

CHAPTER 1

Booting: EFI and Management Processor

Background

The chapter covers booting an HP Itanium-based system. Booting an Itanium-based system requires that you become familiar with two levels of firmware. The first level is HP's implementation of Intel's EFI (Extensible Firmware Interface), *POSSE* (Pre-OS System Environment). The second is HP's independent support processor for the system console, the *Management Processor*. Although functionality varies from one platform to another the topics in this chapter apply to all HP Itanium-based systems at the time of this writing. The systems used throughout this chapter are an rx2600 two-way Itanium system and an rx5670 four-way Itanium system. Since these two systems do not support hard partitions (nPartitions) we don't cover any low-level partition-related commands. When Itanium processors are available on higher end systems, support for hard partitions will be added.

This chapter covers the following topics.

- Extensible Firmware Interface (EFI) and Pre-OS System Environment (POSSE)
- Management Processor (MP)

In addition to providing background on these topics, this chapter also provides lots of examples of running commands and analyzing output for both POSSE and MP.

At the end there is an excellent tear-out card that you can use as a quick reference for many boot-related commands. In addition to the information covered in this chapter, the quick reference card also summarizes a lot of the information related to boot your system and working with partitions, subjects which are covered later in this book.

Extensible Firmware Interface (EFI) and Pre-OS System Environment (POSSE)

EFI is an interface between your operating systems and platform firmware. POSSE is the HP implementation of EFI which contains additional commands beyond the ones available through EFI alone. We'll use the EFI acronym in this chapter but you should be aware that some HP documentation may use POSSE. EFI is a component that is independent of the operating system and provides a shell for interfacing to multiple operating systems. The interface consists of data tables that contain platform-related information along with boot and runtime service calls that are available to the operating system and its loader. These components work together to provide a standard environment for booting multiple operating systems.

If your interested in finding out more about EFI than is documented here, I suggest that you take a look at Intel EFI Web site. At the time of this writing, EFI information can be found at: <http://www.intel.com/technology/efi>.

Fill in URLs from "Using the EFI"

As you can see in Figure 1-1, EFI on HP Itanium-based systems contains several layers. The hardware layer contains disk with an EFI partition, which in turn has in it an operating system loader. This layer also contains one or more operating system partitions.

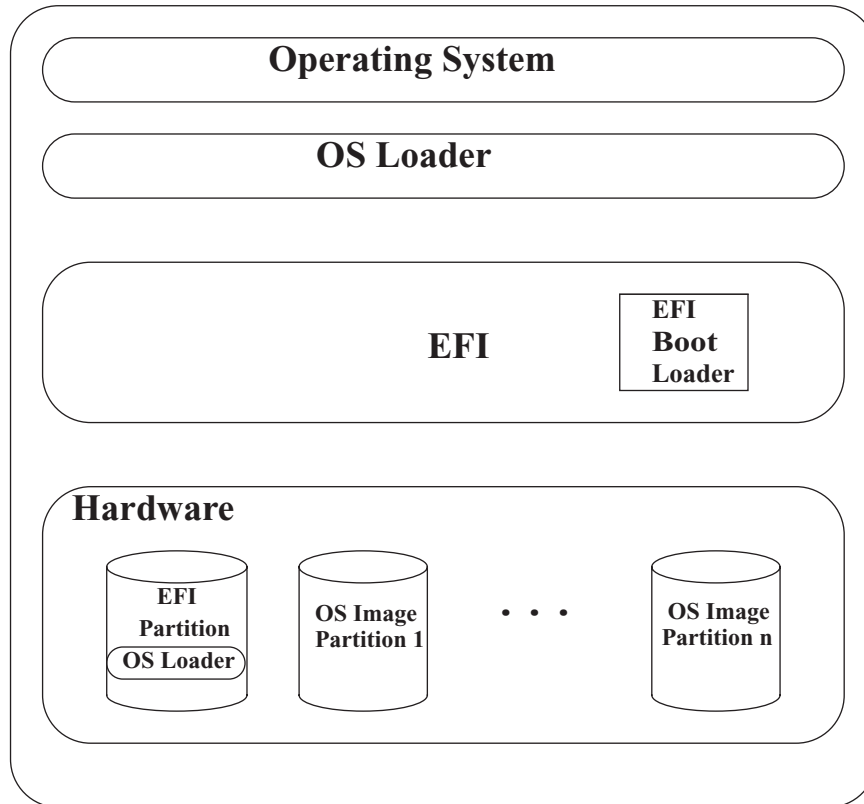


Figure 1-1 EFI on HP Itanium-Based Systems

In addition, the EFI system partition itself consists of several different components as shown in Figure 1-2:

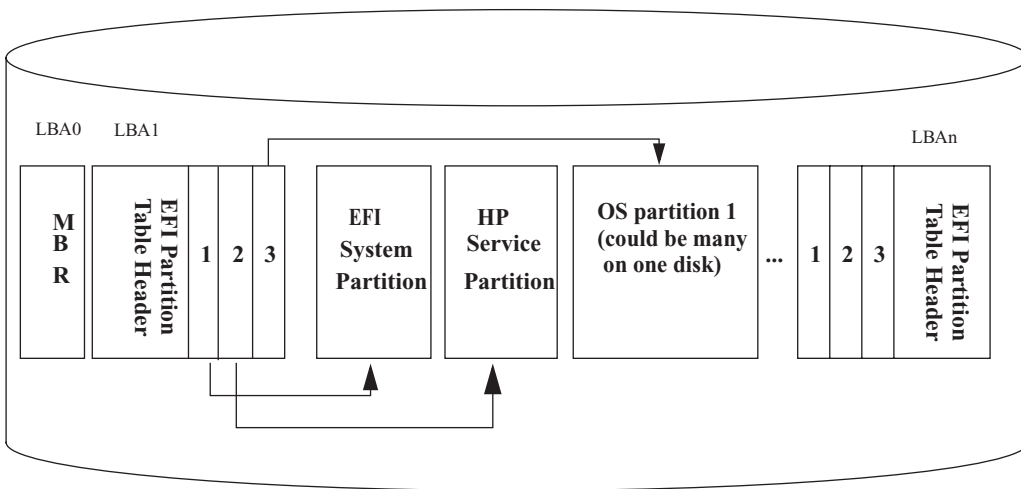


Figure 1-2 EFI System Partition

The Logical Block Addresses (LBAs) are shown across the top of the diagram. The Master Boot Record (MBR) is the first LBA. There is then a partition table. Three partitions are shown on this disk. Note that multiple operating system partitions can be loaded on the same disk. At the time of this writing Windows Server 2003 and Linux can be loaded on the same disk. The EFI partition table on the right is a backup partition table.

