

Chapter 21

Itanium Architecture (IA)



***HP-UX Handbook
Revision 13.00***

TERMS OF USE AND LEGAL RESTRICTIONS FOR THE HP-UX RECOVERY HANDBOOK

ATTENTION: PLEASE READ THESE TERMS CAREFULLY BEFORE USING THE HP-UX HANDBOOK. USING THESE MATERIALS INDICATES THAT YOU ACCEPT THESE TERMS. IF YOU DO NOT ACCEPT THESE TERMS, DO NOT USE THE HP-UX HANDBOOK.

THE HP-UX HANDBOOK HAS BEEN COMPILED FROM THE NOTES OF HP ENGINEERS AND CONTAINS HP CONFIDENTIAL INFORMATION.

THE HP-UX HANDBOOK IS NOT A PRODUCT QUALITY DOCUMENT AND IS NOT NECESSARILY MAINTAINED OR UP TO DATE. THE HP-UX HANDBOOK IS HERE MADE AVAILABLE TO HP CONTRACT CUSTOMERS FOR THEIR INTERNAL USE ONLY AND ON THE CONDITION THAT NEITHER THE HP-UX HANDBOOK NOR ANY OF THE MATERIALS IT CONTAINS IS PASSED ON TO ANY THIRD PARTY.

Use of the HP-UX Handbook: Hewlett-Packard Company ("HP") authorizes you to view and download the HP-UX Handbook only for internal use by you, a valued HP Contract Customer, provided that you retain all copyright and other proprietary notices contained in the original materials on any copies of the materials. You may not modify the HP-UX Handbook in any way or publicly display, perform, or distribute or otherwise use them for any public or purpose outside your own business. The materials comprising the HP-UX Handbook are copyrighted and any unauthorized use of these materials may violate copyright, trademark, and other laws. If you breach any of these Terms, your authorization to use the HP-UX Handbook automatically terminates and you must immediately destroy any downloaded or printed materials.

Links To Other Web Sites: Links to third party Web sites provided by the HP-UX Handbook are provided solely as a convenience to you. If you use these links, you will leave this Site. HP has not reviewed all of these third party sites and does not control and is not responsible for any of these sites or their content. Thus, HP does not endorse or make any representations about them, or any information, software or other products or materials found there, or any results that may be obtained from using them. If you decide to access any of the third party sites linked to this Site, you do this entirely at your own risk.

Disclaimer: THE HP-UX HANDBOOK IS PROVIDED "AS IS" WITHOUT ANY WARRANTIES OF ANY KIND INCLUDING WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT OF INTELLECTUAL PROPERTY. HP further does not warrant the accuracy and completeness of the materials in the HP-UX Handbook. HP may make changes to the HP-UX Handbook at any time without notice. The HP-UX Handbook may be out of date, and HP makes no commitment to update the HP-UX Handbook. Information in the HP-UX Handbook may refer to products, programs or services that are not available in your country. Consult your local HP business contact for information regarding the products, programs and services that may be available to you.

Limitation of Liability: IN NO EVENT WILL HP, ITS SUPPLIERS, OR OTHER ANY THIRD PARTIES MENTIONED IN THE HP-UX HANDBOOK BE LIABLE FOR ANY DAMAGES WHATSOEVER (INCLUDING, WITHOUT LIMITATION, THOSE RESULTING FROM LOST PROFITS, LOST DATA OR BUSINESS INTERRUPTION) ARISING OUT OF THE USE, INABILITY TO USE, OR THE RESULTS OF USE OF THE HP-UX HANDBOOK, WHETHER BASED ON WARRANTY, CONTRACT, TORT OR ANY OTHER LEGAL THEORY AND WHETHER OR NOT ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. THIS DOES NOT APPLY IN CASE OF INTENT OR IF LIABILITY IS LEGALLY STIPULATED. IF YOUR USE OF THE HP-UX HANDBOOK RESULTS IN THE NEED FOR SERVICING, REPAIR OR CORRECTION OF EQUIPMENT OR DATA, YOU ASSUME ALL COSTS THEREOF.

The information contained herein is subject to change without notice. The only warranties for HP products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. HP shall not be liable for technical or editorial errors or omissions contained herein.

Applicable Laws: These Terms will be governed by and construed in accordance with the laws of the State of California, without giving effect to any principles of conflicts of laws.

General: HP may revise these Terms at any time by updating this posting. *Revised Oct 2013*

© Copyright 2000-2006, 2013 Hewlett-Packard Development Company, L.P

FEEDBACK or QUESTIONS:
(please use subject syntax:

please email essam.ackleh@hp.com
HP-UX Handbook v13.00 Chapter <YY> - <Feedback Title>

TABLE OF CONETENTS

Introduction	4
Overview of the Itanium Architecture	4
Main features of EPIC	4
The Itanium processor family (IPF)	6
HP-UX Itanium Releases	8
Differences between HP-UX for PA and IPF	8
Impact of EFI interface and boot disk layout	10
How an IPF system boots	10
Boot disk layout and EFI	10
How to configure the Console input/output devices	12
The boot process	15
How to mirror the root disk	22
How to replace a failed disk (LVM)	25
How to configure additional boot options	26
The ARIES emulator	28
Basics	28
Performance	29
IPF vs. PA-RISC Terminology	30
Introducing UX 11.23PI	31
Update paths	31
Installation Requirements	31
Identifying 11.23 Sep 2004 release	32
New features	33
Additional Information	34

Introduction

When PA-RISC was released, HP began designing the architecture to replace it. Several years into the project, HP determined that the economics of microprocessor manufacturing favored partnership and decided to partner with Intel, the leader in volume IC manufacturing. Intel was working on their next generation architecture as well, so the timing was perfect. This joint development combined HP's strengths in system and architecture design with Intel's strengths in processor design and manufacturing.

Overview of the Itanium Architecture

Traditional microprocessor architectures (CISC and RISC) have fundamental attributes that limit performance. To achieve higher performance, processors must not only execute instructions faster, but also execute more instructions per cycle, referred to as “parallel execution”. Greater parallel execution allows more information to be processed concurrently - thereby increasing overall processor performance. In traditional architectures, the processor is often underutilized because of the compiler's limited ability to organize instructions. Branches (instructions that change the flow of execution within the program) and memory latency (the time for data to arrive from memory) compound the already limited ability of today's processors to achieve parallel execution.

To overcome these limitations, a new architecture was required. Traditional architectures communicate parallelism through sequential machine code that “implies” parallelism to the processor. Intel and HP jointly defined a new architecture technology called *EPIC* (*Explicitly Parallel Instruction Computing*) named for the ability of the software to extract maximum parallelism (potential to do work in parallel) in the original code and “explicitly” describe it to the hardware. Intel and HP have jointly defined a new 64-bit *Instruction Set Architecture (ISA)*, based on EPIC technology, which Intel has incorporated into Itanium, Intel's 64 bit microprocessor architecture. The new 64 bit ISA takes an innovative approach combining explicit parallelism with techniques called predication and speculation to progress well beyond the limitations of traditional architectures. The new architecture is called *Itanium™*, formerly known as IA-64.

Main features of EPIC

The main features of EPIC are *explicit parallelism*, *predication* and *speculation*.

Explicit parallelism

In traditional architectures like RISC the processor receives a sequential stream of instructions from the compiler and must reorder the instructions to prevent functional units from being idle. The processor can only reorder a small, fixed number of instructions. A particular

functional unit may be idle even though there are instructions in the instruction stream destined for that functional unit.

The concept behind explicit parallelism is that instructions arrive at the processor explicitly ordered by the compiler. The compiler organizes the code for an entire program and makes the ordering explicit so the processor can execute instructions in the most efficient manner.

Simpler, smaller chip control structures are possible when parallelism is exposed by the compiler instead of the hardware. Space saved on the chip can be used for additional functional units, large numbers of registers, and large caches -further increasing parallelism and overall performance.

Predication

Another major performance limiter for traditional architectures is branching. A branch is a decision between two sets of instructions. Today's architectures use a method called *branch prediction* to predict which set of instructions to load. When branches are miss-predicted the whole path suffers a time delay. While current architectures may only miss-predict 5-10% of the time, the penalties may slow down the processor by as much as 30-40%. Branches also constrain compiler efficiency and underutilize the capabilities of the microprocessor.

The new 64 bit ISA uses a concept called predication. Predication effectively executes both branches, rather than trying to predict the correct branch. When the correct branch is known, unnecessary results are discarded.

Predication can remove many branches from the code and reduce miss-predicts significantly. A study in [ISCA 1995 by Scott Mahlke and others](#), demonstrated that predication removed over 50% of the branches and 40% of the miss-predicted branches from several popular benchmark programs. Thus, predication enables increased performance resulting from greater parallelism and better utilization of an Itanium based processor's performance capabilities.

