HP-UX System Administrator's Guide: Overview
HP-UX 11i Version 3

Table of Contents

List of Figures

List of Tables

List of Examples

Preface

Intended Audience

The *HP-UX System Administrator's Guide* series is written for administrators of HP-UX systems of all skill levels needing to administer HP-UX systems, beginning with HP-UX Release 11i Version 3.

While many topics in this set apply to previous releases, much has changed in HP-UX 11i as of Version 3. Therefore, for information about prior releases, please refer to *Managing Systems and Workgroups: A Guide for System Administrators*.

About this Series

The *HP-UX System Administrator's Guide* documents the core set of tasks (and associated concepts) necessary to administer HP-UX 11i Version 3.

The *HP-UX System Administrator's Guide* is a set of documents, comprised of the following volumes:

Overview	Provides a high-level view of HP-UX 11i, Version 3, its components, and how they relate to each other.
Configuration Management	Describes many of the tasks you need to perform in order to configure and customize system settings and the behavior of subsystems.
Logical Volume Management	Documents how to configure physical volumes, volume groups, and logical volumes using the HP Logical Volume Manager (LVM).
Security Management	Documents the data and system security features of HP-UX 11i Version 3.
Routine Management Tasks	Documents many of the ongoing tasks you need to perform to keep your system running smoothly.

About this Document:

HP-UX System Administrator's Guide: Overview provides a high level perspective of HP-UX, its components, and how they relate to each other. Where appropriate, this document provides pointers to more detailed information. *Overview* includes the following major topics:

Chapter 1: HP-UX Version 3 at a Glance,	Highlights key features of HP-UX 11i Version 3, important system administration tools, and provides general information about important changes from previous releases of HP-UX.
Chapter 2: HP-UX Virtualization Technologies,	Discusses the many technologies available for use with HP-UX to help you get the most out of your computing resources and provides information to help you understand how these technologies relate to each other.
Chapter 3: Major Components of HP-UX,	Discusses the major pieces of HP-UX and how the pieces relate to each other.
Chapter 4: System Administration Tools,	Describes the tools available for administering HP-UX and its related subsystems from a single system perspective.

Finding HP-UX Information

Table 1Outlines where to find general system administration information for HP-UX. This table does not include information for specific products.

Table 1 Finding HP-UX Information

If you need to. . .	Refer to . . .	Located at . . .
Find out: • What has changed in HP-UX releases • The content of the Operating Environments • Firmware requirements, and supported systems for a specific release	The HP-UX 11i Release Notes specific to your version of HP-UX.	• HP Instant Information (English only) • HP Technical Documentation web site `http://www.hp.com/go/hpux-core-docs`
Install or update HP-UX	• *Read Before Installing or Updating to HP-UX* • *HP-UX 11i Installation and Update Guide*	• Media Kit (supplied with the Operating Environment) • HP Instant Information (English only) • HP Technical Documentation web site `http://www.hp.com/go/hpux-core-docs`
Administer an HP-UX system	For Releases Prior to HP-UX 11i Version 3: • *Managing Systems and Workgroups: A Guide for HP-UX System Administrators* For Releases beginning with HP-UX 11i Version 3: • *HP-UX System Administrator's Guide* (a multivolume set) Other sources of System Administration Information: • *nPartition Administrator's Guide*	• HP Instant Information (English only) • HP Technical Documentation web site `http://www.hp.com/go/hpux-core-docs`

HP-UX 11i Release Names and Operating System Version Identifiers

With HP-UX 11i, HP delivers a highly available, secure, and manageable operating system that meets the demands of end-to-end Internet-critical computing. HP-UX 11i supports enterprise, mission-critical, and technical computing environments. HP-UX 11i is available on both HP 9000 systems and Integrity systems.

Each HP-UX 11i release has an associated release name and release identifier. The uname command with the -r option returns the release identifier. Table 2 shows the releases available for HP-UX 11i.

Table 2 HP-UX 11i Releases

OS Version Identifier	Release Name	Supported Processor Architecture
B.11.11	HP-UX 11i Version 1	HP 9000
B.11.23	HP-UX 11i Version 2	Integrity
B.11.23.0409	HP-UX 11i Version 2 September 2004 Update	HP 9000 and Integrity
B.11.31	HP-UX 11i Version 3	HP 9000 and Integrity
B.11.31.0709	HP-UX 11i Version 3, September 2007 Update	HP 9000 and Integrity

Table 2 HP-UX 11i Releases *(continued)*

OS Version Identifier	Release Name	Supported Processor Architecture
B.11.31.0803	HP-UX 11i Version 3, March 2008 Update	HP 9000 and Integrity
B.11.31.0809	HP-UX 11i Version 3, September 2008 Update	HP 9000 and Integrity
B.11.31.0909	HP-UX 11i Version 3, September 2009 Update	HP 9000 and Integrity
B.11.31.1003	HP-UX 11i Version 3, March 2010 Update	HP 9000 and Integrity
B.11.31.1009	HP-UX 11i Version 3, September 2010 Update	HP 9000 and Integrity

For information on supported systems and processor architecture for different versions of HP-UX, refer to the HP-UX system release notes specific to your version of HP-UX (for example, the *HP-UX 11i Version 3 Release Notes*).

Determining Your System Version

The uname, model, and swlist commands can help you determine information about your system, including its hardware type, machine model, operating system version, and operating environment update status. (See *uname*(1), *model*(1), and *swlist*(1M).)

For OS naming conventions, please see "HP-UX 11i Release Names and Operating System Version Identifiers" (page 12).

Table 3 OS Version, System Architecture, and Machine Model

Topic	Command	Sample Output
OS Version	`$uname -r`	`B.11.31`[1]
Architecture	`$uname -m`	`ia64`[2] `9000/800`[2]
Machine Model	`$model`[3]	`ia64 hp server rx5670` `9000/800/S16K-A`
Operating Environment	`$swlist HPUX*OE*`	`# HPUX11i-OE-MC B.11.31 HP-UX Mission Critical Operating Environment`[1]
OS Version.Update	`$swlist HPUX*OE*`	`# HPUX11i-TCOE B.11.23.0409 HP-UX Technical Computing OE Component`[1]

[1] HP-UX 11i OS version identifiers have the form B.11.23 or B.11.23.0409, where B.11.23 is the OS version and 0409 is the year-month of the operating environment (OE) update.

[2] `ia64` = Integrity. All others = HP 9000.

[3] The getconf MACHINE_MODEL command gives the same output (see *getconf*(1).

Typographic Conventions

This document uses the following typographic conventions.

audit(5)	An HP-UX manpage. *audit* is the name and *5* is the section in the *HP-UX Reference*. On the web and on the Instant Information DVD, it may be a hot link to the manpage itself. From the HP-UX command line, you can enter man audit or man 5 audit to view the manpage. See *man*(1).
Document Title	The title of a document. On the web and on the Instant Information DVD, it may be a hot link to the document itself.
Command	A command name or qualified command phrase.
ComputerOut	Text displayed by the computer.
Emphasis	Text that is emphasized.

Emphasis	Text that is strongly emphasized.
KeyCap	The name of a keyboard key. Note that **Return** and **Enter** both refer to the same key.
Term	The defined use of an important word or phrase.
`UserInput`	Commands and other text that the user types.
Variable	The name of a variable that you may replace in a command or function or information in a display that represents several possible values.
`$`	User command prompt.
`#`	Superuser (`root`) command prompt.

Examples and Shells

This document describes practices used by the system administrator. Since the `root` user, a superuser, is required to use the POSIX shell `/sbin/sh`, all command examples use that shell. The POSIX shell is defined in *sh-posix*(1). For information on other shells, see the *Shells User's Guide* and *sh*(1).

Command Syntax

`Literal`	A word or character that you enter literally.
Replaceable	A word or phrase that you replace with an appropriate value.
-chars	One or more grouped command options, such as `-ikx`. The *chars* are usually a string of literal characters that each represent a specific option. For example, the entry `-ikx` is equivalent to the individual options `-i`, `-k`, and `-x`. The plus character (+) is sometimes used as an option prefix.
-word	A single command option, such as `-help`. The *word* is a literal keyword. The difference from `-chars` is usually obvious and is clarified in an Options description. The plus character (+) and the double hyphen (`--`) are sometimes used as option prefixes.
[ ]	The bracket metacharacters enclose optional content in formats and command descriptions.
{ }	The brace metacharacters enclose required content in formats and command descriptions.
|	The bar metacharacter separates alternatives in a list of choices, usually in brackets or braces.
`...`	The ellipsis metacharacter after a token (*abc*...) or a right bracket ([ ]...) or a right brace ({ }...) metacharacter indicates that the preceding element and its preceding whitespace, if any, may be repeated an arbitrary number of times.
`...`	Ellipsis is sometimes used to indicate omitted items in a range.

Function Synopsis and Syntax

HP-UX functions are described in a definition format rather than a usage format. The definition format includes typing information that is omitted when the function call is actually included in a program.

The general definition form is:

type func (*type param*[, *type param*]...);

For example:

int setuname (const char **name* , size_t *namelen*);

The usage form is:

```
func ( param[, param]...);
```

For example:

```
setuname ( name[, namelen]...);
```

The function syntax elements are the same as for commands, except for the options.

Publication History

The document publication date and part number indicate its current edition. The publication date will change when a new edition is released.

To ensure that you receive the new editions, you should subscribe to the appropriate product support service. See your HP sales representative for details.

First Edition   February 2007
HP Part Number 5991-7436
HP-UX 11i Version 3
Print, HP Technical Documentation web site http://www.hp.com/go/hpux-core-docs

Second Edition   March 2008
HP Part Number 5992-3384
HP-UX 11i Version 3, March 2008 Update
Print, HP Technical Documentation web site http://www.hp.com/go/hpux-core-docs

Third Edition   September 2008, 5992-4580,
HP-UX 11i Version 3 (B.11.31 September 2008 Update)
Printed, DVD (Instant Information), and Web (**http://www.hp.com/go/hpux-core-docs**)

Fourth Edition   September 2009, 5992-6573,
HP-UX 11i Version 3 (B.11.31 September 2009 Update)
Printed, DVD (Instant Information), and Web (**http://www.hp.com/go/hpux-core-docs**)

Fifth Edition   September 2010, B3921-90011,
HP-UX 11i Version 3 (B.11.31 September 2010 Update)
Printed, DVD (Instant Information), and Web (**http://www.hp.com/go/hpux-core-docs**)

NOTE: The documents in the *HP-UX System Administrator's Guide* may be updated independently. Therefore, the latest versions of the documents in the set may vary with time, with respect to each other. The latest versions of each document are available at http://www.hp.com/go/hpux-core-docs.

1 HP-UX Version 3 at a Glance

This chapter highlights some of the key features of HP-UX 11i version 3. However, the *comprehensive source* for what is new and changed in any given HP-UX version, compared with that version's previous releases, is the HP-UX 11i Release Notes. For HP-UX 11i version 3, see *HP-UX 11i Version 3 Release Notes*.

How HP-UX is Delivered

HP-UX is comprised of many components and subsystems. This section covers how those components are packaged and delivered.

Operating Environments

Operating systems are complex software ecosystems that control the usage of resources of your computers (for example memory, peripherals, processors, and applications). HP-UX 11i, HP's version of the UNIX operating system, is highly configurable and has many optional components and applications to enhance the basic functionality of the core operating system. *Operating Environments* are pre-configured combinations of these components, designed and pre-tested to work together.

HP-UX 11i version 3 provides the following operating environments:

- HP-UX 11i v3 Base Operating Environment – (BOE)
- HP-UX 11i v3 Virtual Server Operating Environment – (VS-OE)
- HP-UX 11i v3 High Availability Operating Environment – (HA-OE)
- HP-UX 11i v3 Data Center Operating Environment – (DC-OE)

For complete details about the Operating Environments available for HP-UX 11i version 3, including descriptions of what is included with each, see the version of *HP-UX 11i Version 3 Release Notes* that corresponds to the version of HP-UX 11i that you are using.

Extra Components and Products

Additional Software

In addition to the software components that are shipped as part of HP-UX Operating Environments, HP publishes software components and applications to further enhance the capabilities of HP-UX and your HP servers. Some of these components and applications are available at:

http://software.hp.com

Others are available on the application support media. Contact your HP sales representative, or an HP authorized reseller for details.

Third-Party Products

As with most operating systems, HP-UX 11i is the core of an entire suite of products, not all of which come from HP.

Thousands of applications that run on HP-UX 11i (both on HP 9000 servers and on HP Integrity Servers) are available from software vendors.

HP-UX 11i Version 3 Highlights

With each new release, HP-UX gains new features and enhancements to previously existing features (to enhance their ease of use). Here are a few key features of HP-UX 11i version 3. For the *complete list* of what is new or changed in HP-UX 11i version 3, please see *HP-UX 11i Version 3 Release Notes*.

Processor Terminology

The *HP-UX System Administrator's Guide* uses the following terminology:

> **NOTE:** The items in this list represent an introduction to terminology used throughout the HP-UX System Administrator's Guide. A more comprehensive list of terms is located in the "Glossary" (page 109).

Core

Represented in this document by the double-gear symbol above, a "core," (formerly referred to as a "CPU") is an individual processing unit on a processor chip (see processor). For reasons of clarity, occasionally this document will refer to a core as a "processing core".

Processor

Represented in this document by a solid square surrounding one or more symbols for a core, a "processor" is a physical piece of silicon containing one or more cores.

Hardware Threading

A hardware technique used in Itanium processors to enhance the computational performance of a core. Itanium processors are those used in HP Integrity Servers.

Software Threading

A parallel computing technique used by applications and operating systems to enhance processing efficiency.

Server

Formerly referred to as a "system" or "computer", this document primarily uses the term "server" to describe the physical cabinet containing cell boards, processors, memory, and power supplies.

System

This document occasionally uses the term "system" to define what could be a server *or* a subset of a server (a partition) running an independent copy of HP-UX. It will also use the term system to define what could be a server or another type of system (for example a Windows based PC).

Operating System

Throughout this document, a dotted line surrounding icons for cores and processors indicates an instance of an operating system (HP-UX or otherwise).

Server Nomenclature

When describing servers, cell boards, or other components capable of holding processor chips, HP now uses the nomenclature xP/yC, where x refers to the maximum number of processors the item can hold, and y refers to the maximum number of cores the item can hold.

Examples:

- A server defined as a 1P/1C server (representing one processor/one core) is the simplest of all functional configurations. It contains one, single-core processor.
- A server capable of holding four cell boards, each containing eight dual-core processors is a 32P/64C server. Each cell board is an 8P/16C cell board.

New Device File Format for Addressing Storage

HP-UX 11i version 3 introduces a new device file format that enables many new features, such as:

- Allows you to establish multiple hardware paths to disk and other mass storage devices *using a single device special file*.
- Allows the kernel to automatically bypass failed hardware paths without having to change the device's associated device special file.
- Allows you to relocate (logically or physically) a disk device to a new hardware location *without having to change its device special file*.

Online Activation and Deactivation of PCI Cards

HP-UX 11i version 3 expands your ability to replace hardware components on a running instance of HP-UX without having to shut down or reboot.

PCI Cards

With HP-UX 11i version 3 you can:

- Add PCI / PCI-X cards to a running instance of HP-UX.
- Replace a PCI / PCI-X card (with one of the same type) in a running instance of HP-UX (for example, if a card is not working properly).
- Delete a PCI / PCI-X card from a running instance of HP-UX.

The cards and their associated drivers must support *OL** operations (most do). There are also other important considerations when adding and removing PCI cards. For complete details on PCI / PCI-X Online operations, see the *Interface Card OL* Support Guide (Part Number 5992–1723)*.

Long User and Group Names

In HP-UX 11i version 3, `User` and `Group` names can be up to 255 characters long, and group names can now be all numeric. For information about configuring users and groups, see *HP-UX System Administrator's Guide: Configuration Management*.

> △ **CAUTION:** Be careful, when using the new long names, to insure that all systems needing to interact with a server that uses long user/group names are capable of using the long names. Servers using older versions of HP-UX or servers using non-HP-UX operating systems might not be compatible with this feature.

Concurrent Dump

When an HP-UX system panic (crash) occurs, a memory dump is usually written to disk so that, if necessary, an analysis can be done to determine what caused the crash.[1]

On servers with very large amounts of memory, the process of writing memory contents to disk can take a *very* long time. If you have multiple devices configured for memory dump, HP-UX

1. Whether or not a crash dump is written to disk is configurable.

can be configured to split the task of dumping memory and write to the devices in parallel. This process is called *dump concurrency* and is configured using either the kernel tunable dump_concurrent_on (see *dump_concurrent_on*(5)), or the crash-processing configuration command crashconf (see *crashconf*(1M)).

For additional information on the crash dump sequence, see "Abnormal Shutdowns (System Crashes)" (page 78).

For More Information on HP-UX 11i Version 3

For complete descriptions of what is new/changed in HP-UX 11i version 3, please read *HP-UX 11i Version 3 Release Notes*.

2 HP-UX Virtualization Technologies

Enterprise computing is now heavily focused on enabling you to use your computing resources as efficiently as possible by extracting the most performance from each of your server's resources. Many technologies are available to help you configure HP enterprise servers to suit almost any need, and to adjust your configuration as your computing needs change with minimal disruption to your business.

Collectively, these technologies are known as *virtualization technologies*. They allow you to group and divide your physical computing resources to function as customized virtual resources in order to properly size them for maximum efficiency.

This chapter discusses many of the key virtualization technologies available with HP-UX, what resources they allow you to configure, and how they relate to each other.

Understanding Virtualization Technologies

Because there are now so many ways to configure the resources of your systems and your network, it can be a bit confusing trying to determine which technology (or combination of technologies) best suits your needs.

This document provides general information about each of the technologies and how they relate to each other. Where possible, it provides pointers to where you can find more detailed information.