Booting an operating system with EFI on an Itanium-based system involves several steps. Figure 1-3 depicts the high-level steps:

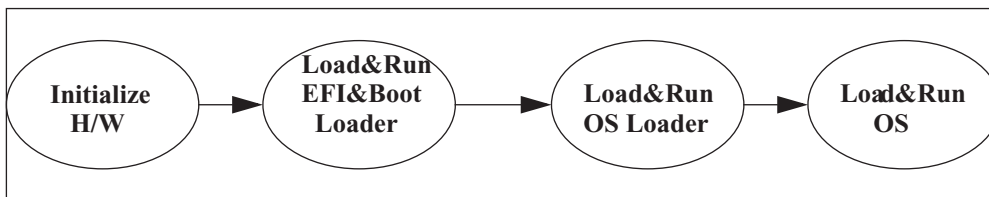


Figure 1-3 Load and Run an Operating System

The first step is to initialize the hardware. This takes place at the lowest level (BIOS) before EFI or the operating systems play any part in the pro-

cess. Next the EFI and boot loader are loaded and run. After an operating system is chosen, the operating system loader is loaded and run for the specific operating system being booted. Finally, the operating system itself is loaded and run. There are no specific operating systems cited in the figure 1-3 because the process is the same regardless of the operating system being loaded. In the examples in this book, Linux, HP-UX, and Windows are used and all of these operating systems would load in the same manner.

Working with EFI

Traversing the EFI menu structure and issuing commands is straight forward. You make your desired selections and then traverse a menu hierarchy. To start EFI, when the system self test is complete, hit any key to break the normal boot process. The main EFI screen appears. Figure 1-4 shows the EFI Boot Administration main screen from which you can make various boot-related selections:



Figure 1-4 EFI Boot Administration Main Screen

Figure 1-4 shows that there are two operating systems installed on this Itanium-based system -- an *HP-UX Primary Boot* and a *Red Hat Linux Advanced Server*. Either of these can be booted. From the main screen you

can also choose either *EFI Shell [Built-in]*, *Boot option maintenance menu*, or *Security/Password menu*. The first item shown in the EFI main screen is the default. Use the arrow keys to scroll and highlight a selection. After the item you need is highlighted, press *Enter* to select it. For example, if you were to select *Boot option maintenance menu* you would see a screen resembling the one shown in Figure 1-5:

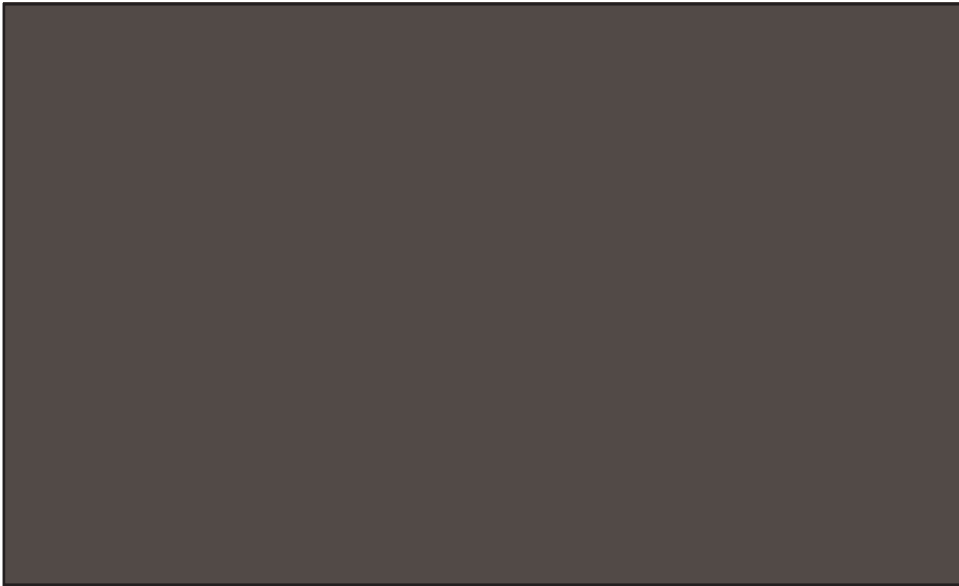


Figure 1-5 EFI Boot Maintenance Manager Main Menu

These two figures (1-4 and 1-5) give you an feeling for the menu-driven nature of EFI and EFI selections.

One of the important things a system administrator might need to know is a given system's device mappings. To view mappings using EFI, you need to get to a console-like *EFI Shell>* prompt. To do so, you must select *EFI Shell* in figure 1-4. Once at the prompt, there are a variety of commands that you can run--including **map**. **map** is the EFI command to show device mapping on our Itanium-based system. The listing below shows the output from the **map** command.

```
Shell> map
fs0 : Acpi (HWP0002,100)/Pci (1|0)/Scsi (Pun0,Lun0)/HD (Part1,Sig8E89981A-0B97-11D7-9C4C
      - AF87605217DA)
fs1 : Acpi (HWP0002,100)/Pci (1|0)/Scsi (Pun1,Lun0)/HD (Part1,Sig7C0F0000)
blk0 : Acpi (HWP0002,0)/Pci (2|0)/Ata (Primary,Master)
blk1 : Acpi (HWP0002,100)/Pci (1|0)/Scsi (Pun0,Lun0)
blk2 : Acpi (HWP0002,100)/Pci (1|0)/Scsi (Pun0,Lun0)/HD (Part1,Sig8E89981A-0B97-11D7-9C4C
      - AF87605217DA)
blk3 : Acpi (HWP0002,100)/Pci (1|0)/Scsi (Pun0,Lun0)/HD (Part2,SigC9D59DF0-0BA7-11D7-9B31
```

```

blk4 : Acpi(HWP0002,100)/Pci(1|0)/Scsi(Pun0,Lun0)/HD(Part3,SigC9D7945C-0BA7-11D7-9B31
- FBA1AECDAF7E)
blk5 : Acpi(HWP0002,100)/Pci(1|0)/Scsi(Pun1,Lun0)
- FBA1AECDAF7E)
blk6 : Acpi(HWP0002,100)/Pci(1|0)/Scsi(Pun1,Lun0)/HD(Part1,Sig7C0F0000)
blk7 : Acpi(HWP0002,100)/Pci(1|0)/Scsi(Pun1,Lun0)/HD(Part2,Sig7C0F0000)

```

The device mappings can be difficult to read. Much of the information is intended for programmers and technicians. As a system administrator, however, you need to know which entries correspond to which devices and which entries are for your partitions and file systems. To determine that, let's take a look at each entry.