Speculation

Memory latency (the time to retrieve data from memory) is yet another performance limitation for traditional architectures. Memory latency stalls the processor, leaving it idle until the data arrives from memory. Because memory latency has not kept up with increasing processor speeds, loads (the retrieval of data from memory) need to be initiated earlier to ensure that data arrives when it is needed.

The new 64-bit ISA uses speculation, a method of allowing the compiler to initiate a load from memory earlier, even before it is known to be needed, thus ensuring data is available for use if needed. As a result, the compiler schedules to allow more time for data to arrive without stalling the processor or slowing its performance.

Because the Itanium ISA allows the compiler to expose maximum parallelism in the code and explicitly describe it to the hardware, simpler and smaller chip control structures are possible. Space saved on the chip can then be used for additional resources, such as larger caches and many more registers and functional units. These, in turn, supply the processor with a steady stream of instructions and data to make full use of its capabilities, greatly increasing parallel execution and overall performance.

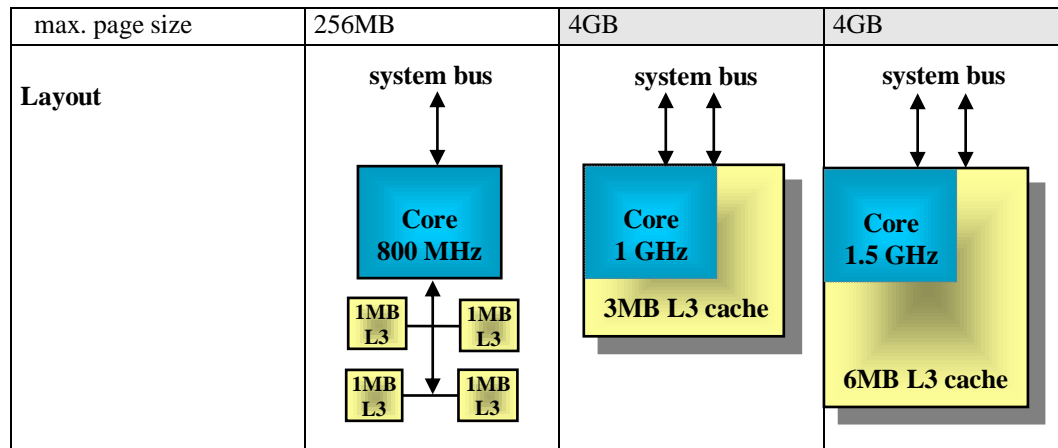
To provide more on-chip resources, Intel's Itanium based processors capitalize on both the strengths of explicit parallelism and the savings in chip space that the 64-bit ISA provides. Itanium based processors have massive resources, with 128 general (integer) registers, 128 floating-point registers, 64 predicate registers, 8 branch registers and 128 control registers. In contrast, today's RISC based processors typically have only 32 general registers and are therefore forced to use register renaming or some other mechanism to create the resources necessary for parallel execution. In Itanium, the functional units attached to the large register file can also be replicated, making Itanium inherently scalable over a wide range of implementations. Of course, since replicated functional units increase the machine width, performance can be increased correspondingly. And with the more sizable caches and the many more read and write ports afforded to memory, the speed of Itanium based processors is no longer limited by the memory latency problems of traditional processors.

A more detailed description of the EPIC features including a code example can be found at <http://www.software.hp.com/products/IA64/arch.html>.

The Itanium processor family (IPF)

Intel announced several generations of the Itanium Processor Family (IPF). The following IPF processors are available as of today.

Feature	1 st generation	2 nd generation	
	Itanium	McKinley	Madison
CPU clock speeds	733/800MHz	900/1000MHz	1.3GHz/1.5GHz
System Bus			
Width	64 bit	128 bit	128 bit
speed/transactions	133MHz/266 MT/s	200MHz/400 MT/s	200MHz/400 MT/s
Bandwidth	2.1 GB/s	6.4 GB/s	6.4 GB/s
Width			
bundles per clock	2	2	2
integer units	4	6	6
loads/stores per clock	2 load or stores	2 loads and 2 stores	2 loads and 2 stores
issue ports	9	11	11
Caches (*)			
level 1 size/latency	2x 16K / 2 cycles	2x 16K / 1 cycles	2x 16K / 1 cycles
level 2 size/latency	96K / 12 cycles	256K / 5-7 cycles	256K / 5-7 cycles
level 3 size/latency	2-4MB off die 20 cycles	1.5-3MB on die 12-15 cycles	3-6MB on die (*) 14-17 cycles
level 3 bandwidth	11.7 GB/s	32 GB/s	48 GB/s
Addressing			
Physical	44 bit	50 bit	50 bit
Virtual	50 bit	64 bit	64 bit



(*): Integrity SuperDome (Orca) has additional 32MB L4 cache. PA8800 has 3-4,5MB L1 only.

The following table shows available HP systems:

gen.	name	processors	clock speeds	chipset	systems
1 st	Itanium	Itanium (fka Merced)	733/800MHz	Intel 82460GX	rx4610 (Ironman) rx9610 (Olympic/Azusa) i2000 (Bigsur)
2 nd	Itanium2	McKinley	900/1000MHz	hp zx1	rx2600/zx6000 (Longs peak) zx2000 (Wilson peak) rx5670 (Everest)
2 nd	Itanium2	Madison	1.5GHz	hp zx1 hp sx1000 (pinnacles)	rx1620 (Nemesis) rx2620 rx4640 (Mt.Diablo) rx7620 (Eiger) rx8620 (Olympia) Integ. SuperDome (Orca) HP Cluster Platform 6000
3 rd	Itanium3	Montecito/ Montvale/ Tukwila	1.42/1.86Ghz	hp zx2 hp sx2000 hp sx3000	Rx2660 Rx3600 Rx6600 Rx7640 Rx8640 Superdome 2

NOTE: 1st generation itanium products were not intended as entry systems and have been discontinued meanwhile.

NOTE: McKinley systems do also support the Madison processor with UX 11.23 (rx2600 PSP)

Specifications of currently available HP Itanium systems can be found at:

http://www.hp.com/products1/itanium/servers_workstations

HP-UX Itanium Releases

These are the different HP-UX releases. The current PA release is UX 11.11. The current IPF release is UX 11.31.

release identifier	official release name	architecture	supported processors
UX 11.00	n/a	PA only	PA7000,PA8000
UX 11.11	11i v1	PA only	PA7000,PA8000
UX 11.20*	11i v1.5	IPF only	Itanium
UX 11.22	11i v1.6	IPF only	Itanium,McKinley
UX 11.23	11i v2	IPF only	McKinley,Madison
UX 11.23PI	11i v2 update 2	IPF & PA	>=PA8700 / >=Madison
UX 11.31	11i v3	IPF & PA	>= Itanium 2

*: UX 11.20 is obsolete. It was intended for prerelease customers only.

Besides HP-UX you can also run Linux, OpenVMS or MS Windows® on HP Itanium systems. Cell based systems (sx1000 chipset) are able to run these operating systems in parallel on different hard partitions. Requirements are:

HP-UX: HP-UX 11i v2 or later
 Linux: Red Hat Advanced Server 2.1 or 3.0, SuSE Enterprise Server 8
 Windows®: Microsoft® Server 2003 Enterprise or Datacenter Edition
 OpenVMS: at least V8.2

Differences between HP-UX for PA and IPF

From the operating systems point of view there should be no noticeable difference between HP-UX for PA and HP-UX for Itanium. The following table highlights the features that are not common between the OSes. Although some applications are not bundled for the Itanium releases they might run in emulation mode through [Aries](#). on the other hand there are some bundled applications, that are not Itanium native but do also run in emulation mode (e.g. SD-UX).