In addition, HP offers technical consulting services to help you tailor these virtualization technologies to best suit your needs.

Virtualization Technology Categories

Virtualization technologies and their associated products fall into six basic categories based on the number of cores, servers, and operating system instances involved. The matrix in Figure 2-1 "Virtualization Technology Categories Matrix" shows these categories and how they align with processing cores, servers, and operating system instances.

Figure 2-1 Virtualization Technology Categories Matrix

Examples:

- One operating system, running on a single one processor/one core (1P/1C) server is a *standalone* system (or a one-node network).
- Multiple operating systems running on one multiple core server utilizes *partitioning* technology.
- One operating system spanning multiple[2] servers represents *clustering* technology

The following sections discuss each of the virtualization categories and describe which HP-UX technologies to use if you need to configure your system(s) to take advantage of each category.

2. Technically, each server *is* running its own operating system, but they are all *functioning as if they were a single instance,* spanning multiple servers.

Standalone Systems (One Single-Core Server, One Operating System Instance)
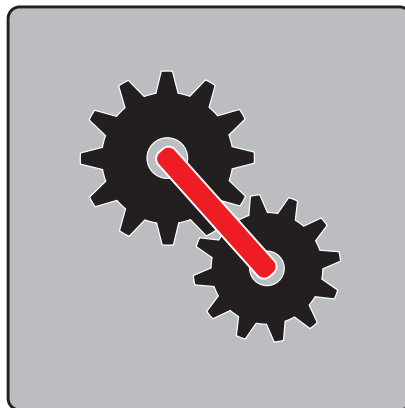
The simplest case, a single 1P/1C server, running a single instance of HP-UX, is a standalone system. It is a computer not connected to a network (or is the sole server on a network).

Today, these are probably dedicated machines, perhaps running highly secure or highly specialized applications. For whatever reason, these are systems in isolation. They are either single user machines or if they support multiple users they require directly connected terminals or modem connections.

By nature, these machines are limited in processing resources and therefore not very flexible in configuration so most of the virtualization technologies are not available to them. However, even these machines can benefit from the following virtualization technologies:

- Integrity Virtual Machines

  Integrity Virtual Machines allow you to share a common set of hardware resources on an HP Integrity Server among multiple operating system instances. The resources are shared on a temporal basis.

- Volume Management using either:
  - The HP Logical Volume Manager (LVM)
  - The VERITAS Volume Manager (VxVM)

- Persistent Device Special Files

  Persistent device special files allow you to have multiple physical paths to mass storage devices. Persistent device special files have virtualized hardware paths (called LUN hardware paths) so that a single persistent device special file can represent *multiple physical paths* to a mass storage device and traffic can be balanced among the physical paths for more efficient data transfer.

- Common Desktop Environment (CDE)

  CDE allows you to configure multiple (virtual) workspaces using the X Window technology. You can group and organize various types of work into these workspaces and name the workspaces for easy identification.

Load Balancing (One Multiple Core Server, One Operating System Instance)
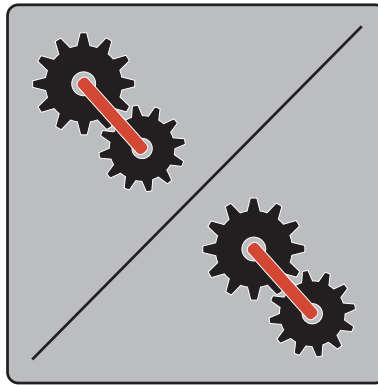
While the HP-UX scheduler does a good job of utilizing multiple cores when they are available, sometimes you need (or want) to override the default scheduling algorithms, for example to dedicate processing resources to an important application.

When your server is equipped with more than one core, the technologies in the load balancing category can help you precisely allocate the resources of multiple cores and other processing resources (memory, disk I/O bandwidth). These technologies include:

- Process Resource Manager (PRM)
- Processor Sets (PSETS)
- Workload Manager (WLM)
- Instant Capacity (iCAP/TiCAP/GiCAP)
- Pay per use (PPU)

Process Resource Manager (PRM)
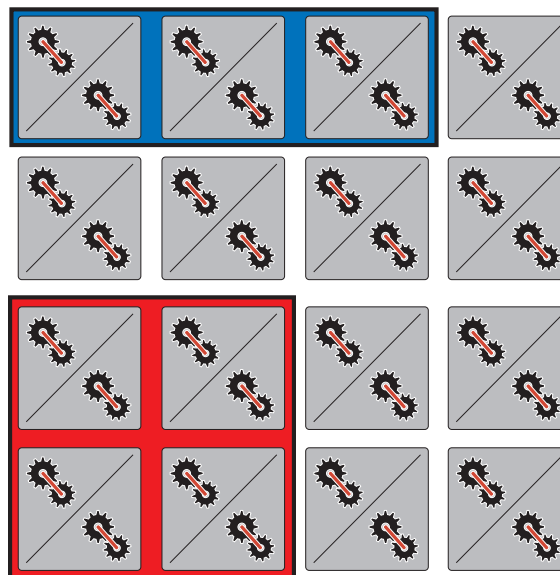
Technology Summary

Process Resource Manager (PRM) is a resource management tool used to control the amount of resources that processes use during peak system load (when the server is at 100% core, 100% memory, or 100% disk bandwidth utilization).

Processes and users are assigned to *PRM Groups*. The PRM Groups are then allocated dedicated cores and memory. This both dedicates resources to the groups and isolates the group's members from using other system resources.

Tools to Administer/Configure

The Process Resource Manager can be configured using the HP System Management Homepage or from a series of PRM management and configuration commands.

How to Obtain

The Process Resource Manager is included in the following operating environments or can be purchased from an HP authorized reseller:
*   BOE - Base Operating Environment

For Further Information

Detailed information about the Process Resource Manager is available on HP's web site from:
*   The document: *HP Process Resource Manager User's Guide* on http://www.hp.com/go/hpux-core-docs.
*   The official Process Resource Manager web site: http://www.hp.com/go/prm

Workload Manager (WLM)

The Process Resource Manager allows you to *manually* dedicate resources to specific applications. However, many factors are likely to change constantly, such as system load, resource utilization, resource needs, and the mix of applications that are running at any given time. If the conditions and needs of your server environment are constantly changing, WLM can continuously monitor and adjust resources to maintain application performance and business goals.

Technology Summary

Workload Manager (WLM) is an automatic resource management tool used for goal-based workload management. A workload is a group of processes that are treated as a single unit for the purposes of resource allocation. For example, a database application that consists of multiple cooperating processes could be considered a workload.

WLM provides automatic resource allocation and application performance management through the use of prioritized *Service Level Objectives* (SLOs). Multiple prioritized workloads can be managed dynamically on a single server based on their reported performance levels.

WLM manages workloads as defined in a configuration file. You assign applications and users to workload groups. WLM automatically allocates processing core resources to achieve the desired SLO. WLM can manage real memory and disk bandwidth but not in response to SLOs. With real memory, WLM allows you to specify lower and upper limits on the amount of memory a workload receives. You can statically assign disk bandwidth shares. If multiple users or applications within a workload are competing for resources, standard HP-UX resource management determines the resource allocation.

Tools to Administer/Configure

The Workload Manager can be configured manually using a configuration file and WLM commands, interactively using the WLM configuration wizard, or by using the WLM graphical user interface.

> 📝 **NOTE:** WLM manages workloads on *individual servers*. To manage workloads on *multiple servers*, install and configure WLM on each of the servers.
>
> WLM can be integrated with HP Serviceguard by storing the WLM configuration file in a file system shared by all nodes in the cluster and then activating the configuration on each node independently.
>
> Alternatively, you can use the HP Integrity Essentials Global Workload Manager (gWLM) to manage workloads across multiple servers. gWLM allows you to define resource-sharing policies that can be used across the servers.

How to Obtain

The Workload Manager is available as a standalone product that can be purchased from an HP authorized reseller or is available in the following operating environments:

- DC-OE - Data Center Operating Environment
- VS-OE - Virtual Server Operating Environment
- MC-OE - Mission Critical Operating Environment

For Further Information

Detailed information about the Workload Manager is available on HP's web site at:

- *HP-UX Workload Manager User's Guide*
- The official Workload Manager web site: http://www.hp.com/go/wlm
- The official HP Integrity Essentials Global Workload Manager web site: http://www.hp.com/go/gwlm

Processor Sets (PSETS)

Processor Sets, a key technology underlying the Process Resource Manager and the Workload Manager, can also be used on its own.

Technology Summary

A *processor set* represents a set of cores grouped together as an independent scheduling domain for use by the HP-UX scheduler for exclusive access to applications assigned to that processor set. Processor Sets allow you to isolate compute intensive or high priority applications from other processes running on a server giving you greater control over the scheduling of critical processes. Processor Sets may be created and re-configured dynamically by users who have the appropriate privileges.

The Processor Sets technology can be used alone, but is often used in conjunction with the Process Resource Manager (PRM).

Tools to Administer/Configure

The `psrset` command creates and manages processor sets.

How to Obtain

The PSET technology is included in HP-UX 11i version 3.

For Further Information

See the following documents (available on the Business Support Center) for further information about PSETS:

- *psrset*(1M) manpage
- *Processor Sets Product Note*

- *Instant Capacity Compatibility with Processor Sets*
- *VSE Concepts and Terminology*

Instant Capacity (iCAP/TiCAP/GiCAP)
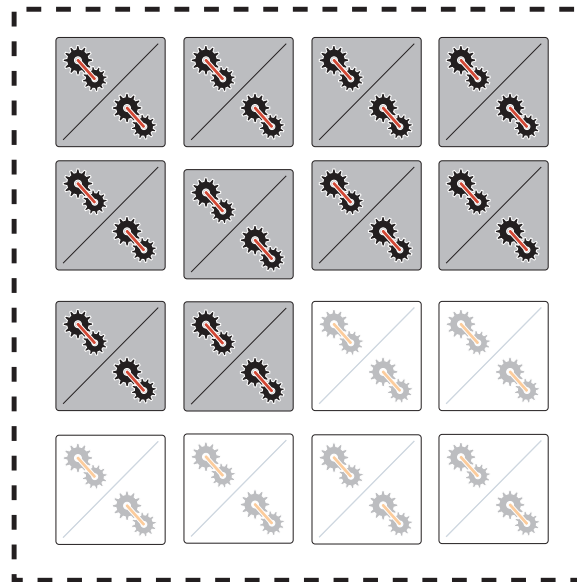
Technology Summary

Instant Capacity technology allows you to purchase at a significantly reduced price: cores, memory, and cell boards that you do not yet own (and cannot yet use). These components, known as *iCAP components* or *components without usage rights*, are standing by, ready for activation when you need them. You then purchase usage rights for some or all of these components and are given the ability through the use of a special codeword to activate (temporarily or permanently) these server components. This gives you the ability to quickly handle peak or unexpected demands for server resources as long as the system remains in compliance with the Instant Capacity contract.

There are three types of Instant Capacity:

iCAP You purchase (for a small price) memory, processors, or cell boards for which you do not yet have usage rights. When you need these resources, you purchase usage rights for them and receive codewords to activate them. From that point on, you own those resources.

TiCAP Similar to iCAP, but the usage rights that you purchase are only temporary (TiCAP is currently sold in 30-day increments). Your usage of the iCAP resources is measured in 30-minute increments and when you've used up the purchased time, the iCAP resources are deactivated until you activate them again by purchasing another right-to-use codeword.

GiCAP GiCAP allows users to share rights for Instant Capacity components within a group of servers, and provides "pooled" temporary capacity across the group.

Tools to Administer/Configure

iCAP Commands

A series of iCAP commands allows you to interact with HP-UX to adjust the processing capacity of a server containing iCAP hardware. The commands allow you to enter purchased codewords to activate your rights to use the stand-by processors, or to decrease the processing resources you are using in order to save money.

The commands to administer iCAP resources are:

`icapmanage`	Global Instant Capacity (GiCAP) management commands for GiCAP groups.
`icapmodify`	icapmodify - Activate and deactivate cores. Specify system contact e-mail address. Change Instant Capacity (iCAP) configuration information. Specify Instant Capacity from e-mail address. Specify system identifier. Specify temporary capacity warning period. Apply iCAP codewords.
`icapstatus`	Displays Instant Capacity status and configuration information, counts, status, and allocation of Instant Capacity components (cores, memory, and cells) for an Instant Capacity system.
`icapnotify`	Tests e-mail connectivity to HP for Instant Capacity (iCAP) systems. Requests a confirmation response E-mail from HP. Turns configuration change notification and asset reporting on or off.

HP System Management Homepage

The iCAP manager (iCAPMgr) version 4.1 runs under the HP System Management Homepage (HP SMH), allowing it to manage iCAP on a single system.

After installing the iCAPMgr bundle, run the following command to configure the iCAPMgr product for HP SMH.

`/opt/icod/bin/icapsmhconfig -a`

How to Obtain

The capability to activate instant capacity components is part of the HP-UX Foundation Operating Environment (and is therefore available in all operating environments that build upon the Foundation Operating Environment). However, in order to use the instant capacity technologies, you must have purchased and installed components without usage rights to activate.

For Further Information

You can find comprehensive information about the various iCAP technologies in the following documents:

- *HP Instant Capacity User's Guide*—See the latest version of this guide at http://www.hp.com/go/hpux-core-docs.
- Detailed information on iCAP, TiCAP, and GiCAP is available at http://www.hp.com/go/icap.
- Information on the iCAP commands and their options are available in the following iCAP manpages:

icap(5)	iCAP overview manpage
icapd(1M)	Instant Capacity Daemon
icapmanage(1M)	Global Instant Capacity management commands for GiCAP groups
icapmodify(1M)	Activate/Deactivate Cores
icapnotify(1M)	Test e-mail connectivity to HP for Instant Capacity systems
icapstatus(1M)	Display Instant Capacity status information

Pay per use (PPU)
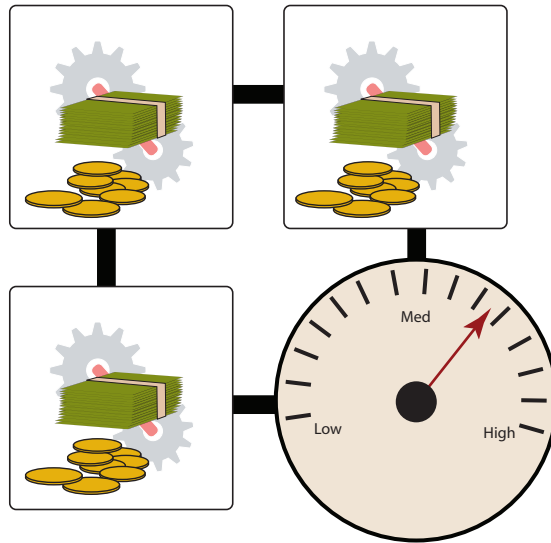
Pay per use (PPU) is a product that enables you to "pay as you go". Pay per use is a pricing model in which you are charged for your actual usage of computing resources.

Technology Summary

With Pay per use, you treat computer processing capacity as you would electricity, water, or other utilities. You acquire a specific hardware platform with a number of cores and are charged for the actual usage based on one of the following HP contractual agreements:

- Core percent utilization (percent core)
- Number of active cores (active core)

Tools to Administer/Configure

The Pay per use program uses a dedicated system known as the Utility Meter that is set up by HP.

For servers you plan to use with Pay per use, you must set up a software agent (the PPU Agent) on every partition (both hardware and software partitions). The PPU Agent reports usage information to the Utility Meter which in turn communicates to HP your actual usage.

A single Utility Meter can service up to 100 servers or partitions running HP-UX 11i or Windows Server 2003.

Most of the configuration of the Pay per use components is done through the `ppuconfig` command. `ppuconfig` can:

- Display current settings
- Test communications to HP
- Set usage caps
- Specify hostnames and system identifiers for use in reporting usage to HP and for your use when viewing usage information on the portal
- Specify which utility meter a server or partition should use to report its usage information

How to Obtain

Pay per use is shipped with the HP-UX Foundation Operating Environment and is customer configurable on all HP-UX servers. However, you must enable the Pay per use system to communicate with a *Utility Meter* (a special system which must be set up by HP). A single Utility Meter can support multiple Pay per use systems (currently up to 100), so you do not need a Utility Meter system for every Pay per use system.

For Further Information

You can find comprehensive information about Pay per use and the Utility Meter in the following documents:

- *Utility Meter User Guide*—See the latest version of this guide at http://www.hp.com/go/hpux-core-docs.
- *HP Pay per use (PPU) User's Guide*—See the latest version of this guide at http://www.hp.com/go/hpux-core-docs.
- You can get additional information and access the sign-in page for the Pay per use Portal at http://www.hp.com/go/payperuse.

Clustering (Multiple Servers, One Operating System)
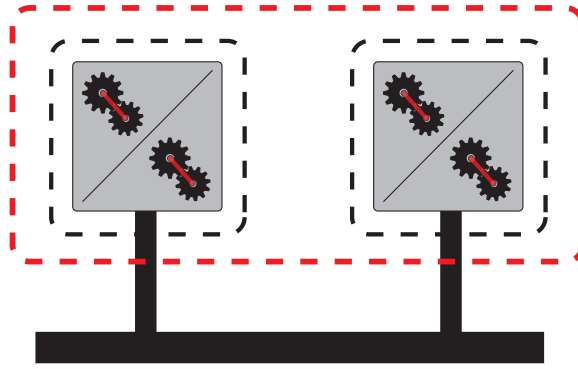
Clustering technologies enable multiple servers to work in unison to present the appearance of a single computing environment. Technically, each server *is* running its own operating system, but they work together as if they were one.