As you probably guessed, file systems begin with *fs* and block devices begin with *blk* in this listing. To understand these entries however, let's take a look at them individually. To make their meaning a little clearer, I've grouped the entries differently than they originally appeared in the EFI listing above.

Red Hat Advanced Server disk and related entries:

```

blk1 physical disk
blk2 is first partition on blk1
blk3 is second partition on blk1
blk4 is third partition on blk1
fs0 is first file system on blk1

```

HP-UX 11i disk and related entries:

```

blk5 physical disk
blk6 is first partition on blk5
blk7 is second partition on blk5
fs1 is first file system on blk5

```

Let's analyze one of the block (*blk*) entries.

```

blk1 : Acpi(HWP0002,100)/Pci(1|0)/Scsi(Pun0,Lun0)

```

blk is the label assigned to the physical drive and the *1* is the number of the physical drive (*blk* can also be a partition on a physical drive as we'll see shortly.) *Acpi(HWP0002,100)* first shows a device type of *HWP0002* with a PCI host number of *100*. This PCI host number is often called the *ROPE*. The *ROPE* is the circuitry that handles I/O for the PCI interface. Although this information is most often used by programmers, it is sometimes handy to know the *ROPE* since it also defines the I/O card slot. The following types of devices are the most common:

HWP0001: Single I/O Controller Single Block Address w/o I/O Controller in the namespace.
HWP0002: Logical Block Address (LBA) device.
HWP0003: AGP LBA device.

After the *ROPE* we find a *Pci* entry. This entry indicates that the device/slot number is *1* and the function number is *0*.

The *Scsi Pun* (physical unit) will be either *0* or *1* depending on which of the SCSI address of the disk. The *Lun* (logical unit) will always be *0* in our case since we're not assigning any *Logical Units* on the disks. Now, lets look at the *blk2* entry.

```
blk2 : Acpi(HWP0002,100)/Pci(1|0)/Scsi(Pun0,Lun0)/
HD(Part1,Sig8E89981A-0B97-11D7-9C4C-AF87605217DA)
```

The *blk2* entry is a partition on the *blk1* device. All of the information is the same for the two entries except for the additional partition-related information beginning with *Part1*. *Part1* indicates this is the first partition on physical device *blk1* with an EFI signature beginning with *Sig*. *blk3* and *blk4* are additional partitions that we created when we loaded Advanced Server on this disk and created three partitions. The first group ends with the *fs0* entry.

```
fs0 : Acpi(HWP0002,100)/Pci(1|0)/Scsi(Pun0,Lun0)/
HD(Part1,Sig8E89981A-0B97-11D7-9C4C-AF87605217DA)
```

Notice that the *fs0* line also matches *blk1* and *blk2* from a path perspective. This is a file system readable by EFI hence it begins with *fs*.

To summarize, what we see is a physical device *blk1* that has on it an EFI partition *blk2* and a file system *fs0*. All three of these are listed as separate entries in EFI. In addition, *blk3* and *blk4* are partitions on the same physical device *blk1*

The same applies to physical unit 1 which is *Pun1*. This is, *blk5*. It too has two partitions at *blk6* and *blk7*. *fs1* is the file system that is on disk. This is the HP-UX disk in our configuration.

```
blk0 : Acpi(HWP0002,0)/Pci(2|0)/Ata(Primary,Master)
```

The *blk0* device in the list is the DVD-ROM. The *Ata* (Advanced Technology Attachment) is the official name that *American National Standards Institute* group X3T10 uses for what the computer industry calls *Integrated Drive Electronics* (IDE) -- in this case a DVD-ROM.

The following table summarizes the fields that we analyzed for all of the entries. Keep in mind that some of this information, such as the *Acpi*

data, is most often necessary to help analyze the system in the event of a problem.

Table1-1 Description of EFI Device Mappings Field

Item	Description
<i>blkn</i>	Physical drive or a partition on a physical drive. A physical drive can be a hard disk drive or a removable media drive. A <i>Partn</i> in parenthesis is shown in our examples when there is a partition on a disk drive.
<i>Acpi(Device,X)</i>	The device type is the first entry in the parenthesis. The second entry, <i>X</i> , is the PCI host number. <i>Acpi</i> is Advance Configuration and Power Interface.
<i>Pci(D/F)</i>	This indicates PCI-related information. <i>D</i> is the PCI device/slot number and <i>F</i> is the PCI function number.
<i>Scsi(Pun.Lun)</i>	The physical characteristic of the disk which is SCSI. The <i>Pun</i> is the SCSI number and the <i>Lun</i> is the LUN number on the physical device.
<i>HD(Part,Sig)</i>	The partition <i>Part</i> and EFI Signature <i>Sig</i> on the partition.

Keep in mind that the file system numbers may change when you remap devices or when components are added or removed such as a DVD device.

There are many more EFI commands that you may want to use in addition to **map**. The following table summarizes many of the most used EFI commands. We'll take a look at some of them in the next section.

Table1-2 Commonly Used EFI Commands

EFI Command ^a	Description
Commands found in all menus	
map	Produce a device map list. (map -r rescans the bus)
info boot	Boot from specified Path
help <command>	Display help for specified command or menu (help bch is handy)

EFI Command^a	Description
reset	Reset the system (to allow reconfiguration of complex)
exit (at the EFI shell)	Return to the Main Menu
Commands found on the MAin menu	
EFI Boot Manager “Change Boot Order”	Display or modify a path (In “Boot option maintenance menu” and then “Change Boot Order” select the desired boot path and use “u” or “d” to move up or down.
bcfg	Search for boot devices
bcfg	search for Boot devices
more, plus many commands offer a [-b] parameter to cause 25 line breaks	Display or change scrolling capability
Commands found on the COnfiguration menu	
autoboot	display or start the auto start flag
info boot	Display or set processor Boot Identifier
EFI Boot Manager	Display boot-related information
autoboot	Seconds allowed for boot attempt
cpuconfig	Config/Deconfig Processor
autoboot	Display or set boot tests execution
time	Read or set the real time clock
Commands found on the Information menu	
info all	Display all system information
info boot	Display boot-related information
info cache	Display cache information
info chiprev	Display revision number of major VLSI
MP command <df>	Display FRU information
info fw	Display firmware version for PDC, ICM, and Complex
info io	Display firmware version for PDC, ICM, and Complex

EFI Command ^a	Description
info mem	Display memory information
info cpu	Display processor information
SERvice	
errdump clear	Clear (zero) the contents of PIM
mm	Read memory locations scope of page Deallocations
PDT	Display or clear the Page Deallocation (Processor Internal Memory)
errdump mca	Display PIM information (Processor Internal Memory)
more [-b]	Display or change scrolling capability

^a An interesting aspect of the EFI commands is that you can accomplish many of the same functions running BCH commands on PA-RISC systems. To those of you who have PA-RISC system experience you'll see many of the same functions performed in EFI.