Feature	UX 11.11	UX 11.22	UX 11.23	UX 11.23PI
---------	----------	----------	----------	------------

Operating Environments	5: BaseOE, EOE, MCOE, MTOE, TCOE	3: BaseOE, MTOE, TCOE	5: FOE (* ⁴), EOE, MCOE, MTOE, TCOE	
Compressed dumps	yes (with product CDUMP11i)	no	yes (by default)	yes (by default)
<i>Kernel configuration</i>	GUI/TUI: SAM CLI: kmtune, mk_kernel	GUI: kcweb CLI: kmtune, mk_kernel	GUI: kcweb (autostart) CLI: kctune, kconfig	GUI+TUI: kcweb(autostart) CLI: kctune, kconfig
Dynamic kernel tunables (* ¹)	12	26 (UX 11.11 + 14)	32 (UX 11.22 + 6)	
Automatic kernel tunables (* ²)	no	no	yes	
OS Update	yes, from >=10.20	no	yes, from 11.22(* ⁵)	yes, from 11.11, 11.22
Peripheral devices configuration	SAM	SAM	pdweb	
System Configuration Repository	yes	replaced by System Inventory Manager (SIM)		
<i>Default file system</i>	JFS 3.3, disk layout 4		JFS 3.5, disk layout 5 (* ³)	
<i>Boot file system /stand</i>	hfs	vxfs		
HP-UX Virtual Partitions (vPars)	yes	no		Q2 05
Instant Capacity on Demand (iCOD)	yes	no	yes B.06.00	yes
MxN threads (* ²)	no	yes		
HP-UX Workload Manager	yes	no	yes	
Secure web console	yes	no		
Central web console	yes	no		
Hard partitions (nPars)	yes	no	yes	
Autoport Aggregation (APA)	yes	no	<i>yes</i>	
Metro-/ContinentalClusters	yes	no	yes	
hptc/ClusterPack	no	yes		
Default Browser	Netscape		Mozilla	
Netscape Directory Server	yes	no		
LDAP-UX Integration	yes	no	yes	
HP-UX Bastille	yes	yes	yes, built into install process	
HP-UX Kerberos Server	yes	no	yes	
HP-UX AAA server	yes	no		

(*¹): Refer to the [Kernel chapter](#) for details

(*²): MxN threading can boost the performance of Java applications

- (*³): JFS 3.5 on >= 11.23 has large file system enablement.
max. 32TB (12TB tested & supported). No disk layout version 5 (i.e. large FS enablement) on JFS 3.5 for UX 11.11
- (*⁴): Foundation OE (FOE) is the new name for the BaseOE as of 11.23
- (*⁵): no OS Update from 11.22 MTOE to 11.23. Update supported on servers only.
Update to equivalent OE first, then to higher OE

NOTE (Ignite-UX): EFI is unable to boot from tape. Anyway - there is a way to use a recovery tape on Itanium systems by booting the Install DVD first and then loading the recovery archive from tape. Refer to Ignite-UX newsletter http://wtec.cup.hp.com/~hpux/sat/products/ignite/news/ignite_ux_news_4.htm (HP internal) for details.

Impact of EFI interface and boot disk layout

Usually you should not recognize if you are working at a PA system or an IPF system. HP-UX has the same “look and feel” on both platforms. The commands and the applications behave exactly the same from the user’s perspective. Anyway there are some specials that need to be considered. The new boot disk layout and the existence of the *Extensible Firmware Interface (EFI)* that is located between firmware and operating system impose some changes in typical admin tasks like setting up a root mirror or replacing a failed disk.

How an IPF system boots

The IPF bootstrap process involves the execution of the following software components in that order:

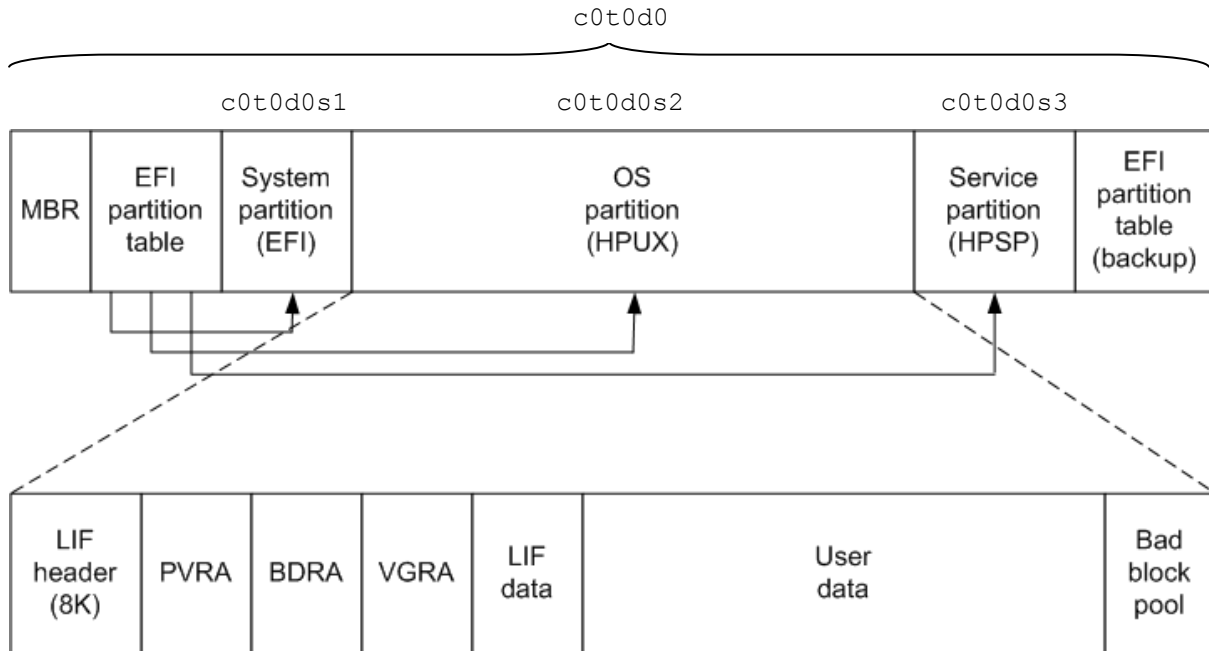
- PAL/SAL initialize hardware (firmware, processor and memory self tests)
- Load EFI Boot Manager (EFI main menu)
- HP-UX Bootloader (hpux.efi)
- HP-UX Kernel (vmunix)

After the processor is reset, firmware initializes and tests processors and platform. It then transfers control to EFI, the Extensible Firmware Interface. EFI, in turn, initializes EFI boot and runtime services and launches the Boot Manager. The Boot Manager, which allows loading of EFI application or drivers from EFI defined file system, loads and transfers control to hpux.efi, the HP-UX-specific bootstrap loader. hpux.efi then loads the HP-UX kernel object file from the HP-UX file system to memory and transfers control to the loaded kernel image.

Boot disk layout and EFI

The EFI (Extensible Firmware Interface) is an interface between HP-UX and the IPF platform firmware (BIOS). The file system supported by the Extensible Firmware Interface is based on the FAT file system. EFI encompasses the use of FAT-32 for a system partition, and FAT-12 or

FAT-16 for removable media. The system partition is required on a bootable disk for the IPF platform. It is used by the IPF BIOS to locate the HP-UX bootloader (hpx). The following picture shows a Itanium boot disk under LVM control:



For a hard disk, the system partition is a contiguous grouping of sectors on the disk, where the starting sector and size are defined by the EFI partition table, which resides on the second logical block of the hard disk, and/or by the Master Boot Record (MBR), which resides on the first sector of the hard disk. MBR is Intel legacy and ignored by EFI firmware.

The EFI System Partition can contain directories, data files, and EFI Images. The EFI system firmware may search the `\EFI` directory of the EFI system partition to find possible EFI images that can be loaded. The HP-UX bootloader is one example of an EFI Image.

From the OS point of view the device file `/dev/dsk/c0t0d0` would represent the whole disk. `c0t0d0s1` would represent the *EFI system partition* (usually 100MB at UX 11.22 and 500MB at UX 11.23) and `c0t0d0s2` would represent the *HP-UX partition*. So the HP-UX partition on an IPF system represents a whole LVM boot disk on a PA system. Hence when dealing with Itanium boot disks you need to specify the s2 device file for e.g. `pvcreate`, `pvdisplay`, etc.

NOTE: Non-bootable disks are treated exactly the same on IPF and PA systems since they are not partitioned and do not contain EFI.

NOTE: As of **UX 11.23** there is a *HP Service Partition* `c0t0d0s3` (FAT-32) right behind the HP-UX partition. The service partition is created by Diagnostics installation. It's default size is 400MB. It can be accessed by the commands `efi_ls(1M)`, `efi_cp(1M)` etc. After cold install it

is empty and must be populated from the Support CD. You can run eDIAG from the HPSP.

The directory structure of a default EFI system partition containing HP-UX boot information is as follows:

<code>/startup.nsh</code>	The startup EFI shell
<code>/EFI/diag</code>	
<code>/EFI/tools</code>	
<code>/EFI/Intel_Firmware/fpswa.efi</code>	
<code>/EFI/HPUX/hpux.efi</code>	the HP-UX bootloader
<code>/EFI/HPUX/nbp.efi</code>	
<code>/EFI/HPUX/AUTO</code>	the AUTO file containing „boot vmunix“
<code>/EFI/hp/tools/network/README.network</code>	
<code>/EFI/hp/tools/network/ftp.efi</code>	
<code>/EFI/hp/tools/network/ifconfig.efi</code>	
<code>/EFI/hp/tools/network/inet.nsh</code>	
<code>/EFI/hp/tools/network/ping.efi</code>	
<code>/EFI/hp/tools/network/route.efi</code>	
<code>/EFI/hp/tools/network/tcpipv4.efi</code>	

These files can also be found below the directory `/usr/lib/efi/` on any Itanium HP-UX file system.

