displays the list of devices managed by EFI drivers
<a href="#">devtree</a>	This command displays the tree of devices
<a href="#">dmem</a>	Displays the contents of system or device memory
<a href="#">dmpstore</a>	Manages all EFI NVRAM variables
<a href="#">edit</a>	Full screen editor for ASCII or UNICODE files
<a href="#">eficompress</a>	Compress a file
<a href="#">efidecompress</a>	Decompress a file
<a href="#">err</a>	Displays or changes the error level in the system
<a href="#">guid</a>	Displays all registered EFI GUIDs
<a href="#">hexedit</a>	Full screen hex editor for files, block devices, or memory
<a href="#">ipconfig</a>	Displays or modifies the current IP configuration
<a href="#">loadpcirom</a>	Loads a PCI Option ROM from the specified file
<a href="#">mem</a>	Displays the contents of system or device memory
<a href="#">memmap</a>	Displays the memory map maintained by the EFI environment
<a href="#">mm</a>	Displays or modifies MEM/MMIO/IO/PCI/PCIE address space
<a href="#">mode</a>	Displays or changes the console output device mode
<a href="#">openinfo</a>	Displays the protocols and agents associated with a handle
<a href="#">pci</a>	Displays PCI device list or PCI function configuration space
<a href="#">sermode</a>	Sets serial port attributes
<a href="#">smbiosview</a>	Displays SMBIOS information
<a href="#">telnetmgmt</a>	Change terminal type
<a href="#">timezone</a>	Displays or sets time zone information

## 2.1.2 Explanation of Command Description Layout

The description of each command is composed of four sections: **Summary**, **EFI Versions**, **Usage**, and **Description**.

**Summary** is a brief explanation of the function of the command. **EFI Versions** are the versions of EFI specification that the command requires for execution. **Usage** describes how the command is used. **Description** describes the details of the command.

## 2.1.3 Supported EFI Protocols in the Shell

The table below shows all the supported protocol information in the EFI Shell. Some Shell commands need this information to operate. For example, the [dh](#) command needs the protocol symbol for its **-p** flag to list all the handles on which the specific protocol was installed.

**Table 3 EFI Shell Protocol Information Table**

Protocol Symbol	EFI Protocol
ARP	ARP Protocol
ARPSb	ARP Service Binding Protocol
BlkIo	Block I/O Protocol
BusSpecificDriverOverride	Bus Specific Driver Override Protocol
ComponentName	Component Name Protocol
Configuration	Driver Configuration Protocol
ConIn	Console-in Device
ConOut	Console-out Device
DebugPort	Debugport Protocol
DebugSupport	Debug Support Protocol
Decompress	Decompress Protocol
DevIo	Device I/O Protocol
DHCPv4	DHCPv4 Protocol
DHCPv4Sb	DHCPv4 Service Binding Protocol
Diagnostics	Driver Diagnostics Protocol
DiskIo	Disk I/O Protocol
Dpath	Device Path Protocol
DriverBinding	Driver Binding Protocol
ErrOutSplit	Error-out Splitter Protocol
ExtScsiPassThru	Extended SCSI Pass Through Protocol
Fs	Simple File System Protocol
Image	Loaded Image Protocol
IPv4	IPv4 Protocol

IPv4Config	IPv4 Configuration Protocol
IPv4Sb	IPv4 Service Binding Protocol
IsaAcpi	ISA ACPI Protocol
Isalo	ISA I/O Protocol
LegacyBoot	Legacy Boot Protocol
Load	Load File Protocol
MNP	Managed Network Protocol
MNPSb	Managed Network Service Binding Protocol
MTFTPV4	MTFTPV4 Protocol
Net	Simple Network Protocol
Nii	Network Interface Identifier Protocol
Pcilo	PCI I/O Protocol
PciRootBridgelo	PCI Root Bridge I/O Protocol
Pxebc	PXE Base Code Protocol
Scsilo	SCSI I/O Protocol
ScsiPassThru	SCSI Pass Thru Protocol
Seriallo	Serial I/O Protocol
SimplePointer	Simple Pointer Protocol
StdErr	Standard Error Device
Tcp	TCP Protocol
TCPv4	TCPv4 Protocol
TCPv4Sb	TCPv4 Service Binding Protocol
Txtin	Simple Text-in Protocol
TxtInSplit	Text-in Splitter Protocol
Txtout	Simple Text Output Protocol
TxtOutSplit	Text-out Splitter Protocol
UDPV4	UDPV4 Protocol
UDPV4Sb	UDPV4 Service Binding Protocol
UgaDraw	UGA Draw Protocol
Ugalo	UGA I/O Protocol
UnicodeCollation	Unicode Collation Protocol
UsbHc	USB Host Controller Protocol
UsbHc2	USB2 Host Controller Protocol
Usblo	USB I/O Protocol
VgaClass	VGA Class Driver Protocol

**Note:** If the **Handle** parameter is required in the Shell command argument—for example, in the [dh](#) or [disconnect](#) command—it actually indicates the handle index, not the real handle. For convenience, the EFI Shell views the handle index as the equivalent of the real handle to operate an image handle, device handle, and so on. As a result, users should also use the handle index instead of the real handle value in Shell commands. The handle index can change after some hardware changes or the execution of some commands, such as [disconnect](#), [connect](#), and [reconnect](#). To get the current handle index for the specified device, driver, or image, use the [dh](#) command.

**Note:** In some file-operation-related Shell commands, such as [load](#) and [attrib](#), an argument is listed as **File...** or **Directory...** in the syntax. The suffix "..." here indicates that the arguments can be one or more files/directories. The command will process these arguments in the order in which they appear on the command line. Any previous execution failures will not prevent the following commands from executing. For example, **attrib +h file1 file2 file3** is a legal command and has the same execution results as the following command sequence:

```
attrib +h file1
```

```
attrib +h file2
```

```
attrib +h file3
```

**Note:** If any execution among them fails, the next command will continue executing until all the arguments in the list have been executed.

## 2.2 Commands from Default Build Shell

### 2.2.1 alias

#### Summary

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

#### EFI Versions

EFI 1.02 and above.

#### Usage

```
ALIAS [-d|-v] [sname] [value]
      -d      - Deletes an alias
      -v      - Volatile variable
      sname   - Alias name
      value   - Original name
```

#### Description

This command displays, creates, or deletes aliases in the EFI Shell environment. An alias provides a new name for an existing EFI Shell command or an EFI application. Once the alias is created, it can be used to run the command or launch the EFI application. There are some aliases that are predefined in the EFI Shell environment. These aliases provide the MS-DOS and UNIX equivalent names for the file manipulation commands. The examples below show the default aliases that are available in the EFI Shell.



## Examples

```
Shell> help alias
Displays, creates, or deletes aliases in the EFI Shell environment.
```

```
ALIAS [-d|-v] [sname] [value]

-d      - Deletes an alias
-v      - Volatile variable
sname   - Alias name
value   - Original name
```

### Note:

1. 'sname' should not be an internal EFI Shell command.
2. 'value' can be an internal EFI Shell command, a script, or an EFI application. However, any other values are also acceptable.
3. ALIAS values are stored in EFI NVRAM and will be retained between boots unless the '-v' option is specified.
4. ALIAS will not add a nonvolatile alias when a volatile alias of the same name already exists, or vice versa.

### Examples:

- \* To display all aliases in the EFI Shell environment:

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

- \* To create an alias in the EFI Shell environment:

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

- \* To delete an alias in the EFI Shell environment:

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

- \* To add a volatile alias in the current EFI environment, which has a star \* at the line head. This volatile alias will disappear at next boot.

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

## 2.2.2 attrib

### Summary

Displays or changes the attributes of files or directories.

### EFI Versions

EFI 1.02 and above.

## Usage

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

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

## Description

This command displays and sets the attributes of files or directories. The following four attribute types are supported in the EFI file system:

- Archive [A]
- System [S]
- Hidden [H]
- Read only [R]

If a file (in general meaning) is a directory, then it is also shown to have the attribute [D].

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

## Examples

```
Shell> help attrib
Displays or changes the attributes of files or directories.

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

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

### Notes:

1. If no attributes parameters are specified, the current attributes of the specified files or directories will be displayed.
2. If no files or directories are specified, then the command applies to all files and sub-directories within the current directory.

### Examples:

- \* To display the attributes of a directory:  
fs0:\> attrib fs0:\  
attrib:D fs0:\
- \* To display the attributes of all files and sub-directories in the current directory:  
fs0:\> attrib \*  
attrib: AS fs0:\serial.efi  
attrib:DA fs0:\test1  
attrib: A HR fs0:\bios.inf  
attrib: A fs0:\VerboseHelp.txt  
attrib: AS fs0:\IsaBus.efi
- \* To add the system attribute to all files with extension '.efi':  
fs0:\> attrib +s \*.efi
- \* To remove the read only attribute from all files with extension '.inf':  
fs0:\> attrib -r \*.inf  
attrib: A H fs0:\bios.inf

## 2.2.3 cd

### Summary

Displays or changes the current directory.

### EFI Versions

EFI 1.02 and above.

## Usage

**CD [path]**  
**path** - The relative or absolute directory path

## Description

This command changes the current working directory that is used by the EFI Shell environment. The table below describes the conventions that are used to refer to the directory, its parent, and its driver mapping in the EFI Shell environment.

**Table 4 Conventions for Directory Names**

Convention	Description
.	Refers to the current directory.
..	Refers to the directory's parent.
\	Refers to the root of the current driver mapping.

The following example shows how to move between the directories on a floppy drive containing an **EFI** directory and a **TOOLS** subdirectory below the **EFI** directory.

## Examples

```
Shell> help cd
Displays or changes the current directory.
```

```
CD [path]
path      - The relative or absolute directory path
```

### Note:

1. Type CD without parameters to display the current fs and directory.
2. There must be at least one blank space between CD and path.
3. The 'path' parameter supports certain special characters:
  - '.' refers to the current directory.
  - '..' refers to the parent directory.
  - '\' used at the beginning of the path refers to the root directory of the current filesystem.
4. CD can only be used to change directories in the current file system.

### Examples:

- \* To change the current filesystem to the mapped fs0 filesystem:  
Shell> fs0:
- \* To change the current directory to subdirectory 'efi':  
fs0:\> cd efi
- \* To change the current directory to the parent directory (fs0:\):  
fs0:\efi> cd ..
- \* To change the current directory to 'fs0:\efi\tools':  
fs0:\> cd efi\tools
- \* To change the current directory to the root of the current fs (fs0):  
fs0:\efi\tools> cd \  
fs0:\>
- \* To change volumes with cd will not work!! For example:  
fs0:\efi\tools> cd fs1:\ !!!! will not work !!!!  
must first type fs1: then cd to desired directory
- \* To move between volumes and maintain the current path.  
fs0:\> cd \efi\tools  
fs0:\efi\tools> fs1:  
fs1:\> cd tmp  
fs1:\tmp> cp fs0:\*. \* .  
copies all of files in fs0:\efi\tools into fs1:\tmp directory

## 2.2.4 cls

### Summary

Clears the standard output and optionally changes the background color.

### EFI Versions

EFI 1.02 and above.

## Usage

```
CLS [color]

color      - New background color
            0   - Black
            1   - Blue
            2   - Green
            3   - Cyan
            4   - Red
            5   - Magenta
            6   - Yellow
            7   - Light gray
```

## Description

This command clears the standard output device with an optional background color attribute. If **color** is not defined, then the background color does not change.

## Examples

```
Shell> help cls
Clears the standard output and optionally changes the background color.
```

```
CLS [color]

color      - New background color
            0   - Black
            1   - Blue
            2   - Green
            3   - Cyan
            4   - Red
            5   - Magenta
            6   - Yellow
            7   - Light gray
```

### Note:

1. If no parameters are specified, this command clears the standard output device. The background color is not changed.

### Examples:

- \* To clear standard output without changing the background color:  
fs0:\> cls
- \* To clear standard output and change the background color to cyan:  
fs0:\> cls 3
- \* To clear standard output and change the background to the default color:  
fs0:\> cls 0

## 2.2.5 connect

### Summary

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

### EFI Versions

EFI 1.10 and above.

## Usage

```
CONNECT [[DeviceHandle] [DriverHandle] | [-c] | [-r]]

-r          - Connect recursively
-c          - Connect console devices described in the EFI
              Environment Variables
DeviceHandle - Device handle, always taken as hexadecimal number
DriverHandle - Driver handle, always taken as hexadecimal number
```

## Description

This command binds a driver to a specific device and starts the driver. If the **-r** flag is used, then the connection is done recursively until no further connections between devices and drivers are made. If the **-c** flag is used, then the **connect** command will bind the proper drivers to the console devices that are described in the EFI environment variables. The example below shows the typical output from the verbose help for this command.

## Examples

```
Shell> help connect
Connects one or more EFI drivers to a device.
```

```
CONNECT [[DeviceHandle] [DriverHandle] | [-c] | [-r]]
```

```
DeviceHandle - Device handle in hexadecimal format
DriverHandle - Driver handle in hexadecimal format
-c           - Connect console devices
-r           - Connect recursively
```

### Note:

1. If no 'DeviceHandle' parameter is specified, all device handles in the current system will be the default.
2. If no 'DriverHandle' parameter is specified, all matched drivers will be bound to the specified device.
3. If 'DriverHandle' parameter is provided, the specified driver will have highest priority on connecting the device(s).
4. If the '-c' option is specified, only console devices described in the EFI Shell environment variables and related devices will be connected.
5. If the '-r' option is specified, the command will recursively scan all handles and check to see if any loaded or embedded driver can match the specified device. If so, the driver will be bound to the device.  
Additionally, if more device handles are created during the binding, these handles will also be checked to see if a matching driver can bind to these devices as well. The process is repeated until no more drivers are able to connect to any devices. However, without the option, the newly created device handles will not be further bound to any drivers.
6. If only a single handle is specified and the handle has an `EFI_DRIVER_BINDING_PROTOCOL` on it, then the handle is assumed to be a driver handle. Otherwise, it is assumed to be a device handle.
7. If no parameters are specified, then the command will attempt to bind all proper drivers to all devices without recursion. Each connection status will be displayed.
8. Output redirection is not supported for 'connect -r' usage.

### Examples:

- \* To connect all drivers to all devices recursively:  
Shell> connect -r
- \* To display all connections:  
Shell> connect  
ConnectController(1) : Status = Success  
ConnectController(2) : Status = Success  
ConnectController(3) : Status = Success  
...  
ConnectController(3D) : Status = Success
- \* To connect drivers with 0x17 as highest priority to all the devices they can manage:  
Shell> connect 17
- \* To connect all possible drivers to device 0x19:  
Shell> connect 19
- \* To connect drivers with 0x17 as highest priority to device 0x19 they can manage:  
Shell> connect 19 17
- \* To connect console devices described in the EFI Shell environment variables:  
Shell> connect -c



## 2.2.6 cp

### Summary

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

### EFI Versions

EFI 1.02 and above.

### Usage

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

-r      - Recursive copy
-q      - Quiet copying (replace existing files without
          prompt)
src     - Source file/directory name (wildcards are
          permitted)
dst     - Destination file/directory name (wildcards are not
          permitted)
```

### Description

This command copies one or more source files or directories to a destination. If the source is a directory, the **-r** flag must be specified. If **-r** is specified, then the source directory will be recursively copied to the destination (which means that all subdirectories will be copied). If a destination is not specified, then the current working directory is assumed to be the destination.

If any target file (not directory) already exists, there will be a prompt asking the user to confirm replacing the file. The following four choices are available:

- Yes: Replace the file.
- No: Do not replace the file.
- All: Replace the existing files in all subsequent cases.
- Cancel: Do not replace any existing files in all subsequent cases.

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

## Examples

```
Shell> help cp
Copies one or more files/directories to another location.
```

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

```

-r      - Recursive copy
-q      - Quiet copying (replace existing files without
          prompt)
src     - Source file/directory name (wildcards are
          permitted)
dst     - Destination file/directory name (wildcards are not
          permitted)
```

### Note:

1. '-r' must be specified if src is a directory. If '-r' is specified, then the source directory will be recursively copied to the destination. Src itself will be copied.
2. If 'dst' parameter is not specified, then the current directory is assumed to be the destination.
3. 'Cp -r src1 src2 dst' is to copy all files and subdirectories in 'src1' and 'src2' to the destination 'dst'. 'Src1' and 'src2' themselves are also copied. 'dst' parameter will be interpreted as a directory.
4. Copying a directory/file to itself is not allowed.
5. If an error occurs, Cp will exit immediately and the remaining files or directories will not be copied.
6. When 'cp' is executed with a script file, it always performs Quiet copy regardless of whether the '-q' option is specified.
7. If you are copying multiple files, the destination must be an existing directory.