Using EFI, you can control the boot-related setup on your Itanium-based system. Because of the number of operating systems you can run on Itanium-based systems, you'll use this interface often to coordinate and manage them.

EFI Command Examples

As I mentioned earlier, traversing the EFI menu structure and issuing commands is straight forward. When the system boots you are given the option to interrupt the autoboot (If you don't interrupt it, the autoboot will load the first operating system listed which, in our case, is *Red Hat Advanced Server*). At system startup, the EFI Boot Manager presents the boot option menu (See below). Here you have 5 seconds to enter a selection before *Red Hat Linux Advanced Server* is started

```
EFI Boot Manager ver 1.10 [ 14.60] Firmware ver 1.61 [ 4241]
Please select a boot option

Red Hat Linux Advanced Server
HP-UX Primary Boot: 0/1/1/0.1.0
EFI Shell [ Built-in]
```

```

Boot option maintenance menu
Security/Password Menu

```

```

Use ^ and v to change option(s). Use Enter to select an option
Default boot selection will be booted in 5 seconds

```

You can use the arrow, or the *u* and *d* keys to move up and down respectively. We used the ↓ key (down arrow) to select *EFI Shell [Built-in]*. This brought us to the *Shell>* prompt. From there, we can issue EFI commands. Similarly, once your at the *Shell>* prompt, help is always available. To get a listing of the classes of commands available in *Shell>*, simply enter **help** and press enter (see below).

```

Shell> help
List of classes of commands:

boot          -- Booting options and disk-related commands
configuration -- Changing and retrieving system information
device        -- Getting device, driver and handle information
memory        -- Memory related commands
shell         -- Basic shell navigation and customization
scripts       -- EFI shell-script commands

Use 'help <class>' for a list of commands in that class
Use 'help <command>' for full documentation of a command
Use 'help -a' to display list of all commands

```

You can also issue *help* requests for any EFI commands at any level. For example, if you want to know more about your current cpu configuration, you would start with **help configuration** to determine the help command for cpu configuration, and then **help cpuconfig** as shown below.

```

Shell> help configuration
Configuration commands:

cpuconfig    -- Deconfigure or reconfigure cpus
date         -- Displays the current date or sets the date in the system
err          -- Displays or changes the error level
esiproc      -- Make an ESI call
errdump      -- View/Clear logs
info         -- Display hardware information
monarch      -- View or set the monarch processor
palproc      -- Make a PAL call.
salproc      -- Make a SAL call
time         -- Displays the current time or sets the time of the system
ver          -- Displays the version information

Use 'help <command>' for full documentation of a command
Use 'help -a' to display list of all commands

Shell> help cpuconfig
cpu          Specifies which cpu to configure

CPUCONFIG [ cpu] [ on|off]

on|off Specifies to configure or deconfigure a cpu

Note:

```

1. Cpu status will not change until next boot.
2. Specifying a cpu number without a state will display configuration status.

Examples:

```
* To deconfigure CPU 0
fs0:\> cpuconfig 0 off
Cpu will be deconfigured on the next boot.
```

```
* To display configuration status of cpus
fs0:\> cpuconfig
PROCESSOR INFORMATION
```

CPU	Speed	Proc Rev	Model	Family	Arch Rev	Processor State
0	560 MHz	B1	0	31	0	Sched Deconf
1	560 MHz	B1	0	31	0	Active

Shell>

As a result of having issued this *help cpuconfig* we now know how to manipulate the CPUs in our system. The following shows what happens we we issue the *cpuconfig* command with a few options.

```
Shell> cpuconfig
```

```
PROCESSOR INFORMATION
```

CPU	Speed	Proc Rev	Model	Family	Arch Rev	Processor State
0	1000 MHz	B3	0	31	0	Active
1	1000 MHz	B3	0	31	0	Active

```
Shell> cpuconfig 1 off
```

```
CPU will be deconfigured on next boot.
```

```
Shell> cpuconfig
```

```
PROCESSOR INFORMATION
```

CPU	Speed	Proc Rev	Model	Family	Arch Rev	Processor State
0	1000 MHz	B3	0	31	0	Active
1	1000 MHz	B3	0	31	0	Sched Deconf

```
Shell> cpuconfig 1 on
```

```
CPU will be configured on next boot.
```

```
Shell> cpuconfig
```

```
PROCESSOR INFORMATION
```

CPU	Speed	Proc Rev	Model	Family	Arch Rev	Processor State
0	1000 MHz	B3	0	31	0	Active
1	1000 MHz	B3	0	31	0	Active

```
Shell>
```

We used **cpuconfig** to view the current CPU configuration showing that both processors are *Active*. Then we turned *off* processor 1 (**cpuconfig 1 off**). We then viewed the CPU configuration again to confirm that processor 1 had been turned off as indicated by the *Sched Deconf* (**cpuconfig**). After that, we turned processor 1 *on* again (**cpuconfig 1 on**). Finally we confirmed that both processors are again *Active* (**cpuconfig**).