HP-UX contains a set of EFI utilities:

<code>idisk (1M)</code>	Creates partitions for IPF disks.
<code>efi_fsinit (1M)</code>	Initialize an EFI partition; i.e. create an EFI file system.
<code>efi_cp (1M)</code>	Copy files to and from an EFI partition.
<code>efi_mkdir (1M)</code>	Create directories in an EFI partition.
<code>efi_ls (1M)</code>	List the contents of an EFI partition.
<code>efi_rm (1M)</code>	Remove files from an EFI partition.
<code>efi_rmdir (1M)</code>	Remove directories from an EFI partition.

The EFI utilities are the only utilities in HP-UX where the internal structure of an EFI volume is known. To the rest of HP-UX, an EFI system partition is simply a partition containing unspecified data. The EFI volume cannot be mounted to HP-UX currently.

The files in the EFI file system can have up to 255 characters, they are not case sensitive. More detailed information can be found in the `efi(4)` manual page. An EFI command overview can be found at http://hprtdt58.grc.hp.com/documents/systems/longspeak/itanium_handly_trifold.pdf (HP internal)

How to configure the Console input/output devices

On Itanium systems there are basically four possible console screens:

- VGA-terminal (monitor)
- On board serial console port (system serial)

- Management port Lan (lan console)
- Management port Serial

NOTE: the Management Port (MP) is only available on servers, not on workstations. From the EFI Boot Maintenance Manager menu you can enable or disable these ports:

```
EFI Boot Maintenance Manager ver 1.10 [14.61]
```

```
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
```

In this example all three devices are enabled. The asterix indicates this.

```
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
```

} system
serial

} MP serial

} VGA or
MP lan

Use the following table to identify the different ports:

Console Port	UART	PCI
VGA	no	yes
MP Lan	no	yes
MP Serial	yes	yes
System Serial	yes	no

UART = Univeral Asynchronous Receiver-Transmitter

KMINE doc [IA64KBRC00011702](#) provides tables that show which of the devices display the output of:

- power on self test (POST)
- EFI menus
- HP-UX system bootup messages (i.e. VG activation, startup-scripts, login prompt)

depending on the different combinations of selected console devices.

Here is the example for the rx2600 server:

Console Port	Console input	Console output	POST messages	EFI menu	Boot messages
--------------	---------------	----------------	---------------	----------	---------------

Serial Port	<i>none</i>	<i>none</i>	no	yes	no
Management Port (MP)	<i>none</i>	<i>none</i>	yes	yes	yes
VGA monitor	<i>none</i>	<i>none</i>	no	yes	no

Serial Port	<i>none</i>	<i>none</i>	no	no	no
Management Port (MP)	<i>none</i>	<i>none</i>	yes	yes	no
VGA monitor	*	*	no	yes	yes

Serial Port	*	*	yes	yes	yes
Management Port (MP)	<i>none</i>	<i>none</i>	yes	yes	no
VGA monitor	<i>none</i>	<i>none</i>	no	no	no

Serial Port	<i>none</i>	<i>none</i>	no	no	no
Management Port (MP)	*	*	yes	yes	yes
VGA monitor	<i>none</i>	<i>none</i>	no	no	no

Serial Port	*	*	yes	yes	no
Management Port (MP)	*	*	yes	yes	yes
VGA monitor	<i>none</i>	<i>none</i>	no	no	no

Serial Port	*	*	yes	yes	no
Management Port (MP)	*	*	yes	yes	yes
VGA monitor	*	*	no	yes	no

This can be summarized as follows:

- If you select no console input & output devices, EFI automatically selects them all.

- The VGA monitor does never show POST output.
- Even if there is no shell login on VGA you will always get a CDE login prompt presented in case the system is equipped with a supported graphics card.
- Boot messages are only printed to one single output device. If multiple devices are enabled the priority is: 1. MP port, 2. Serial port, 3. VGA monitor.

So if you like to have boot messages on VGA you need to enable the VGA port only. EFI's console device settings can be overwritten by specifying the `vga` option at the boot loader:

```
HPUX> boot -vga
```

Now boot messages go to the VGA monitor. The MP port cannot be used after this, only on-board serial port.

NOTE (UX 11.23): This all seems to have changed in UX 11.23. It looks like output is going to all devices.

The boot process

The following section describes the boot process in detail.

NOTE: I recommend to use a vt100 compatible terminal type if you logon to the Management port console (former GSP). After powering on the system will perform the Power On Self Tests (POST) and then launch the EFI Boot Manager:

```
EFI Boot Manager ver 1.10 [14.60]  Firmware ver 2.21 [4306]

Please select a boot option

EFI Shell [Build-in]
HP-UX 11.22
Red Hat Linux Advanced Server
Microsoft Windows
Boot option maintenance menu
Security/Password Menu

Use ^ and v to change option(s). Use Enter to select an option
```

In this case boot option `HP-UX 11.22` is configured. Selecting this will start the OS Loader - continue [here](#).

If no boot option is configured yet you may select `EFI Shell [Build-in]`. The device mapping table will be displayed and the EFI shell script `STARTUP.NSH` will be executed:

```
Loading.: EFI Shell [Build-in]
EFI Shell version 1.10 [14.60]
Device mapping table
fs0  : Acpi(HWP0002,0)/Pci(2|0)/Ata(Primary,Master)/CDROM(Entry0)
```

```

fs1 : Acpi(HWP0002,100)/Pci(1|0)/Scsi(Pun0,Lun0)/HD(Part1,Sig831B783A-...
fs2 : Acpi(HWP0002,100)/Pci(1|0)/Scsi(Pun0,Lun0)/HD(Part3,Sig831B7880-...
fs3 : Acpi(HWP0002,100)/Pci(1|0)/Scsi(Pun1,Lun0)/HD(Part1,SigEDC974D0-...
fs4 : Acpi(HWP0002,100)/Pci(1|0)/Scsi(Pun1,Lun0)/HD(Part4,Sig9912850E-...
fs5 : Acpi(HWP0002,100)/Pci(1|1)/Scsi(Pun2,Lun0)/HD(Part1,Sig9C8E9EE8-...
blk0 : Acpi(HWP0002,0)/Pci(2|0)/Ata(Primary,Master)
...
...

startup.nsh> echo -off

Welcome to HP-UX for IA64
setting hpux path(\EFI\HPUX)...
type 'fs[x]:' where x is your bootdisk (0, 1, 2...)
type 'hpux' to start hpux bootloader
Shell>

```

Now we can correlate the hardware device path with the EFI device mapping, e.g.:

fs5: Acpi(HWP0002,100)/Pci(1|1)/Scsi(Pun2,Lun0)/HD(Part1,Sig9C8E9EE8-...)

Path component	meaning
Acpi(HWP0002,X)	Relevant device is installed to Pci.host bus (X = PCI host bus number)
Pci(D F)	PCI installation info. D = PCI device/slot number. F = PCI function number
Scsi(PunX,LunY)	Physical location of device connected to card. In this case SCSI disk. X = SCSI ID, Y = LUN ID (c#tXdY in device file)
HD(Part1,Sigxxxxxxxx)	SCSI disk drive information. Partition and signature

In this case it is the disk at HW path 0/1 /1/ 1. 2.0 with device file c3t2d0

- ACPI(HWP0002,100) correlates with 0/1
- PCI(D, -) correlates with /1/, which is the PCI device or slot number.
- PCI(-, F) correlates with 1., which is the PCI function number.
- SCSI(Pun, Lun) correlates with the final 2.0 which specifies the physical location of the SCSI device connected to the card.