### Examples:

- \* To display the contents of current directory first of all:  
fs0:\> ls  
Directory of: fs0:\

```

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

- \* To copy a file in the same directory, but change the file name:  
fs0:\> cp temp.txt readme.txt  
copying fs0:\temp.txt -> fs0:\readme.txt  
- [ok]
- \* To copy multiple files to another directory:  
fs0:\> cp temp.txt isaBus.efi \test  
copying fs0:\temp.txt -> fs0:\test\temp.txt  
- [ok]  
copying fs0:\isaBus.efi -> fs0:\test\IsaBus.efi  
- [ok]
- \* To copy multiple directories recursively to another directory:  
fs0:\> cp -r test1 test2 boot \test  
copying fs0:\test1 -> fs0:\test\test1  
copying fs0:\test1\test1.txt -> fs0:\test\test1\test1.txt  
- [ok]

```

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

* To see the results of the above operations:
fs0:\> ls \test
Directory of: fs0:\test

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

```

Shell>

## 2.2.7 date

### Summary

Displays and sets the current date for the system.

### EFI Versions

EFI 1.02 and above.

### Usage

```

DATE [mm/dd/[yy]yy]

mm      - Month of date to be set, Month range: 1 - 12
dd      - Day of date to be set, Day range: 1 - 31
yyyy    - Year of date to be set, Year range: 1998 - 2099

```

### Description

This command displays and/or sets the current date for the system. If no parameters are used, it shows the current date. If a valid month, day, and year are provided, then the system's date will be updated. Detailed rules are listed below:

1. Except for numeric characters and /, all other characters in the argument are invalid. The Shell will report an error if the number is in the wrong month/date/year range.
2. Space before or after the numeric character is not allowed. Inserting a space into the number is invalid.
3. Repeated zeros are allowed before the number. For example:

```
Shell > date 0000008/0000004/000097
Shell > date
08/04/2097
Shell >
```

4. The year range is greater than or equal to 1998. Two numeric characters indicate the year. Numbers below 98 are regarded as 20xx, and numbers equal to or above 98 are regarded as 19xx. 00 means 2000. For example:

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

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

5. The range of valid years is from 1998–2099.

## Examples

```
Shell> help date
Displays the current date or sets the date in the system.
```

```
DATE [mm/dd/[yy]yy]
```

```
mm      - Month of date to be set, Month range: 1 - 12
dd      - Day of date to be set, Day range: 1 - 31
yyyy    - Year of date to be set, Year range: 1998 - 2099
```

Note:

1. yy: 98=1998, 99=1999, 00=2000, 01=2001, ..., 97=2097.
2. yyyy: 1998 - 2099, other values are invalid.
3. EFI may behave unpredictably if illegal date values are used.

Examples:

- \* To display the current date in the system:

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

- \* To set the date with long year format:

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

- \* To set the date with short year format:

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

```
Shell>
```

## 2.2.8 del

### Summary

Deletes one or more files or directories.

## EFI Versions

EFI 1.02 and above.

## Usage

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

-q	- Quiet mode; does not prompt user for a confirmation
file	- File name (wildcards are permitted)
directory	- Directory name (wildcards are permitted)

## Description

This command deletes one or more files or directories. If the target is a directory, it will delete the directory, including all its subdirectories. It is not allowed to redirect a file whose parent directory (or the file itself) is being deleted.

## Examples

```
Shell> help del
Deletes one or more files or directories.
```

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

```

-q           - Quite mode; does not prompt user for a
               confirmation
file         - File name (wildcards are permitted)
directory    - Directory name (wildcards are permitted)
```

### Note:

1. Removing a read-only file/directory will result in a failure. Removing a directory containing read-only file(s) will result in a failure.
2. If an error occurs, DEL will exit immediately and later files/directories will not be removed.
3. You cannot remove a directory when the current directory is itself or its subdirectory.
4. If file contains wildcards, it will not ask user for confirmation.
5. You cannot remove the root directory.
6. You cannot remove the current directory or its ancestor.
7. Redirecting output to a file that exists under the directory that will be removed is not allowed.

### Examples:

- \* To remove multiple directories at a time:

```
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)
```

- \* Error occurs and DEL will exit:

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

- \* To remove multiple directories with wildcards:

```
fs0:\> del 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]
```

- \* Removing a directory that contains a read-only file will fail:

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

```
fs0:\> del test\temp1
rm/del: Cannot open 'readme.txt' under 'fs0:\test\temp1' in
```

```
writable mode
- [error] - Access Denied
Exit status code: Access Denied

Shell>
```

## 2.2.9 dh

### Summary

Displays the device handles in the EFI environment.

### EFI Versions

EFI 1.10 and above.

### Usage

```
DH [-l <lang>] [handle | -p <prot_id>] [-d] [-v]

handle      - Dumps information of a specified handle, always
              taken as hexadecimal number
-p          - Dumps all handles of a protocol specified by
              prot_id
-d          - Dumps EFI Driver Model related information
-l          - Dumps information using the ISO 639-2 language
              specified by lang.
-v          - Dumps verbose information on specified handle
```

### Description

This command displays the device handles in the EFI environment. If this command is used with a specific handle number, the details of all the protocols that are associated with that device handle are displayed. Otherwise, the **-p** option can be used to list the device handles that contain a specific protocol. See [Supported EFI Protocols in the Shell](#) for the abbreviations that are used with this command for EFI protocols. The following examples show how the command can be used.

## Examples

```
Shell> help dh
Displays the handles in the EFI environment.
```

```
DH [-l <lang>] [handle | -p <prot_id>] [-d] [-v]
```

```

    handle      - Dumps information of a specified handle,
                  always taken as hexadecimal number
    -p          - Dumps all handles of a protocol specified by
                  prot_id
    -d          - Dumps EFI Driver Model related information
    -l          - Dumps information using the ISO 639-2 language
                  specified by lang.
    -v          - Dumps verbose information on specified handle
```

### Note:

1. When neither 'handle' nor 'prot\_id' is specified, a list of all the handles in the EFI environment is displayed.
2. Option '-d' can be used to display EFI Driver Model related information, including its parent handles, child handles, all drivers on it, etc.
3. Option '-v' can be used to display verbose information on the specified handle, including all the protocols on it and their details.
4. If option '-p' is specified, all handles containing the specified protocol will be displayed. Otherwise, the 'handle' parameter has to be specified for display. In this case, option '-d' will be enabled automatically if option '-v' is not specified.

### Examples:

- \* To display all handles and display one screen at a time:

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

- \* To display the detailed information on handle 0x30:

```
Shell> dh 30
Handle 30 (01AF5308)
  IsaIo
    ROM Size.....: 00000000
    ROM Location..: 00000000
    ISA Resource List :
      IO  : 000003F8-000003FF  Attr : 00000000
      INT : 00000004-00000000  Attr : 00000000

  dpath
    PNP Device Path for PnP
    HID A0341D0, UID 0x0
    Hardware Device Path for PCI
    PNP Device Path for PnP
    HID 50141D0, UID 0
    AsStr: 'Acpi(PNP0A03,0)/Pci(1F|0)/Acpi(PNP0501,0)'
```

- \* To display all handles with 'diskio' protocol:

```
Shell> dh -p diskio
Handle dump by protocol 'Diskio'
15: DiskIo BlkIo DevPath(..i(3|1)/Ata(Secondary,Master))
```



```

16: DiskIo BlkIo DevPath(..,1)/PCI(0|0)/Scsi(Pun0,Lun0))
44: DiskIo BlkIo Fs DevPath(..ABD0-01C0-507B-9E5F8078F531))
    ESP
45: DiskIo BlkIo Fs DevPath(..i(Pun0,Lun0)/HD(Part4,SigG0))
    ESP
17: DiskIo BlkIo DevPath(..PCI(3|1)/Ata(Primary,Master))

* To display all handles with 'Image' protocol and break when
  the screen is full:
Shell> dh -p Image -b
Handle dump by protocol 'image'
  1: Image(DXE Core)
  5: Image(WinNtThunk)
  7: Image(WinNtBusDriver) DriverBinding
  8: Image(Metronome)
  A: Image(IsaBus) DriverBinding
  B: Image(WinNtConsole) DriverBinding
...
Shell>

```

## 2.2.10 dir

### Summary

Lists directory contents or file information.

### EFI Versions

EFI 1.02 and above.

### Usage

```

DIR [-r] [-a[attrib]] [file]

-r          - Displays recursively (including subdirectories)
attrib      - 'a', 's', 'h', 'r', 'd' or combination of them
               a  - Archive
               s  - System
               h  - Hidden
               r  - Read-only
               d  - Directory
file        - Name of file/directory (wildcards are permitted)

```

### Description

This command lists directory contents or file information. If no file name or directory name is specified, then the current directory is assumed. 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

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]**. If more than one attribute is specified, only the files that have all those attributes will be listed. If **-a** is followed by nothing, then all files/directories are displayed, regardless of their attributes. If **-a** itself is not specified, then all files except system and hidden files are displayed.

## Examples

Shell> help dir  
Displays a list of files and subdirectories in a directory.

DIR [-r] [-a[attrib]] [file]

-r            - Displays recursively (including subdirectories)  
attrib       - 'a', 's', 'h', 'r', 'd' or combination of them  
             a     - Archive  
             s     - System  
             h     - Hidden  
             r     - Read-only  
             d     - Directory  
file         - Name of file/directory (wildcards are permitted)

Examples:

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

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

- \* To display all, except the files/directories with 'h' or 's' attribute:

```
fs0:\> dir
Directory of: fs0:\

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

- \* To display files with all attributes in the current directory:

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

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

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

```
fs0:\> dir -ar
Directory of: fs0:\

06/18/01  11:14p                r    29  readme.txt
          1 File(s)              29 bytes
          0 Dir(s)
```

```

* To display the files with attribute of 's':
fs0:\> dir -as isabus.efi
Directory of: fs0:\

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

* To display all in fs0:\efi directory recursively:
fs0:\> dir -r -a efi

* To search for files with the specified type in the current
  directory recursively:
fs0:\> dir -r -a *.efi -b

Shell>

```

### 2.2.11 disconnect

#### Summary

Disconnects one or more drivers from the specified devices.

#### EFI Versions

EFI 1.10 and above.

#### Usage

```

DISCONNECT DeviceHandle [DriverHandle [ChildHandle]]
DISCONNECT -r

DeviceHandle  - Device handle, always taken as hexadecimal
               number
DriverHandle  - Driver handle, always taken as hexadecimal
               number
ChildHandle   - Child handle of a device, always taken as
               hexadecimal number
-r           - Disconnect drivers from all devices

```

#### Description

This command disconnects one or more drivers from the specified devices. If the **-r** option is used, all drivers are disconnected from all devices in the system. The following example is the typical output from the help for this command.

## Examples

```
Shell> help disconnect
Disconnects one or more EFI drivers from a device.

DISCONNECT DeviceHandle [DriverHandle [ChildHandle]]
DISCONNECT -r

    DeviceHandle - Device handle in hexadecimal format
    DriverHandle - Driver handle in hexadecimal format
    ChildHandle  - Child handle of device in hexadecimal format
    -r           - Disconnect drivers from all devices

Note:
    1. If the 'DriverHandle' parameter is not specified, the default is
       to disconnect 'DeviceHandle'.
    2. If the 'ChildHandle' parameter is not specified, the default is
       to disconnect all child handles of the 'DeviceHandle'.
    3. If the '-r' option is specified, all drivers will be
       disconnected from all devices in the system. In this case, no
       other parameters are allowed.
    4. This command does not support output redirection.

Examples:
    * To disconnect all drivers from all devices:
      Shell> disconnect -r

    * To disconnect all drivers from device 0x28:
      fs0:\> disconnect 28

    * To disconnect driver 0x17 from device 0x28:
      fs0:\> disconnect 28 17

    * To disconnect driver 0x17 from controlling the child 0x32 of device
      0x28
      fs0:\> disconnect 28 17 32
```

## 2.2.12 drivers

### Summary

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

### EFI Versions

EFI 1.10 and above.

### Usage

```
DRIVERS [-l XXX]

    -l           - Displays drivers using the ISO 639-2
                  language specified by XXX
```

### Description

This command displays a list of information for drivers that follow the EFI Driver Model in EFI environment. The list has the following columns:

- **DRV:** The handle number of the EFI driver.

- **VERSION:** The version number of the EFI driver.
- **TYPE:** The driver type. A **B** in this column indicates a bus driver, and **D** indicates a device driver.
- **CFG:** Indicates that the driver supports the Driver Configuration Protocol.
- **DIAG:** Indicates that the driver supports the Driver Diagnostics Protocol.
- **#D:** The number of devices that this driver is managing.
- **#C:** The number of child devices that this driver has produced.
- **DRIVER NAME:** The name of the driver from the Component Name Protocol.
- **IMAGE NAME:** The file path from which the driver was loaded.

## Examples

```
Shell> help drivers
Displays the list of drivers that follow the EFI Driver Model

DRIVERS [-l XXX]

    -l          - Displays drivers using the ISO 639-2
                  language specified by XXX

Display Format:
DRV - The handle number of the EFI driver
TYPE - The driver type
      [B] Bus driver
      [D] Device driver
CFG - The driver supports the Driver Configuration Protocol
DIAG - The driver supports the Driver Diagnostics Protocol
#D - The number of devices that this driver is managing
#C - The number of child devices that this driver has produced
DRIVER NAME - The name of the driver from the Component Name Protocol
IMAGE NAME - The file path from which the driver was loaded

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

Shell>
```

### 2.2.13 drvcfg

#### Summary

Invokes the Driver Configuration Protocol.

#### EFI Versions

EFI 1.10 and above.

## Usage

```
DRVCFG [-l XXX] [-c] [-f <Type>|-v|-s] [DriverHandle [DeviceHandle  
[ChildHandle]]]
```

```
-c          - Configure all child devices
-l          - Configure using the ISO 639-2 language
              specified by XXX
-f          - Force defaults
-v          - Validate options
-s          - Set options
Type        - The type of default configuration options to
              force on the controller.
              0 - Safe Defaults.
              1 - Manufacturing Defaults.
              2 - Custom Defaults.
              3 - Performance Defaults.
DriverHandle - The handle of the driver to configure
DeviceHandle - The handle of a device that DriverHandle is
              managing
ChildHandle  - The handle of a device that is a child of
              DeviceHandle
```

## Description

This command invokes the Driver Configuration Protocol. The table below describes the values for the **Type** parameter. Other values depend on the driver's implementation.

**Table 5 Default Values for the Type Parameter**

Value	Type of Default	Description
0x0000	Safe Defaults	Places a controller in a safe configuration that has the greatest probability of functioning correctly in a platform.
0x0001	Manufacturing Defaults	Optional type that places the controller in a configuration that is suitable for a manufacturing and test environment.
0x0002	Custom Defaults	Optional type that places the controller in a custom configuration.
0x0003	Performance Defaults	Optional type that places the controller in a configuration that maximizes the controller's performance in a platform.



## Examples

```
Shell> help drvcfg
Invokes the Driver Configuration Protocol

DRVCFG [-l XXX] [-c] [-f <Type>|-v|-s] [DriverHandle [DeviceHandle
[ChildHandle]]]

    -c          - Configure all child devices
    -l          - Configure using the ISO 639-2 language
                  specified by XXX
    -f          - Force defaults
    -v          - Validate options
    -s          - Set options
    Type        - The type of default configuration options to
                  force on the controller.
                  0 - Safe Defaults.
                  1 - Manufacturing Defaults.
                  2 - Custom Defaults.
                  3 - Performance Defaults.
    DriverHandle - The handle of the driver to configure
    DeviceHandle - The handle of a device that DriverHandle is
                  managing
    ChildHandle  - The handle of a device that is a child of
                  DeviceHandle
```

### Notes:

1. Default Type.
  - 0 - Safe Defaults. It places a controller in a safe configuration that has the greatest probability of functioning correctly in a platform.
  - 1 - Manufacturing Defaults. Optional type that places the controller in a configuration suitable for a manufacturing and test environment.
  - 2 - Custom Defaults. Optional type that places the controller in a custom configuration.
  - 3 - Performance Defaults. Optional type that places the controller in a configuration that maximizes the controller's performance in a platform.Other values depend on the driver's implementation.

### Examples:

- \* To display the list of devices that are available for configuration:

```
Shell> drvcfg
```
- \* To display the list of devices and child devices that are available for configuration:

```
Shell> drvcfg -c
```
- \* To force defaults on all devices:

```
Shell> drvcfg -f 0
```
- \* To force defaults on all devices that are managed by driver 0x17:

```
Shell> drvcfg -f 0 17
```
- \* To force defaults on device 0x28 that is managed by driver 0x17:

```
Shell> drvcfg -f 0 17 28
```
- \* To force defaults on all child devices of device 0x28 that is managed by driver 0x17:

```
Shell> drvcfg -f 0 17 28 -c
```

```
* To force defaults on child device 0x30 of device 0x28 that is
managed by driver 0x17:
Shell> drvcfg -f 0 17 28 30

* To validate options on all devices:
Shell> drvcfg -v

* To validate options on all devices that are managed by driver
0x17:
Shell> drvcfg -v 17

* To validate options on device 0x28 that is managed by driver
0x17:
Shell> drvcfg -v 17 28

* To validate options on all child devices of device 0x28 that
is managed by driver 0x17:
Shell> drvcfg -v 17 28 -c

* To validate options on child device 0x30 of device 0x28 that
is managed by driver 0x17:
Shell> drvcfg -v 17 28 30

* To set options on device 0x28 that is managed by driver 0x17:
Shell> drvcfg -s 17 28

* To set options on child device 0x30 of device 0x28 that is
managed by driver 0x17:
Shell> drvcfg -s 17 28 30

* To set options on device 0x28 that is managed by driver 0x17
in English:
Shell> drvcfg -s 17 28 -l eng

* To set options on device 0x28 that is managed by driver 0x17
in Spanish:
Shell> drvcfg -s 17 28 -l spa

Shell>
```

## 2.2.14 drvdiag

### Summary

Invokes the Driver Diagnostics Protocol.