Examples of clustering technologies include:
- Serviceguard Clusters
- Extended Campus / Extended Distance Clusters
- Metropolitan Clusters
- Continental Clusters

Serviceguard

A Serviceguard cluster is a networked group of HP Integrity or HP 9000 servers (known to Serviceguard as *nodes*) having sufficient redundancy of software and hardware that a single point of failure will not significantly disrupt service. The ability to continue operating in spite of a software or hardware failure makes Serviceguard clusters *highly available*.

Technology Summary

When you configure a Serviceguard cluster, you set up as many redundancies as possible, both in hardware and software. To provide the greatest degree of availability, Serviceguard typically works in conjunction with other high availability products such as:
- MirrorDisk/UX
- VERITAS Volume Manager (VxVM)
- Disk arrays using various RAID levels
- HP Powertrust power supplies (Uninterruptable Power Supplies)

With Serviceguard, you configure **packages** (collections of HP-UX and application processes) on the nodes of your cluster. There are three types of Serviceguard packages:

Failover	The most common type of Serviceguard package is the **Failover Package**. A Failover Package is installed on multiple nodes in a Serviceguard cluster, but runs only on one node at a time. The node running the primary copy of a Failover package is called the **primary node**. Should the primary node fail, a copy of the package on another node (known as an **adoptive node**) takes over with little or no interruption in service until the primary node can be restored to service. Multiple adoptive nodes can be defined for extra redundancy.
Multi-node	**Multi-node Packages** run (simultaneously) on one or more nodes in a Serviceguard cluster. A multi-node package can be configured to remain running as long as at least one copy of the package remains running. Multi-node packages do not fail over. HP supports them only for specific applications.

System Multi-node	System Multi-node Packages run (simultaneously) on all nodes in Serviceguard cluster. They remain running as long as all copies of the package contents remain running. All copies of System Multi-node packages halt if even one of the copies in the cluster halts. System Multi-node packages do not fail over. HP supports them only for specific applications.

Serviceguard clusters use TCP/IP networking services for reliable communication among nodes in the cluster. This includes **heartbeat messages** (signals from each functioning node that are central to the operation of the cluster).

Tools to Administer/Configure

Serviceguard clusters can be configured and managed using Serviceguard Manager or using Serviceguard commands, including:

`cmviewcl`	View information about a high availability cluster
`cmrunnode`	Run a node in a high availability cluster
`cmhaltnode`	Halt a node in a high availability cluster
`cmruncl`	Run a high availability cluster
`cmhaltcl`	Halt a high availability cluster

How to Obtain

Serviceguard is included with the Mission Critical Operating Environment and with various specialized mission-critical and virtualization bundles. Consult your HP Sales Representative or an HP authorized reseller for more information.

For Further Information

The following Serviceguard document contains detailed information on configuring Serviceguard clusters:

- *Managing Serviceguard, Thirteenth Edition, B3936-90105*

Other Cluster Types

Serviceguard clusters can be located in a single room or single building; additionally, they can participate in clusters spread over vast distances. The distance between nodes largely determines which form of clustering you are using.

More information about the following cluster types is available in the Business Support Center at http://www.hp.com/go/hpux-serviceguard-docs.

Extended Distance Clusters

An *extended distance cluster* contains alternate nodes located in different data centers separated by some distance. Extended distance clusters are connected using a high speed cable that guarantees network access between the nodes as long as all guidelines for disaster tolerant architecture are followed. The maximum distance between nodes in an extended distance cluster is set by the limits of the data replication and networking technology used.

Sometimes referred to as an *extended campus cluster*, this cluster type services an area approximately the size of a college or corporate campus.

Metropolitan Clusters

A metropolitan cluster is geographically dispersed within the confines of a metropolitan area requiring right-of-way to lay cable for redundant network and data replication components.

Sometimes referred to by the shorter term *metrocluster*, this cluster type services an area approximately the size of a town or city, or several closely located cities.

Continental Clusters

A continental cluster is a group of clusters that use routed networks or common carrier networks for data replication and cluster communication to support package failover between separate clusters in different data centers. Continental clusters are often located in different cities or different countries, and can span hundreds or even thousands of miles.

Multiple Boot (One Single-Core Server, Multiple Operating Systems)
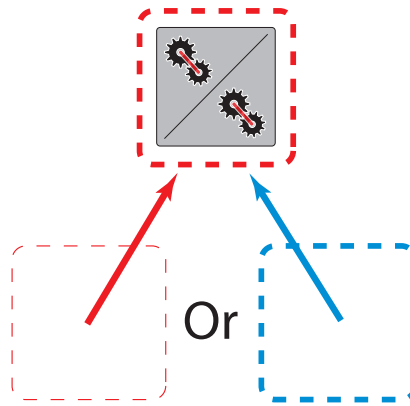
HP Integrity Servers are capable of running multiple types of operating systems (for example, HP-UX 11i, Linux, and Microsoft Windows). Depending on your server model, multiple versions of these operating systems might also be supported.

If you only have a single available core, you cannot have multiple operating systems running simultaneously. However, you can have multiple boot disks (or logical boot volumes), each containing a different operating system, and you can choose which to boot from on any given boot.

On HP 9000 servers, you can also have multiple versions of HP-UX (or other supported operating systems) installed on different disks (or logical boot volumes) and select your desired boot volume at boot time.

NOTE: This multiple boot capability works equally well for servers with more than one active core. It is presented here because this is an available function for servers containing only one active core.

Partitioning (Multiple Operating Systems, One Multi-processor Server)

When you have multiple cores available, more virtualization possibilities become available as well. This area of virtualization technology is known as **partitioning**. There are several types of partitioning available on HP systems, but they fall into two major categories—*hardware partitioning* and *software partitioning*.

Hardware Partitioning

nPartition 1   nPartition 2
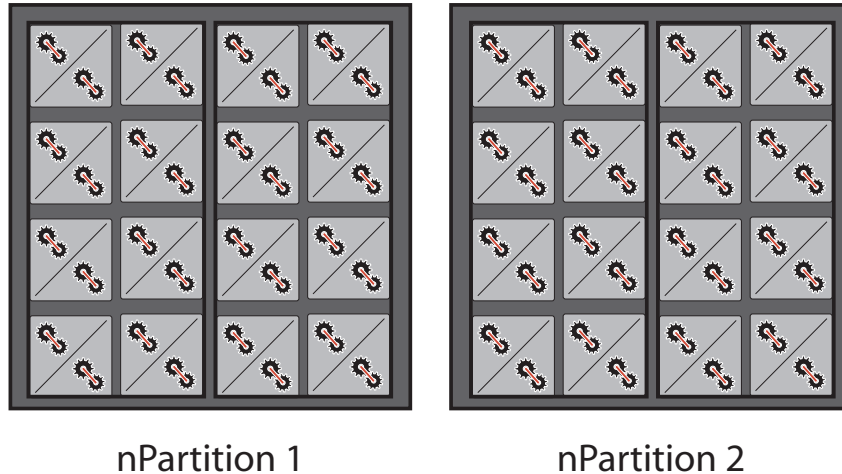
Hardware partitioning is accomplished at the cell board or Superdome 2 blade level, using an HP technology called *nPartitions*.

Implemented on servers that support multiple cell boards and on Superdome 2 blade-based systems, nPartitions isolate (both logically and electrically) multiple operating system instances. That is, cell boards/Superdome 2 blades, cores, I/O components, and memory assigned to one nPartition are available only to operating systems running in that partition.

Should problems occur with an operating system, software, or even hardware in one partition, operating systems and software running in other partitions are unaffected.

Key Features of Hardware Partitioning   Important features of hardware partitioning include:

- Implemented at the cell board / Superdome 2 blade level.
- Provides both functional and electrical isolation.
- Partitions are referred to as nPartitions.
- nPartitions can be further sub-divided using Software Partitioning.
- On servers that support them, nPartitions can run HP-UX, Linux, Microsoft Windows, or all of these operating systems (in separate partitions).

Software Partitioning
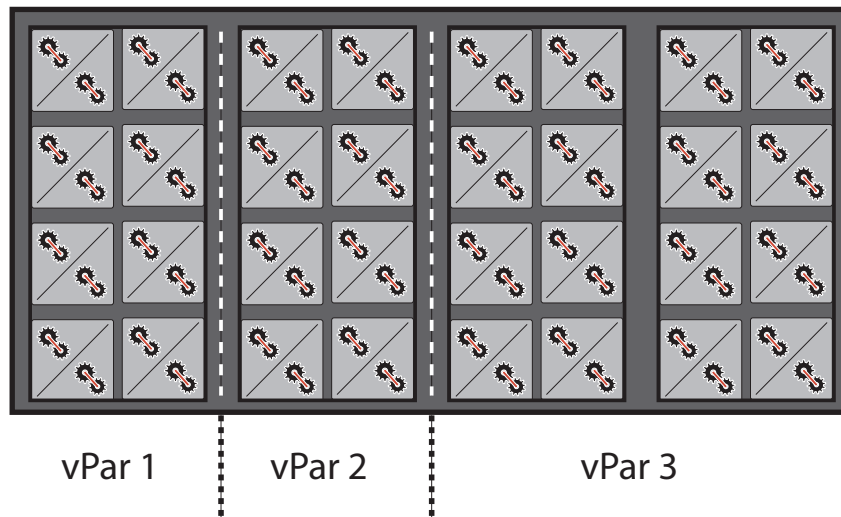
Software partitioning provides for finer grained partitioning than nPartitions. HP offers two products to accomplish software partitioning:

1. *vPars* - Virtual Partitions
2. **Integrity VM** - Integrity Virtual Machines

Using either of these two products, you can partition your server at the processing core level. Integrity VM even allows sub-core-level partitioning.

Key Features of Software Partitioning   Important features of software partitioning include:

- Implemented at the processing core level (for vPars), or by time slices (for Integrity VM)
- Provides functional but *not* electrical isolation
- Partitions are referred to as:
    - *vPars* (if implemented using the Virtual Partitions product)
    - *guest operating systems* (if implemented using the Integrity Virtual Machines product)
- nPartitions can be further subdivided using software partitioning
- vPars only supports instances of HP-UX

Comparing Partitioning Technologies

The following table compares the three partitioning technologies (nPartitions, vPars, and Integrity Virtual Machines):

Table 2-1 A Partitioning Technology Comparison

	nPartitions	vPars	Integrity Virtual Machines
Isolation	Operating System *and Electrical*	Operating System	Operating System
Partition boundaries	Cell Board / Superdome 2 Blade Groups	Cores and Memory Chunks	Time Slices
Partitions configured using...	Parmgr or the Superdome 2 Onboard Administrator (OA)	vPars Monitor (and its partition database) or the Superdome 2 Onboard Administrator (OA)	The `hpvmcreate` command
Partitions are called...	nPartitions or hard partitions	Virtual Partitions	Guest Operating Systems

Combining Partitioning Technologies for Greater Flexibility

By themselves, the various server partitioning technologies give you a very flexible compute environment, but you can combine them for even greater flexibility and control.

The following illustration shows how nPartitions, vPars, and Integrity Virtual Machines can be combined on a single server to support a large amount of work.

Figure 2-2 The Virtualization Technology Stack
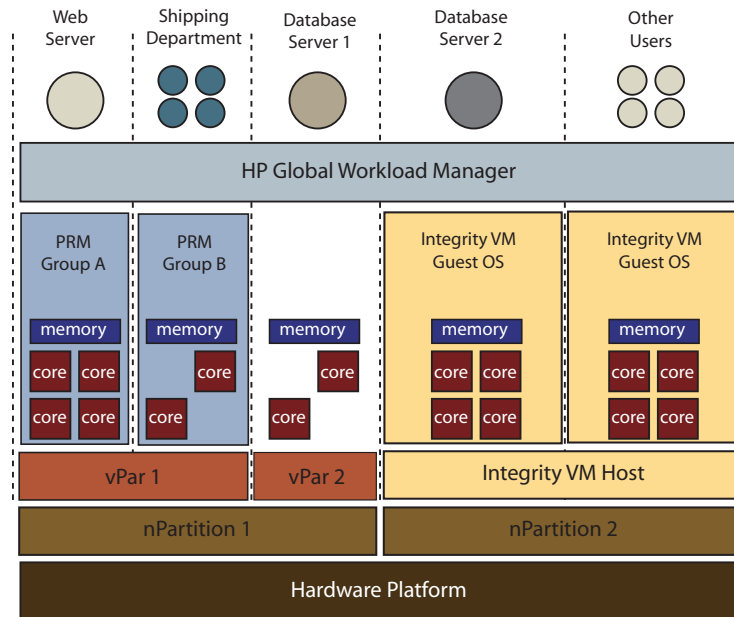
Networking (Multiple Operating Systems, Multiple Servers)
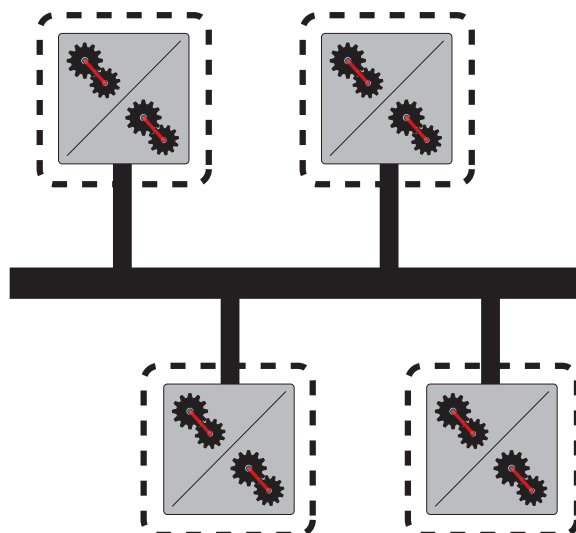
Unless the physical servers are each running in standalone mode — "Standalone Systems (One Single-Core Server, One Operating System Instance)") — when you have multiple operating systems running on an equal or lesser number of physical servers, the virtualization technologies that can be used to flexibly use their resources fall into the *networking* category of virtualization technologies.

There are many well established applications, protocols, and networking technologies. They work between multiple HP-UX operating systems, and between HP-UX and other operating systems as well. Table 2-2 lists a few common networking technologies.

Table 2-2 Common Networking Technologies

Technology	Used For...
NFS	Mounting remote file systems
ftp, rcp	Copying files between networked operating systems
sftp, scp	Securely copying files between networked operating systems
telnet, rlogin, ssh	Logging into remote operating systems
HTTP	Web browser access to remote files

Combining Virtualization Technologies

All of the HP virtualization technologies work together to give you even greater flexibility.

For example, you can use WLM within a whole server that can be clustered in an HP *Serviceguard* cluster, *extended campus cluster*, *metropolitan cluster*, or *continental cluster* (high availability clusters of various geographical separation).

You can also use WLM on an Integrity VM host and within any individual Integrity VM (guest). You can use WLM within nPartitions and virtual partitions as well as across partitions. Many such combinations are possible.

3 Major Components of HP-UX

Operating systems are complex pieces of software designed to control the many resources of a computer so that many users and processes can use those resources cooperatively and efficiently.

HP-UX 11i, a version of the UNIX operating system, is comprised of many components that work together to control the resources of HP Integrity Servers, HP 9000 servers, and others.

This chapter describes the main components of HP-UX, what they do, and how they relate to each other.

The HP-UX Kernel

The *kernel* is the central part of the HP-UX operating system; everything else around it depends on and interacts with. When you start up (boot) HP-UX, it is the kernel that is copied from disk into memory by the *boot loader* and initiated.

The kernel is made up of many subcomponents. In the case of an HP-UX kernel, these subcomponents are called kernel modules and kernel tunables.

Kernel Modules

Kernel *modules* are sections of kernel code dedicated to specific purposes, such as memory management, class drivers, or interface drivers.

Adding/Removing Kernel Modules

Many kernel modules are configured into the kernel when it is built and are always present in the kernel whenever it is running. Other modules are added when they are needed, and some can be removed when not needed.

The kernel modules that are only required some of the time fall into two categories:
- Modules that can be added to, or removed from, the kernel *while it is running*
- Modules that *require a reboot* to be added to, or removed from, the kernel

You can use the `Modules` tool in the `Kernel Configuration` toolbox in the System Management Homepage, or the `kcmodule` command from a shell command line, to:
- view the current list of modules in the kernel,
- view the state of specific modules in the kernel,
- determine whether modules require a reboot to be added to or removed from the kernel,
- change the state of a module, either currently (if supported by the module) or at the next boot.

For details about kernel configuration, see: "Configuring the Kernel" in *HP-UX System Administrator's Guide: Configuration Management*.

Kernel Tunables

Kernel tunables are settings that determine things like how many processes can simultaneously be active or how much memory can be allocated for certain data structures within the kernel. For more information on kernel tunables, what they are, and how to adjust them, see the *HP-UX System Administrator's Guide: Configuration Management*.

The HP-UX Directory Structure

HP-UX 11i, like all versions of UNIX and many other operating systems, is based on a hierarchical directory structure that contains all of the operating system directories and files, as well as all user and application files.

The directory structure can be contained entirely within a single *file system*, but it is usually comprised of multiple file systems that attach to the *root file system* at special directories known as *mount points*.

The top of the tree is known as the *root directory* and is represented by the *directory path* "/".

All other directories and files reside under the root directory. Each layer of the directory tree is represented in directory path names (pathnames) by adding its name to the end of the path, with each layer separated by a slash character ("/"). See the following examples:

Example 3-1 Directory Path Name Nomenclature

/	The root directory
/usr	The `usr` directory within the root directory
/home/guest27	The `guest27` account within the `home` directory within the root directory
/usr/share/man/man1.Z/cat.1	`cat.1` is a file within the `man1.Z` directory, within the `man` directory, within the `share` directory, within the `usr` directory, within the root directory (/). It just happens to be the source file for the *cat*(1) manpage.