As of UX 11.23 there is a new -e option for ioscan(1M) that displays EFI partitions:

```

# ioscan -e -fnkCdisk
Class      I  H/W Path      Driver      S/W State  H/W Type  Description
=====
disk       0  0/0/2/0.0.0.0  sdisk       CLAIMED    DEVICE     TEAC      DV-28E-B
              /dev/dsk/c0t0d0  /dev/rdsd/c0t0d0
              Acpi(000222F0,0)/Pci(2|0)/Ata(Primary,Master)/\EFI\HPUX\HPUX.EFI
disk       1  0/1/1/0.0.0.0  sdisk       CLAIMED    DEVICE     HP 36.4GST336753LC

```



```

/dev/rdisk/c2t0d0s2          /dev/dsk/c2t0d0          /dev/dsk/c2t0d0s2          /dev/rdisk/c2t0d0
/dev/rdisk/c2t0d0s3          /dev/dsk/c2t0d0s1        /dev/dsk/c2t0d0s3          /dev/rdisk/c2t0d0s1
/dev/rdisk/c2t0d0s3
Acpi (000222F0,100)/Pci (1|0)/Scsi (Pun0,Lun0)/HD (Part1,Sig831B783A-296A-11D8-
8002-D6217B60E588)/\EFI\HPUX\HPUX.EFI
disk      2  0/1/1/0.1.0      sdisk      CLAIMED      DEVICE      HP 36.4GST336753LC
/dev/dsk/c2t1d0          /dev/dsk/c2t1d0s2          /dev/rdisk/c2t1d0
/dev/rdisk/c2t1d0s2
/dev/dsk/c2t1d0s1        /dev/dsk/c2t1d0s3          /dev/rdisk/c2t1d0s1
/dev/rdisk/c2t1d0s3
Acpi (000222F0,100)/Pci (1|0)/Scsi (Pun1,Lun0)/HD (Part1,SigEDC974D0-3AA3-11D8-
83EC-9C7DD0B67B8B)/\EFI\HPUX\HPUX.EFI
disk      3  0/1/1/1.2.0      sdisk      CLAIMED      DEVICE      HP 36.4GST336753LC
/dev/dsk/c3t2d0          /dev/rdisk/c3t2d0
Acpi (000222F0,100)/Pci (1|1)/Scsi (Pun2,Lun0)/\EFI\HPUX\HPUX.EFI

```

NOTE: You may enter help at the shell prompt:

```

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

Shell>

```

If you do not know which fs represents your HP-UX boot disk just try all of them and search for the HP-UX loader:

```

Shell> fs2:

fs2:\> ls
Directory of: fs2:\

10/28/02  11:10p <DIR>          512  EFI
10/28/02  11:10p              336  STARTUP.NSH
      1 File(s)              336 bytes
      1 Dir(s)

```

```

fs2:\> type startup.nsh
File: fs2:\startup.nsh, Size 336

#
# Copyright (c) 2000 Hewlett Packard Corporation
#
# HP-UX EFI start-up shell script
#
echo -off
echo ""

```

```

echo "Welcome to HP-UX for IA64"
echo "  setting hpux path(\EFI\HPUX)..."
set path ".;\efi\hpux;\efi\tools;"
echo "  type 'fs[x]:' where x is your bootdisk (0, 1, 2...)"
echo "  type 'hpux' to start hpux bootloader"
echo -on

fs2:\> cd efi

fs2:\EFI> ls
Directory of: fs2:\EFI

10/28/02  11:10p <DIR>          512  .
10/28/02  11:10p <DIR>          0    ..
10/28/02  11:10p <DIR>          512  HPUX
10/28/02  11:10p <DIR>          512  Intel_Firmware
10/28/02  11:10p <DIR>          512  DIAG
10/28/02  11:10p <DIR>          512  HP
10/28/02  11:10p <DIR>          512  TOOLS
11/19/02  11:22a <DIR>          512  FC
      0 File(s)                0 bytes
      8 Dir(s)

fs2:\EFI> cd hpux

fs2:\EFI\HPUX> ls
Directory of: fs2:\EFI\HPUX

10/28/02  11:10p <DIR>          512  .
10/28/02  11:10p <DIR>          512  ..
10/28/02  11:10p                419,545 HPUX.EFI
10/28/02  11:10p                24,576 NBP.EFI
10/28/02  11:10p                12    AUTO
      3 File(s)              444,133 bytes
      2 Dir(s)

fs2:\EFI\HPUX> type auto
File: fs2:\EFI\HPUX\auto, Size 12

boot vmunix

fs2:\EFI\HPUX>

```

yes. This looks like the HP-UX boot disk.

Switch from the EFI shell to fs2:

```

Shell> fs2:

fs2:\>

```

From here you are able to switch to any other boot disk that has an EFI file system, e.g to switch to fs1 type `fs1:.` In order to go back to the EFI boot manager menu type `exit`.

To continue, invoke the HP-UX boot loader. It's an EFI executable (`hpux.efi`). EFI executables (`.efi`) and EFI shell scripts (`.nsh`) can be invoked without specifying the appendix. This is DOS like ;-)

```
fs2:\> hpux

(c) Copyright 1990-2002, Hewlett Packard Company.
All rights reserved

HP-UX Boot Loader for IA64  Revision 1.713

Press Any Key to interrupt Autoboot
\efi\hpux\AUTO ==> boot vmunix
Seconds left till autoboot - 3
```

Pressing any key at this point interrupts the 10 seconds autoboot timeout and invokes the OS Loader `hpux.efi` in interactive mode. At the `HPUX>` prompt you have several possibilities to interact, such as

Changing boot options:

<code>HPUX> boot -is</code>	for single user mode
<code>HPUX> boot -lq</code>	for LVM quorum mode
<code>HPUX> boot -lm</code>	for maintenance mode boot
<code>HPUX> boot -tm</code>	for failsave mode (no DLKMs etc,)
<code>HPUX> boot backup</code>	for booting backup kernel configuration
<code>HPUX> boot vmunix.prev</code>	for booting previous kernel
<code>HPUX> boot -vga</code>	redirect console output (e.g. rc scripts) to VGA monitor.

Some of the options may be combined.

NOTE: At UX 11.22 you need to have [PHKL 28787](#) or later installed to be able to boot to maintenance mode.

Getting help:

```
HPUX> help
-- HPUX bootloader for IA64 Help --
list of supported commands:

boot [kernel] - boots HPUX kernel
help [-d]      - help screen (-d to list debug commands)
ls [-b] [dir]  - lists directory (-b for screen break)
ll [-b] [dir]  - lists directory in long detail
mmap           - show current EFI memory map
setauto [-d] [str] - sets AUTO file (-d to delete AUTO file)
showauto       - shows AUTO file
ver            - prints version numbers
exit           - exits bootloader
```

Gathering firmware versions:

```
HPUX> ver
Bootloader Version => 1.713
EFI Specification Revision => 1.10
Firmware Vendor   => HP
Firmware Revision=> 14.60
CPU Revision => 7
```

Displaying the content of the AUTO file:

```
HPUX> showauto
\efi\hpux\AUTO => boot vmunix
```

Listing the content of the boot directory /stand:

```
HPUX> ll
FILENAME                                SIZE    DATE (UTC)
.
..
.kminstall_lock                         0       2002/11/13
.kmmodreg_lock                         0       2002/11/13
.kmsystune_lock                        0       2002/08/09
boot.3D5429A80B4D/                     0       2002/08/09
boot.3DBE377F0857/                     0       2002/10/29
boot.sys/                              0       2002/08/09
bootconf                               21      2002/10/29
build/                                 0       2003/02/21
dlkm(LINK)                             24      2002/10/29
dlkm.3D5429A80B4D/                     0       2002/08/09
dlkm.3DBE377F0857/                     0       2003/02/21
ioconfig                               3,076   2003/02/21
kernrel                                82      2002/08/09
krs/                                   0       2003/02/21
krs_lkg/                              0       2003/02/21
krs_tmp/                              0       2003/02/21
lost+found/                           0       2002/10/29
rootconf                               12      2003/02/21
system                                1,041   2002/10/29
system.d/                             0       2002/11/13
vmunix                                39,775,528 2002/10/29
```

If autoboot timeout was not interrupted the OS loader will be launched with the options as specified in the AUTO file and try to load the kernel:

```
AUTOBOOTING...
AUTO BOOT> boot vmunix
> System Memory = 2041 MB
loading section 0 ..... (complete)
loading section 1 ..... (complete)
loading symbol table
loading System Directory(boot.sys) to MFS .....
loading Kernel Boot Directory(boot.3DBE377F0857) to MFS. ....
Launching /stand/vmunix
```

```
SIZE: Text:19787K + Data:2610K + BSS:2381K = Total:24779K
```

If you are sitting in front of the VGA monitor and your system seems to hang at this stage the console output may not be configured properly. The “SIZE:” message above is printed by HP-UX bootloader who prints console output to all available devices. The subsequent message “Console is on ...” is printed by the HP-UX kernel’s function `printf()`. This goes only to the configured console path. Refer to the [section above](#) in order to configure the console for VGA.