### EFI Versions

EFI 1.10 and above.

## Usage

```
DRVDIAG [-c] [-l XXX] [-s|-e|-m] [DriverHandle [DeviceHandle  
[ChildHandle]]]
```

-c	- Diagnose all child devices
-l	- Diagnose using the ISO 639-2 language specified by XXX
-s	- Run diagnostics in standard mode
-e	- Run diagnostics in extended mode
-m	- Run diagnostics in manufacturing mode
DriverHandle	- The handle of the driver to diagnose
DeviceHandle	- The handle of a device that DriverHandle is managing
ChildHandle	- The handle of a device that is a child of DeviceHandle

## Description

This command invokes the Driver Diagnostics Protocol.

## Examples

```
Shell> help drvdiag
Invokes the Driver Diagnostics Protocol
```

```
DRVDIAG [-c] [-l XXX] [-s|-e|-m] [DriverHandle [DeviceHandle
[ChildHandle]]]
```

```

-c          - Diagnose all child devices
-l          - Diagnose using the ISO 639-2 language
              specified by XXX
-s          - Run diagnostics in standard mode
-e          - Run diagnostics in extended mode
-m          - Run diagnostics in manufacturing mode
DriverHandle - The handle of the driver to diagnose
DeviceHandle - The handle of a device that DriverHandle is
              managing
ChildHandle  - The handle of a device that is a child of
              DeviceHandle
```

### Examples:

- \* To display the list of devices that are available for diagnostics:  
Shell> drvdiag
- \* To display the list of devices and child devices that are available for diagnostics:  
Shell> drvdiag -c
- \* To run diagnostics in standard mode on all devices:  
Shell> drvdiag -s
- \* To run diagnostics in standard mode on all devices in English:  
Shell> drvdiag -s -l eng
- \* To run diagnostics in standard mode on all devices in Spanish:  
Shell> drvdiag -s -l spa
- \* To run diagnostics in standard mode on all devices and child devices:  
Shell> drvdiag -s -c
- \* To run diagnostics in extended mode on all devices:  
Shell> drvdiag -e
- \* To run diagnostics in manufacturing mode on all devices:  
Shell> drvdiag -m
- \* To run diagnostics in standard mode on all devices managed by driver 0x17:  
Shell> drvdiag -s 17
- \* To run diagnostics in standard mode on device 0x28 managed by driver 0x17:  
Shell> drvdiag -s 17 28
- \* To run diagnostics in standard mode on all child devices of device 0x28 managed by driver 0x17:  
Shell> drvdiag -s 17 28 -c
- \* To run diagnostics in standard mode on child device 0x30 of device 0x28 managed by driver 0x17:  
Shell> drvdiag -s 17 28 30

```
Shell>
```

## 2.2.15 echo

### Summary

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

### EFI Versions

EFI 1.02 and above.

### Usage

```
ECHO [-on|-off]  
ECHO [message]
```

- |         |  |
|---------|--|
| -on     | - Displays when reading command lines from batch files |
| -off    | - Does not display when reading batch command lines    |
| message | - Displays a message string                            |

### Description

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

## Examples

```
Shell> help echo
Displays a message, or turns command echoing on or off in batch files.
```

```
ECHO [-on|-off]
ECHO [message]
```

```
-on           - Displays when reading command lines from batch
               files
-off          - Does not display when reading batch command
               lines
message       - Displays a message string
```

Note:

1. Echo -off means to not display the command line when reading from batch files. This command is not like the MS-DOS echo.
2. Echo without a parameter shows the current echo setting.

Examples:

- \* To display a message string of 'Hello World':  
fs0:\> echo Hello World  
Hello World
- \* To turn command echoing on:  
fs0:\> echo -on
- \* To execute HelloWorld.nsh, and display when reading lines from the batch file:  
fs0:\> HelloWorld.nsh  
+HelloWorld.nsh> echo Hello World  
Hello World
- \* To turn command echoing off:  
fs0:\> echo -off
- \* To display the current echo setting:  
fs0:\> echo  
Echo is off

```
Shell>
```

## 2.2.16 exit

### Summary

Exits the EFI Shell environment and returns control to the parent that launched the EFI Shell.

### EFI Versions

EFI 1.02 and above.

### Usage

```
EXIT
```

### Description

This command exits the EFI Shell environment and returns control to the parent that launched the EFI Shell.

## Examples

```
Shell> help exit
Exits the EFI Shell environment and returns control to its parent.

EXIT

Examples:
  Shell> exit
```

## 2.2.17 help

### Summary

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

### EFI Versions

EFI 1.02 and above.

### Usage

```
HELP [cmd | pattern]

  cmd      - Shell command
  pattern  - Wildmatch pattern
```

### Description

The help command displays the list of commands that are built into the EFI Shell. It also supports displaying the verbose help information for a specified command.

You can also use `cmd -?` to display the verbose help of a command, where `cmd` is the name of the EFI Shell command or application. This syntax does the same thing as `help cmd` and can be used for both internal and external Shell commands. The form of `help cmd`, however, can be used only for internal Shell commands.

The following example shows the output from this command.

## Examples

Shell> help help  
Displays the list of commands or verbose help of a command in the EFI Shell.

```
HELP [cmd | pattern]

cmd      - Shell command
pattern - Wildmatch pattern
```

**Note:**

1. 'cmd -?' also displays the verbose help of cmd, the same as 'help cmd'.
2. If cmd has no verbose help, its line help will be displayed instead.
3. HELP will only show commands that were documented in the Shell.

**Examples:**

- \* To display the list of commands in the EFI Shell and break after one screen:  
Shell> help -b  
?  
- Displays commands list or verbose help of a command  
alias - Displays, creates, or deletes aliases in the EFI shell  
attrib - Displays or changes the attributes of files or directories  
cd - Displays or changes the current directory  
cls - Clears the standard output with an optional background color  
connect - Binds an EFI driver to a device and starts the driver  
copy - Copies one or more files/directories to another location  
...
- \* To display help information of a Shell command - ls:  
Shell> help ls  
Shell> ? ls  
Shell> ls -?
- \* To display the list of commands that start with character 'p':  
Shell> help p\*  
pause - Prints a message and suspends for keyboard input

## 2.2.18 load

### Summary

Loads an EFI driver into memory.

### EFI Versions

EFI 1.10 and above.



## Usage

```
LOAD [-nc] file [file...]

-nc      - Load the driver, but do not connect the driver.
file     - File that contains the image of the EFI driver
           (wildcards are permitted)
```

## Description

This command loads an EFI driver into memory. It can load multiple files at one time, and the file name supports the asterisk wildcard. If the **-nc** flag is not specified, this command will try to connect the driver to a proper device; meanwhile it may cause loaded drivers be connected to their corresponding devices. This action is not a bug but the implementation policy.

## Examples

```
Shell> help load
Loads EFI drivers and then they can provide available services.

LOAD [-nc] file [file...]

-nc      - Load the driver, but do not connect the driver.
file     - File that contains the image of the EFI driver
           (wildcards are permitted)

Note:
1. LOAD can deal with multiple files and supports wildcards.
2. Use the 'UNLOAD' command to unload a driver if it supports
   unloading.
3. If option -nc is not specified, then the loaded drivers will
   be automatically connected. If -nc is specified, then none of
   the loaded drivers will be connected. Loading without -nc
   could cause the previously loaded drivers to be connected.
   This is not a bug, and it complies with the EFI Specification.

Examples:
fs0:\> load Isabus.efi
load: Image 'fs0:\Isabus.efi' loaded at 18FE000 - Success

fs0:\> load Isabus.efi IsaSerial.efi
load: Image 'fs0:\Isabus.efi' loaded at 18E5000 - Success
load: Image 'fs0:\IsaSerial.efi' loaded at 18DC000 - Success

fs0:\> load Isa*.efi
load: Image 'fs0:\IsaBus.efi' loaded at 18D4000 - Success
load: Image 'fs0:\IsaSerial.efi' loaded at 18CB000 - Success

fs0:\> load -nc IsaBus.efi
load: Image 'fs0:\Isabus.efi' loaded at 18FE000 - Success

Shell>
```

## 2.2.19 ls

### Summary

Lists a directory's contents or file information.

## EFI Versions

EFI 1.02 and above.

## Usage

```
LS [-r] [-a[attrib]] [file]

-r          - Displays recursively (including subdirectories)
attrib     - 'a', 's', 'h', 'r', 'd' or combination of them
              a      - Archive
              s      - System
              h      - Hidden
              r      - Read-only
              d      - Directory
file       - Name of file/directory (wildcards are
              permitted)
```

## Description

This command lists directory contents or file information. If no file name or directory name is specified, then the current directory is assumed. 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

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]**. If more than one attribute is specified, only the files that have all those attributes will be listed. If **-a** is followed by nothing, then all files/directories are displayed, regardless of their attributes. If **-a** itself is not specified, then all files except system and hidden files are displayed.

## Examples

Shell> help ls  
Displays a list of files and subdirectories in a directory.

LS [-r] [-a[attrib]] [file]

-r            - Displays recursively (including subdirectories)  
attrib       - 'a', 's', 'h', 'r', 'd' or combination of them  
              a     - Archive  
              s     - System  
              h     - Hidden  
              r     - Read-only  
              d     - Directory  
file         - Name of file/directory (wildcards are permitted)

Examples:

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

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

- \* To display all, except the files/directories with 'h' or 's' attribute:

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

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

- \* To display files with all attributes in the current directory:

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

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

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

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

06/18/01 11:14p                r    29  readme.txt
          1 File(s)              29 bytes
          0 Dir(s)
```

```

* To display the files with attribute of 's':
fs0:\> ls -as isabus.efi
Directory of: fs0:\

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

* To display all in fs0:\efi directory recursively:
fs0:\> ls -r -a efi

* To search for files with the specified type in the current
directory recursively:
fs0:\> ls -r -a *.efi -b

Shell>

```

## 2.2.20 map

### Summary

Defines a mapping between a user-defined name and a device handle.

### EFI Versions

EFI 1.02 and above.

### Usage

```

MAP [-d <sname>]
MAP [[-r | -u] [-v] [-c] [-f] [-t <type[,type...]>] [sname]]
MAP [sname handle | mapname]

-d      - Deletes a mapping
-r      - Resets to default mappings
-u      - Update mappings
-v      - Lists verbose information of mappings
sname   - Defines a name for the mapping by users
handle  - The number of handle, which is same as dumped from
           'dh'
           command
-c      - Shows the consistent mapping name.
-f      - Shows the normal mapping name(not consiste
           mapping).
-t      - Shows the device mapping name according to the
           device type.
type     - The device type. The current supported types are:
           fp (Floppy)
           hd (Hard Disk)
           cd (CD Rom)
           Types can be combined by putting a comma between
           two types. Spaces are not allowed between types.
mapname - The device's mapped name. Use this parameter to
           assign a new mapping name to a device. There is a
           postfix ':' after the mapname.

```

### Description

This command is used to define a mapping between a user-defined name and a device handle. The most common use of this command is to assign drive letters to device

handles that support a file system protocol. Once these mappings are created, the drive letters can be used with all the file manipulation commands.

The EFI Shell environment creates default mappings for all the device handles that support a recognized file system.

This command can be used to create additional mappings, or it can be used to delete an existing mapping with the **-d** option. If the **map** command is used without any parameters, all the current mappings will be listed. If the **-v** option is used, the mappings will be shown with additional information about each mapped handle. The **-r** option is used to regenerate all the default mappings in a system; this option is useful if the system configuration has changed since the last boot.

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

Each device in the system has a consistent mapping name. If the hardware configuration has not changed, the device's consistent mapping names do not change. If two or more machines have the same hardware configurations, the device's consistent mapping will be the same. Use the **-c** option to list all the consistent mapping names in the system.

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

This command support wildcards. You can use the wildcards to delete or show the mapping name. However, when you assign the mapping name, wildcards are forbidden.

## Examples

```
Shell> help map
Displays or defines mappings between user-defined names and device handles.
```

```
MAP [-d <sname>]
MAP [[-r | -u] [-v] [-c] [-f] [-t <type[,type...]>] [sname]]
MAP [sname handle | mapname]

    -d      - Deletes a mapping
    -r      - Resets to default mappings
    -u      - Update mappings
    -v      - Lists verbose information of mappings
    sname   - Defines a name for the mapping by users
    handle  - The number of handle, which is same as dumped from
               'dh'
               command
    -c      - Shows the consistent mapping name.
    -f      - Shows the normal mapping name (not consistent
               mapping).
    -t      - Shows the device mapping name according the device
               type.
    type    - The device type.The currrent supported types are:
               fp (Floppy)
               hd (Hard Disk)
               cd (CD Rom)
               Types can be combined by putting a comma between
               two types. Spaces are not allowed between types.
    mapname - The device's mapped name. Use this parameter to
               assign a new mapping name to a device. There is a
               postfix ':' after the mapname.
```

### Note:

1. Consistent mapping is persistent across the 'map -r' command and a system reboot.
2. Only characters and numbers are allowed inside of sname.
3. Redirection is not allowed when running map, because we do not know the file system before mapping is done.
4. Output redirection is not supported for 'map -r' usage.
5. Option '-u' will only add/delete the mappings for new/removed devices. The unchanged device mappings with user-defined mappings are unchanged. The current dir, if current device is not changed, will also be preserved.

### Examples:

```
* To reset the mapping table to the default mappings:
shell> map -r
Device mapping table
f4      :UnknownDevice - Alias fs0 blk0
        Device Path  VenHw(58C518B1-76F3-11D4-BCEA-0080C73C8881)
        /VenHw(0C95A92F-A006-11D4-BCFA-0080C73C8881)
fs0     :UnknownDevice - Alias f4 blk0
        Device Path  VenHw(58C518B1-76F3-11D4-BCEA-0080C73C8881)
        /VenHw(0C95A92F-A006-11D4-BCFA-0080C73C8881)
blk0    :UnknownDevice - Alias f4 fs0
        Device Path  VenHw(58C518B1-76F3-11D4-BCEA-0080C73C8881)
        /VenHw(0C95A92F-A006-11D4-BCFA-0080C73C8881)
```

```

* To display all mappings in the device mapping table:
Shell> map
Device mapping table
  f4      :UnknownDevice - Alias fs0 blk0
           Device Path  VenHw(58C518B1-76F3-11D4-BCEA-0080C73C8881)
                           /VenHw(0C95A92F-A006-11D4-BCFA-0080C73C8881)
  fs0     :UnknownDevice - Alias f4 blk0
           Device Path  VenHw(58C518B1-76F3-11D4-BCEA-0080C73C8881)
                           /VenHw(0C95A92F-A006-11D4-BCFA-0080C73C8881)
  blk0    :UnknownDevice - Alias f4 fs0
           Device Path  VenHw(58C518B1-76F3-11D4-BCEA-0080C73C8881)
                           /VenHw(0C95A92F-A006-11D4-BCFA-0080C73C8881)

* To display mapping table verbosely:
Shell> map -v
Device mapping table
  f4      Consist Name  f4
           Other Name   fs0 blk0
           Handle       5F: Fs DiskIo BlkIo WinNtDriverIo
           Media Type   UnknownDevice
           Removeable   NO
           Current Dir  \
           Device Path  VenHw(58C518B1-76F3-11D4-BCEA-0080C73C8881)
                           /VenHw(0C95A92F-A006-11D4-BCFA-0080C73C8881)
  fs0     Consist Name  f4
           Other Name   blk0
           Handle       5F: Fs DiskIo BlkIo WinNtDriverIo
           Media Type   UnknownDevice
           Removeable   NO
           Current Dir  \
           Device Path  VenHw(58C518B1-76F3-11D4-BCEA-0080C73C8881)
                           /VenHw(0C95A92F-A006-11D4-BCFA-0080C73C8881)
  blk0    Consist Name  f4
           Other Name   fs0
           Handle       5F: Fs DiskIo BlkIo WinNtDriverIo
           Media Type   UnknownDevice
           Removeable   NO
           Current Dir  \
           Device Path  VenHw(58C518B1-76F3-11D4-BCEA-0080C73C8881)
                           /VenHw(0C95A92F-A006-11D4-BCFA-0080C73C8881)

* To assign fs0 another name:
Shell> map floppy fs0:
floppy:UnknownDevice - Alias f4 fs0 blk0
           Device Path  VenHw(58C518B1-76F3-11D4-BCEA-0080C73C8881)
                           /VenHw(0C95A92F-A006-11D4-BCFA-0080C73C8881)

* To display the information of the mapped name:
Shell> map floppy
floppy:UnknownDevice - Alias f4 fs0 blk0
           Device Path  VenHw(58C518B1-76F3-11D4-BCEA-0080C73C8881)
                           /VenHw(0C95A92F-A006-11D4-BCFA-0080C73C8881)

* To operate with the mapped name:
Shell> floppy:
floppy:\> ls

```

```

* To delete a mapped name:
Shell> map -d floppy
Shell> map
Device mapping table
  f4      :UnknownDevice - Alias fs0 blk0
          Device Path  VenHw(58C518B1-76F3-11D4-BCEA-0080C73C8881)
                      /VenHw(0C95A92F-A006-11D4-BCFA-0080C73C8881)
  fs0     :UnknownDevice - Alias f4 blk0
          Device Path  VenHw(58C518B1-76F3-11D4-BCEA-0080C73C8881)
                      /VenHw(0C95A92F-A006-11D4-BCFA-0080C73C8881)
  blk0    :UnknownDevice - Alias f4 fs0
          Device Path  VenHw(58C518B1-76F3-11D4-BCEA-0080C73C8881)
                      /VenHw(0C95A92F-A006-11D4-BCFA-0080C73C8881)

* To display all the mapped names that start with 'f':
Shell> map f*
Device mapping table:
  f4      :UnknownDevice - Alias fs0 blk0
          Device Path  VenHw(58C518B1-76F3-11D4-BCEA-0080C73C8881)
                      /VenHw(0C95A92F-A006-11D4-BCFA-0080C73C8881)
  fs0     :UnknownDevice - Alias f4 blk0
          Device Path  VenHw(58C518B1-76F3-11D4-BCEA-0080C73C8881)
                      /VenHw(0C95A92F-A006-11D4-BCFA-0080C73C8881)

```

### 2.2.21 mkdir

#### Summary

Creates one or more new directories.

#### EFI Versions

EFI 1.02 and above.

#### Usage

```

MKDIR dir [dir...]

dir      - Name of a directory to be created (wildcards are
          not allowed).

```

#### Description

This command creates one or more new directories.



## Examples

```
Shell> help mkdir
Creates one or more directories.