Figure 3-1 Directory Tree Example
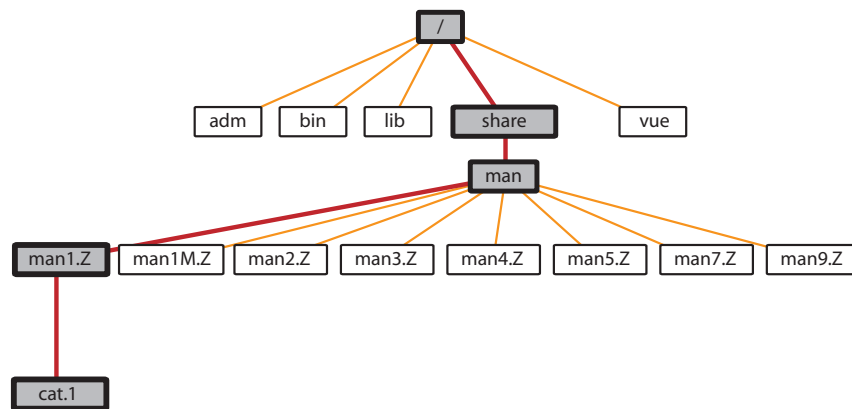
Key HP-UX Directories

This section describes many of the key directories in the HP-UX directory structure and what they are used for.

/dev	Contains *device special files*. Though they appear in the directory tree like regular disk files, device special files are associated with physical devices or pseudodevices. They are portals to the devices and device drivers they are associated with.
	Device special files are written to or read from by software applications or operating system components to access data on the

devices, get status of the devices, or to control those devices in some other way.

There are two classes of device special files (based on how they are used to transfer data):

- *block special files* (transfer data in blocks by means of the system's normal buffering mechanism, and are primarily used to mount file systems)
- *character special files* (transfer data in an unbuffered stream, and are used for nearly everything else)

In HP-UX 11i version 3 there are also three types of device special files (based on how they refer to their associated devices):

- *legacy* device special files reference their corresponding devices by *the hardware paths to those devices*. Legacy device special files are the type of device special files that have always been part of HP-UX. They are still supported in HP-UX 11i version 3 and function as they always have.
- *persistent* device special files reference their corresponding devices based on a world-wide unique identifier that is built into, or associated with, the device. Because persistent device special files are not dependent on specific hardware paths, they allow for multiple hardware paths to be represented by *a single device special file*. This opens up many new HP-UX features in the areas of I/O performance, reliability, and flexibility.
- Device special files for pseudodevices. The majority of device special files are not directly associated with actual hardware devices but are instead used to access *pseudodevices* that allow HP-UX to introduce a virtualization layer between processes and disk storage (like those for LVM or VxVM), simulate hardware devices like terminals (ptys), or provide useful abstractions like `/dev/null` (a device file often referred to as the "bit bucket", used to receive and discard output that is not needed) There are many pseudo-devices in HP-UX, used for a variety of purposes. Most of these are described in the manpages of Section 7 of the *HP-UX Reference Manual*.

`/etc`	The `/etc` directory stores system-wide configuration files including files required for:

- customizing boot and shutdown activities
- networking configuration
- configuring which file systems to mount
- defining users and groups
- defining logical volumes

The previous list is only partial. For compatibility reasons, `/etc` also contains a number of symbolic links to commands in other directories (commands that once resided in the `/etc` directory).

`/etc/opt/product`	Some optional products, when added to a server, create subdirectories under the directory `/etc/opt` to store product specific configuration information.
`/home`	The `/home` directory is the default location for the home directories for user accounts. For example: If a user "Thomas" with user name `thomas` is added to an HP-UX server (and if the home directory

	location is left at its default), the home (login) directory for Thomas will be /home/thomas.
/opt	The /opt directory contains application software and other system components of the HP-UX 11i operating system that are not considered to be part of the minimum-level installation needed for a viable system.
/sbin	Contains statically linked versions of critical programs needed at boot time or when important shared libraries have become corrupted. /sbin is available when the system is not in multiuser mode and before the /usr file system has been mounted.
/stand	The /stand directory is part of the root file system (the very first file system that is mounted during the system startup sequence). /stand is a special directory that is used by the boot loader to read the kernel file from disk and start it running.
/tmp	The /tmp directory is a temporary scratch directory, particularly used before the file system mounted at /var is available during system startup.
	Some installations delete the contents of /tmp at every boot; others choose to manually maintain /tmp. Either way, do not store files in /tmp for long periods of time that would be detrimental to lose. See also /var/tmp.
/usr	Contains many HP-UX resources that are typically not needed to boot the system.
	The files in the / and /usr directories are intended to be for read-only use so that these directories can, if necessary, be mounted from a network resource in read-only mode. When HP-UX components and applications need to update files, such as log files, those files should be placed in a writable file system such as /var.
/usr/bin	Contains *dynamically linked* non-essential commands and programs used by most users once the system is in multiuser mode. The commands and programs in /usr/bin are *not needed in order to boot the system*. These commands are available only when the file system that contains them is mounted.
/usr/lib	Contains shared libraries used by the binaries in /usr/bin.
/usr/sbin	Contains *dynamically linked* non-essential commands used to administer the system once it is in multiuser mode. These commands are available only when the file system that contains them is mounted.
/var	"var" stands for variable. This directory (usually associated with a mountable file system) contains variable data; that is, files that need to change as the system is running (for example, log files that need to be written to).
/var/opt/*product*	Some optional products, when added to a server, create subdirectories under the directory /var/opt to store product specific information.
/var/tmp	The /var/tmp directory is a temporary scratch directory, preferred over the /tmp directory after the system has been booted (primarily because /var is the intended location for transient data and temporary data is by definition transient).
	As with the /tmp directory, /var/tmp should only be used for temporary files (files that are not intended to be stored for long periods of time and files that are not essential). Unlike /tmp, /var/

`tmp` files are not usually removed during system startup, but when (or whether) files in either of the `tmp` directories are removed is configurable and installation specific.

Storage on HP-UX

Of all the resources managed by any operating system, arguably the most important is storage. **Storage** is a generic term referring to devices that store data. Storage can take many forms, including:

- Physical disk drives locally attached to a server:
    - SCSI hardware protocol disks
    - Fibre Channel disks
    - USB 2.0 disks
- Drive enclosures containing multiple physical disks locally attached to a server
- Disk arrays (drive enclosures as with the previous item but with a disk controller added to the enclosure to manage disks within the enclosure) attached locally to a server (sometimes called JBODs—for **J**ust a **B**unch **O**f **D**isks—and sometimes called RAIDs—for **R**edundant **A**rrays of **I**nexpensive **D**isks)
- Storage Area Networks (SANs), physical disks or disk arrays as above, but configured and accessed through a special high speed network. SAN storage can be physically near the servers accessing it or physically distant for disaster protection. Storage Area Networks work at the block I/O layer, below the file system layer.
- Network Attached Storage (NAS), an alternate network storage solution, working at the file system layer using standard network protocols (NFS, CIFS)
- Off line storage or removable media. Data are stored on:
    - Tapes (DLT, DDS, Reel, and other tape formats)
    - Optical Media (CD, Magneto-optical, DVD-ROM, and other optical formats)
    - Removable disk drives

Offline storage is often used for off site storage of data (usually backups) for purposes of disaster recovery.

Storage Uses

In the HP-UX operating system, storage can be used in many ways; among them:

- Files and Directories stored locally in file systems
- Databases stored on disk volumes in raw form (managed by the database application rather than by HP-UX) for speed
- Swap space (used by HP-UX for paging purposes)
- Dump space (used to capture the state of HP-UX following a system panic or other significant event)

How Storage is Organized

Like networking and many other subsystems in the computing world, storage is comprised of many layers, from the physical devices to the applications that read and write data to those devices. Collectively these layers are known as the *storage stack*.

The following sections cover the various components of the HP-UX storage stack.

Physical Storage Devices

At the lowest level of the storage stack are the physical devices that store and retrieve data. Usually these are disk drives, but they can be other storage devices as well, including:

- DLT tape drives / libraries
- Magneto-optical drives / libraries
- DDS tapes

Disk drives can be:

- Individual drives
- Drive enclosures (groups of multiple disk drives that are treated as individual drives)
- Disk Arrays (like drive enclosures but with an added disk controller for local intelligence in managing the contained storage (for example, RAIDs)
- SAN - Storage Area Networks (physical drives attached to a dedicated network)
- NAS - Network Attached Storage (storage attached to dedicated servers accessed through standard network file system protocols)

Individual disk drives (whether standalone or in an array or enclosure) are often referred to as **LUNs**. The term "LUN" stands for "Logical Unit", and while it is often associated with a physical disk drive (drive unit) within a larger array device, LUNs can point to other (logically defined) subsets of a larger device.

Volume Managers

Physical disk drives can be used in a standalone mode; that is, they can be partitioned, formatted with a file system, used for paging, or used by database applications as units of storage. However, physical disk drives are usually grouped into larger pools of space that can then be divided into logical storage containers. These containers (called volumes or logical volumes, depending on which volume manager you are using) are not required to respect the boundaries of the physical drives in the group. That is, they can span multiple physical devices.

Volume managers enable you to divide and allocate these pools of space into logical storage containers.

The pools of space are called *volume groups* in the Logical Volume Manager (LVM) and *disk groups* in the VERITAS Volume Manager (VxVM).

The logical storage containers are referred to as *logical volumes* in LVM or simply *volumes* in VxVM. To applications, file systems, and databases, these volumes appear to be physical disks and are treated as such.

HP-UX 11i version 3 supports the following volume managers:

LVM   The Logical Volume Manager (LVM) is detailed in *HP-UX System Administrator's Guide: Logical Volume Management*. LVM is the default volume manager for HP-UX 11i.

VxVM   The VERITAS Volume Manager (VxVM) has many features, some of which are not available with LVM or MirrorDisk/UX (the companion product to LVM that allows you to mirror data onto multiple physical disks).

The version of VxVM that ships with HP-UX is a base version containing a subset of the features offered in the full version (which requires an additional license). For complete information about which features are included with the base version and the full version of VxVM see the *VERITAS Volume Manager Releases Notes* corresponding to the version of the VERITAS Volume Manager you are using.

NOTE: The pools of space are called *volume groups* in the Logical Volume Manager and *disk groups* in the VERITAS Volume Manager.

Both volume managers can co-exist on a server. Each volume manager keeps track of which disks it is controlling and any given physical disk can only be a controlled by one volume manager at a time. The utility `vxvmconvert` can convert an LVM physical volume to a VxVM disk if you want to migrate a disk from LVM to VxVM for greater configuration flexibility.

Selecting a Volume Manager

With HP-UX 11i version 3, there are two volume managers to choose from:
- The HP Logical Volume Manager (LVM)
- The VERITAS Volume Manager (VxVM)

You can use both simultaneously (on different physical disks), but usually you will choose one or the other and use it exclusively. Table 3-1 provides a comparison of the two volume managers to help you determine which will best suit your needs.

Table 3-1 Volume Manager Feature and Terminology Comparison

Feature	HP Logical Volume Manager (LVM)	VERITAS Volume Manager (VxVM)
Obtaining...	LVM is included in the HP-UX 11i version 3 Foundation Operating Environment	Check HP-UX 11i version 3 Release Notes to see if VERITAS Volume Manager is included with your chosen Operating Environment
Physical Disks are called...	Physical Volumes	VERITAS Volume Manager Disks
Groups of Physical Disks are called...	Volume Groups	Disk Groups
Logical Storage Containers allocated from the Volume/Disk Groups are called...	Logical Volumes	Volumes
User data within (Logical) Volumes is allocated using...	Physical Extents (fixed size chunks of disk space, the size of which must be the same for all physical volumes in a given volume group)	Subdisks (arbitrary in size, subdisks represent a set of disk blocks, allocated from VERITAS Volume Manager Disks in a disk group)
Volume and Volume Group configuration information is stored in...	a special reserved area at the beginning of each physical volume.	a special area known as a "private region" of each physical volume.
Mirrors (copies of data)	You need to add the MirrorDisk/UX product to your system to support mirroring. Mirrordisk/UX supports up to three copies of data if you are using LVM with Version 1 volume groups, and up to six copies of data if you are using LVM with Version 2 volume groups.	Mirrors consist of **plexes**, each of which is a copy of the volume being mirrored. The base version of the VERITAS Volume Manager allows you to mirror your root file system (only). Like LVM, VERITAS Volume Manager requires an additional license to support mirroring (beyond the mirroring of your root file system), but with the additional license VxVM supports up to 32 copies of your data.

NOTE: VERITAS Volume Manager is available in several versions as of the Release of HP-UX 11i version 3. Values and features shown are for VxVM Version 4.1. Consult the VERITAS Volume Manager Release Notes of the appropriate version for that version's specifications.

The VERITAS Volume Manager has two licensing levels, base and full. Unless stated otherwise, the features listed in the previous table are for the base level license. See the VERITAS Volume Manager documentation for the additional features supported with the full license level.

Volume Groups

When using a volume manager, the first step is to group physical drives into pools of disk space called:

- *disk groups*, if you are using the VERITAS Volume Manager (VxVM)

or

- *volume groups*, if you are using the Logical Volume Manager (LVM)

NOTE: The individual disks in the disk/volume groups are called:
- **VM disks**, if you are using the VERITAS Volume Manager (VxVM).

or

- **physical volumes**, if you are using the Logical Volume Manager (LVM)

(Logical) Volumes

Once you have grouped physical disk drives into disk/volume groups, the collective space can be divided into logical storage containers that can be smaller or larger than any individual drive in the group. These logical storage containers are called:

- *volumes*, if you are using the VERITAS Volume Manager (VxVM)

or

- *logical volumes*, if you are using the Logical Volume Manager (LVM)

Once defined, individual volumes or logical volumes can be used for:

- booting (they can contain bootstrap loaders, offline diagnostics, and other software needed for server administration purposes)
- file systems (traditional file storage)
- swap space (virtual memory / paging)
- dump space (memory dumps following a system panic)
- raw disk access (for use by database applications, and other applications that manage their own disk space)

Logical volumes can be made larger or smaller as needed (if the data they contain support these operations).

Figure 3-2 Logical Volumes can be Resized
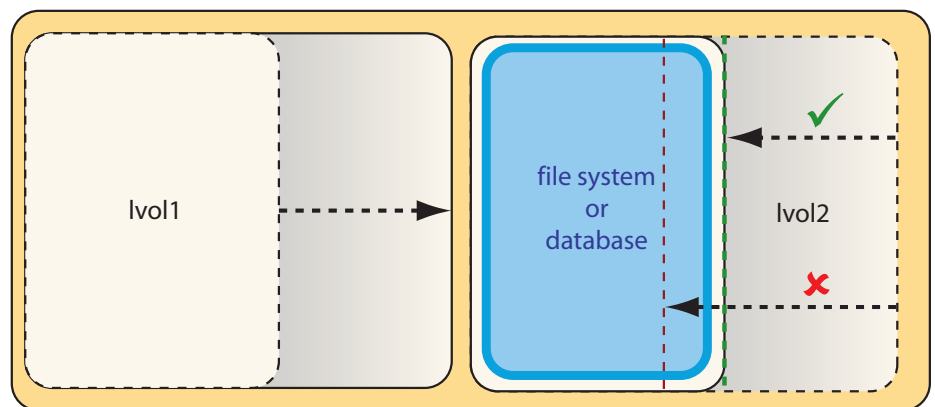
File Systems

If you are not using a volume (or physical disk) for swap space or raw disk access (for example disk space managed by a database application), you are probably using it for file storage.

The directories, files, and file data are usually distributed throughout a volume or disk drive. In the same way a card catalog allows you to locate a specific book in a large library, the primary function of a file system is to maintain the pointers to the files stored in a volume or on a physical disk so that those files can be later retrieved. These are not merely pointers indicating which file is in which directory; these are the low level pointers and other vital information, for example:

- which disk blocks belong to which file
- which disk blocks are currently unused (so that specific disk blocks are not simultaneously used for more than one purpose and order is maintained)
- linked lists of directory navigation information

File systems also have other important functions, for example maintaining ownership and access privilege information so that HP-UX security functions can ensure that only those authorized to access a file or directory can do so.

As with volume managers, HP-UX offers you several choices of file system types to choose from. Specifically:

HFS   The HP proprietary High-Performance File System supports files and file system sizes up to 128GB.

VxFS   The VERITAS File System Version 4.1 supports file sizes up to 16TB and file system sizes up to 40TB. VxFS is also known as OnlineJFS/JFS 4.1.

In addition to the previous file system types, HP-UX 11i version 3 supports:

MEMFS   This memory-based file system provides fast access to temporary files by storing them in memory.

CDFS   The CD file system allows you to read and write compact disc media using ISSO-9660 format (either with or without the Rockridge Extensions).

The `ISOIMAGE-ENH` bundle provides support for mounting and unmounting ISO Image files. This bundle is newly delivered in March, 2009 as an optional product for all Operating Environments.

Efficient Data Access

If your operations depend on high performance disk I/O, in addition to ensuring that you are using high-speed interfaces (for example, Fibre Channel), consider the following:

- Disk Striping
- Distributing Disk Access
- File System Type
- Establishing Multiple Paths to a Device (for efficiency)
- Disk Mirroring (for performance)

Disk Striping

Disk striping spreads data over multiple physical devices in such a way that successive writes occur on different devices. In this way, the second chunk of data to be written does not have to wait for the device writing the first chunk to finish. In essence, if you have n devices striped together, then you can write n chunks of data simultaneously (or nearly simultaneously) without having to wait for devices to become ready for subsequent data.

Striping can be performed at the device level if you are using a disk array, RAID array, or other hardware that supports RAID operations. Other types of disks striping can be performed by LVM or the VERITAS Volume Manager (VxVM). For information about performing striping using one of the volume managers, see *lvcreate*(1M) or *vxassist*(1M).

You can specify the size of data chunks to use with striping.