If boot messages are set up to be directed to your current console device, the boot processes will continue as follows:

```
Console is on a Serial Device
Booting kernel...
```

Here it takes some time, then the normal boot process as described in the [Boot chapter](#) continues until you get the Console login:

```
Loaded ACPI revision 2.0 tables.
NOTICE: nfs3 link(): File system was registered at index 4.
NOTICE: autofs_link(): File system was registered at index 5.
NOTICE: cacheefs_link(): File system was registered at index 6.
td: claimed Tachyon XL2 Fibre Channel Mass Storage card at 0/7/1/0
Boot device's HP-UX HW path is: 0.0.2.0.2.0
legacyio_cdio_end: WARNING isc for function 0 not found!!!

    System Console is on the Built-In Serial Interface
igelan0: APA NOT SUPPORTED on HP A6794-60001 PCI 1000Base-T at hardware path
0/1/1/0/4/0
igelan0: INITIALIZING HP A6794-60001 PCI 1000Base-T at hardware path 0/1/1/0/4/0
Entering cifs_init...
Initialization finished successfully... slot is 10
Logical volume 64, 0x3 configured as ROOT
Logical volume 64, 0x2 configured as SWAP
Logical volume 64, 0x2 configured as DUMP
    Swap device table: (start & size given in 512-byte blocks)
        entry 0 - major is 64, minor is 0x2; start = 0, size = 8388608
Starting the STREAMS daemons-phase 1
Checking root file system.
file system is clean - log replay is not required
Root check done.
..... OK

...
...
...
...

The system is ready.

GenericSysName [HP Release B.11.22] (see /etc/issue)
Console Login:
```

How to mirror the root disk

The following procedure shows how to mirror the root disk. Let c1t0d0 be the existing primary disk and c2t1d0 the new mirror boot disk:

For LVM

1) Setup the disk partitions

At a cold-installed UX 11.23 system the partition sizes are different compared to UX 11.22. Use diskinfo(1M) to check the partition sizes. E.g. for a UX 11.23 system you would get:

```
# diskinfo /dev/rdisk/c1t0d0s1 | grep size
size: 512000 Kbytes

# diskinfo /dev/rdisk/c1t0d0s3 | grep size
size: 409600 Kbytes
```

Create a partition description file:

UX 11.22	UX 11.23 (cold-installed)
# vi /tmp/partitionfile 2 EFI 100MB HPUX 100%	# vi /tmp/partitionfile 3 EFI 500MB HPUX 100% HPSP 400MB

Use idisk(1M) command to partition the disk according to this file:

```
# idisk -wf /tmp/partitionfile /dev/rdisk/c2t1d0
idisk version: 1.2
***** WARNING *****
If you continue you may destroy all data on this disk.
Do you wish to continue(yes/no)? yes
...
```

2) Create the new device files for the new partitions (c2t1d0s1, s2, (s3))

```
# insf -e -Cdisk
```

3) Use efi_fsinit(1M) to initialize the FAT filesystem on the EFI partition:

```
# efi_fsinit -d /dev/rdisk/c2t1d0s1
```

NOTE: This step is not necessary if it can be guaranteed that the mirror disk does not contain a valid EFI filesystem. In this case efi_fsinit(1M) will be done automatically by the subsequent

mkboot(1M) command. But if you take e.g. an old UX 11.22 boot disk as mirror disk, mkboot will not automatically run efi_fsinit. As a result only 100MB of the 500MB EFI partition (s1) can be used.

- 4) Use mkboot(1M) to format the EFI partition (s1) and populate it with the EFI files below /usr/lib/efi/ and to format the LIF volume (part of s2) and populate it with the LIF files (ISL, AUTO, HPUX, LABEL) below /usr/lib/uxbootlf:

```
# mkboot -e -l /dev/rdisk/c2t1d0
# efi_ls -d /dev/rdisk/c2t1d0s1 (to check EFI)
FileName                               Last Modified                               Size
EFI/                                    11/ 5/2003                                   0
STARTUP.NSH                             11/ 5/2003                                296

total space 523251712 bytes, free space 520073216 bytes
```

```
# lifls -l /dev/rdisk/c2t1d0s2 (to check LIF)
```

- 5) Check the content of AUTO file on EFI partition:

```
# efi_cp -d /dev/rdisk/c2t1d0s1 -u /EFI/HPUX/AUTO /tmp/x; cat /tmp/x
```

NOTE: Specify the -lq option if prefer that your system boots up without interruption in case of a disk failure:

```
# mkboot -a "boot vmunix -lq" /dev/rdisk/c2t1d0
# mkboot -a "boot vmunix -lq" /dev/rdisk/clt0d0
```

- 6) Copy the HP service partition (UX 11.23 only):
(skip this, if you don't have a service partition)

```
# dd if=/dev/rdisk/clt0d0s3 of=/dev/rdisk/c2t1d0s3 bs=1024k
```

- 7) Initialize the LVM partition (s2) and add it to vg00:

```
# pvcreate [-f] -B /dev/rdisk/c2t1d0s2 (take care to use s2)
# vgextend vg00 /dev/dsk/c2t1d0s2
```

- 8) Mirror the LVs to the s2 partition:

```
# for i in lvoll lvoll2 ... lvoll8 (specify each LV)
> do lvextend -m 1 /dev/vg00/$i /dev/dsk/c2t1d0s2
> done
```

- 9) Check if content of LABEL file
(i.e. root, boot, swap and dump device definition) has been initialized (done by lvextend) on the mirror disk:

```
# lvinboot -v
Boot Definitions for Volume Group /dev/vg00:
Physical Volumes belonging in Root Volume Group:
    /dev/dsk/clt0d0s2 (0/1/1/1.2.0) -- Boot Disk
    /dev/dsk/c2t1d0s2 (0/1/1/0.1.0) -- Boot Disk
Boot: lvoll      on:      /dev/dsk/clt0d0s2
```

```

Root: lvol3      on:      /dev/dsk/c2t1d0s2
                  /dev/dsk/c1t0d0s2
Swap: lvol2      on:      /dev/dsk/c1t0d0s2
                  /dev/dsk/c2t1d0s2
Dump: lvol2      on:      /dev/dsk/c1t0d0s2, 0

```

If not, then set it:

```

# lvinboot -r /dev/vg00/lvol3
# lvinboot -b /dev/vg00/lvol1
# lvinboot -s /dev/vg00/lvol2
# lvinboot -d /dev/vg00/lvol2

```

- 8) Add a line to /stand/bootconf for the new boot disk (the letter l is for LVM)

```

# vi /stand/bootconf
l /dev/dsk/c2t1d0s2
l /dev/dsk/c1t0d0s2

```

NOTE: This is not necessary to be able to boot from the mirror in the first place. But if you omit it then you will run into trouble when installing an EFI boot loader patch (e.g. PHKL_34088). This patch updates the hpux.efi boot loader on all disks listed in the bootconf file.

- 9) Specify the mirrored disk as alternate bootpath

```

# setboot -a <HW path of mirror>
# setboot -h <HW path of mirror>    (for UX 11.23 use the HA alternate)
# setboot                          (to check it)

```

NOTE: At UX 11.22 setboot did not automatically create an EFI boot option for the mirror. Follow the steps in the “How to configure additional boot options” section [below](#) to do this.

For VxVM

For VxVM most of the work is done automatically by the vxrootmir(1M) script.

- 1) Have VxVM see the new c2t1d0 disk (if not already booted with it):

```
# vxdctl enable
```

- 2) Confirm that you can see the new disk as “online” or “online invalid”:

```
# vxdisk list
```

- 3) Mirror the root disk. This may take some time:

```
# /etc/vx/bin/vxrootmir -v -b c2t1d0
```

- 4) Verify that all volumes are mirrored:

```
# vxprint -g rootdg
```


5) Verify the LABEL files on root and mirror disk:

```
# vxvmboot -v /dev/rdisk/clt0d0s2

LIF Label File @ (1k) block # 834 on VxVM Disk /dev/rdisk/clt0d0s2:
Label Entry: 0, Boot Volume start:      3168; length: 300 MB
Label Entry: 1, Root Volume start:    4504672; length: 400 MB
Label Entry: 2, Swap Volume start:     310368; length: 4096 MB
Label Entry: 3, Dump Volume start:     310368; length: 4096 MB

# vxvmboot -v /dev/rdisk/c2t1d0s2

LIF Label File @ (1k) block # 834 on VxVM Disk /dev/rdisk/c2t1d0s2:
Label Entry: 0, Boot Volume start:      3168; length: 300 MB
Label Entry: 1, Root Volume start:    4504672; length: 400 MB
Label Entry: 2, Swap Volume start:     310368; length: 4096 MB
Label Entry: 3, Dump Volume start:     310368; length: 4096 MB
```

6) Verify the setboot path:

```
# setboot
```

How to replace a failed disk (LVM)

There is no difference between PA and IPF systems when talking about replacing *non-root* disks. Even for root disks the procedure is quite similar, so for details refer to the [LVM chapter](#). The partitioned disk layout and the existence of EFI needs to be considered though. Hence there are three major differences compared to the usual “hot-swap” procedure.

1) Partition the new disk

You have to run `idisk(1M)` first before performing `mkboot(1M)`, `lvlnboot(1M)` and `vgcfgrestore(1M)` in order to partition the new disk. Note that `vgcfgrestore(1M)` does not restore EFI or LIF. This is done by `mkboot(1M)`. `idisk(1M)` usage is explained in [step 1](#) of the LVM mirror procedure above.

2) Recreate EFI boot option

`idisk(1M)` creates a new disk identifier (GUID) on the disk. If an EFI boot option entry (held in NVRAM) exists for the replaced disk it is still linked to the old GUID. Hence you have to remove the old boot option entry and add a new one. Refer to the additional boot options section [below](#).