MKDIR dir [dir...]

    dir        - Name of a directory to be created(wildcards are not
                  allowed)

Note:
1. The parent directory must already exist.
2. If the directory already exists, it will fail to make such a
   directory.
3. The directories in the command line should not rely on the
   creation of other directories in the command line. For
   example, 'mkdir new new\test' is not allowed.
4. Redirecting output to a file that exists under the directory
   specified on the command line is not allowed.

Examples:
* To create a new directory:
fs0:\> mkdir rafter
fs0:\> ls
Directory of: fs0:\

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

* To create multiple directories:
fs0:\> mkdir temp1 temp2
fs0:\> ls
Directory of: fs0:\

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

Shell>
```

## 2.2.22 mv

### Summary

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

### EFI Versions

EFI 1.02 and above.

## Usage

```
MV src [src...] [dst]
```

```
src      - Source file/directory name (wildcards are
           permitted)
dst      - Destination file/directory name (wildcards are not
           permitted)
```

## Description

This command moves one or more files to a destination within a file system. Moving between file system volumes is not supported. If the destination is an existing directory, then the sources are moved into that directory. Otherwise, the sources are moved to the destination, as if the directory has been renamed. If a destination is not specified, the current directory is assumed to be the destination.

## Examples

```
Shell> help mv
```

```
Moves one or more files/directories to destination within fs.
```

```
MV src [src...] [dst]
```

```
src      - Source file/directory name (wildcards are
           permitted)
dst      - Destination file/directory name (wildcards are not
           permitted)
```

Note:

1. If 'dst' is not specified, then the current directory is assumed to be the 'dst'.
2. If there is more than one argument in the command line, the last one will be taken as 'dst' unconditionally. If there is more than one source file/directory to move, the 'dst' should be an existing directory.
3. Attempting to move a read-only file/directory will result in failure.
4. Moving a directory that contains read-only file(s) is allowed.
5. You cannot move a directory into itself or its subdirectories.
6. You cannot move a directory if the current directory is itself or its subdirectory.
7. Redirecting output to a file that exists under the directory that will be moved is not allowed.
8. If an error occurs, the remaining files or directories will still be moved.

Examples:

- \* To rename a file:  

```
fs0:\> mv IsaBus.efi Bus.efi
moving fs0:\IsaBus.efi -> \Bus.efi
- [ok]
```
- \* To move a directory to the current directory:  

```
fs0:\> mkdir test1\temp
fs0:\> mv test1\temp
moving fs0:\test1\temp -> \.\temp
- [ok]
```
- \* To rename a directory:  

```
fs0:\> mv efi efi1.1
moving fs0:\efi -> \efi1.1
- [ok]
```

```

* To move multiple directories at a time:
fs0:\> mv test1 test2 test
moving fs0:\test1 -> \test\test1
- [ok]
moving fs0:\test2 -> \test\test2
- [ok]

* Moving a read-only directory will result a failure:
fs0:\test> attrib +r temp1
DA R fs0:\test\temp1
fs0:\test> mv temp1 temp2
moving fs0:\test\temp1 -> \test\temp2
- error - Invalid Parameter

Shell>

```

### 2.2.23 reconnect

#### Summary

Reconnects drivers to the specific device.

#### EFI Versions

EFI 1.10 and above.

#### Usage

```

RECONNECT DeviceHandle [DriverHandle [ChildHandle]]
RECONNECT -r

DeviceHandle    - Device handle, always taken as hexadecimal
                  number
DriverHandle    - Driver handle, always taken as hexadecimal
                  number
ChildHandle     - Child handle of device, always taken as
                  hexadecimal number
-r             - Reconnect drivers to all devices

```

#### Description

This command reconnects drivers to the specific device. It will first disconnect the specified driver from the specified device and then connect the driver to the device recursively.

If the **-r** option is used, then all drivers will be reconnected to all devices. See the **connect** and **disconnect** commands for more details. The example below is the typical output from the help for this command.

## Examples

```
Shell> Help Reconnect
Reconnects one or more EFI drivers to a device.
```

```
RECONNECT DeviceHandle [DriverHandle [ChildHandle]]
RECONNECT -r
```

```
DeviceHandle - Device handle in hexadecimal format
DriverHandle - Driver handle in hexadecimal format
ChildHandle  - Child handle of device in hexadecimal format
-r           - Reconnect drivers to all devices
```

### Note:

1. This command disconnects the drivers from the controller similar to 'DISCONNECT'. The main difference is this command then immediately reconnects all drivers recursively.
2. If no 'DriverHandle' parameter is specified, all drivers on the specified device will be the default.
3. If no 'ChildHandle' parameter is specified, all child handles of the specified device will be the default.
4. If 'DriverHandle' parameter is provided, the specified driver will have highest priority on connecting the device(s).
5. If the '-r' option is specified, any drivers that are binding to any devices will be disconnected first and then connected recursively.
6. This command is a great way to test if drivers are compliant with the EFI 1.10 Driver Model.
7. This command does not support output redirection.

### Examples:

- \* To reconnect all drivers to all devices:  
Shell> reconnect -r
- \* To reconnect all drivers to device 0x28:  
fs0:\> reconnect 28
- \* To disconnect 0x17 from 0x28 then reconnect drivers with 0x17 as highest priority to device 0x28:  
fs0:\> reconnect 28 17
- \* To disconnect 0x17 from 0x28 destroying child 0x32 then reconnect drivers with 0x17 as highest priority to device 0x28  
fs0:\> reconnect 28 17 32

## 2.2.24 reset

### Summary

Resets the system.

### EFI Versions

EFI 1.02 and above.

## Usage

```
RESET [-w [string]]
RESET [-s [string]]

-w      - Performs a warm reset
-s      - Performs a shutdown
string  - String to be passed to reset service
```

## Description

This command resets the system. The default is to perform a cold reset unless the **-w** parameter is specified. If the reset **string** is specified, then it is passed into the **Reset()** function, so the system can know the reason for the system reset.

## Examples

```
Shell> help reset
Resets the system.

RESET [-w [string]]
RESET [-s [string]]

-w      - Performs a warm reset
-s      - Performs a shutdown
string  - String to be passed to reset service

Note:
1. Not all systems implement '-w' flag. This may mean different
   things depending on which BIOS EFI is implemented.
2. Reset will be guaranteed to reset the chipset as well as the
   Processor when cold reset is called.
3. This command does not support output redirection.

Shell>
```

## 2.2.25 rm

### Summary

Deletes one or more files or directories.

### EFI Versions

EFI 1.02 and above.

## Usage

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

-q          - Quiet mode; does not prompt user for a
              confirmation
file        - File name (wildcards are permitted)
directory  - Directory name (wildcards are permitted)
```

## Description

This command deletes one or more files or directories. If the target is a directory, it will delete the directory, including all its subdirectories. It is not allowed to redirect a file whose parent directory (or the file itself) is being deleted.

## Examples

```
Shell> help rm
Deletes one or more files or directories.
```

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

```
    -q          - Quite mode; does not prompt user for a
                  confirmation
    file         - File name (wildcards are permitted)
    directory    - Directory name (wildcards are permitted)
```

### Note:

1. Removing a read-only file/directory will result in a failure. Removing a directory containing read-only file(s) will result in a failure.
2. If an error occurs, RM will exit immediately and later files/directories will not be removed.
3. You cannot remove a directory when the current directory is itself or its subdirectory.
4. If file contains wildcards, it will not ask user for confirmation.
5. You cannot remove the root directory.
6. You cannot remove the current directory or its ancestor.
7. Redirecting output to a file that exists under the directory that will be removed is not allowed.

### Examples:

- \* To remove multiple directories at a time:

```
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)
```

- \* Error occurs and RM will exit:

```
fs0:\> rm test\temp1 temp2
rm/del: Cannot find 'fs0:\test\temp1' - Not Found
```

- \* To remove multiple directories with wildcards:

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

- \* Removing a directory that contains a read-only file will fail:

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

```
fs0:\> rm test\temp1
```

```

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

Shell>

```

## 2.2.26 set

### Summary

Used to maintain the environment variables that are available from the EFI environment.

### EFI Versions

EFI 1.02 and above.

### Usage

```

SET [-v] [sname [value]]
SET [-d <sname>]

-d      - Deletes the environment variable
-v      - Volatile variable
sname   - Environment variable name
value   - Environment variable value

```

### Description

This command is used to maintain the environment variables that are available from the EFI environment. This command can do the following:

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

The **set** command will set the environment variable that is specified by **sname** to **value**. This command can be used to create a new environment variable or to modify an existing environment variable. If the **set** command is used without any parameters, then all the environment variables are displayed. If the **set** command is used with the **-d** option, then the environment variable that is specified by **sname** will be deleted. The following examples show how this command can be used to create, modify, and delete the environment variable **DiagnosticPath**.



## Examples

```
Shell> help set
Displays, creates, changes, or deletes EFI environment variables.
```

```
SET [-v] [sname [value]]
SET [-d <sname>]
```

```
    -d      - Deletes the environment variable
    -v      - Volatile variable
    sname   - Environment variable name
    value   - Environment variable value
```

### Notes:

1. Size of NVRAM for set command will depend on the system implementation.
  2. May send NVRAM variables to /efi/boot/bootstr.nvr on the file system if no NVRAM is implemented in the core EFI routines.
  3. SET values are stored in EFI NVRAM and will be retained between boots unless the option -v is specified.
  4. A \* in front of sname means that this variable is a volatile variable.
- To make any changes to it, please use the command with the -v option.

### Examples:

```
* To add an environment variable:
Shell> set DiagnosticPath fs0:\efi\diag;fs1:\efi\diag

* To display environment variables:
Shell> set
    * path          : .
      diagnosticPath : fs0:\efi1.1\diag;fs1:\efi1.1\diag

* To delete an environment variable:
Shell> set -d diagnosticpath
Shell> set
    * path          : .

* To change an environment variable:
fs0:\> set src efi
fs0:\> set
    * path : .;fs0:\efi\tools;fs0:\efi\boot;fs0:\
      src  : efi
fs0:\> set src efi1.1
fs0:\> set
    * path : .;fs0:\efi\tools;fs0:\efi\boot;fs0:\
      src  : efi1.1

* To append an environment variable:
Shell> set
    * path          : .
Shell> set path %path%;fs0:\efi\tools;fs0:\efi\boot;fs0:\
Shell> set
    * path          : .;fs0:\efi\tools;fs0:\efi\boot;fs0:\

* To set a volatile variable that will disappear at the next
boot:
Shell> set -v EFI_SOURCE c:\project\EFI1.1
Shell> set
    * path          : .;fs0:\efi\tools;fs0:\efi\boot;fs0:\
    * EFI_SOURCE    : c:\project\EFI1.1
```

```
Shell>
```

## 2.2.27 time

## Summary

Displays or sets the current time for the system.

## EFI Versions

EFI 1.02 and above.

## Usage

**TIME** [hh:mm[:ss]]

```
hh      - Hour of time, Hour range: 0 - 23
mm      - Minute of time, Minute range: 0 - 59
ss      - Second of time, Second range: 0 - 59
```

## Description

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

- Except for numeric characters and the : character, all other characters in the argument are invalid. The Shell will report an error if the number is in the wrong hour/minute/second range.
- Spaces before or after the numeric character are not allowed. Spaces inserted into the number are not allowed either.
- Repeated zeros are allowed before the number. For example:

```
Shell > time 00000017:000004:0000
Shell > time
17:04:00 (GMT+08:00)
Shell >
```

- The seconds parameter is optional. If there is no seconds number, it will set to zero by default.

```
Shell > time 17:23
Shell > time
17:23:00 (GMT+08:00)
Shell > time 17:23:
Shell > time
17:23:00 (GMT+08:00)
```

## Examples

```
Shell> help time
Displays the current time or sets the time of the system.
```

```
TIME [hh:mm[:ss]]
```

```
hh      - Hour of time, hour range: 0 - 23
mm      - Minute of time, minute range: 0 - 59
ss      - Second of time, second range: 0 - 59
```

Note:

1. Hour and minute are required to set the time.
2. If second is not specified, 0 will be used as default.

Examples:

```
* To display current time:
fs0:\> time
16:51:03 (GMT+08:00)

* To set the system time:
fs0:\> time 9:51:30
fs0:\> time
09:51:31 (GMT+08:00)
```

```
Shell>
```

## 2.2.28 touch

### Summary

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

### EFI Versions

EFI 1.02 and above.

### Usage

```
TOUCH [-r] file [file ...]
```

```
-r      - Recursive to subdirectories
file    - The name or pattern of the file or directory. There
           can be multiple files on the command line.
```

### Description

This command updates the time and date on the file that is specified by the **file** parameter to the current time and date.

## Examples

```
Shell> help touch
Updates time and date with current time.

TOUCH [-r] file [file ...]

    -r    - Recursive to subdirectories
    file  - The name or pattern of the file or directory. There
            can be multiple files on the command line.

Notes:
1. If multiple files are specified on the command line, it will
   continue processing. It will touch the files one by one and
   errors will be ignored.
2. Touch cannot change the time and date of read-only files and
   and directories.

Examples:
* To touch a file (the time and date of the file will be
  changed after
  TOUCH):
  fs0:\> ls for.nsh
  Directory of: fs0:\
    06/18/01  09:32p                153  for.nsh
                1 File(s)          153 bytes
                0 Dir(s)

  fs0:\> touch for.nsh
  touch: fs0:\for.nsh [ok]

  fs0:\> ls for.nsh
  Directory of: fs0:\
    06/19/01  09:54a                153  for.nsh
                1 File(s)          153 bytes
                0 Dir(s)

* To touch a directory recursively:
  fs0:\> touch -r efil.1
  touch: fs0:\efil.1 [ok]
  touch: fs0:\efil.1\boot [ok]
  touch: fs0:\efil.1\boot\nshell.efi [ok]
```

### 2.2.29 type

#### Summary

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

#### EFI Versions

EFI 1.02 and above.

## Usage

```
TYPE [-a|-u] file [file...]  
  
-a      - Displays the file as ASCII characters  
-u      - Displays the file as Unicode characters  
file    - Name of file to display
```

## Description

This command sends the contents of a file to the standard output device. If no options are used, then the file type is automatically detected and sent to the standard output device. If the **-a** option is specified, the file is sent to the standard output device as a stream of ASCII characters. If the **-u** option is specified, the file is sent to the standard output device as a stream of Unicode characters. The example below shows how to send the ASCII file **README.TXT** on the floppy drive to the standard output device and the Unicode file **READMEU.TXT** on the floppy drive to the standard output device.

Type redirection to the same file is not supported currently. Wildcards are permitted in the file name. If the **type** command is used in conjunction with an ASCII file with the **-u** flag, this flag will be ignored and the contents of the file will be sent to the standard output device as ASCII characters. If the **type** command is used in conjunction with a Unicode file with the **-a** flag, the file will be sent as ASCII characters but with an additional space between each character.

## Examples

Displays file contents on the standard output device.

TYPE [-a|-u] file [file...]

```
-a      - Display file in ASCII format
-u      - Display file in Unicode format
file    - Name of file to display
```

Examples:

```
* To display a file in Unicode format:
fs0:\> type -u pause.nsh
File: fs0:\pause.nsh, Size 204
#
# Example script for 'pause' command
#
echo pause.nsh begin..
date
time
pause
echo pause.nsh done.

* To display a file in ASCII format:
fs0:\> type -a pause.nsh
File: fs0:\pause.nsh, Size 204
#
# Example script for 'pause' command
#
echo pause.nsh begin..
date
time
pause
echo pause.nsh done.

* To display multiple files:
fs0:\> type test.*
File: fs0:\test.txt, Size 23
      How to Install?
File: fs0:\test.nsh, Size 48
time
stall 3000000
time
```

### 2.2.30 unload

#### Summary

Unloads a driver image that was already loaded.