As you can see, a lot of useful configuration information about your system is available using EFI. In addition to **cpuconfig**, you can also use **info** to get important system information. The following listing first shows the results of the **info** command. **info** with no argument lists all the differing information options available (such as **all**, **boot**, **cache**, and so on). After we see all the **info** options, we use **info all** to get a complete rundown on our system.

```
Shell> info
Usage: INFO [-b] [target]
target : all, boot, cache, chiprev, cpu,
        fw, io, mem, sys, warning

Shell>
Shell> info all
SYSTEM INFORMATION
    Product Name: server rx2600
    Serial Number: US24758356
    UUID: B831CE57-19C2-11D7-A034-3483D23C4340

PROCESSOR INFORMATION
```

CPU	Speed	Proc Rev	Model	Family	Arch Rev	Processor State
0	1000 MHz	B3	0	31	0	Active
1	1000 MHz	B3	0	31	0	Active

```

CACHE INFORMATON
```

CPU	Instruction	Data	Unified	
	L1	L1	L2	L3
0	16 KB	16 KB	256 KB	3072 KB
1	16 KB	16 KB	256 KB	3072 KB

```

MEMORY INFORMATION
```

	---- DIMM A ----		---- DIMM B ----	
	DIMM	Current	DIMM	Current
0	256MB	Active	256MB	Active
1	256MB	Active	256MB	Active
2	256MB	Active	256MB	Active
3	256MB	Active	256MB	Active
4	----		----	
5	----		----	

```

Active Memory : 2048 MB
```

Installed Memory : 2048 MB

I/O INFORMATION

BOOTABLE DEVICES

Order	Media Type	Path
1	HARDDRIVE	Acpi (HWP0002,100) / Pci (1 0) / Scsi (Pun0, Lun0) / HD (Part1, Sig8E89981A-0B97-11D7-9C4C-AF87605217DA)
2	HARDDRIVE	Acpi (HWP0002,100) / Pci (1 0) / Scsi (Pun1, Lun0) / HD (Part1, Sig7C0F0000)

Seg #	Bus #	Dev #	Fnc #	Vendor ID	Device ID	Slot #	Path
00	00	01	00	0x1033	0x0035	XX	Acpi (HWP0002,0) / Pci (1 0)
00	00	01	01	0x1033	0x0035	XX	Acpi (HWP0002,0) / Pci (1 1)
00	00	01	02	0x1033	0x00E0	XX	Acpi (HWP0002,0) / Pci (1 2)
00	00	02	00	0x1095	0x0649	XX	Acpi (HWP0002,0) / Pci (2 0)
00	00	03	00	0x8086	0x1229	XX	Acpi (HWP0002,0) / Pci (3 0)
00	20	01	00	0x1000	0x0030	XX	Acpi (HWP0002,100) / Pci (1 0)
00	20	01	01	0x1000	0x0030	XX	Acpi (HWP0002,100) / Pci (1 1)
00	20	02	00	0x14E4	0x1645	XX	Acpi (HWP0002,100) / Pci (2 0)
00	40	01	00	0x1011	0x0019	03	Acpi (HWP0002,200) / Pci (1 0)
00	80	01	00	0x1011	0x0019	01	Acpi (HWP0002,400) / Pci (1 0)
00	E0	01	00	0x103C	0x1290	XX	Acpi (HWP0002,700) / Pci (1 0)
00	E0	01	01	0x103C	0x1048	XX	Acpi (HWP0002,700) / Pci (1 1)
00	E0	02	00	0x1002	0x5159	XX	Acpi (HWP0002,700) / Pci (2 0)

BOOT INFORMATION

Monarch CPU:

Current Monarch	Preferred Monarch	Possible Warnings
0	0	

AutoBoot: ON - Timeout is : 10 sec

Boottest:

BOOTTEST Settings Default Variable

OS is not speedy boot aware.

Selftest	Setting
early_cpu	Skip this test
late_cpu	Skip this test
platform	Skip this test
chipset	Skip this test
io_hw	Skip this test
mem_init	Skip this test
mem_test	Skip this test

LAN Address Information:

LAN Address	Path
Mac (00306E39B724)	Acpi (HWP0002,0) / Pci (3 0) / Mac (00306E39B724)
*Mac (00306E3927B0)	Acpi (HWP0002,100) / Pci (2 0) / Mac (00306E3927B0)

FIRMWARE INFORMATION

Firmware Revision: 1.61 [4241]

```

PAL A Revision: 7.31
PAL B Revision: 7.36

SAL Spec Revision: 0.20
SAL A Revision: 2.00
SAL B Revision: 1.60

EFI Spec Revision: 1.10
EFI Intel Drop Revision: 14.60
EFI Build Revision: 1.22

POSSE Revision: 0.10

ACPI Revision: 7.00

BMC Revision 1.30
IPMI Revision: 1.00
SMBIOS Revision: 2.3.2a
Management Processor Revision: E.02.07

WARNING AND STOP BOOT INFORMATION

```

CHIP REVISION INFORMATION

Chip Type	Logical ID	Device ID	Chip Revision
Memory Controller	0	122b	0022
Root Bridge	0	1229	0022
Host Bridge	0000	122e	0032
Host Bridge	0001	122e	0032
Host Bridge	0002	122e	0032
Host Bridge	0003	122e	0032
Host Bridge	0004	122e	0032
Host Bridge	0006	122e	0032
Host Bridge	0007	122e	0032
Other Bridge	0	0	0002
Other Bridge	0	0	0007
Baseboard MC	0	0	0130

```
Shell>
```

As you can see **info all** produces a great overview of our system configuration. Note that two bootable devices are listed in the order in which they appear in the main EFI screen (See “EFI Boot Administration Main Screen” on page 6).

In addition to selecting the *EFI Shell*, which we have been doing in our examples to this point, we also have other options. Selecting *Boot option maintenance menu* produces the selections shown below.

```

EFI Boot Maintenance Manager ver 1.10 [ 14.60]
Manage BootNext setting.  Select an Operation

Red Hat Linux Advanced Server
HP-UX Primary Boot: 0/1/1/0.1.0
EFI Shell [ Built-in]
Reset BootNext Setting
Save Settings to NVRAM
Help
Exit

```

We have some of the same selections that we had at the main menu level, including our two installed operating systems and the EFI shell, but we also have some new selections. If we were to select *Reset BootNext Setting* the following selections would be produced.

```
EFI Boot Maintenance Manager ver 1.10 [ 14.60]
```

```
Main Menu. Select an Operation
```

```

Boot from a File
Add a Boot Option
Delete Boot Option(s)
Change Boot Order

Manage BootNext setting
Set Auto Boot TimeOut

Select Active Console Output Devices
Select Active Console Input Devices
Select Active Standard Error Devices

Cold Reset
Exit
```

At this point you could perform a variety of functions. On our system, for instance, we could use *Change Boot Order* to make the HP-UX partition the default boot selection instead of the Linux Advanced Server.

Although we are going to perform a console-only installation of Red Hat Advanced Server in Chapter 2, the rx2600 we're working on does indeed have a built-in VGA port and graphics display attached. If we wanted to enable the graphics display we would select *Select Active Console Output Devices* from the menu above.