For UX 11.23 this is done automatically by the `setboot(1M)` command:

```
# setboot -h 0/1/1/0.1.0
```

-h means high available alternate

3) Be careful to specify the `s2` device file (`c#t#d#s2`) when performing LVM commands.

How to configure additional boot options

It would be nice to have additional boot options configured in the EFI menu, e.g. for boot from mirror disk, boot to single user mode, etc. Since the Boot options (stored in EFI NVRAM) which can be specified in the Boot Option Maintenance Menu do not seem to have effect, it has to be done in a different way.

Here's how to add a boot option for single user mode boot:

- 1) Go to EFI shell and switch to the primary boot disk (e.g. c1t0d0):

```
shell> fs1:
```

- 2) Go to the EFI directory, create another subdirectory and copy the bootloader

```
fs1:\> cd EFI
fs1:\EFI\> mkdir HPUX_single_user
fs1:\EFI\> cp HPUX\HPUX.EFI HPUX_single_user\
fs1:\EFI\> cd HPUX_single_user
```

- 3) Create an AUTO file for the bootloader that contains the boot options:

```
fs1:\EFI\> edit AUTO
```

enter the line:

```
boot vmunix -is
```

or (to boot without quorum check):

```
boot vmunix -is -lq
```

Type ESC-s to save and ESC-q to quit the editor.

- 4) Now exit the EFI shell (`exit`) and go to the boot maintenance menu in order to add a boot option for this:

```
EFI Boot Maintenance Manager ver 1.10 [14.61]

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

```

Select the disk, here c1t0d0. This is Pun0,Lun0 (SCSI target 0, Lun 0):

```

EFI Boot Maintenance Manager ver 1.10 [14.61]

Add a Boot Option.  Select a Volume

    IA64_EFI                                [Acpi (HWP0002,100) /Pci (1|0) /Scsi (Pun1,Lun0) /HD
(Part1,SigB45A0000)
    IA64_EFI                                [Acpi (HWP0002,100) /Pci (1|1) /Scsi (Pun0,Lun0) /HD
(Part1,Sig958B0000)
    EFI      DISK                            [Acpi (HWP0002,600) /Pci (1|0) /Pci (0|0) /Pci (0|0) /Pci (0|0) /
Scsi (Pun0,Lun0) /HD (Part1,Sig119E1A60-0B4C-01C3-507B-9E5F8078F531)
    Removable Media Boot [Acpi (HWP0002,0) /Pci (2|0) /Ata (Primary,Master)
    Load File [EFI Shell [Built-in]]
    Load File [Acpi (HWP0002,0) /Pci (3|0) /Mac (00306E3809C6) ]
    Load File [Acpi (HWP0002,100) /Pci (2|0) /Mac (00306E3889E3) ]
    Exit

```

Now navigate to the HP-UX bootloader HPUX.EFI on the disk:

```

EFI Boot Maintenance Manager ver 1.10 [14.61]

Select file or change to new directory:

    05/28/03  09:38a <DIR>                512 EFI
    [Treat like Removable Media Boot]
    Exit

Select file or change to new directory:

    05/28/03  09:38a <DIR>                512 .
    05/28/03  09:38a <DIR>                0  ..
    05/28/03  09:38a <DIR>                512 HPUX
    05/28/03  09:38a <DIR>                512 HPUX single user
    05/28/03  09:38a <DIR>                512 Intel Firmware
    05/28/03  09:38a <DIR>                512 DIAG
    05/28/03  09:38a <DIR>                512 HP
    05/28/03  09:38a <DIR>                512 TOOLS
    Exit

```

```

Select file or change to new directory:

    05/28/03  09:38a <DIR>                512 .

```

```

05/28/03 09:38a <DIR>          512 ..
05/28/03 11:52a          419,545 HPUX.EFI
Exit

Filename: \EFI\HPUX_single_user\HPUX.EFI
DevicePath:[Acpi(HWP0002,100)/Pci(1|0)/Scsi(Pun1,Lun0)/HD(Part1,SigB45A0000)/\EFI\HPUX_single_user\HPUX.EFI]
IA-64 EFI Application 05/28/03 11:52a 419,545 bytes

BootFFFF:
Acpi(HWP0002,100)/Pci(1|0)/Scsi(Pun1,Lun0)/HD(Part1,SigB45A0000)/\EFI\HPUX_single_user\HPUX.EFI

```

Now enter a description for this boot option, e.g. *HP-UX mirror boot disk*:

```

Enter Description: HP-UX single user boot

Current BootOption-->Main Menu. Select an Operation
New BootOption Data. ASCII/Unicode strings only, with max of 240 characters
Enter BootOption Data Type [A-Ascii U-Unicode N-No BootOption] : N

```

Finally save the setting to NVRAM:

```

Save changes to NVRAM [Y-Yes N-No] : Y

```

The ARIES emulator

Basics

Aries is the HP-UX PA-RISC to HP-UX IPF binary emulator or *dynamic code translator*. Aries transparently emulates both 32-bit and 64-bit HP-UX PA-RISC 2.0 applications on HP-UX IPF, thus providing binary compatibility for PA-RISC binaries on IPF systems. It is transparent in that users need not do anything special to simply run their PA programs on IPF.

The HP-UX/IPF kernel (exec(2)) will recognize PA executable files (both 32- and 64-bit fields) and invoke Aries to emulate these processes. Aries will handle all instruction set architecture (ISA) emulation and environment emulation on behalf of the PA program, and present a virtual PA machine architecture to the emulated program.

There are actually two versions of Aries: one for emulating PA-32 processes, and one for PA-64 processes. This is necessary because Aries must keep the address space layout that the emulated process expects; natively, then Aries-32 is a 32-bit process, and Aries-64 is a 64-bit process.

Performance

Because an emulator can never be as performant as native code Aries should be used where performance is not critical or where it is not possible to create a native IPF family binary.

Depending on the type of application you will see more or less impact on performance compared to PA-RISC. Here's an extract of the aries man page:

Aries performance can be characterized by following points:

- 1) For PA-RISC applications that have an **average mix of system intensive, I/O intensive, and memory intensive operations**, Aries performance on an Itanium2-based machine running at 1500 MHz is of the order of or better than a PA8700-based machine. For such PA-RISC applications, Aries performance may be closer to an Itanium2-based machine's native performance.
- 2) For PA-RISC applications that have **integer intensive operations**, Aries performance on an Itanium2-based machine running at 1500 MHz is about 70-80% of a PA8700-based machine and is 10-15% better than PA8600-based machine. For such PA-RISC applications, Aries performance is 35-40% of Itanium2-based machine's native performance.
- 3) For PA-RISC applications that have **floating point intensive operations**, Aries performance on an Itanium2-based machine running at 1500 MHz is about 50% of a PA8700-based machine and about 70% of PA8600 based machine. For such PA-RISC applications, Aries performance is only 10-15% of an Itanium2-based machine's native performance.
- 4) For PA-RISC virtual machines like **JVM (Java Virtual Machine)**, Aries performance is slower of the order of 15-20 times compared to an Itanium2-based machine's native performance. Such applications are not good candidates for execution in emulation mode under Aries.
- 5) Aries performance for **multi-threaded PA-RISC applications** is poorer than the non-threaded counterpart of the same application. This is because Aries implements internal mutex locks to protect Aries private data structures. Aries threads need to reach a safe point before they can be suspended. This condition adds to the performance penalty as the calling thread will have to wait until the target thread reaches the safe zone before it can be suspended. This is required, as before suspending a thread, Aries has to make sure that PA-RISC application's emulated register context is consistent in registers and memory.
- 6) PA-RISC applications which make use of **performance libraries** like HP mlib are not good candidates to run under Aries. Performance library (for example, HP mlib) kernels are hand-coded in assembly and are tuned to the cache behavior and instruction resources as to achieve closer to theoretical machine peak rate. It is not possible for a software emulator like Aries to emulate this detailed behavior of the application for maximum hardware resource

utilization.

- 7) PA-RISC applications that make extensive use of OpenGL experience slow performance. This is because the ogld daemon process on an Itanium-based machine is a native process, and an emulated application cannot send its graphics output to a native graphics card by directly communicating with the ogld daemon process. Such PA-RISC applications send their graphics data to the Itanium-based machine's graphics card through a virtual memory driver (VMD) which is emulated under Aries. This process of displaying the graphics data is slow. Instead, such applications should make use of OpenGL display lists. This causes the PA-RISC OpenGL library to use GLX protocol mode to the X server and the ogld process. This process of displaying graphics data with OpenGL can be significantly faster than the virtual memory driver route.

There are some Aries options available to optimize Aries for certain applications. Refer to the aries man page.

If you intend to run PA binaries on IPF it is strongly recommended to apply the latest available Aries patch:

UX 11.22: [PHSS 30214](#) or newer

UX 11.23: [PHSS 30015](#) or newer

Further information about Aries can be found on the websites mentioned in the [Additional information](#) section.