#### EFI Versions

EFI 1.10 and above.

## Usage

```
UNLOAD [-n] [-v] Handle

-n          - Without prompt
-v          - Dump Verbose information
Handle      - Handle of driver to unload, always taken as
              hexadecimal number
```

## Description

This command unloads a driver image that was already loaded.

## Examples

```
Shell> help unload
Unloads a driver image.
```

```
UNLOAD [-n] [-v] Handle

-n          - Without prompt
-v          - Dump Verbose information
Handle      - Handle of driver to unload, always taken as
              hexadecimal number
```

### Note:

1. Option '-n' can be used to skip all the prompts during the unloading so that it can be used in a batch file.
2. If option '-v' is specified, the verbose information on the image will be displayed before the image is unloaded.
3. Only those images that can support the unloading operation can be unloaded successfully.
4. The '[load](#)' command does the opposite opposite of the 'UNLOAD' command.

### Examples:

- \* To find the handle index driver image to unload:

```
Shell> dh -b
Handle dump
  1: Image(DXE Core)
  2: FwVol FwFileSys FwVolBlk DevPath(MemMap(11:1760000-
    189FFC8))
  ...
 27: Image(Reset)
 28: Image(WinNtBlockIo) DriverBinding
 29: Image(Timer)
  ...
```

- \* To unload the driver image of 'Reset':

```
Shell> unload 27
 27: Image(Reset)
Unload driver image (y/n)? n
Exit status code: Aborted
```

## 2.2.31 ver

## Summary

Displays the version information for this EFI firmware.

## EFI Versions

EFI 1.02 and above.



## Usage

```
VER [-s]
```

```
-s      - Displays only the EFI Shell version
```

## Description

This command displays the version information for this EFI Firmware or the version information for the EFI Shell itself. The information is retrieved through the EFI System Table or the Shell image.

## Examples

```

Shell> help ver
Displays the version information for this EFI Firmware.

VER [-s]

    -s          - Displays only the EFI Shell version

Examples:
* To display the version information of a platform:
fs0:\> ver
EFI Specification Revision : 1.10
EFI Vendor                 : INTEL
EFI Revision                : 14.5

* To display the version information of another platform:
Note that the version information may vary from different
platforms due to some platform-specific information.

fs0:\> ver
EFI Specification Revision : 1.02
EFI Vendor                 : INTEL
EFI Revision                : 12.38

SAL Specification Revision      3. 0
  SAL_A Revision               = 1. 1
  SAL_B Revision               = 1. 1

PAL_A Revision                 66.23
PAL_B Revision                 66.23

Other modules mentioned in FIT (Firmware Interface Table)
FIT_Entry Type    0, Revision    2.60
FIT_Entry Type   15, Revision    66.23
FIT_Entry Type   16, Revision    0.90
FIT_Entry Type   32, Revision    0.30
FIT_Entry Type   30, Revision    1. 0
FIT_Entry Type   17, Revision    0.90
FIT_Entry Type   18, Revision    6. 0
FIT_Entry Type   20, Revision    0.80

SalProc Entry 000000003FE3F720 and GP 000000003FF22480
PalProc Entry 000000003FF48010 IO Port Base 00000FFFC000000
Cache Enabled

* To display the version information of the EFI Shell:
Shell> ver -s
EFI Shell Revision 1.0
EFI Shell Signature: D2C18636-40E5-4EB5-A31B-36695FD42C87

Shell>

```

### 2.2.32 vol

#### Summary

Displays the volume information for the file system that is specified by **fs**.

#### EFI Versions

EFI 1.02 and above.

## Usage

```
VOL [fs] [-n <Volume Label>]
VOL [fs] [-d]

fs           - The name of the file system
Volume Label - New volume label
-d          - Empty volume label
```

## Description

This command displays the volume information for the file system that is specified by **fs**. If **fs** is not specified, the current file system will be taken as the correct **fs**. If **Volume Label** is specified, then the volume label for **fs** will be set to **Volume Label**. The maximum length for **Volume Label** is 11 characters.

## Examples

```
Shell> help vol
Displays volume information for the file system specified by fs.
```

```
VOL [fs] [-n <Volume Label>]
VOL [fs] [-d]

fs           - The name of the file system
Volume Label - New volume label
-d          - Empty volume label
```

### Notes:

1. The following characters cannot be used in Volume Label:  
% ^ \* + = [ ] | : ; " < > ? / ..
2. No space is allowed in Volume Label.
3. If **fs** is not specified, the current file system will be taken as **fs**.

### Examples:

- \* To display the volume of the current fs:  
fs0:\> vol  
Volume has no label (rw)  
1,457,664 bytes total disk space  
1,149,440 bytes available on disk  
512 bytes in each allocation unit
- \* To change the label of fs0:  
shell> vol fs0 -n help\_test  
Volume HELP\_TEST (rw)  
1,457,664 bytes total disk space  
1,149,440 bytes available on disk  
512 bytes in each allocation unit
- \* To get rid of the label of fs0:  
fs0:\> vol fs0 -d  
Volume has no label (rw)  
1,457,664 bytes total disk space  
220,160 bytes available on disk  
512 bytes in each allocation unit

```
Shell>
```

## 2.3 Additional Commands from Full Build Shell

### 2.3.1 **comp**

#### Summary

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

#### EFI Versions

EFI 1.10 and above.

#### Usage

```
COMP [-b] file1 file2
```

```
-b      - Display one screen at a time
file1   - First file name (directory name or wildcards not
          permitted)
file2   - Second file name (directory name or wildcards not
          permitted)
```

#### Description

This command compares the contents of two files in binary mode. It displays up to 10 differences between the two files. For each difference, up to 32 bytes from the location where the difference starts is dumped.

## Examples

```
Shell> help comp
Compares the contents of two files on a byte for byte basis.
```

```
COMP [-b] file1 file2
```

```
    -b          - Display one screen at a time
file1          - First file name  (directory name or wildcards not
                  permitted)
file2          - Second file name (directory name or wildcards not
                  permitted)
```

### Note:

1. COMP will compare files in binary mode.
2. COMP will exit immediately if the lengths of the compared files are different.
3. COMP will exit if 10 differences encountered.

### Examples:

- \* To compare two files with different lengths:

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

- \* To compare two files with the same contents:

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

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

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

## 2.3.2 dblk

### Summary

Displays the contents of one or more blocks from a block device.

### EFI Versions

EFI 1.10 and above.

## Usage

```
DBLK device [Lba] [blocks] [-b]

    -b          - Display one screen at a time
    device      - Block device name
    Lba         - Index of the first block to be displayed in hex format
    blocks      - Number of blocks to be displayed in hex format
```

## Description

This command displays the contents of one or more blocks from a block device. **Lba** and **blocks** should be typed in hex value. If **Lba** is not specified, block #0 is assumed. If **blocks** is not specified, then only 1 block will be displayed. The maximum number of blocks that can be displayed at one time is 0x10.

If an MBR is found on the block, the partition information will be printed after all the block contents have been displayed.

If the block is a FAT partition, some FAT parameters will be displayed (label, systemid, oemid, sectorsize, clustersize, media etc) after all the blocks have been displayed.

## Examples

Shell> help dblk

Displays the contents of one or more blocks from a block device.

DBLK device [Lba] [blocks] [-b]

-b            - Display one screen at a time  
device       - Block device name  
Lba          - Index of the first block to be displayed in hex format  
blocks       - Number of blocks to be displayed in hex format

Note:

1. The 'Lba' and 'blocks' parameters are in hexadecimal format.
2. The 'Lba' parameter defaults to 0 if not specified.
3. The 'blocks' parameter defaults to 1 if not specified.
4. If 'blocks' is larger than 0x10, only the first 0x10 blocks will be displayed.
5. Lba + blocks should not be larger than the last block of the device.
6. If a FAT file system is detected, some FAT parameters will also be displayed (label, systemid, oemid, sectorsize, clustersize, media etc) after all the blocks have been displayed.
7. If a MBR is detected on a FAT file system, the partition information will be displayed after all the block contents have been displayed.

Examples:

- \* To display one block of blk0, beginning from block 0:

Shell>dblk blk0

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

Shell>dblk fs0 2

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

Shell>dblk fs0 12 5

- \* To display 0x10 blocks of fs0, beginning from block 0x12:

Shell>dblk fs0 12 10

- \* The attempt to display more than 0x10 blocks will display only 0x10 blocks:

Shell>dblk fs0 12 20

- \* To display one block of blk2, beginning from the first block (blk0):

fs1:\tmps1> dblk blk2 0 1

```
LBA 0000000000000000 Size 00000200 bytes BlkIo 3F0CEE78
00000000: EB 3C 90 4D 53 44 4F 53-35 2E 30 00 02 04 08 00 *.<.MSDOS5.0.....*
00000010: 02 00 02 00 00 F8 CC 00-3F 00 FF 00 3F 00 00 00 *.....?....?...*
00000020: 8E 2F 03 00 80 01 29 2C-09 1B D0 4E 4F 20 4E 41 */.....),...NO NA*
00000030: 4D 45 20 20 20 20 46 41-54 31 36 20 20 20 33 C9 *ME      FAT16   3.*
00000040: 8E D1 BC F0 7B 8E D9 B8-00 20 8E C0 FC BD 00 7C *.....*
00000050: 38 4E 24 7D 24 8B C1 99-E8 3C 01 72 1C 83 EB 3A *8N$.$.<..r...:*
00000060: 66 A1 1C 7C 26 66 3B 07-26 8A 57 FC 75 06 80 CA *f...&f;.&.W.u...*
00000070: 02 88 56 02 80 C3 10 73-EB 33 C9 8A 46 10 98 F7 *..V....s.3..F...*
00000080: 66 16 03 46 1C 13 56 1E-03 46 0E 13 D1 8B 76 11 *f..F..V..F...v.*
00000090: 60 89 46 FC 89 56 FE B8-20 00 F7 E6 8B 5E 0B 03 *^..F..V.. ..^...*
000000A0: C3 48 F7 F3 01 46 FC 11-4E FE 61 BF 00 00 E8 E6 *.H...F..N.a.....*
000000B0: 00 72 39 26 38 2D 74 17-60 B1 0B BE A1 7D F3 A6 *.r9&8-t..^.....*
000000C0: 61 74 32 4E 74 09 83 C7-20 3B FB 72 E6 EB DC A0 *at2Nt... ;.r....*
000000D0: FB 7D B4 7D 8B F0 AC 98-40 74 0C 48 74 13 B4 0E *.....@t.Ht....*
000000E0: BB 07 00 CD 10 EB EF A0-FD 7D EB E6 A0 FC 7D EB *.....*
000000F0: E1 CD 16 CD 19 26 8B 55-1A 52 B0 01 BB 00 00 E8 *.....&.U.R.....*
00000100: 3B 00 72 E8 5B 8A 56 24-BE 0B 7C 8B FC C7 46 F0 *;.r.[.V$......F.*
00000110: 3D 7D C7 46 F4 29 7D 8C-D9 89 4E F2 89 4E F6 C6 *=..F.)....N..N...*
00000120: 06 96 7D CB EA 03 00 00-20 0F B6 C8 66 8B 46 F8 *.....f.F.*
00000130: 66 03 46 1C 66 8B D0 66-C1 EA 10 EB 5E 0F B6 C8 *f.F.f..f....^....*
```

```

00000140: 4A 4A 8A 46 0D 32 E4 F7-E2 03 46 FC 13 56 FE EB *JJ.F.2....F..V..*
00000150: 4A 52 50 06 53 6A 01 6A-10 91 8B 46 18 96 92 33 *JRP.Sj.j...F...3*
00000160: D2 F7 F6 91 F7 F6 42 87-CA F7 76 1A 8A F2 8A E8 *.....B...v.....*
00000170: C0 CC 02 0A CC B8 01 02-80 7E 02 0E 75 04 B4 42 *.....u..B*
00000180: 8B F4 8A 56 24 CD 13 61-61 72 0B 40 75 01 42 03 *...V$.aar.@u.B.*
00000190: 5E 0B 49 75 06 F8 C3 41-BB 00 00 60 66 6A 00 EB *^.Iu...A...`fj...*
000001A0: B0 4E 54 4C 44 52 20 20-20 20 20 20 0D 0A 52 65 *.NTLDR ..Re*
000001B0: 6D 6F 76 65 20 64 69 73-6B 73 20 6F 72 20 6F 74 *move disks or ot*
000001C0: 68 65 72 20 6D 65 64 69-61 2E FF 0D 0A 44 69 73 *her media....Dis*
000001D0: 6B 20 65 72 72 6F 72 FF-0D 0A 50 72 65 73 73 20 *k error...Press *
000001E0: 61 6E 79 20 6B 65 79 20-74 6F 20 72 65 73 74 61 *any key to resta*
000001F0: 72 74 0D 0A 00 00 00 00-00 00 00 AC CB D8 55 AA *rt.....U.*

Fat 16 BPB  FatLabel: 'NO NAME ' SystemId: 'FAT16 ' OemId: 'MSDOS5.0'
SectorSize 200 SectorsPerCluster 4 ReservedSectors 8 # Fats 2
Root Entries 200 Media F8 Sectors 32F8E SectorsPerFat CC
SectorsPerTrack 3F Heads 255

```

### 2.3.3 devices

#### Summary

Displays the list of devices managed by EFI drivers.

#### EFI Versions

EFI 1.10 and above.

#### Usage

```

DEVICES [-b] [-l XXX]

    -b                - Display one screen at a time
    -l XXX            - Display devices using the specified ISO 639-2 language

```

#### Description

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

**Table 6 Display Format of `devices`**

Columns	Description
CTRL	The handle number of the EFI device
TYPE	The device type: <ul style="list-style-type: none"> <li>• [R] – Root Controller</li> <li>• [B] – Bus Controller</li> <li>• [D] – Device Controller</li> </ul>
CFG	A managing driver supports the Driver Configuration Protocol
DIAG	A managing driver supports the Driver Diagnostics Protocol
#P	The number of parent controllers for this device
#D	The number of child controllers produced by this device
Device Name	The name of the device from the Component Name Protocol



## Examples

```
Shell> help devices
Displays the list of devices managed by EFI drivers.

DEVICES [-b] [-l XXX]

    -b          - Display one screen at a time
    -l XXX      - Display devices using the specified ISO 639-2
                  language

Display Format:
CTRL          - The handle number of the EFI device
TYPE          - The device type:
                  [R] - Root Controller
                  [B] - Bus Controller
                  [D] - Device Controller
CFG           - A managing driver supports the Driver Configuration
                  Protocol
DIAG          - A managing driver supports the Driver Diagnostics
                  Protocol
#P            - The number of parent controllers for this device
#D            - The number of drivers managing the device
#C            - The number of child controllers produced by this
                  device
DEVICE NAME   - The name of the device from the Component Name
                  Protocol

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

### 2.3.4 devtree

## Summary

This command displays the tree of devices compliant with the EFI Driver Model.

## EFI Versions

EFI 1.10 and above.

## Usage

```
DEVTREE [-b] [-d] [-l XXX] [DeviceHandle]

    -b          - Display one screen at a time
    -d          - Display device tree using device paths
    -l          - Display device tree using the specified ISO 639-2
                  language
    DeviceHandle - Display device tree below a certain handle
```

## Description

This command prints a tree of devices that are being managed by drivers that follow the EFI Driver Model. By default, the devices are printed in device names that are retrieved from the Component Name Protocol. If the option **-d** is specified, the device paths will be printed instead.

## Examples

```
Shell> help devtree
Displays the tree of devices compliant with the EFI Driver Model.