The *Select the Console Output Device(s)* menu appears listing all possible console devices. The first device with an * is the serial port that was selected by default. To enable the graphics as well, you must select the last console device as well. We know that this is the graphics device port because it does not contain *Uart* as part of the selection. Note that *Uart* devices are always serial devices. Also note that in the below example the graphical device port is already selected (indicated by an * in front of it). We then select *Save Settings to NVRAM* which saves the console device settings, and then *Exit*.

```
EFI Boot Maintenance Manager ver 1.10 [ 14.60]
```

```
Select the Console Output Device(s)
```

```

Acpi (PNP0501,0)/Uart (9600 N81)/VenMsg (PcAnsi)
Acpi (PNP0501,0)/Uart (9600 N81)/VenMsg (Vt100)
Acpi (PNP0501,0)/Uart (9600 N81)/VenMsg (Vt100+)
Acpi (PNP0501,0)/Uart (9600 N81)/VenMsg (VtUtf8)
Acpi (HWP0002,700)/Pci (1|1)/Uart (9600 N81)/VenMsg (PcAnsi)
Acpi (HWP0002,700)/Pci (1|1)/Uart (9600 N81)/VenMsg (Vt100)
```

```
* Acpi (HWP0002,700)/Pci (1|1)/Uart (9600 N81)/VenMsg (Vt100+)  
  Acpi (HWP0002,700)/Pci (1|1)/Uart (9600 N81)/VenMsg (VtUtf8)  
* Acpi (HWP0002,700)/Pci (2|0)  
Save Settings to NVRAM  
Exit
```

This setting enables both the serial and graphics consoles. The early boot messages will go to the serial console. After the graphics server is started the Advanced Server-related boot messages will display graphically. After installation is complete, this would be true during the load of Advanced Server as well. The early selections would take place on the console and then the majority of the load selections would display graphically.

As we have seen EFI is a very useful and relatively easy to use tool. Of course, no one expects you to remember all you have seen here. If you need help, refer to “Commonly Used EFI Commands” on page 10. You can also ask for **help** at the *Shell*> prompt, or take a look at the tear out card that summarizes EFI. Between them all, you’ll be able to perform many useful functions using the EFI interface.

Configuring the Management Processor (MP)

There is a built-in processor on most HP systems, including Itanium-based systems, that can be used for either local or remote system administration functions called the Management Processor (MP). Occasionally, you may hear it referred to as the Service Processor (SP) or Guardian Service Processor (GSP) but MP is by far the most commonly used name. MP functionality gets enhanced on a regular basis so what you’ll see in the upcoming examples may not be the same as the MP functionality you see on your system. Still, it should be very close. In what follows, I’ll provide a quick overview of MP including listing the overview supplied with the tool. Note that initial MP configuration is important because the first person to gain access to MP before it has been configured is a MP administrator by default.

You gain access to the MP with **^b** (control b). If MP has not been configured, anyone who gets access to the console can type **^b** and gain access to the system. Because of this, I recommend you perform MP configuration as soon as possible after installing your system.

After you issue **^b**, you will see the *MP*> prompt. The following listing shows the output from the **he ov** (help overview) command.

```
-----
Service Processor login:
Service Processor password:
```

Hewlett-Packard Management Processor

(c) Copyright Hewlett-Packard Company 1999-2002. All Rights Reserved.

System Name: uninitialized

```
*****
MP ACCESS IS NOT SECURE
No MP users are currently configured and remote access is enabled.
Set up a user with a password (see SO command)
OR
Disable all types of remote access (see EL and ER commands)
*****
```

MP Host Name: uninitialized

MP> **he**

Hardware Revision a1 Firmware Revision E.02.07 Aug 8 2002,23:00:38

MP Help System

Enter a command at the help prompt:

```
Overview : Launch the help overview
List      : Show the list of MP commands
<COMMAND> : Enter the command name for help on individual command
TOPics    : Show all MP Help topics and commands
HElp      : Display this screen
Q         : Quit help
```

====

MP HELP: **ov**

ov

==== MP Help Overview =====

The Management Processor (MP) is an independent support processor for the system console. MP provides services that facilitate the management of the host system. Its major features are:

- * Always-on capability: The MP is alive as long as the power cord is plugged in.
- * User/password access control: Supports operator and administrator users
- * Multiple access methods:
 - Local Port - use terminal or laptop computer for direct connect
 - Remote/modem Port - use dedicated modem RS-232 port and external modem
 - LAN - use telnet or web to access MP LAN
- * Mirrored console: the system console output stream is reflected to all of the connected console users, and any user can provide input.
- * Display and/or logging of:
 - The system console, System event logs (chassis codes), Virtual Front Panel (VFP), and system power and configuration status
- * An independent, non-mirrored session:
 - Available from local and modem ports for MP connection (CSP) or OS

```

login (SE).
* Power control, system reset, and TOC capabilities.

MORE Help (Q to go back to main, <CR> for more):

==== MP Help =====
                                HELP TOPICS

The following topics can be entered for general information:

* ADMINISTRator          LIst of commands          PASSword resetting
* CHASSis codes          * MODEM                      * PORT summary
* COMMAND summary        * MODES of the MP           * SESSion
* CONSOLE                 * OVERview                   * TOPics
* HPterm and VT100

(* topics which are included in the OVERview)

The following commands can be entered for help on the command:

AC AR CA CL CO CSP DC DF DI EL ER EX HE IT LC LS MR
MS PC PG PS RS SDM SE SL SO SR SS TC TE VFP WHO XD XU

MORE Help (Q to go back to main, <CR> for more):

```

```

==== MP Help Overview =====
                                MP MODES

THE MIRRORED MP SESSION:
When first logging on, the user is part of the mirrored MP session. The
mirrored MP session has three modes:
  1) MP Command Mode  2) Console Mode  3) VFP or Alert Mode

  1. You are in the MP Command Mode right now. In this mode all the MP
     commands can be executed. Typically this mode is entered by typing
     CTRL-B from console mode.
  2. In Console Mode the user gets access to the Unix console. This mode is
     entered from the MP Command Mode using the CO command.
  3. In the VFP or Alert Mode the MP displays a representation of the front
     pannel leds, called the Virtual Front Panel (VFP). This mode is entered
     from MP Command Mode using the VFP command.
See the help on those commands for more information.

LEAVING MIRRORED MP SESSION:
From the local or remote/modem ports, it is also possible to leave the
mirrored MP session and connect to either the OS (see the "SE" command)
or to another MP on the network (see the "CSP" command).

MORE Help (Q to go back to main, <CR> for more):