IPF vs. PA-RISC Terminology

PA term	IPF term
PA-RISC	Itanium
Precision Architecture (PA)	Itanium Processor Family (IPF)
HP 9000	HP Integrity
Chassis Code	Event ID
Activity Log	Forward Progress Log (FPL)
Errorlog	System Event Log (SEL)
Boot Control Handler (BCH)	Extensible Firmware Interface (EFI)
Processor Dependant Code (PDC)	Processor Abstraction Layer (PAL)
IO Dependant Code (IODC)	System Abstraction Layer (SAL)
High Priority Machine Check (HPMC)	Machine Check Abort (MCA)
Transfer of Control (TOC)	Non-Maskable Interrupt (NMI) or INIT
LPMC (corrected by CPU) e.g. Level 2 Cache Single Bit Error (SBE)	Correctable Machine Check (CMC)
LPMC (corrected by CEC)	Correctable Platform Error (CPE)

e.g. Memory SBE	
-----------------	--

CEC (Core Electronic Complex) is the chipset (zx1/sx1000)

PAL is delivered by Intel, SAL by HP. PAL and SAL together represent the PDC on PA-RISC.

EFI represents the BCH and IPL.

The firmware releases for PAL, SAL and EFI are installed as a single bundle.

Introducing UX 11.23PI

The latest HP-UX release UX 11.23PI was released in Sep. 2004.

The official name is HP-UX 11i v2 update 2 or UX 11i v2 UPD2. The HP internal names are 11.23PI or 11.23 0409.

PI stands for PA-RISC & Itanium, so it is the first OS release that supports both platforms. PA-RISC customers can benefit from the new features of 11.23.

Update paths

You can update from UX 11.11, 11.22 or 11.23.

- 1) For UX 11.11 you need to install the Update-UX product from the 11.23PI install media to update.
- 2) For UX 11.22 you need to install the Aug 2004 BUNDLE11i patch bundle before installing the Update-UX product.
- 3) For UX 11.23 you need to install the Aug 2004 BUNDLE11i patch bundle before installing the Update-UX product.

ATTENTION: The Aug 2004 bundle contains 7 patches. Although each of them has been replaced by the Sep 2004 base patch patch PHKL_31500 which itself is included in the Sep bundle, the Aug bundle is still required because it solves kernel build problems. The Aug bundle is required before installing PHKL_31500 or the Sep bundle which includes it.

ATTENTION: Even in case you are updating from previous 11.23 to 11.23PI you need to run update-ux. Just installing the Aug and Sep patch bundles is not sufficient since these bundles do not update the HPUXBaseAux bundle. If you omit update-ux then you won't have the kweb TUI feature from SAM, e.g.

Installation Requirements

To cold-install or update to 11.23PI, you must have

- 1) Supported HP-UX system
 - Itanium-based servers include HP Integrity c8000, cx2600, rx2600, rx4640, rx5670, rx7620, rx8620, Superdome (16-, 32-, 64-way)
 - Itanium-based workstations include zx2000 and zx6000
 - PA-RISC servers include rp2400, rp3400, rp4400, rp5400, rp7400, rp8400, Superdome (up to 128-way), A-, L-, and N-class
- 2) Minimum firmware revision required
- 3) HP-UX 11i v2 OE and application media (two DVD set)
- 4) 512 MB memory, minimum
- 5) 1 GB swap space, minimum
- 6) 1 GB disk space to accommodate EFI Boot Disk & HP Service Partition
- 7) 12 GB of disk space to install server OEs

Note that documented minimums may be low compared to typical actual configurations

Identifying 11.23 Sep 2004 release

To determine which OS release is running, use `uname` first

PA-RISC platform

```
# uname -a
HP-UX krmt03b B.11.23 U 9000/800 2272434998 unlimited-user license
```

9000/... indicates PA-RISC. B.11.23 indicates 11.23 0409 because 11.23 was not available on PA-RISC before 11.23 0409.

Itanium-based systems

```
# uname -a
HP-UX fcewdu42 B.11.23 U ia64 2548321655 unlimited-user license
```

If output is `ia64` (Itanium) and `B.11.23`, then more data is needed.

```
# swlist | grep HP-UX
HP-UX11i-TCOE      B.11.23.0403    HP-UX Technical Computing OE Component
HP-UXBaseAux       B.11.23.0403    HP-UX Base OS Auxiliary
HP-UXBaseOS        B.11.23         HP-UX Base OS
...
```

B.11.23.0409 would indicate 11.23PI. In this case we have 0403 (March) release and therefore no 11.23PI. This is true even if the Sep 04 bundle was installed (refer to ATTENTION above):


```
# swlist | grep BUNDLE
BUNDLE11i          B.11.23.0409.3 Required Patch Bundle for HP-UX 11i v2
(B.11.23), September 2004
```

New features

Read the release notes of 11i v2 upd 2 on docs.hp.com to learn about all the new features. Here are some remarkable changes.

ps(1) command

before 11.23PI ps (1) displayed a maximum of 64 characters in the command field. At 11.23PI, the default width is set to 128 characters and a default file has been provided (/etc/default/ps) to define the length of the command field. The valid value of the command field width is between 64-1024. The default file will be part of the OS depot.

fuser(1M) command

In earlier releases, the fuser command can require several minutes to complete when there are hundreds of memory-mapped files. At 11.23PI HP has provided a new support from the kernel to identify memory-mapped files or file structure quickly to improve fuser performance.

pvchange(1M) command

pvchange command has a new -a option to temporarily disable a hardware path to a disk device and re-enable it.

LVM – no more bad block relocation

Bad block relocation will no longer be performed.

Mixed Serviceguard clusters

Itanium and PA-RISC nodes can be members in one cluster.

kcweb TUI for kernel configuration

With 11.23PI kcweb provides a TUI for kernel configuration management from a console or a non-graphical display. This will be useful in an environment where graphical display is not possible or is slow. The /opt/hpsmh/lbin/cmenu, part of SysMgmtBase product is used to launch the TUI. The look and feel of the TUI is different. In 11.11 (06.2004) where in a non-graphical environment the TUI is invoked using ObAM 4.2

Cell Local Memory (CLM) support for PA systems

UX 11.23PI introduces CLM support also for PA-RISC platforms. CLM was not available on UX 11.11.

ccNUMA support

HP's cell-based platforms are ccNUMA (cache coherent, Non-Uniform Memory Access) capable, i.e. latency and bandwidth of a same-cell memory access is better than accessing memory on a different cell. For ccNUMA platforms, 11.23PI is ccNUMA-aware. This maintains good behavior for applications that are not ccNUMA-aware and gives applications, that are ccNUMA-aware the ability to control their memory placement as well as processor placement. This is a core feature of 11.23PI.

Additional Information

General

HP's Itanium homepage

<http://hp.com/go/itanium>

Intel's Itanium homepage

<http://developer.intel.com/design/itanium/family/> (non HP)

WTEC Itanium website

<http://wtec.uksr.hp.com/~hpux/docs/ia64> (HP internal)

Itanium Center of Expertise (Canada)

<http://ia64.canada.hp.com/> (HP internal)

IA64 Virtual Competency Center EMEA (hardware)

<http://hpux.uksr.hp.com/ceasst/ia64/> (HP internal)

IA64 Workstation CC (including IPF Hardware Handbook)

<http://www.grc.hp.com/docs/wkstcc/ia64/index.html> (HP internal)

UNIX Support Knowledge Team website

(contains Itanium related training material such as future HP-UX releases, EFI, ...)

<http://cso.fc.hp.com/ssil/uxsk/hpux> (HP internal)

Software Development & Performance

Aries homepage

<http://cllweb.cup.hp.com/migration> (HP internal)

Application availability (ISV listing)

<http://isvappstatus.cac.cpqcorp.net/ISV/ISV.asp> (HP internal)

Software Transition Kit (STK)

<http://devsrc1.external.hp.com/STK/>

SPEC Performance Benchmarks

<http://www.spec.org> (non HP)

TPC Performance Benchmarks

<http://www.tpc.org/> (non HP)

Performance Tuning Guide for Itanium systems

http://www.hp.com/products1/unix/operating/infolibrary/whitepapers/7206_IPF_tuning_wp_051403.pdf

UX 11.23 at docs.hp.com

<http://docs.hp.com/hpux/os/11iV2>

EFI

Intel's EFI homepage:

<http://developer.intel.com/technology/efi/>

Hands on EFI training:

http://cso.fc.hp.com/ssil/uxsk/hpux/hpux_releases/11.22

EFI command overview (includes also POSSE, MP, BMC/CLI commands):

http://hprtdt58.grc.hp.com/documents/systems/longspeak/itanium_handly_trifold.pdf
internal)

(HP