DEVTREE [-b] [-d] [-l XXX] [DeviceHandle]

    -b          - Display one screen at a time
    -d          - Display device tree using device paths
    -l          - Display device tree using the specified ISO 639-2
                  language
    DeviceHandle - Display device tree below a certain handle

Examples:
* To display the tree of all devices compliant with the EFI Driver
  Model:
  Shell> devtree

* To display the tree of all devices below device 28 compliant with
  the EFI Driver Model:
  Shell> devtree 28

* To display the tree of all devices compliant with the EFI Driver
  Model one screen at a time:
  Shell> devtree -b
```

### 2.3.5 dmef

## Summary

Displays the contents of system or device memory.

## EFI Versions

EFI 1.10 and above.

## Usage

```
DMEM [-b] [Address] [Size] [-MMIO]
```

```
-b          - Display one screen at a time
address     - Starting address in hexadecimal format
size        - Number of bytes to display in hexadecimal format
-MMIO       - Forces address cycles to the PCI bus
```

## Description

This command displays the contents of system memory or device memory. **Address** and **Size** should be typed in hex value. If Address is not specified, then the contents of the EFI System Table are displayed. Otherwise, memory starting at Address is displayed. **Size** specifies the number of bytes to display. If **Size** is not specified, then it defaults to 512 bytes. If **MMIO** is not specified, then main system memory is displayed. Otherwise, device memory is displayed through the use of the [EFI PCI ROOT BRIDGE IO PROTOCOL](#).

## Examples

```
Shell> dmem
Displays the contents of system or device memory.
```

```
DMEM [-b] [Address] [Size] [-MMIO]
```

```
-b          - Display one screen at a time
address     - Starting address in hexadecimal format
size        - Number of bytes to display in hexadecimal format
-MMIO       - Forces address cycles to the PCI bus
```

### Note:

1. All units are in hexadecimal format.
2. Address must be aligned on an even processor address boundary.
3. If the 'address' parameter is not specified, DMEM will display the all system table pointer entries by default.

### Examples:

```
* To display the EFI system table pointer entries:
fs0:\> dmem
```

```
Memory Address 000000003FF7D808 200 Bytes
3FF7D808: 49 42 49 20 53 59 53 54-02 00 01 00 78 00 00 00 *IBI SYST....x...*
3FF7D818: 5C 3E 6A FE 00 00 00 00-88 2E 1B 3F 00 00 00 00 *\>j.....?....*
3FF7D828: 26 00 0C 00 00 00 00 00-88 D3 1A 3F 00 00 00 00 *&.....?....*
3FF7D838: A8 CE 1A 3F 00 00 00 00-88 F2 1A 3F 00 00 00 00 *...?.....?....*
3FF7D848: 28 EE 1A 3F 00 00 00 00-08 DD 1A 3F 00 00 00 00 *(..?.....?....*
3FF7D858: A8 EB 1A 3F 00 00 00 00-18 C3 3F 3F 00 00 00 00 *...?.....*
3FF7D868: 00 4B 3F 3F 00 00 00 00-06 00 00 00 00 00 00 *..K.....*
3FF7D878: 08 DA F7 3F 00 00 00 00-70 74 61 6C 88 00 00 00 *...?....ptal....*
3FF7D888: 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 *.....*
3FF7D898: 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 *.....*
3FF7D8A8: 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 *.....*
3FF7D8B8: 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 *.....*
3FF7D8C8: 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 *.....*
3FF7D8D8: 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 *.....*
3FF7D8E8: 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 *.....*
3FF7D8F8: 00 00 00 00 00 00 00 00-70 68 06 30 88 00 00 00 *.....ph.0....*
3FF7D908: 65 76 6E 74 00 00 00 00-02 02 00 60 00 00 00 00 *evnt.....~....*
3FF7D918: 18 6F 1A 3F 00 00 00 00-10 E0 3F 3F 00 00 00 00 *.o.?.....*
3FF7D928: 10 00 00 00 00 00 00 00-40 C0 12 3F 00 00 00 00 *.....@..?....*
3FF7D938: 10 80 13 3F 00 00 00 00-00 00 00 00 00 00 00 *...?.....*
3FF7D948: 00 00 00 00 00 00 00 00-40 7D 3F 3F 00 00 00 00 *.....@.....*
3FF7D958: 50 6F 1A 3F 00 00 00 00-00 00 00 00 00 00 00 *Po.?.....*
3FF7D968: 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 *.....*
3FF7D978: 00 00 00 00 00 00 00 00-70 74 61 6C 88 00 00 00 *.....ptal....*
3FF7D988: 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 *.....*
3FF7D998: 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 *.....*
3FF7D9A8: 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 *.....*
3FF7D9B8: 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 *.....*
3FF7D9C8: 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 *.....*
3FF7D9D8: 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 *.....*
3FF7D9E8: 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 *.....*
3FF7D9F8: 00 00 00 00 00 00 00 00-70 68 06 30 A0 00 00 00 *.....ph.0....*
```

```
Valid EFI Header at Address 000000003FF7D808
```

```
-----
System: Table Structure size 00000078 revision 00010002
ConIn (3F1AD388) ConOut (3F1AF288) StdErr (3F1ADD08)
Runtime Services 000000003F3FC318
Boot Services 000000003F3F4B00
SAL System Table 000000003FF22760
ACPI Table 000000003FFD9FC0
ACPI 2.0 Table 00000000000E2000
MPS Table 000000003FFD0000
SMBIOS Table 00000000000F0020
```

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

```
Shell> dmem 1af3088 16
Memory Address 000000001AF3088 16 Bytes
01AF3088: 49 42 49 20 53 59 53 54-00 00 02 00 18 00 00 00 *IBI SYST.....*
```

```

01AF3098: FF 9E D7 9B 00 00
*.....*
* To display memory mapped IO contents from 1af3088 with size of 16
bytes:
Shell> dmem 1af3088 16 -MMIO

```

## 2.3.6 dmpstore

### Summary

Manages all EFI NVRAM variables.

### EFI Versions

EFI 1.10 and above.

### Usage

```

DMPSTORE [-b] [-d] [Variable]
DMPSTORE [Variable] -s file
DMPSTORE [Variable] -l file

    -b          - Display one screen at a time
Variable       - Display the specified variable name
    -d          - Delete variables
    -s          - Save variables to file
    -l          - Load and set variables from file

```

### Description

This command is used to manage the EFI NVRAM variables. All EFI NVRAM variables are displayed when **Variable** is not specified. The variable value is printed as HEX dump.

Option **-d** is used to delete variables. Option **-s** and **-l** are used to save and load variables to and from file. The variable name can be specified when using these variables so that the operation only takes effect on that variable.

## Examples

```
Shell> help dmpstore
Displays all EFI NVRAM variables.
```

```
DMPSTORE [-b] [-d] [Variable]
DMPSTORE [Variable] -s file
DMPSTORE [Variable] -l file
```

```
-b      - Display one screen at a time
Variable - Display the specified variable name
-d      - Delete variables
-s      - Save variables to file
-l      - Load and set variables from file
```

### Note:

1. If the 'variable' parameter is not specified, all variables will be displayed.
2. The variable name is not case sensitive.

### Examples:

\* To display all EFI NVRAM variables:

```
Shell> dmpstore
```

```
Dump NVRAM
```

```
Variable RT+BS 'Efi:BootCurrent' DataSize = 2
00000000: FF FF
Variable NV+RT+BS 'Efi:LangCodes' DataSize = 2A
00000000: 65 6E 67 65 6E 6D 61 6E-67 63 68 69 7A 68 6F 64 *engenmangchizhod*
00000010: 65 75 67 65 6D 67 65 72-67 6D 68 67 6F 68 66 72 *eugemgergmhghohfr*
00000020: 61 66 72 65 66 72 6D 66-72 6F
*afrefrmfro*
Variable NV+RT+BS 'Efi:Lang' DataSize = 3
00000000: 65 6E 67
*eng*
...
Variable NV+BS 'ShellAlias:copy' DataSize = 6
00000000: 63 00 70 00 00 00
*c.p...*
Variable NV+BS 'SEnv:path' DataSize = 4
00000000: 2E 00 00 00
*....*
```

\* To display the Boot0000 EFI NVRAM variable:

```
Shell> dmpstore Boot0000
```

```
Dump Variable Boot0000
```

```
Variable NV+RT+BS 'Efi:Boot0000' DataSize = 48
00000000: 01 00 00 00 30 00 42 00-6F 00 6F 00 74 00 30 00 *....0.B.o.o.t.0.*
00000010: 30 00 30 00 30 00 00 00-01 04 14 00 B1 18 C5 58 *0.0.0.....X*
00000020: F3 76 D4 11 BC EA 00 80-C7 3C 88 81 01 04 18 00 *.v.....<.....*
00000030: 2F A9 95 0C 06 A0 D4 11-BC FA 00 80 C7 3C 88 81 */.....<...*
00000040: 00 00 00 00 7F FF 04 00-
```

## 2.3.7 edit

### Summary

Full screen editor for ASCII or UNICODE files.

### EFI Versions

EFI 1.10 and above.

## Usage

```
EDIT [file]

file      - Name of file to be edited
```

## Description

This command allows a file to be edited using a full screen editor. The editor supports both Unicode and ASCII file types. The following example shows typical output for help on this command.

## Examples

```
Shell> help edit
Full screen editor for ASCII or UNICODE files.

EDIT [file]

file      - Name of file to be edited

Note:
  1. If the file is not specified, NewFile.txt is edited.
  2. The size of file shall not be larger than 16 Mbytes.

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

## 2.3.8 eficompress

### Summary

Compress a file using EFI Compression Algorithm.

### EFI Versions

EFI 1.10 and above.

### Usage

```
EFICOMPRESS infile outfile

infile    - Filename for uncompressed input file
outfile   - Filename for compressed output file
```

### Description

This command is used to compress a file using EFI Compression Algorithm and write the compressed form out to a new file.

**Note:** *The Shell binary before (and including) Edk-Dev-Snapshot-20061228 uses Tiano Compression Algorithm which is not compatible with EFI Compression Algorithm. Files compressed with the EFI Compression Algorithm can't be decompressed with those previous versions of Shell binaries.*

## Examples

```
Shell> help eficompress
Compress a file.
```

```
EFICOMPRESS infile outfile
```

```
infile    - Filename for uncompressed input file
outfile    - Filename for compressed output file
```

Examples:

```
* To compress a file named 'uncompressed' to file 'compressed':
fs0:\> eficompress uncompressed compressed
```

### 2.3.9 efidecompress

#### Summary

Decompress a file using EFI Decompression Algorithm.

#### EFI Versions

EFI 1.10 and above.

#### Usage

```
EFIDECOMPRESS infile outfile
```

```
infile    - Filename of compressed input file
outfile    - Filename of decompressed output file
```

#### Description

This command is used to decompress a file using EFI Decompression Algorithm and write the decompressed form out to a new file.

**Note:** *The Shell binary before (and including) Edk-Dev-Snapshot-20061228 uses Tiano Compression Algorithm which is not compatible with EFI Compression Algorithm. The files that are compressed with previous version of Shell binaries have to be decompressed with those previous Shell binaries.*



## Examples

```
Shell> help efidecompress
Decompress a file.

EFIDECOMPRESS infile outfile

    infile    - Filename of compressed input file
    outfile   - Filename of decompressed output file

Examples:
* To decompress a file named 'compressed' to file 'uncompressed':
fs0:\> efidecompress compressed uncompressed
```

## 2.3.10 err

### Summary

Displays or changes the error level in the system.

### EFI Versions

EFI 1.10 and above.

### Usage

```
ERR [ErrorLevel]
ERR -dump [Handle]
ERR Handle ErrorLevel

    ErrorLevel    - New error level bit mask
    -dump         - Display debug mask of all handles or specified
                   handle
    Handle        - Device handle
```

### Description

This command sets the current debug error level in the system by setting the variable "EFIDebug" to the value of *ErrorLevel*. Error Level is a bit mask for different types of error messages.

This command can also be used to dump or change the error level of a specific device handle.

## Examples

```
Shell> help err
Displays or changes the error level in the system.
```

```
ERR [ErrorLevel]
ERR -dump [Handle]
ERR Handle ErrorLevel
```

```
ErrorLevel    - New error level bit mask
-dump         - Display debug mask of all handles or specified
               handle
Handle        - Device handle
```

### Note:

1. Saving to NVRAM will cause the error level to be saved and used on all future reboots. The EFI Core will use the new error level as system reboots. All core EFI routines will then output using the new error level.
2. Error console must be set to a device path (i.e. com port or console).
3. Not all EFI implementations include an error console or support debug output. Consult the BIOS release notes for this support.
4. In debug version, the message whose error level is higher than the specified level will be displayed.
5. To add your own errors/error level see debug macro in sample implementation source in efidebug.h.

### Examples:

- \* To display the current error message output level:

```
Shell> err
EFI ERROR 80000000
00000001  EFI_D_INIT
00000002  EFI_D_WARN
00000004  EFI_D_LOAD
00000008  EFI_D_FS
00000010  EFI_D_POOL
00000020  EFI_D_PAGE
00000040  EFI_D_INFO
00000100  EFI_D_VARIABLE
00000400  EFI_D_BM
00001000  EFI_D_BLKIO
00004000  EFI_D_NET
00010000  EFI_D_UNDI
00020000  EFI_D_LOADFILE
00080000  EFI_D_EVENT
80000000  EFI_D_ERROR
```

- \* To change the error message output level:

```
Shell> err 80000107
Make this change and save to NVRAM? [Y/N]y

Shell> _
```

- \* To dump the debug mask of all handles which support debug mask protocol:

```
Shell> err -dump
Handle      Mask
=====
1: 80000004
9: 80000004
B: 80000004
D: 80000004
F: 80000004
11: 80000004
13: 80000004
45: 80000004
```

```

62: 80000004
63: 80000004
64: 80000004

* To display the debug mask of handle 0x13:
Shell> err -dump 13
  Handle      Mask
  =====
    13: 80000004

* To change the debug mask of handle 0x13:
Shell> err 13 80000007

Shell>

```

### 2.3.11 guid

#### Summary

Displays all registered EFI GUIDs.

#### EFI Versions

EFI 1.10 and above.

#### Usage

```

GUID [-b]

  -b          - Display one screen at a time

```

#### Description

This command displays a list of all the GUIDs that have been registered with the EFI environment. The following example shows the output from this command.

## Examples

```
Shell> help guid
Displays all registered EFI GUIDs.

GUID [-b]

        -b          - Display one screen at a time

Note:
  1. Only displays the GUIDs included in the core EFI build at the
     time the core was built. Additional GUIDs may have been added by
     the BIOS integrator. Any GUIDs that are not in the original core
     build or were added by a new protocol loaded by the user will
     show up as an Unknown Device.
  2. The GUID with a '*' at end means that there will probably be
     dump information or token available for the protocol to this
     GUID. The 'dh' command can be used to dump this information.

Examples:
fs0:\> guid -b
DevIo          : AF6AC311-84C3-11D2-8E3C-00A0C969723B
diskio         : CE345171-BA0B-11D2-8E4F-00A0C969723B
blkio          : 964E5B21-6459-11D2-8E39-00A0C969723B *
txtin          : 387477C1-69C7-11D2-8E39-00A0C969723B
txtout         : 387477C2-69C7-11D2-8E39-00A0C969723B *
fs             : 964E5B22-6459-11D2-8E39-00A0C969723B
load           : 56EC3091-954C-11D2-8E3F-00A0C969723B
image          : 5B1B31A1-9562-11D2-8E3F-00A0C969723B *
...
```

## 2.3.12 hexedit

### Summary

Full screen hex editor for files, block devices, or memory.

### EFI Versions

EFI 1.10 and above.

### Usage

```
HEXEDIT [[-f] FileName | [-d DiskName Offset Size] | [-m Address Size]]

-f          - Name of file to edit
-d          - Disk block to edit:
               DiskName - Name of disk to edit (for example fs0)
               Offset   - Starting block number (beginning from 0)
               Size      - Number of blocks to be edited
-m          - Memory region to edit:
               Address   - Starting 32-bit memory address (beginning
                           from 0)
               Size      - Size of memory region to be edited in bytes
```

### Description

This command allows a file, block device, or memory region to be edited. The region being edited is displayed as hexadecimal bytes, and the contents can be modified and saved. The following example shows typical output for help on this command.

## Examples

```
Shell> help hexedit
Full screen hex editor for files, block devices, or memory.

HEXEDIT [[-f] FileName | [-d DiskName Offset Size] | [-m Address Size]]

    -f          - Name of file to edit
    -d          - Disk block to edit:
                   DiskName - Name of disk to edit (for example fs0)
                   Offset   - Starting block number (beginning from 0)
                   Size     - Number of blocks to be edited
    -m          - Memory region to edit:
                   Address   - Starting 32-bit memory address (beginning
                               from 0)
                   Size     - Size of memory region to be edited in bytes

Examples:
* To edit a file in hex mode:
  fs0:\> hexedit test.bin

* To edit block device fs0 starting at block 0 with size of 2 blocks:
  fs0:\> hexedit -d fs0 0 2

* To edit memory region starting at address 0x00000000 with size of 2
  bytes:
  fs0:\> hexedit -m 0 2
```

### 2.3.13 ipconfig

#### Summary

Displays or modifies the current IP configuration.

#### EFI Versions

EFI 1.10 and above.

#### Usage

```
IPCONFIG [-r] | [-b] [-c Instance] [IpAddress [-m NetMask]]

    -r          - Restart the PXE base code and DHCP settings
    -b          - Display one screen at a time
    Instance    - Zero-based Simple Network Protocol instance
    IpAddress    - IP address in a.b.c.d format
    NetMask     - Network mask in 255.255.255.0 format
```

#### Description

This command displays or modifies the IP configuration of `EFI_PXE_BASE_CODE_PROTOCOL`. If no parameter is specified in the command line, it just displays the IP configuration of `EFI_PXE_BASE_CODE_PROTOCOL`. If **IpAddress** or **NetMask** is specified, the according configuration of `EFI_PXE_BASE_CODE_PROTOCOL` will be changed.

## Examples

Displays or modifies the current IP configuration.

```
IPCONFIG [-r] | [-b] [-c Instance] [IpAddress [-m NetMask]]
```

```
-r          - Restart the PXE base code and DHCP settings
-b          - Display one screen at a time
Instance    - Zero-based Simple Network Protocol instance
IpAddress    - IP address in a.b.c.d format
NetMask      - Network mask in 255.255.255.0 format
```

Examples:

- \* To restart the PXE base code and refresh the DHCP settings:  
shell:\> IpConfig -r
- \* To display the current IP configuration:  
Shell:\> IpConfig
- \* To modify the IP address without changing the network mask:  
shell:\> IpConfig 192.168.10.30
- \* To modify the IP address and network mask:  
shell:\> IpConfig 192.168.10.30 -m 255.255.255.0

## 2.3.14 loadpcirom

### Summary

Loads a PCI Option ROM from the specified file.