Hardware data striping is accomplished by using certain RAID configurations. Striping related RAID levels commonly used on HP disk arrays that support RAID are:

RAID 0   Striping using data blocks with no parity disks in the stripe. For performance reasons, data blocks are usually a multiple of 512 bytes (the physical sector size of most hard disks).

An important consideration when using RAID 0 is the number of disks in the stripe. The more disks you have in the stripe, the greater the chance of one of them failing. With no parity disk included in the stripe, missing data cannot be reconstructed and therefore must be restored from a backup or other source.

RAID 5   Striping using data blocks with parity information evenly distributed across the devices in the stripe set. The parity information can be used to reconstruct the missing data if a drive should fail.

The stripe set can function with one missing drive and, when the failed drive is replaced, the parity information on the remaining drives can be used to reconstruct the missing data (formerly on the failed drive). Once reconstruction is complete, the new disk drive fully participates in the set to help protect against data loss should a different drive fail later on.

NOTE: Not every device supports every RAID level. Check the hardware documentation for your disk arrays, RAID arrays, disk drives, or other storage equipment for information about which RAID levels are supported by your devices.

Distributing Disk Access

For the same reasons as described in Disk Striping, the more you can balance disk access, the better performance you will achieve from disk reads and writes. This reduces the chance that a given device will be busy servicing another data access operation, causing additional reads and writes to wait.

File System Type

Your choice of file system type can also affect the efficiency of accessing your data. The VERITAS File System (VxFS) is generally faster than using the HFS file system.

Establishing Multiple Paths to a Device (for efficiency)

Beginning with HP-UX 11i version 3, HP-UX 11i supports *device multipathing*, a new technology that associates device files with devices by using unique device IDs rather than the hardware path to the devices. This means that *a single device file can represent multiple hardware paths to a given device* which, when combined with hardware that has multiple ports (supporting multiple physical connections), yields not only redundant paths to the device, but greater I/O bandwidth. HP-UX 11i version 3 can automatically load balance between multiple physical connections to a device, improving I/O efficiency.

For more information on device multipathing, see "How Storage is Addressed" (page 51).

Disk Mirroring (for performance)

Though the topic of disk mirroring is more of a data redundancy topic, there are performance reasons for mirroring. If you use a RAID 1 (mirroring) disk configuration in an environment more focused on *reading* data from the disks rather than writing data to the disks in the mirror, you can significantly speed up data input because subsequent disk blocks can be retrieved in parallel from alternating devices. For additional benefits of using RAID 1 configurations, see "Disk Mirroring".

Storage and Data Redundancy

The value of most data in the information age ranges from important to critical. The importance of data redundancy is directly proportional to the importance of the data being protected.

Data redundancy can take many forms but in every form multiple copies of your data exist so that if the primary copy of the data is damaged or destroyed another copy of those data can be used to continue your operations.

When choosing a data redundancy technology consider the following:
- How quickly do you need to recover from data loss?
- How easy is it to switch over to the alternate copy of your data should the primary become unusable?

- Do you need off site copies of your data?
- How often do the data change?

Establishing Multiple Paths to a Device (for redundancy)

One of the key points to protecting your data is eliminating single points of failure. RAIDs and other Disk Arrays, Disk Mirroring, and Data Backups, and Serviceguard are all about eliminating single points of failure.

Beginning with HP-UX 11i version 3, HP-UX 11i supports *device multipathing*, a technology that associates device files with devices by using unique device IDs rather than the hardware path to the devices. This means that *a single device file can represent multiple hardware paths to a given device* which, when combined with hardware that has multiple ports (supporting multiple physical connections), yields not only greater I/O bandwidth but also redundant paths to the device. HP-UX 11i can now automatically failover to an alternate hardware path should an interface card, cable, or other piece of hardware fail, and it can do this with little or no interruption to applications and users accessing the device.

RAIDs and other Disk Arrays

Disk arrays, and collections of independent disks configured using RAID configurations are capable of mirroring data from one physical disk to one or more additional physical disks, thereby giving you additional copies of the data should one drive mechanism fail. Simply having a second copy of the data exponentially decreases the chance of a failure of all copies of your data.

Hardware data mirroring is accomplished by using certain RAID configurations. Mirroring related RAID levels commonly used on HP disk arrays that support RAID are:

RAID 1   Mirroring data to one or more *additional* disks provides redundancy and the ability to take a copy of your data offline for example to snapshot the current state of the disk to a backup set for offline/off site storage.

> **NOTE:**   Not every device supports every RAID level. Check the hardware documentation for your disk arrays, RAID arrays, disk drives, or other storage equipment for information about which RAID levels are supported by your devices.

Disk Mirroring

The previous section discussed disk arrays and RAID configurations primarily from a hardware perspective. Disk mirroring can also be accomplished in software. The volume managers LVM and VERITAS Volume Manager can be used to mirror data.

In order to implement disk mirroring using LVM, you need to install the MirrorDisk/UX product (which is available as an optional product in the following Operating Environments):

- HP-UX 11i v3 BOE
- HP-UX 11i v3 VS-OE
- HP-UX 11i v3 HA-OE
- HP-UX 11i v3 DC-OE

Mirrordisk/UX supports up to three copies of data if you are using LVM with Version 1 volume groups, and up to six copies of data if you are using LVM with Version 2 volume groups.

Using the *base* version of VERITAS Volume Manager you can mirror only your root file system. By purchasing and installing the *full* version of VERITAS Volume Manager, you can mirror other disk groups and have up to 32 mirror copies of a volume's address space.

Data Backups

At any point in time, data can be copied using any one of a number of utilities. The destination for the copies of data can be removable media that can be stored off site or shipped to another location for safe keeping. Removable media that can be used for backups include:

- other disks
- magnetic tapes
  - DLT
  - DDS
- optical discs
  - recordable DVDs
  - recordable CDs
  - magneto-optical disk libraries

You can even back up files to a file on an alternate disk (as in the case of a tar archive).

Backup Utilities

There are many utilities in HP-UX to backup your data:

- `pax`

  The `pax` command extracts, writes, and lists archive files and copies files and directory hierarchies. A more contemporary utility, `pax` performs basically the same functions as the older (still available) utilities `cpio` and `tar`. For details about `pax`, see *pax*(1).

- `shar`

  The `shar` command bundles the named files and directories into a single distribution package suitable for mailing or moving. The files can contain any data, including executables. The resulting package, written to standard output, is a shell script file that can be edited (for example, to add messages at the beginning).

- `vxdump`

  `vxdump` copies to magnetic tape all files in a VxFS file system that have been changed after a certain date. See *vxdump*(1M).

- `fbackup` (recover data using `frecover`), an HP-UX specific backup utility for backing data up to the previous media types.

- `tar`

  `tar` (called the "tape archiver") can write to disk archive files or optical media. `tar` is compatible with many other operating systems, including other versions of UNIX, Linux, and Microsoft Windows.

- `cpio`

  The `cpio` command saves and restores archives of files on magnetic tape, other devices, or a regular file, and copies files from one directory to another while replicating the directory tree structure. When `cpio` completes processing the files, it reports the number of blocks written.

In addition to backing up to removable media, you can copy important files to another system using `ftp`, `rcp`, or (for secure copies) `sftp`.

How Storage is Addressed

The various components that comprise the HP-UX storage stack are addressed in different ways:

Table 3-2 Storage Components and how they are Addressed

Stack Component	How it is addressed
File System	Once created, a file system is usually addressed by its *mount point*, the directory in the HP-UX directory tree that represents the root of the files *in that file system*.
RAW access	Not all logical / physical volumes contain file systems. Other uses of these volumes include: • swap space • dump space • database managed space Swap and dump space are managed by the kernel, and database managed space is usually managed by the database application.
(Logical) Volumes	The logical containers allocated from space in a volume group or disk group are addressed by their volume name. Because these volumes are disk drives from the perspective of the operating system, they have associated device files. LVM logical volumes have device files with names of the form: • `/dev/vgxx/lvoln` • `/dev/vgxx/rlvoln` where *xx* represents the volume group that the logical volume belongs to, and *n* represents the logical volume number within that volume group. The directory `lvoln` contains the block device special files and the directory `rlvoln` contains the character device special files. VxVM volumes have device files with names of the form: • `/dev/vx/dsk/diskgroupname/volnn` • `/dev/vx/rdsk/diskgroupname/volnn` where `diskgroupname` is the name assigned to the volume group associated with the device files and `nn` represents the volume number.
Volume / Disk Groups	LVM Volume Groups have names, usually of the form "vg*nn*" (where *nn* represents the volume group number). The volume group that contains the root file system (if the root file system is contained in an LVM logical volume) is `vg00`. If you are using the VERITAS Volume Manager (and your root file system is contained in a VxVM volume), the root VxVM disk group is usually called `rootdg`[1].
Physical Disks	The fundamental building blocks of both LVM Volume Groups and VxVM Disk Groups are physical disk drives.

1   In releases of VERITAS Volume Manager (VxVM) prior to release 4.0, a system installed with VxVM was configured with a default disk group, `rootdg`, that had to contain at least one disk. By default, operations were directed to the `rootdg` disk group. Beginning with VxVM release 4.0, VxVM can function with no disk groups configured. There is no longer a requirement that you name any disk group `rootdg`, and any disk group that is named `rootdg` has no special properties because of that name.

Device Special Files

HP-UX, applications, and other processes communicate with devices and pseudo-devices by writing to and reading from *device special files*. Device special files are in a special format that tells HP-UX:

• Whether to communicate with the device using character or block transmissions
• Which device driver to use when communicating with the associated device
• How to locate/identify the device
• Any driver-specific attributes needed for communicating with a device

The first two items in the list above are determined by the *major number* of a device special file, the latter two items are determined by the *minor number* of a device special file.

The Anatomy of a Device Special File

Device Special Files (DSFs) are comprised of the following parts:

file name	This is the name of the file that appears in the /dev directory tree
major number	A number used to identify which driver to use to communicate with the device/LUN associated with that device special file
	The major number is an index for the device driver into one of two kernel tables — bdevsw, the block device switch table and cdevsw, the character device switch table.
	Drivers that support both block and character I/O (such as SCSI disk driver and optical autochanger) have both a block major number and a character major number. Devices that support only character-mode access have only a character major number.
minor number	A number that identifies the hardware location and sometimes driver-dependent characteristics (usually organized by bit assignments).

The three parts of a device file can be viewed using the ll (ls -l) command. For example:

Figure 3-3 Device Special File Components

```
$ ll /dev/dsk /dev/rdsk /dev/disk /dev/rdisk

/dev/disk:
total 0
brw-r-----   1 bin      sys          1 0x000000 Aug 10 16:26 disk2
brw-r-----   1 bin      sys          1 0x000001 Aug 10 16:26 disk3
brw-r-----   1 bin      sys          1 0x000002 Sep 20 17:27 disk4
brw-r-----   1 bin      sys          1 0x000003 Sep 20 17:27 disk5

/dev/dsk:
total 0
brw-r-----   1 bin      sys         31 0x006000 Aug 10 16:26 c0t6d0
brw-r-----   1 bin      sys         31 0x026000 Aug 10 16:26 c2t6d0

/dev/rdisk:
total 0
crw-r-----   1 bin      sys         11 0x000000 Aug 10 16:26 disk2
crw-r-----   1 bin      sys         11 0x000001 Sep 20 16:03 disk3
crw-r-----   1 bin      sys         11 0x000002 Sep 20 17:27 disk4
crw-r-----   1 bin      sys         11 0x000003 Sep 20 17:27 disk5

/dev/rdsk:
total 0
crw-r-----   1 bin      sys        188 0x006000 Aug 10 16:26 c0t6d0
crw-r-----   1 bin      sys        188 0x026000 Aug 10 16:26 c2t6d0
```

Major Number
Minor Number
File Name

b = block device special file   c= character device special file

You can view, in more human readable form, information contained in the device special file using the /usr/sbin/lssf command. For example:

```
# /usr/sbin/lssf /dev/rdsk/*
sdisk card instance 0 SCSI target 6 SCSI LUN 0 section 0 at address 0/0/0/2/0.6.0
  /dev/rdsk/c0t6d0sdisk card instance 2 SCSI target 6 SCSI LUN 0 section 0 at
  address 0/0/0/3/0.6.0 /dev/rdsk/c2t6d0
```

```
# /usr/sbin/lssf  /dev/disk/*
esdisk section 0 at address 64000/0xfa00/0x0 /dev/disk/disk2 esdisk section 0 at
  address 64000/0xfa00/0x1 /dev/disk/disk3
```

The /usr/sbin/ioscan command will also show you hardware path information about the devices on your system.

Here is the legacy view of the disk drives on a system:

```
# /usr/sbin/ioscan -C disk
H/W Path          Class             Description
=======================================================
0/0/0/2/0.6.0     disk              HP 36.4GMAN3367MC
0/0/0/3/0.6.0     disk              HP 36.4GMAN3367MC
```

Here is the agile view of the same disk drives (showing the virtual LUN hardware paths rather than the actual hardware paths):

```
# /usr/sbin/ioscan -N -C disk
H/W Path          Class             Description
```

```
=====================================================
64000/0xfa00/0x2  disk                HP 36.4GMAN3367MC
64000/0xfa00/0x3  disk                HP 36.4GMAN3367MC
```

Legacy versus Agile Device Addressing

Beginning with HP-UX 11i version 3, mass storage devices are referenced by device instances rather than by the hardware paths to the devices. This has many benefits over the previous addressing scheme which associated a given device special file with the hardware path to a device. Device hardware addressing for mass storage devices is now fluid, automatic, and transparent. This has many benefits.

NOTE: For compatibility with previous HP-UX releases, the previous (legacy) device addressing scheme for mass storage devices is still supported under HP-UX 11i version 3; therefore scripts, configurations, and other uses of device special files you have previously created will continue to work.

For transition purposes, you can use *both* legacy device addressing (using legacy device special files) *and* agile device addressing (using persistent device special files) simultaneously, but to take advantage of the many benefits of agile device addressing, and for *future compatibility* reasons, you should transition to using persistent device special files going forward once all of the underlying file systems and technologies that you use can support them.

Greater Configuration Stability

Agile device addressing allows for hardware paths to change between system boots (for example, if a LUN is moved from one HBA to another while a server is shutdown) and for SAN configurations to change without requiring changes to device special files (and therefore without requiring changes to other configuration files). If you replace a disk associated with a persistent device special file (the type of device special files that provide agile device addressing), you can use the io_redirect_dsf command to update the persistent device special file to reference the replacement disk. For details, see the manpage *io_redirect_dsf*(1M).

Scalability

Because of a limitation in device special file minor numbers, a server was previously limited to 256 bus instances. Using agile device addressing you can now address more than 256 bus instances.

Agile device addressing also allows you to address a greater number of LUNs. HP-UX 11i version 3 supports up to 16,384 LUNs.

Multi-pathing

Each LUN can have up to 32 physical I/O paths. HP-UX 11i version 3 automatically discovers and configures new physical I/O paths to a LUN and balances data flowing through the various paths to a given device using one of the following load-balancing policies:

closest_path	This load balancing policy selects a LUN based on its affinity with the processor core issuing the I/O, minimizing latency.
cl_round_robin	This load balancing policy is applicable to HP cell-based platforms. The LUN paths are selected in a round robin manner within the locality of CPU on which the I/O was initiated, to ensure that memory access latencies are optimized.
least_cmd_load	Directs I/O requests through the hardware path with the least outstanding I/O requests.
round_robin	Cycles I/O requests through available hardware paths in round robin fashion.
path_lockdown	Directs I/O requests through a single hardware path; specifically, the path with the fewest outstanding I/O requests when the device is opened. This

	is the default load balancing algorithm for serial devices (for example tape drives).
`preferred_path`	The I/O path set in the *preferred_path* attribute is preferably used for I/O transfer. If this I/O path is not available or if the *preferred_path* attribute was not set, any other path is selected for I/O transfer. This policy is useful for certain disk arrays, which may exhibit some performance degradation if I/Os are transferred via several I/O paths to a LUN simultaneously.
`pref_tport`	This load balancing policy selects LUN paths from a preferred list unless none are available (or defined), in which case other LUN paths are selected.
`wt_round_robin`	This load balancing policy selects an I/O path based on a weighted round robin algorithm

Use the `scsimgr` command to specify which of the previous policies should be used for a given device.

> **NOTE:** Not every device supports every load balancing policy. The type of device determines which of the previous policies you can use. For details see the following manpages:
> - *scsimgr*(1M)
> - *scsimgr_eschgr*(7)
> - *scsimgr_esdisk*(7)
> - *scsimgr_estape*(7)

HP-UX can also automatically re-balance the loads on remaining data paths to a LUN should one or more of those paths fail.

> **NOTE:** Use the `scsimgr` command (see *scsimgr*(1M) for details) to set the load balancing algorithm. An algorithm choice can be set individually for each LUN, or for all LUNs on the server. Also, the choice can be permanently set (value retained across reboots), or temporarily set (until the next reboot).

Device Special File Directories (and Name Formats)

Device Special Files are located in the `/dev` directory and many are organized in a series of sub-directories within `/dev`. Two of these directories contain the *persistent* device special files defining the physical disk drives on a server:

`/dev/disk`	Contains *persistent* device special files for *block mode* access to physical disk devices on a server.
`/dev/rdisk`	Contains *persistent* device special files for *character mode* access to physical disk devices on a server.

Within the previous directories, files have names in the format "disk*N*" (where "*N*" is the instance number of the disk).

Examples:

```
/dev/disk/disk15
/dev/rdisk/disk7
```

An optional part of a device file name for a disk can be appended to represent disk partition numbers. By convention, in the absence of this optional part of the device file name the name represents an entire disk. This optional part expands the format of the name to be diskN_p# (where p# represents the partition number, tape density, or other information).