```

```

==== MP Help Overview =====
                                MP COMMAND HELP

The MP commands can be grouped into the following categories:
* STATUS COMMANDS- Provide status on the server and the MP.
  CL, DF, LS, MS, PS, SL, SR, SS, VFP

* SERVER CONTROL- Alter the state of the server
  MR, PC, RB, RS, TC

* GENERAL MP CONFIGURATION
  AC, BP, CG, DC, IT, LOC, PR, SDM, SO, XD, XU

* MP PORT CONFIGURATION- Configure LAN/WEB, remote/modem, and local ports
  CA, EL, ER, LC, PG

* CONNECTIONS- Examine and make connections or change mode
  CO, CSP, DI, EX, SE, TE, VFP, WHO

```

MORE Help (Q to go back to main, <CR> for more):

```
==== MP Help Overview =====
                        OPERATOR / ADMINISTRATOR HELP
```

Administrators have more capabilities than Operators:

Administrator-Only Commands:

AR (Automatic system Restart), DC (Default Configuration), IT (Inactivity Timeout), LC (LAN Configuration), PG (PaGing), SO (Security Options)

The MP Command interface permission/access level:

The MP Command interface permission/access level is set by the first user that types CTRL-B to initiate it. If that user is an operator-user, then the command interface runs at operator access level. Even if an administrator-user logs in later or the operator-user disconnects, the access level stays the same. The access level can be changed by leaving MP command mode and typing CTRL-B again to return.

MORE Help (Q to go back to main, <CR> for more):

```
==== MP Help Overview =====
                        CONSOLE MODE HELP
```

Typing "CO" from the MP Command interface provides a mirrored version of the OS console. All mirrored users see the same output. At any time, only one of the mirrored users has write access to the console. To get write access to the console, type CTRL-e c f (not CTRL-e CTRL-c CTRL-f).

SEE ALSO: CO (Console)

MORE Help (Q to go back to main, <CR> for more):

```
==== MP Help Overview =====
                        SESSION MODE HELP
```

Typing "SE" from the MP Command interface provides a non-mirrored, normal OS login. The session is not the console; it is a separate login to the OS, and the messages that the OS sends to the console will not be seen.

RECOMMENDATION:

Use the session for ASCII screen-oriented applications (SAM) or file transfer programs (ftp) from the local or modem ports. To use these applications from the LAN, telnet directly to the system for a private login.

SEE ALSO: SE (SEssion)

MORE Help (Q to go back to main, <CR> for more):

```
==== MP Help Overview =====
                        CHASSIS CODES
```

Chassis codes are encoded data that provide system information to the user. Some well-known names for similar data would be Event Logs or Post Codes. Chassis codes are produced by intelligent hardware modules, the O/S, and system firmware. Use VFP to view the live chassis codes. Use SL to view the chassis code log.

The following severity (or alert) levels are defined:

0) Minor Forward Progress 4) Reserved

- | | |
|---------------------------|-------------|
| 1) Major Forward Progress | 5) Critical |
| 2) Informational | 6) Reserved |
| 3) Warning | 7) Fatal |

SEE ALSO: SL, VFP (Show Logs, Virtual Front Panel)

MORE Help (Q to go back to main, <CR> for more):

==== MP Help Overview =====

PORTS HELP

The local, remote/modem, and LAN ports are actual connections on the back of the server. All ports can be used even when the server is non-functional.

- * Local port : Provides serial port terminal access. To configure, see the CA (Configure Asynchronous) command.
- * Remote/Modem port: Provides external modem access. Related commands are ER, MS, and MR (Enable Remote/modem, Modem Status, Modem Reset)
- * LAN port : Provides telnet and web access into the MP. Related commands are EL, LC, and LS (Enable LAN, LAN Configuration, and LAN Status).

MORE Help (Q to go back to main, <CR> for more):

==== MP Help Overview =====

REMOTE/MODEM PORT HELP

The remote/modem port can be used for dial-in access to the MP or the OS, and can be configured so that the OS or the MP can activate a pager.

The remote/modem port is configured via the CA command. The ER command enables/disables the port and gives the modem control over to the MP or the OS. See the help on those commands for more details.

SEE ALSO: CA, ER, DI, MR, MS, PG (Configure Asynchronous, Enable Remote, Disconnect remote or LAN, Modem Reset, Modem Status, PaGing)

MORE Help (Q to go back to main, <CR> for more):

==== MP Help Overview =====

INTERNAL PORT HELP

The Internal Port has been obsoleted.

MORE Help (Q to go back to main, <CR> for more):

==== MP Help Overview =====

HPTERM & VT100 HELP

RECOMMENDATION: Do not mix HP and vt100 terminal types at the same time.

The MP mirrors the system console to the MP local, remote/modem, and LAN ports. One console output stream is reflected to all of the connected console users. If several different terminal types are used simultaneously by the users, some users may see strange results.

HP-UX example:

Applications which care about the terminal type (install, SAM, vi, etc.) running on HP-UX use three methods to determine the terminal type:

- 1) The application takes the terminal information from the OS. This value is set in the CA command and takes effect for all MP ports.
- 2) The \$TERM shell environment variable
- 3) The application directly queries the terminal (in this case, the write enabled terminal will establish the terminal type.)

```

Make sure that settings #1 and #2 agree with your terminal type.

====
(HE for main help, enter command name, or Q to quit)
MP HELP:

==== MP Help =====(Administrator)===
      Hardware Revision a1 Firmware Revision E.02.07 Aug  8 2002,23:00:38

                MP Help System

Enter a command at the help prompt:
  OVerview  : Launch the help overview
  LIst      : Show the list of MP commands
  <COMMAND> : Enter the command name for help on individual command
  TOPics    : Show all MP Help topics and commands
  HELp      : Display this screen
  Q         : Quit help

====
MP HELP:

```

It is very important to note that neither a login name or password were required to log in after the **^b** was issued. This is because at startup no MP users have yet been configured. At this point, any user can get access to this menu by typing **^b** at the system console.