### EFI Versions

EFI 1.10 and above.

### Usage

```
LoadPciRom [-nc] romfile [romfile...]
```

```
-nc          - Load the ROM image but do not connect the driver
romfile      - PCI option ROM image file (wildcards are permitted)
```

### Description

This command is used to load PCI option ROM images into memory for execution. The file can contain legacy images and multiple PE32 images, in which case all PE32 images will be loaded. The example below shows typical output from help for this command.

## Examples

```
Shell> help LoadPciRom
Loads a PCI Option ROM from the specified file.

LoadPciRom [-nc] romfile [romfile...]

        -nc          - Load the ROM image but do not connect the driver
romfile  - PCI option ROM image file (wildcards are permitted)

Examples:
* To load a rom file 'rom.bin':
fs0:\> LoadPciRom rom.bin
* To load '*.bin' files but do not connect the driver
fs0:\> LoadPciRom -nc *.bin
```

## 2.3.15 mem

### Summary

Displays the contents of system or device memory.

### EFI Versions

EFI 1.10 and above.

### Usage

```
MEM [-b] [Address] [Size] [-MMIO]

        -b          - Display one screen at a time
address  - Starting address in hexadecimal format
size     - Number of bytes to display in hexadecimal format
-MMIO    - Forces address cycles to the PCI bus
```

### Description

This command displays the contents of system memory or device memory. **Address** and **Size** should be typed in hex value. If Address is not specified, then the contents of the EFI System Table are displayed. Otherwise, memory starting at Address is displayed. **Size** specifies the number of bytes to display. If **Size** is not specified, then it defaults to 512 bytes. If **MMIO** is not specified, then main system memory is displayed. Otherwise, device memory is displayed through the use of the [EFI PCI ROOT BRIDGE IO PROTOCOL](#).

## Examples

```
Shell> help mem
Displays the contents of system or device memory.
```

```
MEM [-b] [Address] [Size] [-MMIO]
```

```
-b          - Display one screen at a time
address     - Starting address in hexadecimal format
size        - Number of bytes to display in hexadecimal format
-MMIO       - Forces address cycles to the PCI bus
```

### Note:

1. All units are in hexadecimal format.
2. Address must be aligned on an even processor address boundary.
3. If the 'address' parameter is not specified, DMEM will display the all system table pointer entries by default.

### Examples:

```
* To display the EFI system table pointer entries:
fs0:\> mem
```

```
Memory Address 000000003FF7D808 200 Bytes
3FF7D808: 49 42 49 20 53 59 53 54-02 00 01 00 78 00 00 00 *IBI SYST....x...*
3FF7D818: 5C 3E 6A FE 00 00 00 00-88 2E 1B 3F 00 00 00 00 *\>j.....?....*
3FF7D828: 26 00 0C 00 00 00 00 00-88 D3 1A 3F 00 00 00 00 *&.....?....*
3FF7D838: A8 CE 1A 3F 00 00 00 00-88 F2 1A 3F 00 00 00 00 *...?.....?....*
3FF7D848: 28 EE 1A 3F 00 00 00 00-08 DD 1A 3F 00 00 00 00 *(..?.....?....*
3FF7D858: A8 EB 1A 3F 00 00 00 00-18 C3 3F 3F 00 00 00 00 *...?.....*
3FF7D868: 00 4B 3F 3F 00 00 00 00-06 00 00 00 00 00 00 *..K.....*
3FF7D878: 08 DA F7 3F 00 00 00 00-70 74 61 6C 88 00 00 00 *...?....ptal...*
3FF7D888: 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 *.....*
3FF7D898: 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 *.....*
3FF7D8A8: 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 *.....*
3FF7D8B8: 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 *.....*
3FF7D8C8: 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 *.....*
3FF7D8D8: 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 *.....*
3FF7D8E8: 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 *.....*
3FF7D8F8: 00 00 00 00 00 00 00 00-70 68 06 30 88 00 00 00 *.....ph.0....*
3FF7D908: 65 76 6E 74 00 00 00 00-02 02 00 60 00 00 00 00 *evnt.....`....*
3FF7D918: 18 6F 1A 3F 00 00 00 00-10 E0 3F 3F 00 00 00 00 *.o.?......*
3FF7D928: 10 00 00 00 00 00 00 00-40 C0 12 3F 00 00 00 00 *.....@..?....*
3FF7D938: 10 80 13 3F 00 00 00 00-00 00 00 00 00 00 00 *...?.....*
3FF7D948: 00 00 00 00 00 00 00 00-40 7D 3F 3F 00 00 00 00 *.....@.....*
3FF7D958: 50 6F 1A 3F 00 00 00 00-00 00 00 00 00 00 00 *Po.?......*
3FF7D968: 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 *.....*
3FF7D978: 00 00 00 00 00 00 00 00-70 74 61 6C 88 00 00 00 *.....ptal...*
3FF7D988: 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 *.....*
3FF7D998: 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 *.....*
3FF7D9A8: 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 *.....*
3FF7D9B8: 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 *.....*
3FF7D9C8: 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 *.....*
3FF7D9D8: 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 *.....*
3FF7D9E8: 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 *.....*
3FF7D9F8: 00 00 00 00 00 00 00 00-70 68 06 30 A0 00 00 00 *.....ph.0....*
```

```
Valid EFI Header at Address 000000003FF7D808
```

```
-----
System: Table Structure size 00000078 revision 00010002
ConIn (3F1AD388) ConOut (3F1AF288) StdErr (3F1ADD08)
Runtime Services 000000003F3FC318
Boot Services   000000003F3F4B00
SAL System Table 000000003FFD2760
ACPI Table      000000003FFD9FC0
ACPI 2.0 Table  00000000000E2000
MPS Table       000000003FFD0000
SMBIOS Table    00000000000F0020
```

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

```
Shell> mem 1af3088 16
Memory Address 000000001AF3088 16 Bytes
01AF3088: 49 42 49 20 53 59 53 54-00 00 02 00 18 00 00 00 *IBI SYST.....*
```



```
01AF3098: FF 9E D7 9B 00 00 *.....*
* To display memory mapped IO contents from 1af3088 with size of 16
bytes:
Shell> mem 1af3088 16 -MMIO
```

## 2.3.16 memmap

### Summary

Displays the memory map maintained by the EFI environment.

### EFI Versions

EFI 1.10 and above.

### Usage

```
MEMMAP [-b]
-b          - Display one screen at a time
```

### Description

This command displays the memory map that is maintained by the EFI environment. The EFI environment keeps track all the physical memory in the system and how it is currently being used. The EFI Specification defines a set of Memory Type Descriptors. Please see the EFI Specification for a description of how each of these memory types is used. The following example shows typical output for help on this command.

## Examples

```
Shell> help memmap
Displays the memory map maintained by the EFI environment.
```

```
MEMMAP [-b]
```

```
-b          - Display one screen at a time
```

Note:

1. The EFI environment keeps track all the physical memory in the system and how it is currently being used.
2. Total memory is the physical memory size not including the MemMapIO and MemPortIO size.
3. Refer to the EFI specification for memory type definitions.

Examples:

```
* To display the system memory map:
fs0:\> memmap
```

Type	Start	End	# Pages	Attributes
available	0000000000750000	00000000001841FFF	00000000000010F2	0000000000000009
LoaderCode	00000000001842000	000000000018A3FFF	0000000000000062	0000000000000009
available	000000000018A4000	000000000018C1FFF	000000000000001E	0000000000000009
LoaderData	000000000018C2000	000000000018CAFFF	0000000000000009	0000000000000009
BS_code	000000000018CB000	00000000001905FFF	000000000000003B	0000000000000009
BS_data	00000000001906000	000000000019C9FFF	00000000000000C4	0000000000000009
...				
RT_data	00000000001B2B000	00000000001B2BFFF	0000000000000001	8000000000000009
BS_data	00000000001B2C000	00000000001B4FFFF	0000000000000024	0000000000000009
reserved	00000000001B50000	00000000001D4FFFF	0000000000000200	0000000000000009

  

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

## 2.3.17 mm

### Summary

Displays or modifies MEM/MMIO/IO/PCI/PCIE address space.

### EFI Versions

EFI 1.10 and above.

## Usage

```
MM Address [Value] [-w 1|2|4|8] [-MEM | -MMIO | -IO | -PCI | -PCIE] [-n]

Address - Starting address
Value   - The value to write
-MEM    - Memory Address type
-MMIO   - Memory Mapped IO Address type
-IO     - IO Address type
-PCI    - PCI Configuration Space Address type:
          Address format: 0x000000ssbbddffrr
          ss - Segment
          bb - Bus
          dd - Device
          ff - Function
          rr - Register
-PCIE   - PCIE Configuration Space Address type:
          Address format: 0x000000ssbbddffrrr
          ss - Segment
          bb - Bus
          dd - Device
          ff - Function
          rrr - Register
-w      - Unit size accessed in bytes:
          1 - 1 byte
          2 - 2 bytes
          4 - 4 bytes
          8 - 8 bytes
-n      - Non-interactive mode
```

## Description

This command allows the user to display or modify I/O register, memory contents, or PCI configuration space. The user can specify the start address and the access size they wish to perform using the **Address** parameter and **-w** option. **Address** should be typed in hex value. **-MEM** accesses system memory, **-MMIO** accesses device memory, **-IO** accesses device I/O ports, **-PCI** accesses PCI Configuration Space, and **-PCIE** accesses PCIE Configuration Space. If **Value** is specified which should be typed in hex value, this command will write this value to specified address. Otherwise when this command is executed, the current contents of **Address** are displayed. If **-n** is not specified, the command will run in interactive mode and the user has the option of modifying the contents by typing in a hex value. When the user pressed 'ENTER', then next address is displayed. This is continued until the user enters 'q'.

## Examples

```
Shell> help mm
```

```
Displays or modifies MEM/MMIO/IO/PCI/PCIE address space.
```

```
MM Address [Value] [-w 1|2|4|8] [-MEM | -MMIO | -IO | -PCI | -PCIE] [-n]
```

```

Address - Starting address
Value   - The value to write
-MEM    - Memory Address type
-MMIO   - Memory Mapped IO Address type
-IO     - IO Address type
-PCI    - PCI Configuration Space Address type:
          Address format: 0x000000ssbbddffrr
          ss - Segment
          bb - Bus
          dd - Device
          ff - Function
          rr - Register
-PCIE   - PCIE Configuration Space Address type:
          Address format: 0x000000ssbbddffrr
          ss - Segment
          bb - Bus
          dd - Device
          ff - Function
          rrr - Register
-w      - Unit size accessed in bytes:
          1 - 1 byte
          2 - 2 bytes
          4 - 4 bytes
          8 - 8 bytes
-n      - Non-interactive mode
```

### Note:

1. If the address type parameter is not specified, address type defaults to the 'MEM' type.
2. If the 'Value' parameter is specified, the '-n' option will be used automatically. In this case, this command will write the value to the specified address in non-interactive mode. If the 'Value' parameter is not specified, only the current contents in the address are displayed.
3. If the '-w' option is not specified, unit size defaults to 1 byte.
4. If the PCI address type is specified, the 'Address' parameter should follow the PCI Configuration Space Address format above. The 'PCI' command can be used to determine the address for a specified device. It is listed in the PCI configuration space dump information, in the following format:  
"[EFI 0x000000ssbbddffxx]".
5. If the PCIE address type is specified, the 'Address' parameter should follow the PCIE Configuration Space Address format above.
6. In interactive mode, type a hex value to modify, 'q' or '.' to exit. If the '-n' option is specified, it will run in non-interactive mode which supports batch file operation without user intervention.
7. Not all PCI configuration register locations are writable.
8. MM will only write the specified value. Read-modify-write operations are not supported.
9. The 'Address' parameter should be aligned on a boundary of the specified width.
10. Not all addresses are safe to access. Access to any improper address can bring unexpected results.

### Examples:

- \* To display or modify memory:  
Address 0x1b07288, default width=1 byte:

```

fs0:\> mm 1b07288
MEM 0x0000000001B07288 : 0x6D >
MEM 0x0000000001B07289 : 0x6D >
MEM 0x0000000001B0728A : 0x61 > 80
MEM 0x0000000001B0728B : 0x70 > q

fs0:\> mm 1b07288
MEM 0x0000000001B07288 : 0x6D >
MEM 0x0000000001B07289 : 0x6D >
MEM 0x0000000001B0728A : 0x80 > *Modified
MEM 0x0000000001B0728B : 0x70 > q

* To modify memory:
Address 0x1b07288, width = 2 bytes:
Shell> mm 1b07288 -w 2
MEM 0x0000000001B07288 : 0x6D6D >
MEM 0x0000000001B0728A : 0x7061 > 55aa
MEM 0x0000000001B0728C : 0x358C > q

Shell> mm 1b07288 -w 2
MEM 0x0000000001B07288 : 0x6D6D >
MEM 0x0000000001B0728A : 0x55AA > *Modified
MEM 0x0000000001B0728C : 0x358C > q

* To display IO space:
Address 80h, width = 4 bytes:
Shell> mm 80 -w 4 -IO
IO 0x0000000000000080 : 0x000000FE >
IO 0x0000000000000084 : 0x00FF5E6D > q

* To modify IO space using non-interactive mode:
Shell> mm 80 52 -w 1 -IO
Shell> mm 80 -w 1 -IO
IO 0x0000000000000080 : 0x52 > FE *Modified
IO 0x0000000000000081 : 0xFF >
IO 0x0000000000000082 : 0x00 >
IO 0x0000000000000083 : 0x00 >
IO 0x0000000000000084 : 0x6D >
IO 0x0000000000000085 : 0x5E >
IO 0x0000000000000086 : 0xFF >
IO 0x0000000000000087 : 0x00 > q

* To display PCI configuration space, ss=00, bb=00, dd=00, ff=00,
rr=00:
Shell> mm 0000000000 -PCI
PCI 0x0000000000000000 : 0x86 >
PCI 0x0000000000000001 : 0x80 >
PCI 0x0000000000000002 : 0x30 >
PCI 0x0000000000000003 : 0x11 >
PCI 0x0000000000000004 : 0x06 >
PCI 0x0000000000000005 : 0x00 > q
* These contents can also be displayed by 'PCI 00 00 00'.

* To display PCIE configuration space, ss=00, bb=06, dd=00, ff=00,
rrr=000:
Shell> mm 000600000000 -PCIE
PCIE 0x0000000060000000 : 0xAB >
PCIE 0x0000000060000001 : 0x11 >
PCIE 0x0000000060000002 : 0x61 >
PCIE 0x0000000060000003 : 0x43 >
PCIE 0x0000000060000004 : 0x00 > q

```

## 2.3.18 mode

## Summary

Displays or changes the console output device mode.

## EFI Versions

EFI 1.10 and above.

## Usage

```
MODE [row col]

row      - Number of rows
col      - Number of columns
```

## Description

This command is used to change the display mode for the console output device. When this command is used without any parameters, it shows the list of modes that the standard output device currently supports. The mode command can then be used with the **row** and **col** parameter to change the number of rows and columns on the standard output device. The following examples show how the mode command can be used. The first example lists all modes that are currently available, and the current selected mode is indicated by an '\*'. The second example changes the mode to an 80 X 50 text mode display. The display is cleared every time the mode command is used to change the currently selected display mode.

## Examples

```
Shell> help mode
Displays or changes the console output device mode.
```

```
MODE [row col]
```

```
row      - Number of rows
col      - Number of columns
```

Note:

1. The mode with a star at line end is the current mode setting.
2. Only 80x25 mode is supported if console redirection is enabled.
3. Most EFI command output was designed for 80x50 mode (50 rows).

Examples:

- \* To display all available modes on standard output:

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

- \* To change the current mode setting:

```
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
```

## 2.3.19 openinfo

### Summary

Displays the protocols and agents associated with a handle.

### EFI Versions

EFI 1.10 and above.

### Usage

```
OPENINFO Handle [-b]
```

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

### Description

This command is used to display the open protocols on a given handle. The example below is typical output from help for this command.