Examples:

```
/dev/disk/disk15_p3
/dev/rdisk/disk7_p1
```

The following directories still exist in HP-UX 11i version 3 for backwards compatibility. They contain the *legacy* device special files defining the physical drives on a server (the legacy form):

/dev/dsk   Contains *legacy* device special files for *block mode* access to physical disk devices on a server.

/dev/rdsk   Contains *legacy* device special files for *character mode* access to physical disk devices on a server.

Within these directories, files have names in the format "c#t#d#" (where c# represents the controller instance number, t# represents the SCSI target number, and d# represents the SCSI LUN number).

Examples:

```
/dev/dsk/c3t7d0
/dev/rdsk/c3t15d5
```

With legacy device special files that are associated with disks an optional part of the filename can represent disk partition numbers. By convention, in the absence of this optional part of the device file name the name represents an entire disk. This optional part expands the format of the name to be c#t#d#s# (where s# represents the partition number, tape density, or other information).

Examples:

```
/dev/dsk/c3t7d0s3
/dev/rdsk/c3t15d7s1
```

NOTE: You can use both legacy and persistent device special files simultaneously but to gain the many benefits of persistent device special files, you need to use those.

For example, using legacy device special files you can only define and address up to 256 external buses on a server. If you have more than 256 you will need to use persistent device special files to access the devices beyond the 256 address boundary.

For Additional Information on Naming Device Special Files

Additional information on naming device special files is located in the following manpages:
- *autochanger*(7)
- *disk*(7)
- *intro*(7)
- *mt*(7)

Mass Storage Hardware Paths (three formats)

Hardware paths, as the name implies, define the physical paths that data travels to reach devices. In HP-UX 11i version 3 there are three formats to specific hardware paths to mass storage devices.

Legacy Hardware Paths

This is the format used in releases prior to HP-UX 11i Version 3. It is composed of a series of bus-nexus addresses separated by slash characters ("/") leading to the host bus adapter (HBA); beneath the HBA, additional address elements are separated by periods (".").

For directly connected devices, the addressing can be a simple target and LUN:
0/0/2/0.1.7.0

For SCSI-3 devices connected through a storage area network (SAN), legacy addressing is emulated with a *domain*, *area*, *port*, *virtual bus*, *virtual target*, and *virtual LUN*:
0/2/1/0.1.5.0.0.3.7

Lunpath Hardware Paths

This format is used for addressing LUNs in agile mode. It is identical to the legacy hardware path format up to the host bus adapter (HBA). Beneath the HBA, two additional address elements are represented (in hexadecimal):

Target address   A transport dependent address identifying the physical device associated with the hardware path

LUN address   A 64-bit representation of the LUN identifier reported by the target.

`0/2/1/0.0x50001fe1500170ac.0x4017000000000000` is an example of a SCSI-3 hardware path.

Figure 3-4 Lunpath Hardware Path Components

0/2/1/0.0x50001fe1500170ac.0x4017000000000000

    HBA Path   Target Address   LUN Address
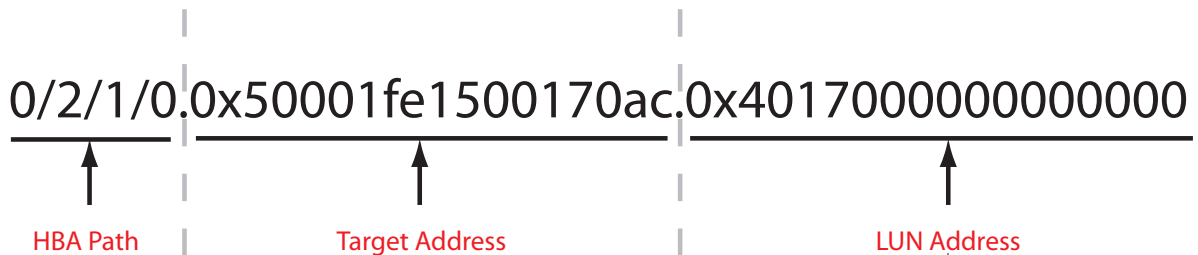
LUN Hardware Paths

Because devices can have multiple physical hardware paths, a *virtualized LUN hardware path* is needed so that the hardware path that a persistent device special file maps to remains the same even when the underlying physical hardware path elements change.

Instead of a series of bus-nexus addresses (corresponding to specific hardware paths) leading to the HBA, virtual hardware paths use a virtual bus-nexus (known as the *virtual root node*) with an address of 64000. Addressing beneath that virtual root node consists of a **virtual bus address** and a *virtual LUN ID*, delimited by slash characters ("/").

`64000/0xfa00/0x22` is an example of a virtual hardware address.

Figure 3-5 LUN Hardware Path Components

64000/0xfa00/0x22

    Virtual   Virtual   Virtual
    Root   Bus   LUN ID
    Node   Address
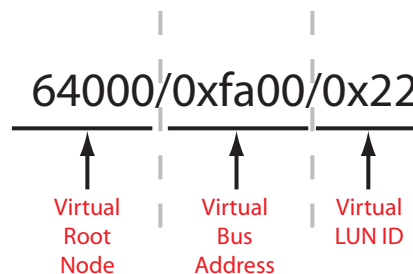
Example 3-2 Hardware Path Format Summary

The three formats described previously are different ways of referring to the same LUN, so a single LUN could have all of the following addresses:

```
0/2/1/0.1.4.0.0.2.7
0/2/1/0.1.5.0.0.2.7
0/4/1/0.1.4.0.0.2.7
0/4/1/0.1.5.0.0.2.7
0/2/1/0.0x50001fe1500170ac.0x4017000000000000
0/2/1/0.0x50001fe1500170ad.0x4017000000000000
0/4/1/0.0x50001fe1500170ac.0x4017000000000000
0/4/1/0.0x50001fe1500170ad.0x4017000000000000
64000/0xfa00/0x22
```

In the previous example the LUN has four physical hardware paths. The first four lines represent them using the legacy hardware path format, the next four lines represent them using the SCSI-3 hardware path format, and the final line represents the *single* virtual hardware path (used for all four physical paths).

HP-UX 11i version 3 commands will accept any of these three formats to specify a hardware path to a LUN.

Commands Associated with Device Special Files

Here is a list of key commands used for managing device special files:

insf	Used during boot to create both legacy and persistent device special files for new devices. insf can also be used manually, while the system is running, to create such device special files (for example, if ioscan caused new hardware to be discovered insf could be run to create device special files for that new hardware without having to wait for a reboot of the system)[3]. See *insf*(1M) for details.
	With the -L option, insf enables the legacy naming model.
mksf	Used to create a single device special file, usually with non-default characteristics. For details on specifying the parameters for your device special file, see *mksf*(1M).
	If you wish to create a *single* device special file, perhaps with specific parameters, use mksf instead of insf. For details on specifying the parameters for your device special file, see *mksf*(1M).
lssf	lssf lists information about specified device special files (in a friendly, human readable form). For example, lssf can list *stale device special files*. You can specify legacy or persistent device special files. For details and examples of lssf output, see *lssf*(1M).
rmsf	rmsf removes specified device special files. For information about the many rmsf options for specifying which device special files to remove (and from where to remove them), see the *rmsf*(1M) manpage.
	With the -L option, rmsf disables the legacy naming model, removing all legacy I/O nodes and their device special files from the system.
io_redirect_dsf	io_redirect_dsf reassigns an existing device special file from one device to another. Because this takes advantage of one of the primary benefits of persistent device special files, it will not work with legacy device special files. See *io_redirect_dsf*(1M) for details.

---

3.   In some cases, new hardware is automatically discovered and associated persistent device special files are created, even before ioscan is run. However, if you have new hardware with no associated device special files, insf can create them for you.

`ioscan`	`ioscan` performs many functions. Primarily it scans a server's hardware, binding new hardware with appropriate drivers. With regard to device special files, `ioscan` can display the mapping between legacy device special files and persistent device special files. See *ioscan*(1M) for details on `ioscan`'s many functions and options.
`scsimgr`	`scsimgr`, like `ioscan`, has many functions other than working with device special files. But, you can use `scsimgr` to validate the change of a LUN associated with a legacy device special file (replace a legacy device special file). If you have replaced a disk drive and are using legacy device special files, refer to the *scsimgr*(1M) manpage for assistance with remapping the device special file to the new device. The `scsimgr` command to do this is `replace_leg_dsf`.

For More Information on the Next Generation Mass Storage Stack

The following resources contain much more information on the components of the Next Generation Mass Storage Stack:

- Technical White Paper: *The Next Generation Mass Storage Stack HP-UX 11i v3*
- The following manpages:
  — *scsimgr*(1M)
  — *io_redirect_dsf*(1M)
  — *insf*(1M), *lssf*(1M), *mksf*(1M), *rmsf*(1M)
  — *iobind*(1M), *iofind*(1M), *ioscan*(1M)
  — *intro*(7)

Superdome 2 Resource Paths

On Superdome 2 systems, the *Onboard Administrator (OA)*, used to configure nPartitions and Virtual Partitions (vPars), uses a special form of addressing to reference the components of the system. Known as *resource paths*, these addresses reference system components using operating system agnostic path elements called entities.

The Superdome 2 OA works whether or not any operating systems are installed or booted; it therefore needs its own mechanism for identifying system components. Resource paths provide that mechanism.

Resource paths are associated with easily identifiable system components (for example, enclosures, blades, I/O bays, and I/O slots), and are therefore a more "human friendly" way of representing addresses.

Once booted, HP-UX uses its own hardware addressing mechanism (described in the previous sections of this chapter). These hardware addresses, in either legacy or agile form, allow HP-UX to locate its resources; but, they are much less suitable for human interpretation.

On Superdome 2 systems, where nPartitions and Virtual Partitions are created using the Onboard Administrator, there are occasions when you need to map a hardware address (for example, as reported by the `ioscan` command) to the corresponding resource path or physical location needed to properly partition your system. The `ioscan` command, when used with the "`-m resourcepath`" option provides this mapping. For examples of how this works see the *HP Superdome 2 Partitioning Administrator Guide*, and the *resourcepath*(5) and *ioscan*(1M) manpages.

Managing HP-UX Swap Space

Swap space is where HP-UX stores unneeded pages of memory from running processes, a process called virtual memory paging (or simply paging) because the chunks of data moved in and out of physical RAM are called pages. This enables HP-UX to use much more memory than physically exists on a server.

Types of Swap Space

There are three types of swap space used for paging operations:
- Device Swap
- File System Swap
- Pseudo Swap

Device Swap

Swap space is initially allocated when you configure your disks. *Device swap* space occupies a logical volume or disk partition that is typically reserved expressly for paging purposes. This space may also be configured as a dump area but doing so has implications for memory dump integrity if a crash occurs. See "Using a Device for Both Paging and Dumping (System Recovery Time)" (page 82). Because, when HP-UX is running, the device is used exclusively for paging, you cannot also store files on it.

> **NOTE:** There is one exception to the rule that a given logical volume cannot be used for both file system space and device swap. If you have unused space between the end of a file system and the end of the logical volume in which it resides (that is, the file system is smaller than the logical volume it is in), the unused space (not allocated to the file system), can be used as device swap space.

Device swap can only be used locally; it cannot be accessed remotely by clients using network disk access protocols.

Device swap is quickly accessed because HP-UX can get to the logical volume or disk partition directly to perform large writes or reads.

File System Swap

If the device swap space you have configured on your server is not enough and you have no more devices that you can dedicate for that device swap space, you can configure file system swap space.

File system swap allows for extra swap space if there is an *occasional need* for more than is allocated as device swap space. It is used only when device swap space is insufficient. File system swap space is configured as additional swap space to be allocated from unused space *within a file system*.

Because file system swap requires the system to perform a greater amount of processing it is usually slower than device swap and should not be used as a permanent replacement for a sufficient amount of device swap space. It is best for the occasional overflow of device swap space.

The file system used for swap can be either a local or a remote file system. Cluster clients can use remote file system swap for their swap needs. Swapping to a remote file system is slower than swapping to a local file system and is *not* encouraged if local device swap or local file system swap is available.

Pseudo Swap

Pseudo Swap is very different than device swap space or file system swap space. It is one of the technologies that allow you to more efficiently utilize the resources on your system.

Pseudo swap "space" does not really exist; HP-UX just behaves as though it has an extra amount of swap space. Pseudo swap takes advantage of the fact that not all swap space that is reserved is actually used. This allows a greater number of processes to run in memory than could be supported by configured swap devices. Pseudo swap is best used with large memory systems.

If you choose to use the pseudo swap capability (actually, it is enabled by default), an amount of pseudo swap space equal to 7/8ths of the amount of physical ram available to your server, nPartition, or virtual partition is used as pseudo swap.

Lazy Swap

Another technology that takes advantage of the fact that not all swap space that is reserved is actually used is *lazy swap*. The lazy swap feature causes HP-UX to *not* reserve swap space for a process-private page until the associated process actually modifies the page. This can significantly reduce the amount of allocated swap space.

Lazy swap is configured on a process by process basis. There are programmatic ways to enable lazy swap or a user can modify a binary executable file to enable lazy swap by using the +z option to the chatr command. See the *chatr*(1) manpage for details.

Primary and Secondary Swap Space

HP-UX must have at least one device swap area available when it boots. This area is known as the *primary swap* area.[4] Primary swap, by default, is located on the same disk as the root file system (though in a different logical volume). Use the swapon command (see *swapon*(1M)) to define swap space.

Other swap space may be used in addition to primary swap. This is known as *secondary swap* space. If you are using device swap as your secondary swap space, for better performance allocate the secondary swap space on a disk drive other than where the primary swap is located.

File system swap is always secondary swap.

Figure 3-6 Swap Space - Possible Locations for Paging
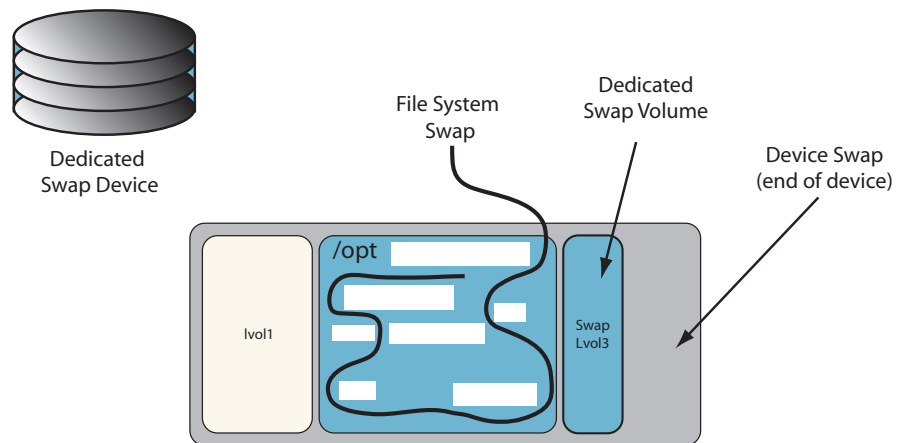
Estimating Your Swap Space Needs

Your swap space must be large enough to hold all the processes that could be running at your system's peak usage times.

If your system performance is good, and, in particular, if you are not getting swap errors such as "Out of Memory" or those to the effect that a process was killed due to no swap space, then your system has adequate swap space.

Unless the amount of physical memory on your system is extremely large, the minimum amount of swap space should equal the amount of physical memory on the system. In general, size your server's swap space to be roughly two to four times the amount of physical memory that is used by HP-UX on your server, nPartition, or virtual partition.

Swap space usage increases with system load. If you are adding (or removing) a large number of users or applications, you will probably need to re-evaluate your swap space needs.

---

4.   Primary swap is not mandatory if pseudo swap is enabled, however, it is strongly recommended.

NOTE: To get a snapshot of the total amount of swap space currently being used, use the command:

```
# /usr/sbin/swapinfo -tam
             Mb       Mb       Mb     PCT    START/      Mb
TYPE      AVAIL     USED     FREE    USED    LIMIT   RESERVE   PRI   NAME
dev        4096        0     4096      0%        0         -     1   /dev/vg00/lvol2
reserve       -      257     -257
memory     1940      562     1378     29%
total      6036      819     5217     14%        -         0     -
```

This number will vary over time, depending on the current mix of applications running, but if the total percentage used is regularly high, roughly 90% or greater, then you probably need to add more swap space.

Once you know or suspect that you will have to increase (or decrease) your swap space, you should estimate your swap space requirements.

You can estimate the amount of swap space you need by adding the memory requirements of the applications you expect to run (simultaneously) on your system to the amount of physical memory you have.

Enabling Swap Space

When HP-UX is in the earliest stages of the boot sequence, the system begins by paging on only a single device so that only one disk is required at boot time.

During the processing of the startup script /sbin/init.d/swap_start, calls to swapon enable any additional paging areas, if any are defined in the file /etc/fstab.

Should you find the need for additional paging space while your system is running, you can also run the swapon command to manually enable additional space. See the *swapon*(1M) and *fstab*(4) manpages for details.

Disabling Swap Space

Sometimes, you can disable swap space. In order to be able to disable paging to swap areas:
- your system must be running HP-UX 11i Version 3 September 2008 Update, or later
- sufficient swap space must remain active (following the change) to continue to operate your system

Use the swapoff command to disable paging to a specific swap area. For example:

/usr/sbin/swapoff /dev/vg00/lvol2

Guidelines for Setting Up Swap Areas

There are some guidelines to consider when configuring swap space on your system. Most of these are focused on maximizing the performance of HP-UX.

Device Swap Guidelines

Use the following guidelines to configure the most commonly used type of swap space, device swap:
- Interleave device swap areas for better performance.

   Two swap areas on different disks perform better than one swap area with the equivalent amount of space. This allows *interleaved swapping* which means the swap areas are written to concurrently, minimizing disk head movement, thus enhancing performance.