The MP overview provides a wealth of information. Although this was a long output, I included it because it describes many MP commands and functions. As I mentioned, at startup any user can get access to the MP menu. Because you don't want unauthorized users to have access to many of these commands, the first step we'll take is to setup security options using the Security Options command, **SO**. **SO** begins by summarizing its parameters after which it asks you a series of security related questions. I have listed them below. In the example, I answered *y* when asked if I wanted to modify the parameters so that we could see each one individually. I answered *n* when asked if I wanted to modify each individual parameter. Your answers should reflect your own security preferences.

```
MP> SO
```

```
This command allow you to modify the security options and access control.
```

```

MP wide parameters are:
. Login Timeout: 1 minutes.
. Number of Password Faults allowed: 3
. Flow Control Timeout: 5 minutes.
. SSL for Web Access: Disabled
. Contact Information:
  ->contact person :
  ->contact phone number :
  ->contact email address :
  ->contact pager number :
. Location Information:
  ->server location :
  ->server rack ID :

```

```
->server position :
Do you want to modify the MP wide parameters? (Y/[ N ]) y

Current Login Timeout: 1 minutes.
Do you want to modify it? (Y/[ N ]) n

Current Number of Password Faults allowed: 3
Do you want to modify it? (Y/[ N ]) n

Current Flow Control Timeout: 5 minutes.
Do you want to modify it? (Y/[ N ]) n

Current SSL for Web Access: Disabled
Do you want to modify it? (Y/[ N ]) n

Do you want to modify the contact information? (Y/[ N ]) n

Do you want to modify the location information? (Y/[ N ]) n

User number 1 parameters are:
. User's Name:
. User's Login:
. Organization's Name:
. Dial-back configuration: Disabled
. Access Level: Operator
. Mode: Single
. User's state: Disabled

Do you want to modify the user number 1 parameters? (Y/[ N ]/Q to quit) q

MP Host Name: uninitialized
MP>
```

You can see that there are many *Security Options*. Because you have unlimited access to the system through the console you will want to review these to ensure that they meet your requirements. You can add users and perform other security-related tasks.

In addition to configuring users and security, there are many useful features of MP. I encourage you to experiment with it. One command I often use is **cl**. I use **cl** to display a console history as shown in the following listing for the rx2600 system used throughout the examples in this chapter:

```
MP> cl
EFI Boot Maintenance Manager ver 1.10 [ 14.60]
Select the Console Output Device(s)

Acpi (PNP0501,0)/Uart (9600 N81)/VenMsg (PcAnsi)
Acpi (PNP0501,0)/Uart (9600 N81)/VenMsg (Vt100)
Acpi (PNP0501,0)/Uart (9600 N81)/VenMsg (Vt100+)
Acpi (PNP0501,0)/Uart (9600 N81)/VenMsg (VtUtf8)
Acpi (HWP0002,700)/Pci (1|1)/Uart (9600 N81)/VenMsg (PcAnsi)
```

```

    Acpi (HWP0002,700)/Pci (1|1)/Uart (9600 N81)/VenMsg (Vt100)
*   Acpi (HWP0002,700)/Pci (1|1)/Uart (9600 N81)/VenMsg (Vt100+)
    Acpi (HWP0002,700)/Pci (1|1)/Uart (9600 N81)/VenMsg (VtUtf8)
*   Acpi (HWP0002,700)/Pci (2|0)
    Save Settings to NVRAM
    Exit

```

```
EFI Boot Maintenance Manager ver 1.10 [14.60]
```

```
Select the Console Output Device(s)
```

```
Press Q/q to quit, Enter to continue:
```

This listing shows the last console task we completed, which was to enable the graphics console (which is the second entry with an *** in front of it) which we did at the end of the last section.

You can also use MP to obtain the status of power modules with **ps** as shown in the following listing:

```

MP> ps

PS
System Power state: On          Power Switch      : On
Temperature           : Low OverTemp Selected fan state: Normal

#   State   Power supplies      | Fan
      Type                                | States
-----|-----
0 Normal    Type 0              | Normal
1 Not Installed -                | Normal
2 -        -                | Normal
3 -        -                | Normal
4 -        -                | Normal
5 -        -                | Normal
6 -        -                | Normal
7 -        -                | Normal

MP Host Name: uninitialized
MP>

```

This shows that we have one power supply and that our many fans are operating *Normal*. The second power supply is not installed.

You can view processor status with **ss** as shown in the following listing:

```

MP> ss          (This command will not run on Itanium-based
                Superdome in the future)

SS

```

```

System Processor Status:

  Monarch Processor: 0

  Processor 0 is : Installed and Configured
  Processor 1 is : Installed and Configured

MP Host Name:  uninitialized
MP>

```

This listing shows that there are two processors installed and the *Monarch Processor* is 0. This is the main controlling processor from the perspective of the operating system. This processor is designated as *CPU 0*. The Low Priority Machine Check (LPMC) monitor will not deactivate or replace a failing monarch processor.

Although they weren't shown in any examples you can power on or off the system using the *PC* command, for Power Control.

After you have completed your MP-related work you can return to console mode from the MP prompt simply by issuing **co**, as shown in the following listing:

```
MP> co
```

Anytime that you are on the console, you can issue **^b** and get access to MP and then get back to console mode using **co**.

Table 1-4 contains a list of MP commands available on Itanium-based systems at the time of this writing. Keep in mind the MP commands for your Itanium system may be somewhat different from those listed.

TABLE 1-4 Management Processor Commands

Command (Function)	Explanation
AC	Alert Display Configuration
CA	Configure Asynchronous and Modem parameters.
CL	View console log.
CO	Return to console mode.
CSP	Connect to remote management processor.
DC	Default configuration.

Command (Function)	Explanation
DI	Disconnect remote or LAN console.
EL	Enable or Disable LAN or Web access.
ER	Enable remote or LAN console access.
EX	Exit MP and disconnect.
HE	Display help for menu or command.
IT	Set inactivity timeout settings.
LC	Configure LAN connected and PPP console.
LOC	Locator LED control.
LS	LAN status.
MR	Modem Reset.
MS	Display the status of the modem.
PC	Remote power control.
PG	Configure Paging.
PR	Power restore policy configuration.
PS	Display the status of the power.
RB	Reset BMC through the toggle CPIO pin.
RS	Request BMC to reset system through RST signal.
SE	Activate a system session on local or remote port.
SL	Display status logs.
SO	Configure security options and access control.
SS	Display the status of the system processors.
TC	Request BMC to reset with a TOC.
TE	Sends a message to other terminals.
VFP	Virtual Front Panel.
WHO	Display a list of connected MP users.
XD	Diagnostics and/or reset of MP.
XU	Upgrade the MP firmware.