**Table 7 Open Protocol Information Layout**

Column Index	Description
1	Agent handle that opens the protocol
2	Controller handle that requires the protocol interface

3	Open count
4	Open type: <b>HandProt</b> , <b>GetProt</b> , <b>TestProt</b> , <b>Child</b> , <b>Driver</b> , <b>Exclusive</b> , <b>DriverEx</b> or <b>Unknown</b>
5	Name of image of the agent if available

## Examples

```
Shell> help openinfo
Displays the protocols and agents associated with a handle.

OPENINFO Handle [-b]

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

Examples:
* To show open protocols on handle 0x23:
  It shows that the PCI Root Bridge is being managed by the PCI
  Bus Driver, and the PCI Bus contains 7 PCI child controllers.
  Shell> openinfo 23
  Handle 23 (07DEE108)
  PciRootBridgeIo
    Drv[1D] Ctrl[23] Cnt(01) Driver   Image(PciBus)
    Drv[1D] Ctrl[28] Cnt(01) Child   Image(PciBus)
    Drv[1D] Ctrl[29] Cnt(01) Child   Image(PciBus)
    Drv[1D] Ctrl[2A] Cnt(01) Child   Image(PciBus)
    Drv[1D] Ctrl[2B] Cnt(01) Child   Image(PciBus)
    Drv[1D] Ctrl[2C] Cnt(01) Child   Image(PciBus)
    Drv[1D] Ctrl[2D] Cnt(01) Child   Image(PciBus)
    Drv[1D] Ctrl[2E] Cnt(01) Child   Image(PciBus)
    Drv[00] Ctrl[ ]  Cnt(01) HandProt
  dpath
    Drv[1D] Ctrl[23] Cnt(01) Driver   Image(PciBus)
    Drv[00] Ctrl[ ]  Cnt(0D) HandProt
```

### 2.3.20 pci

#### Summary

Displays PCI device list or PCI function configuration space.

#### EFI Versions

EFI 1.10 and above.



## Usage

```
PCI [Bus Dev [Func] [-s Seg] [-i]]

Bus      - Bus number
Dev      - Device number
Func     - Function number
-s       - Optional segment number specified
Seg      - Segment number
-i       - Information interpreted
```

## Description

This command will display all the PCI devices found in the system. And it can also display the configuration space of PCI device according to specified bus (**Bus**), device (**Dev**), and function (**Func**) addresses. If the function address is not specified, it will default to 0. The **-i** option is used to display verbose information for the specified PCI device. The PCI configuration space for the device will be dumped with a detailed interpretation.

## Examples

```
Shell> help pci
Displays PCI device list or PCI function configuration space.
```

```
PCI [Bus Dev [Func] [-s Seg] [-i]]
```

```
Bus      - Bus number
Dev      - Device number
Func     - Function number
-s       - Optional segment number specified
Seg      - Segment number
-i       - Information interpreted
```

### Note:

1. If no parameters are specified all PCI devices will be listed.
2. If the Bus and Device number parameters are specified while the Function or Segment parameters are not, Function or Segment will be set as default value 0.
3. The '-i' option can be used to display verbose information for the specified PCI device. The PCI configuration space for the specified device will be dumped with a detailed interpretation.

### Examples:

\* To display all PCI devices in the system:

```
Shell> PCI
```

Seg	Bus	Dev	Func	
00	00	00	00	==> Bridge Device - Host/PCI bridge
				Vendor 8086 Device 1130 Prog Interface 0
00	00	01	00	==> Bridge Device - PCI/PCI bridge
				Vendor 8086 Device 1131 Prog Interface 0
00	00	1E	00	==> Bridge Device - PCI/PCI bridge
				Vendor 8086 Device 244E Prog Interface 0
00	00	1F	00	==> Bridge Device - PCI/ISA bridge
				Vendor 8086 Device 2440 Prog Interface 0
00	00	1F	01	==> Mass Storage Controller - IDE controller
				Vendor 8086 Device 244B Prog Interface 80
00	00	1F	02	==> Serial Bus Controllers - USB
				Vendor 8086 Device 2442 Prog Interface 0
00	00	1F	03	==> Serial Bus Controllers - System Management Bus
				Vendor 8086 Device 2443 Prog Interface 0
00	00	1F	04	==> Serial Bus Controllers - USB
				Vendor 8086 Device 2444 Prog Interface 0
00	00	1F	05	==> Multimedia Device - Audio device
				Vendor 8086 Device 2445 Prog Interface 0
00	00	1F	06	==> Simple Communications Controllers - Modem
				Vendor 8086 Device 2446 Prog Interface 0
00	01	00	00	==> Display Controller - VGA/8514 controller
				Vendor 1002 Device 5246 Prog Interface 0
00	02	07	00	==> Multimedia Device - Audio device
				Vendor 1274 Device 1371 Prog Interface 0
00	02	0A	00	==> Bridge Device - CardBus bridge
				Vendor 1180 Device 0476 Prog Interface 0
00	02	0A	01	==> Bridge Device - CardBus bridge
				Vendor 1180 Device 0476 Prog Interface 0

\* To display the configuration space of Bus 0, Device 0, Function 0:

```
Shell> PCI 00 00 00 -i
```

```
PCI Segment 00 Bus 00 Device 00 Func 00
00000000: 86 80 30 11 06 00 90 20-02 00 00 06 00 00 00 00 *..0....*
00000010: 08 00 00 20 00 00 00 00-00 00 00 00 00 00 00 00 *...*
00000020: 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 *.....*
00000030: 00 00 00 00 88 00 00 00-00 00 00 00 00 00 00 00 *.....*
```

```

00000040: 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 *.....*
00000050: 50 00 09 38 00 00 00 00-00 00 00 00 00 00 00 00 *P..8.....*
00000060: 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 *.....*
00000070: 00 00 18 00 00 00 00 00-00 00 00 00 00 00 00 00 *.....*
00000080: DE 2C CF 00 00 00 00 00-09 A0 04 F1 00 00 00 00 00 *.....*
00000090: 00 00 D6 FF FE FF 00 00-33 80 33 80 85 84 C4 00 *.....3.3.....*
000000A0: 02 00 20 00 07 02 00 1F-00 00 00 00 00 00 00 00 *.. .....*
000000B0: 00 00 00 00 30 00 00 00-00 00 00 00 00 00 08 00 *....0.....*
000000C0: 00 00 00 00 00 00 00 00-00 08 00 00 00 00 00 00 *.....*
000000D0: 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 *.....*
000000E0: 00 00 00 00 00 00 00 00-00 00 90 14 00 00 00 00 *.....*
000000F0: 00 00 00 00 74 F8 00 00-00 00 00 00 08 00 00 00 *....t.....*

Vendor ID(0): 8086                      Device ID(2): 1130

Command(4): 0006
(00)I/O space access enabled:          0 (01)Memory space access enabled:      1
(02)Behave as bus master:              1 (03)Monitor special cycle enabled:    0
(04)Mem Write & Invalidate enabled:    0 (05)Palette snooping is enabled:      0
(06)Assert PERR# when parity error:    0 (07)Do address/data stepping:        0
(08)SERR# driver enabled:              0 (09)Fast back-to-back transact....:  0

Status(6): 2090
(04)New Capabilities linked list:      1 (05)66MHz Capable:                    0
(07)Fast Back-to-Back Capable:        1 (08)Master Data Parity Error:          0
(09)DEVSEL timing:                    Fast (11)Signaled Target Abort:            0
(12)Received Target Abort:             0 (13)Received Master Abort:            1
(14)Signaled System Error:             0 (15)Detected Parity Error:            0

Revision ID(8):      02                  BIST(0F): Incapable
Cache Line Size(C): 00                  Latency Timer(D): 00
Header Type(0E):     0, Single function, PCI device
Class: Bridge Device - Host/PCI bridge -
Base Address Registers(10):
      Start  Type   Space   Prefetchable?   Size      Limit
-----
20000000  Mem     32 bits   YES              04000000  24000000
-----

No Expansion ROM(30)

Cardbus CIS ptr(28): 00000000
Sub VendorID(2C):    0000              Subsystem ID(2E):    0000
Capabilities Ptr(34): 88
Interrupt Line(3C):  00              Interrupt Pin(3D):    00
Min_Gnt(3E):         00              Max_Lat(3F):         00

* To display configuration space of Segment 0, Bus 0, Device 0, Function 0:
Shell> PCI 00 00 00 -s 0

PCI Segment 00 Bus 00 Device 00 Func 00
00000000: 86 80 30 11 06 00 90 20-02 00 00 06 00 00 00 00 *..0.... .....*
00000010: 08 00 00 20 00 00 00 00-00 00 00 00 00 00 00 00 *... .....*
00000020: 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 *.....*
00000030: 00 00 00 00 88 00 00 00-00 00 00 00 00 00 00 00 *.....*
00000040: 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 *.....*
00000050: 50 00 09 38 00 00 00 00-00 00 00 00 00 00 00 00 *P..8.....*
00000060: 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 *.....*
00000070: 00 00 18 00 00 00 00 00-00 00 00 00 00 00 00 00 *.....*
00000080: DE A8 CE 00 00 00 00 00-09 A0 04 F1 00 00 00 00 00 *.....*
00000090: 00 00 D6 FF FE FF 00 00-33 80 33 80 85 84 C4 00 *.....3.3.....*
000000A0: 02 00 20 00 07 02 00 1F-00 00 00 00 00 00 00 00 *.. .....*
000000B0: 00 00 00 00 30 00 00 00-00 00 00 00 00 00 08 00 *....0.....*
000000C0: 00 00 00 00 00 00 00 00-00 08 00 00 00 00 00 00 *.....*
000000D0: 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 *.....*
000000E0: 00 00 00 00 00 00 00 00-00 00 A0 18 00 00 00 00 *.....*
000000F0: 00 00 00 00 74 F8 00 00-00 00 00 00 08 00 00 00 *....t.....*

```

### 2.3.21 sermode

#### Summary

Sets serial port attributes.

## EFI Versions

EFI 1.10 and above.

## Usage

```
SERMODE [handle [baudrate parity databits stop bits]]

handle    - Device handle for a serial port
baudrate  - Baud rate for specified serial port
parity    - Parity bit settings for specified serial port:
             d - Default parity
             n - No parity
             e - Even parity
             o - Odd parity
             m - Mark parity
             s - Space parity
databits  - Data bits for specified serial port
stopbits  - Stop bits for specified serial port
```

## Description

This command displays or sets baud rate, parity attribute, data bits and stop bits of serial ports.

## Examples

```
Shell> help sermode
Sets serial port attributes.
```

```
SERMODE [handle [baudrate parity databits stop bits]]
```

```
handle   - Device handle for a serial port
baudrate - Baud rate for specified serial port
parity   - Parity bit settings for specified serial port:
            d - Default parity
            n - No parity
            e - Even parity
            o - Odd parity
            m - Mark parity
            s - Space parity
databits - Data bits for specified serial port
stopbits - Stop bits for specified serial port
```

### Note:

1. The 'handle' parameter is the device handle of the desired serial port. The 'DH' command can be used to retrieve this information.
2. The 'stopbits' parameter supports the following settings:
  - 0 (0 stop bits - default setting)
  - 1 (1 stop bit)
  - 2 (2 stop bits)
  - 15 (1.5 stop bits)All other settings are invalid.
3. The 'baudrate' parameter supports the following settings:
  - 50, 75, 110, 150, 300, 600, 1200, 1800, 2000, 2400, 3600, 4800, 7200, 9600(default), 19200, 38400, 57600, 115200, 230400, 460800All other values will be converted to the next highest setting.
4. The 'databits' parameter supports the following settings:
  - 4
  - 7
  - 8 (default)All other settings are invalid.
5. Parity attributes are mutually exclusive.

### Examples:

- \* To display the settings for all serial port devices:

```
Shell> sermode
4F06B08 - (115200, N, 8, 1)
4F05F88 - (115200, N, 8, 1)
```
- \* To display the settings for the serial port device whose handle is 0x6B:

```
Shell> sermode 6B
4F06B08 - (115200, N, 8, 1)
```
- \* To configure the serial port settings for handle 0x6B to 9600bps, even parity, 8 data bits, and 1 stop bit:

```
Shell> sermode 6B 9600 e 8 1
sermode: Mode set on handle 04F06B08
```

## 2.3.22 smbiosview

### Summary

Displays SMBIOS information.

## EFI Versions

EFI 1.10 and above.

## Usage

```
SMBIOSVIEW [-t SmbiosType] | [-h SmbiosHandle] | [-s] | [-a]

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

## Description

This command displays the SMBIOS information. Users can display the information of SMBIOS structures specified by type or handle.

## Examples

```
Shell> help smbiosview
Displays SMBIOS information.
```

```
SMBIOSVIEW [-t SmbiosType] | [-h SmbiosHandle] | [-s] | [-a]

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

### Notes:

1. The SmbiosType parameter supports the following types:
  - 0 - BIOS Information
  - 1 - System Information
  - 3 - System Enclosure
  - 4 - Processor Information
  - 5 - Memory Controller Information
  - 6 - Memory Module Information
  - 7 - Cache Information
  - 8 - Port Connector Information
  - 9 - System Slots
  - 10 - On Board Devices Information
  - 15 - System Event Log
  - 16 - Physical Memory Array
  - 17 - Memory Device
  - 18 - 32-bit Memory Error Information
  - 19 - Memory Array Mapped Address
  - 20 - Memory Device Mapped Address
  - 21 - Built-in Pointing Device
  - 22 - Portable Battery
  - 34 - Management Device
  - 37 - Memory Channel
  - 38 - IPMI Device Information
  - 39 - System Power Supply
2. The SmbiosHandle parameter can be specified in either decimal or hexadecimal format. Use the '0x' prefix format for hexadecimal values.
3. Internal commands:
  - :q ----- quit smbiosview
  - :0 ----- Change smbiosview display NONE info
  - :1 ----- Change smbiosview display OUTLINE info
  - :2 ----- Change smbiosview display NORMAL info
  - :3 ----- Change smbiosview display DETAIL info
  - /? ----- Show help

## 2.3.23 telnetmgmt

### Summary

Change terminal type

### EFI Versions

EFI 1.10 and above.

## Usage

```
TELNETMGMT [-t <TerminalType>]

-t          - Specifies the terminal type
              Available types:
                0 - PcAnsi type
                1 - vt100 type
                2 - vt100 plus type
                3 - VTUTF8 type
```

## Description

This command sets the terminal type for the system.

## Examples

```
Shell> help telnetmgmt
Change terminal type

TELNETMGMT [-t <TerminalType>]

-t          - Specifies the terminal type
              Available types:
                0 - PcAnsi type
                1 - vt100 type
                2 - vt100 plus type
                3 - VTUTF8 type
```

```
Examples:
* Set the terminal type to PcAnsi:
Shell> telnetmgmt -t 0
```

## 2.3.24 timezone

### Summary

Displays or sets time zone information.

### EFI Versions

EFI 1.10 and above.

### Usage

```
TIMEZONE [-s hh:mm | -l] [-b] [-f]

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

### Description

This command displays and sets the current time zone for the system. If no parameters are used, it shows the current time zone. If valid **hh:mm** parameter is provided, then the system's time zone information will be updated.



## Examples

```
Shell> timezone
Displays or sets time zone information.
```

```
TIMEZONE [-s hh:mm | -l] [-b] [-f]
```

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

### Examples:

- \* To display all available time zones:

```
Shell> timezone -l
GMT-12:00, International Date Line West
GMT-11:00, Midway Island, Samoa
GMT-10:00, Hawaii
GMT-09:00, Alaska
GMT-08:00, Pacific Time(US & Canada); Tijuana
GMT-07:00, Arizona, Chihuahua, La Paz, Mazatlan
GMT-06:00, Central America, Central Time(US & Canada)
GMT-05:00, Bogota, Lima, Quito, Eastern Time(US & Canada)
GMT-04:00, Atlantic Time(Canada), Caracas, Santiago
GMT-03:30, Newfoundland
GMT-03:00, Brasilia, Buenos Aires, Georgetown, Greenland
GMT-02:00, Mid-Atlantic
GMT-01:00, Azores, Cape Verde Is.
GMT, Greenwich Mean Time, Casablanca, Monrovia, Dublin, London
GMT+01:00, Amsterdam, Berlin, Bern, Rome, Paris, West Central Africa
GMT+02:00, Athens, Istanbul, Bucharest, Cairo, Jerusalem
GMT+03:00, Baghdad, Kuwait, Riyadh, Moscow, Nairobi
GMT+03:30, Tehran
GMT+04:00, Abu Dhabi, Muscat, Baku, Tbilisi, Yerevan
GMT+04:30, Kabul
GMT+05:00, Ekaterinburg, Islamabad, Karachi, Tashkent
GMT+05:30, Chennai, Kolkata, Mumbai, New Delhi
GMT+05:45, Kathmandu
GMT+06:00, Almaty, Novosibirsk, Astana, Dhaka, Sri Jayawardenepura
GMT+06:30, Rangoon
GMT+07:00, Bangkok, Hanoi, Jakarta, Krasnoyarsk
GMT+08:00, Beijing, Chongqing, Hong Kong, Urumqi, Taipei, Perth
GMT+09:00, Osaka, Sapporo, Tokyo, Seoul, Yakutsk
GMT+09:30, Adelaide, Darwin
GMT+10:00, Canberra, Melbourne, Sydney, Guam, Hobart, Vladivostok
GMT+11:00, Magadan, Solomon Is., New Caledonia
GMT+12:00, Auckland, Wellington, Fiji, Kamchatka, Marshall Is.
GMT+13:00, Nuku'alofa
```

- \* To set the time zone:

```
Shell> timezone -s -7:00
Shell> timezone
GMT-07:00
```

```
Shell> timezone -s 5:00
Shell> timezone
GMT+05:00
```

- \* To display detailed information for the current time zone:

```
Shell> timezone -f
GMT+05:00, Ekaterinburg, Islamabad, Karachi, Tashkent
Shell> timezone
GMT+05:00
```