   When using LVM, you should set up secondary swap areas within logical volumes that are on *different* disks (physical volumes) using lvextend.

If you have only one disk and need to increase swap space, try to move the primary swap area to a larger region on the disk.

To see which devices are already being used for device swap use the command: `swapinfo -d`

- Try to keep multiple device swap areas similar in size.

  Device swap areas should have similar sizes for best performance. When you configure swap areas of different sizes, when all space in the smaller device swap area is used only the larger swap area is available making interleaving no longer possible and slowing down paging performance.

- The `nswapdev` tunable system parameter controls the maximum number of swap devices. Although the default value for `nswapdev` is large enough to accommodate nearly all HP-UX systems, verify that it is large enough to accommodate the number of swap areas you require.

File System Swap Guidelines

When you need more swap space and you have no devices available for additional device swap, or if you need to swap to a remote system, you can dynamically add file system swap to your system. Use the following guidelines:

- Interleave file system swap areas for best performance.

  Two swap areas on different disks perform better than one swap area with the equivalent amount of space. Multiple devices allow for *interleaved swapping* which means the swap areas are written to concurrently, minimizing disk head movement, thus enhancing performance. This applies as much to file system swap space as it does to device swap space, so the same guideline applies.

  To see which devices are already being used for file system swap use the command: `swapinfo -f`

- If possible, avoid configuring heavily used file systems. Heavily used has two meanings here:
  1. *Actively used file systems* (for example, the root file system, or those file systems used most frequently by your primary applications). This will slow down the performance of your server as paging activities compete with your applications and user file access.
  2. *Very full file systems*. Because file system swap uses unused space within file systems, if the file systems are very full there is not much unused space for paging use (and it is probably very fragmented within the file system). To gauge how full a file systems are, use the `bdf` command.

Guidelines for Assigning Swap Priority

When you add swap areas, you can assign a priority to each. Priorities range from 0 (the highest) to 10 (the lowest). HP-UX uses swap areas with higher priority first. And, HP-UX gives device swap priority over file system swap when each has the same priority. Here are the guidelines you should use:

- Given multiple swap devices with identical performance, assign each an identical priority. By so doing, you will allow the system to use each of them on an interleaved basis which enhances performance.
- Assign higher priorities to the swap areas that have faster performance and lower priorities to areas that are slower.
- Do not give file system swap areas priority over device system swap areas. Although this is not absolutely necessary it enhances the `swapinfo` output.
- Give lower use file systems priority over higher use file systems.

For More Information on Configuring and Managing Swap Space

The following manpages contain important information on configuring swap space:

fstab(4)   The file `/etc/fstab` not only defines which file systems should be mounted to which mount points in the directory tree (see "The HP-UX Directory Structure" (page 40)), it is also one of the key places you configure swap space.

lvlnboot(1M)   `lvlnboot` prepares an LVM logical volume to be root, boot, primary swap, or a dump volume.

swapinfo(1M)   `swapinfo` prints information about device and file system paging space.

swapon(1M)   The `swapon` command enables devices or file systems on which paging is to take place.

swapoff(1M)   The `swapoff` command disables active paging areas, if possible.

The following kernel tunables affect paging activities on HP-UX:

`nswapdev`   Maximum number of swap devices that can be enabled for swap.

HP-UX Input and Output

Input and Output (I/O) on HP-UX is accomplished by reading from and writing to special files known as *device special files*. Starting with HP-UX 11i version 3, there are two types of device special files:

- *Legacy* Device Special Files
- *Persistent* Device Special Files

Persistent device special files offer many advantages over legacy device special files:

- With *legacy* device special files, if you move the device associated with them (logically or physically) you will need to address those devices via new or different device special files. This is because the device special files are associated with the *physical path to the device* (the sequence of internal components that data must travel). If any part of the path changes, a different device special file is needed.

- *Persistent* device special files are instead associated with a *unique address on the device* (known as a worldwide identifier — WWID) rather than the path to the device. This means that even if the path to the device changes, no changes are needed to the device special files associated with that device.

  Persistent device special files are only associated with mass storage devices. Generally these are disk drives, some tape drives, and similar mass storage devices.

  For non-mass storage devices, the traditional form of device special files (based on hardware paths as with legacy device special files for mass storage devices) still applies.

For more information on device special files, "How Storage is Addressed".

Printing

Printing on HP-UX is accomplished through a subsystem known as the HP-UX Line Printer Spooling System.

Overview of the HP-UX Line Printer Spooling System

The *Line Printer Spooling System* (*spooler*) is a set of programs, shell scripts, and directories that controls your printers and the flow of data going to them.

The spooler ensures that output from multiple users or processes doesn't arrive on a printed page intermixed, yielding a printout that is useful to no one. With the line printer spooling system you can also:

- Give priority to print requests from certain users/processes
- Group printers so that they share a common input print queue
- Give priority to some printers over others
- Define a system default printer (to be used whenever a print request does not specify a print destination)
- Control the acceptance or rejection of incoming print requests to specific print queues
- Control the printing of jobs that have already been submitted
- Submit print requests to spoolers on remote systems (remote printing of local jobs)
- Accept print requests from spoolers on remote systems (local printing of remote jobs)
- Cancel previously submitted print jobs

To understand the flow of data through the spooler, think of it as a plumbing system, as shown in Figure 3-7 (page 66). The data in the form of *print requests* (print jobs) enters the system like the "water" in the plumbing system. Directories known as *print queues* serve as temporary holding tanks for the print requests until they are sent to a printer to be printed. The print queues and the scheduler control the flow of print jobs to the defined printers:

Accepting, rejecting, enabling, or disabling print requests controls the data flowing through the spooler as valves would control the flow of water in a real plumbing system.

- the commands *accept* and *reject* control the flow of print requests *into* the print queues
- the commands *enable* and *disable* control the flow of print requests *out of* the print queues to the printers

Based on the status of the various print queues and printers, the line printer **scheduler** (called `lpsched`) accepts incoming print requests, routing them to the print queues; and it functions as an automated flow controller in the "plumbing" system, routing print requests from the print queues to the physical printers on a first-in-first-out basis (while accounting for the priority of print requests and printers).

Interface files (written as shell scripts) near the end of the data flow serve as pumps that "pump" the orderly flow of data to the printers.

The line printer scheduler:

- prevents intermixed listings
- monitors printer/print request priorities
- adjusts printer status and availability
- logs spooler activities

If one printer's "drain gets clogged", you can reroute a print request from that printer to another by using the `lpmove` command. Unwanted data can be "flushed" from the spooling system with the `cancel` command.

Figure 3-7 Line Printer Spooler "Plumbing" Diagram

Print Requests (lp command)

Reject
(closed)

lpmove

Accept
(open)

color_lj

dept_27

Printer
Class

printer_27a

printer_27b

Disable
(closed)

Enable
(open)

Disable
(closed)
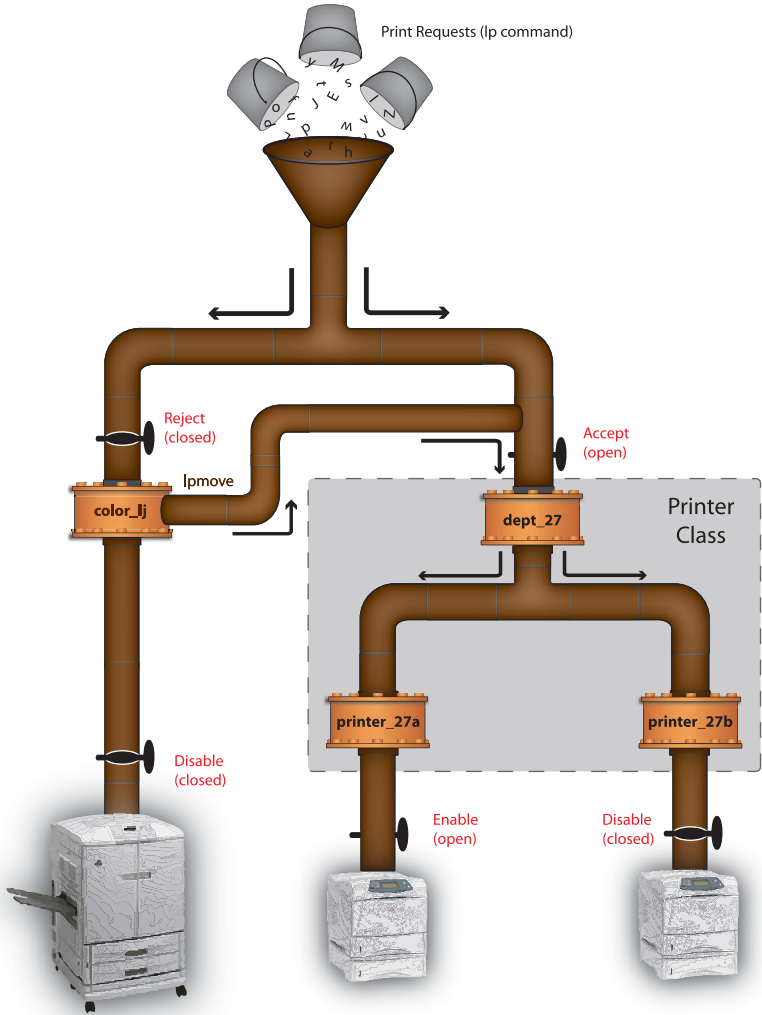
Remote Spooling

You can also send print requests to a printer configured on a **remote system**, using *remote spooling*. When you use remote spooling, a shell script ("pump") sends data to a remote system via the `rlp` command.

A remote spooling program (a daemon called `rlpdaemon`), running on the remote system, receives data and directs it into the remote system's spooler. `rlpdaemon` can also run on your local system to receive requests from remote systems. Remote spooling is carried out by communication between the local spooler and the remote spooler.

If some of your systems have printers configured and others do not (but all systems are networked), you can have the systems share use of available printers. To do so, set up the spoolers of the systems lacking printers to automatically send print jobs via the network to the spooler of a system equipped with the printer.

The `rlpdaemon` program runs in the background of the printer's system, monitoring the incoming network traffic for remote print requests from other systems. When these requests arrive, the `rlpdaemon` submits them to its local spooler on behalf of the remote user.

In addition to handling remote print requests, `rlpdaemon` handles cancel and status requests from remote systems, using special interface scripts much like printer interface scripts.

Configuring a remote printer into your spooler requires that you supply the following additional information beyond what you supply to configure a local printer:

- name of the system with the printer
- interface script to use when issuing a remote cancel request
- interface script to use when issuing a remote status request
- printer name, as defined in the spooler of the remote system

To configure remote spooling, see *HP-UX System Administrator's Guide: Configuration Management*.

Network Printing

Network Printing refers to printing to a printer that is directly connected to the network, usually via an HP JetDirect interface card or built-in network connectivity. This is different from remote spooling in that another computer is not involved.

Printer Model Files and Interface Files

Printer interface files are special script files used by the spooler to communicate with the printers during the actual printing of a print job. These interface files are very specific to the type of printer they are sending data to and they can be customized, for example to provide unique banner/separator pages.

HP-UX provides a library of sample files corresponding to most HP printers (or printer families) -- examples: "`laserjet`", "`colorlaserjet`", "`PCL5`" -- and some sample files corresponding to generic printer families (for example "`postscript`").

When you set up a printer in the line printer spooling system (whether through a tool like the System Management Homepage or directly by using the shell based `lpadmin` command), you specify a *printer model script* to be associated with the printer you are setting up. A copy of the specified model script (one of the previously mentioned sample files) is copied from the `/usr/spool/lp/model` directory into the `/usr/spool/lp/interface` directory (which is actually a symbolic link to the directory `/etc/lp/interface`[5] where the copy will physically reside).

---

5. Because everything under the `/usr` mount point is supposed to be treated as read-only and the contents of the interface directory are supposed to be editable. See `/usr` in the section Key HP-UX Directories.

Printer model files are required in the following procedures:

- Adding a Local Printer to the LP Spooler — see *HP-UX System Administrator's Guide: Configuration Management (Chapter 6: Configuring Printers)*
- Adding a Remote Printer to the LP Spooler — see *HP-UX System Administrator's Guide: Configuration Management (Chapter 6: Configuring Printers)*
- Adding a Network-Based Printer — see *HP-UX System Administrator's Guide: Configuration Management (Chapter 6: Configuring Printers)*

The model scripts are ASCII files that you can read to find one that matches the printer's capabilities. Protocols like the Printer Command Language (PCL) or PostScript are understood by many non-HP printers as well. The printer's user guide may reveal details like the PCL language level supported by the printer that are necessary to find the model script that allows to make the best use of the printer's features.

Even model scripts written to support a specific printer type might not support all features offered by it. Therefore, they might benefit from customizing. The scripts can be edited though it is probably better practice to copy one to a file of a new name and edit the copy (in order to prevent your customization from being overwritten should a new version of the original script be delivered in a future update to HP-UX).

If there is no model file to match the protocol used by your printer, the one named "dumb" offers some basic functionality and should work with most non-HP printers. There is also a "dumbplot" model script for plotters.

The `/usr/sbin/lpadmin` command copies the identified model script to `/etc/lp/interface/printername`. See *lpadmin*(1M) for information on the command options.

Printer Types

A *local printer* is physically connected to your system. To configure a local printer, see *HP-UX System Administrator's Guide: Configuration Management (Chapter 6: Configuring Printers)*.

A *remote printer* may be physically connected or simply configured to a computer and accessed over a network. To access the remote printer, your system sends requests over the network to another system. To configure a remote printer into your local spooler, you must be able to access the remote system via the network. To configure a remote printer, see *HP-UX System Administrator's Guide: Configuration Management (Chapter 6: Configuring Printers)*.

A *network-based printer* differs from a remote printer in that it is connected directly to the network rather than to a server. Network printers do not use device special files, but have their own IP address and LANIC identification. To configure a network-based printer, see *HP-UX System Administrator's Guide: Configuration Management (Chapter 6: Configuring Printers)*.

Printer Name

When you configure a printer into the spooler, you assign it a printer name, to which you direct print requests. A printer name may have up to 256 alphanumeric characters, however for maximum compatibility with other subsystems that interact with the spooler try to keep the names to 14 characters. Printer names may include underscores. The following are sample valid printer names: `laser1`, `letterhead`, `invoices`, `check_printer`. The printer names you assign are listed in the directory `/usr/spool/lp/interface`. Each file in that directory is a copy of the model file (printer interface script) that enables you to print to the named printer.

Printer Class

You can make efficient use of multiple printers by grouping them as though logically they were a single printer. To do this, you create a *printer class*. A printer class is a collective name for a group of printers. The printer class is retained in the directory `/usr/spool/lp/class`. For example, the sample printers named `laser1` and `letterhead` might be assigned a printer class called "VIP", while the printers named `invoices` and `check_printer` might be assigned a

printer class called "accounts". A printer can belong to more than one class, however remote printers cannot belong to a printer class. It is not necessary to have every printer be part of a class; you can have some of your printers grouped in classes while others remain independent.

To use a printer class you direct print requests to the class, name rather than to a specific printer. The print request is spooled to a single print queue and printed by the first available printer in the class. Thus, printer usage can be balanced and reliance on a particular printer can be minimized.

To create a printer class, see the procedure "Creating a Printer Class" in the *HP-UX System Administrator's Guide: Configuration Management (Chapter 6: Configuring Printers)*. Also see the procedures "Removing a Printer from a Printer Class" and "Removing a Printer Class" in the same chapter.

Print Destination

The *print destination* is the printer or printer class where a file containing the print job will be queued. Several commands for the spooler require you to specify a print destination. You can appoint one print destination in your spooler to be the *system default printer*. Each user can also personalize which printer is their default printer by setting a shell environment called LPDEST. When LPDEST is defined in a user's environment, the printer represented by that variable takes precedence over the system default printer.

Example 3-3 Default Printer (example)

For example, if the system default printer for a server is defined as laser1 and a user has defined the environment variable LPDEST with the value ceo_print, then unless the user specified a print destination their print requests will be sent to the destination ceo_print. A different user on the same server with no LPDEST environment variable defined will have their print jobs directed to laser1 unless they specifically identify a print destination.

Priorities of Printers and Print Requests

In an environment where there is a lot of competition for a given printer or group of printers, the line printer spooling system offers several ways to prioritize print jobs so that critical print jobs can effectively "jump to the head of the line".

Both printers and print requests have priority values associated with them. Typically, print requests are handled by a printer in the order they are received. By default, print requests have the printer's default priority and are printed on a first-in-first-out basis. However, print jobs can be assigned priority values to raise or lower their priority, using the -p option of the lp command. Priority values range from 0 to 7, with 7 being the highest priority. See *lp*(1) for details.

The priority of a print request can be altered by using the lpalt command. A printer's default request priority can be set using the lpadmin command. See *lpadmin*(1M) and *lpalt*(1) for details.

If multiple print requests are waiting to be printed on a specific printer and all have priorities high enough to print, the printer will print the next print request with the highest priority. If more than one print request has the same priority, print requests with that priority will print in the order they were received by the spooler.

Similarly, a priority fence value can be assigned to each printer to set the minimum priority that a print request must have to print on that printer. A printer's **fence priority** is used to determine which print requests get printed; only requests with priorities equal to or greater than the printer's fence priority get printed. The idea is to lower the fence outside peak usage hours, perhaps via crontab scripts. For more information, see *crontab*(1M) and *cron*(1M). This would allow low priority print jobs submitted during the day to be printed in the evening when the fence priority was lowered. See *lpadmin*(1M) and *lpfence*(1M) for additional details.

Printer Logging

Every spooling system request is logged in a log file located in `/usr/spool/lp/log`. The file contains a record of each spooling system request, including request ID, user name, printer name, time, error messages, and reprints due to failure.

Summary of Line Printer Spooling System Commands

These are the commands associated with the line printer spooling system and a summary of what they do. See their respective manpages for complete details. In addition to these commands, most features of the spooling system can be controlled using one of the system administration tools such as the System Management Homepage.

lpadmin
: `lpadmin` allows you to:
  - Add/remove printers to/from the spooler
  - Define printer classes
  - Set the default priority for a printer

lpsched
: `lpsched` (the scheduler) is the heart of the line printer spooling system. When it is running, the line printer spooling system is considered to be running; when it is not running the line printer spooling system is considered to be not running. The `lpsched` command starts the scheduler running.

lpshut
: Performing the opposite function of the `lpsched` command is the `lpshut` command. `lpshut` stops the scheduler and therefore stops all printing on all printers.

lp
: The `lp` command submits a new print request to the line printer spooling system. When you want to print something from a shell environment (from a command line), use the `lp` command.

lpstat
: The `lpstat` command reports the current status of the scheduler ("running" or "stopped"), whether or not a print queue (printer or class queue) is currently accepting new print requests, whether or not printers in the spooler are currently enabled, and which print requests are queued up or printing on each print queue.

lpmove
: The `lpmove` command moves a print request from one print queue to another or to move all queued print requests on one queue to a different queue. The `lpmove` command can only be used when the scheduler is *not running*.

lpalt
: The `lpalt` command allows you to change the attributes of specified print requests. Using `lpalt` you can:
  - Move specified print requests from one queue to another *without having to stop the scheduler* (Note: the print requests cannot be actively printing when you move them)
  - Change the priority of a queued print request
  - Change the number of copies that will print (of the specified print requests)
  - Change the print options (for example, change portrait to landscape printing) for queued print requests
  - Change the title that will print on a banner page associated with specified print requests

lpfence
: Each print request has a priority (from 0 to 7) associated with it. The `lpfence` command, which can only be used when the scheduler is *not running*, specifies for a given printer the minimum priority a print request must have in order to print on the specified printer.

cancel
: Cancels queued or printing print requests.

accept
: Allows new print requests to be submitted to a print queue (associated with a printer or class).

reject	Prohibits new print requests from being submitted to a print queue (associated with a printer or class).
enable	Allows the printing of queued print requests to the specified printer.
disable	Inhibits the printing of queued print requests to the specified printer. If the print queue associated with the printer is still accepting new requests the requests will build up in the queue until the printer is again enabled. If the queue is associated with a class and other printers in the class remain enabled, disabling a specific printer only inhibits printing to that printer. Print requests coming to the class queue will still print on the other active printers in the class.

LP Spooler Integration with LDAP-UX

If you have configured your HP-UX system to use LDAP-UX Client Services, the daemon `ldapclientd` initializes printer configurator services when it starts running.

Printer configurator services will periodically scan an LDAP Directory Server for printer entries. If it finds any, it will extract the necessary information to compare against printers currently configured in the local spooler (on the client system).

If the Printer Configurator Service discovers that new entries for printers have been configured on the LDAP Directory Server, the printer configurator service will configure those new printers into the local spooler automatically.

If Printer Configurator Service discovers printer configuration entries have been removed from the LDAP Directory Server, the Printer Configurator Service will automatically remove the corresponding entries from the local spooler.

In this way, by configuring printers in one place (the LDAP server), you can automatically add printers to (or remove printers from) numerous client systems automatically.

NOTE: Even if a client system is configured to use LDAP-UX and its Printer Configurator Services, a system administrator can still configure printers in the client's spooler manually.

For More Information on Printer-Related Tasks

Refer to the following documents for additional information:

- *HP JetDirect Network Interface Configuration Guide* — for configuring network printers on the HP JetDirect Network Interface
- *LDAP-UX Client Services B.04.15 Administrator's Guide* — for details on configuring LDAP-UX Client Services and the Printer Configurator Services

Security and Access Control

HP-UX has many tools for securing your servers and data. Threats to your servers and data can be either malicious or accidental, as well as physical (fires, earthquakes, failing hardware, and so on). Or, they can be logical (misbehaving software, hacking, and so on).

For information about the tools available to protect your servers and data against loss from the threats mentioned previous, see "Data Protection Tools" (page 104).

Controlling Access to Data Using Legacy Unix File Ownership and Privileges

HP-UX has the ability to control access to directories and files using a combination of:
- User and Group ownership of files and directories
- File and Directory mode

Using these, a file or directory is assigned an owner, a group, and an access mask called a **mode**, which collectively determine:

Files   Who can read, write, or attempt to execute the file.

Directories	Who can search the contents of the directory, add files to, remove files from, or rename files in the directory, and who can cd to the directory.

There is a *lot* more to the topic of legacy Unix file ownership and privileges and there are other, more powerful, mechanisms that allow you to carefully control and monitor who is accessing the files and directories on your system. An entire volume of the *HP-UX System Administrator's Guide* is devoted to the topic of security. For extensive coverage of the topic of controlling access to the files and directories of your system and other security related topics, see *HP-UX System Administrator's Guide: Security Management*.

Controlling Access to Data Using Security Containment Technologies

Traditional UNIX file access mechanisms are adequate for many basic installations, but today's security and privacy conscious world requires a lot more control over who has access to which data.

With traditional security methods, a typical weak link in the mechanism is the superuser (or root user). The term **superuser** refers to any account with a User ID (or any program or process with an effective User ID) of "0" (zero). These special accounts allow anyone who has access to them complete access to every local file on the entire server. Should the password for a superuser account fall into the wrong hands, *the security of the entire server* becomes compromised.

In many installations, it is not desirable to give any one person access to every file on a server. In particular, the role of system administrator might be sub-divided into more specific roles that are assigned to different people. Others may need to administer specific applications, or a database or other entity. Perhaps, for security reasons, it is desirable to give a person access to certain files or capabilities only during certain hours of the day.

Technologies for Greater Access Control

HP-UX 11i version 3 has security technologies that, when used together, provide significantly greater access control of the data files and user privileges on your servers when HP-UX is running in Standard Mode:[6]

Compartments	*Compartments* isolate unrelated resources on a server to help prevent catastrophic damage to the server if one compartment is penetrated.
	When configured in a compartment, an application has restricted access to resources (processes, binaries, data files, and communication channels used) outside its compartment. This restriction is enforced by the HP-UX kernel and cannot be overridden unless specifically configured to do so. If the application is compromised, it will not be able to damage other parts of the system because it is isolated by the compartment configuration.
Fine-Grained Privileges	Traditional UNIX privileges grant "all or nothing" administrative privileges based on the effective UID of the process that is running. If the process is running with the effective UID=0, it is granted *all* privileges. With *fine-grained privileges*, processes are granted *only the privileges needed for the task* and, optionally, only for the time needed to complete the task. Applications that are privilege-aware can elevate their privilege to the required level for the operation and lower it after the operation completes.
Role-Based Access Control	Typically, UNIX system administration commands must be run by a superuser (root user). Similar to kernel level

6. These security technologies are also available in HP-UX 11i version 2. For more information on Standard Mode vs. Trusted Mode, see "Protecting Against Unauthorized Access to Your Servers and Data" (page 104).

system call access, access is usually "all or nothing" based on the user's effective UID.

HP-UX *Role-Based Access Control* (HP-UX RBAC) enables you to group common or related tasks into a role. For example, a common role might be User and Group Administration. Once the role is created, you assign to specific users a role or set of roles that enables them to run the commands defined by those roles.

When you implement HP-UX RBAC, you enable non-root users to perform tasks previously requiring superuser privileges without granting those users complete superuser privileges.

Auditing
: The HP-UX auditing system records security-related events for later analysis. Administrators use auditing to detect and analyze security breaches. Auditing is available on both Standard Mode and Trusted Mode HP-UX systems.

User Database
: Previously, all Standard Mode HP-UX security attributes and password policy restrictions were set on a system-wide basis. The introduction of the user database enables you to set security attributes on a per-user basis that overrides system defaults.

Further Information

For more information on the enhanced security containment features introduced above, see the following resources:

- *HP-UX System Administrator's Guide: Security Management*
- *HP-UX 11i Security Containment Administrator's Guide*
- The *privileges*(5) manpage

Start-up and Shutdown

Whenever you turn on (or reset) your computer, the hardware, firmware, and software must be initialized in a carefully orchestrated sequence of events known as the boot sequence. A similar sequence, known as the shutdown sequence, refers to the orderly sequence of steps needed to halt HP-UX. The shutdown sequence ensures all running processes are properly stopped and any data in memory that needs to be written to disk is not lost when the operating system is halted and power to the server is lost.

Run Levels

When HP-UX (or any operating system) is up and running, it is said to be booted. When HP-UX is not running, it is said to be halted. HP-UX, like most Unix based operating systems, has several levels of the "booted" state known as *run levels*. As HP-UX starts up or shuts down, it transitions through the various run levels until it reaches its targeted run level. The various run levels determine what aspects of HP-UX are running.

At boot time, a daemon known as `init` is started. Its primary role is to create processes from a script stored in the file`/etc/inittab` (see *inittab*(4)). The`/etc/inittab` file is one of the mechanisms used to configure what aspects of HP-UX are running at any given run level. The `inittab` file can also specify the initial run level that the system will boot to.

The following list describes the general characteristics of each HP-UX run level:

run level 0
: When run level 0 is initiated, HP-UX transitions from whatever run level it is currently in through all lower run levels and halts. In the process of

transitioning down through the run levels it cleanly terminates all running processes and writes any memory based information to disk, ensuring properly structured file system linkages on disk.

run level s — Also known as single user mode, run level s allows input only from the terminal (or pseudo-terminal) known as the system console. This allows one user, usually a system administrator, to have exclusive access to the server usually for maintenance operations that must be done on a quiescent system.

By default, in run level s, only the root file system is mounted and many subsystems such as the line printer spooling system and networking are not running.

> **NOTE:** There is a different but similar run level known as S (upper-case S). It is functionally the same as run level s (lowercase) with the exception that the capabilities of the true system console are switched to the terminal where you are logged in making it a virtual system console. With modern remote access to a server through its management processor, the distinction between run levels s and S is pretty much semantic.

run level 1 — Just above run level s is run level 1. In run level 1 the system is still dedicated to one user but all file systems are mounted and a process known as the syncer is running. The syncer periodically writes any cached memory based file system changes to disk to make sure that the disk based view of a file system's state matches that of the memory based view of the file system's state. See *sync*(1M).

run level 2 — Multiuser level. Run level 2 is the first of the run levels that allow multiple users to log in simultaneously from different locations. Run levels 3, 4, 5, and 6 also allow this and each of those run levels adds additional capabilities over all previous run levels.

run level 3 — At run level 3, the ability to export NFS file systems is activated. If your server has file systems that you need to access from other servers via NFS mounts, use at least run level 3. Additionally, web-based administration and graphical presentation managers such as CDE begin at run level 3.

run level 4 — Currently undefined. Available for user customization.

run level 5 — Currently undefined. Available for user customization.

run level 6 — Currently undefined. Available for user customization.

> **NOTE:** As described in the previous list, run levels appear to be additive, and based on the default contents of the /etc/inittab file, they generally are. However, it is possible to have processes started at a lower run level that are not available at higher run levels. Each process represented in the /etc/inittab file specifies at which levels it will be active.

Startup and Kill Scripts (Run Level Transitions)

In the past, much more of the system startup process was configured in the/etc/inittab file. Currently, most system services are started and stopped by the /sbin/rc daemon which is called by init each time you change the system run level.

/sbin/rc (the "rc daemon") performs the following actions:

1. Runs the script /sbin/rc.utils which is responsible for preparing your system console to display the one-line messages you see on the system console during run level transitions.

/sbin/rc.utils also logs output from startup and shutdown scripts to the file /etc/rc.log.

2. The rc daemon then runs/etc/rc.config which processes all scripts in the directory /etc/rc.config.d. The scripts in /etc/rc.config.d set variables that control the execution of the startup and shutdown scripts that are subsequently run by the rc daemon.

IMPORTANT: You control what the startup and shutdown (kill) scripts do by setting variables in their corresponding scripts in the /etc/rc.config.d directory.

Do not edit the scripts in the /sbin/init.d *directory directly* (These scripts might be replaced during a patch installation or product update, causing you to lose your changes).

---

3. Searches the appropriate/sbin/rc#.d directory for scripts to run (and runs them).
   - If the run level being transitioned to is *higher* than the current run level, then the # in rc#.d represents the run level one *higher* than the current run level and the scripts in the rc#.d directory with names beginning with "S" are run.
   - If the run level being transitioned to is *lower* than the current run level, then the # in rc#.d represents the run level one *lower* than the current run level and the scripts in the rc#.d directory with names beginning with "K" are run.

   This step is repeated for each run level between the current run level and the target run level.

4. Each startup (or kill) script is first run with the *start_msg* (or *stop_msg*) parameter causing it to output its one line message on the system console, and then run again with the *start* (or *stop*) parameter to perform its function (based on what variables were previously set from the file in the /etc/rc.config.d directory).

Example 3-4 Run Level Transition Examples

The following two examples show what happens during two typical situations:

Transition up
The file `/etc/inittab` contains an entry telling `init` that the initial run level for the system during boot-up should be run level 3:

```
init:3:initdefault:
```

To reach run level 3, the system transitions:

- From run level 0 (the halted state)
- to run level 1 (running scripts pointed to by links in the `/sbin/rc1.d` directory whose names begin with the letter S — for example, `/sbin/rc1.d/S100localmount`, `/sbin/rc1.d/S520syncer`, and others).
- to run level 2 (running scripts pointed to by links in the `/sbin/rc2.d` directory whose names begin with the letter S — for example, `/sbin/rc2.d/S500inetd`, `/sbin/rc2.d/S900samba`, and others).
- and finally to run level 3 (running scripts pointed to by links in the `/sbin/rc3.d` directory whose names begin with the letter S — for example, `/sbin/rc3.d/S823hpws_webmin`, `/sbin/rc3.d/S823hpws_webproxy`, and others).

Transition down
If HP-UX is currently in run level 3 and an system administrator with appropriate privileges executes the command:

`/sbin/init 1`

The system will transition:

- From run level 3
- to run level 2 (running scripts pointed to by links in the `/sbin/rc2.d` directory whose names begin with the letter K — for example, `/sbin/rc2.d/K177hpws_tomcat`, and others).
- and finally to run level 1 (running scripts pointed to by links in the `/sbin/rc1.d` directory whose names begin with the letter K — for example, `/sbin/rc1.d/K500inetd`, and others).

Commands for Manipulating System Run Levels

The following commands can be used to set, change, and view HP-UX run levels:

init
`init` is both a daemon and a command.

The `init` *command* interacts with the `init` *daemon*. You use the `init` command to set or change run levels.

The `init` *daemon*, started at boot time spawns processes as defined in the `/etc/inittab` file. These processes in turn control how HP-UX interacts with the outside world (for example, which terminals to accept input from, and whether or not to export local file systems via NFS for use by other servers).

NOTE: If your goal is to transition HP-UX to single-user mode from a higher run level, do not use `init s`. This could leave processes running and disks mounted that you do not want present.

Use the `shutdown` command with no parameters to transition to run level s, or to be absolutely certain no undesirable processes or mounted file systems are present, reboot the system to single-user mode by interrupting the boot process and using the secondary boot loader (`hpux.efi` for Integrity servers or `hpux` for HP 9000 servers) to override the default run level.

who -r   The `-r` option of the who command displays the current system run level, the date and time the current run level was entered, and three state fields representing the current run level, how many times that run level was previously entered (since the system was booted), and the previous run level (from which the current run level was entered).
Example:

who -r

    .   run-level 3 Jun 27 06:22 3   1   4

This output indicates:

- The system is currently in run level 3.
- It entered the current run level on June 27th at 22 minutes after six in the morning.
- The current run level is (3) was previously entered one (1) time since the system was last booted, and the current run level was entered from the previous run level which was four (4).

Starting (Booting) HP-UX

HP-UX based systems go through the following sequential steps when you power them on or reset them:

1. Hardware and/firmware-based routines on-board the processors and I/O cards perform self-tests and initialize those items along with enough memory to continue the boot process. They also locate and initialize communications with console display and keyboard devices, and a boot device.
2. Pre-boot firmware/software routines then load and execute the HP-UX boot loader.
3. The HP-UX boot loader:
   — Locates, opens, and reads the kernel file and copies the kernel into memory
   — Initiates the HP-UX kernel
4. HP-UX goes through its initialization process and begins normal operation.

For complete details on the HP-UX boot process and its possible variations, see *HP-UX System Administrator's Guide: Routine Management Tasks*.

Stopping (Shutting Down) HP-UX

"READY . . . SET . . . GO!" As with the famous phrase, there is a definite order that you should follow to shut down your system—or you may encounter problems.

When shutting down an HP-UX system:

1. First, notify everyone who is likely to be affected by the shutdown, giving them a chance to complete work in progress, and if necessary unmount file systems that were NFS-mounted from your system.
2. Then, shutdown any programs you might be running that would not be safely shutdown by one of the system's kill scripts ("Startup and Kill Scripts (Run Level Transitions)").
3. Finally, use the `shutdown` command to shut down the system. The `shutdown` command:
   a. allows you to notify the users of the system of the shutdown in progress if you have not previously done so, or to remind those users that the shutdown is imminent.
   b. transitions backward through the run levels (executing the kill links in the directories `/sbin/rc[0-4].d`)
   c. and finally calls `reboot` to perform a `sync` operation that insures memory structures are written to disk before memory is overwritten by the subsequent boot.

For details on the HP-UX shut down process, see *HP-UX System Administrator's Guide: Routine Management Tasks*.

Abnormal Shutdowns (System Crashes)

When your system crashes, it is important to know why so that you can take actions to prevent it from happening again. Sometimes, it is easy to determine why: for example, if somebody trips over the cable connecting your computer to the disk containing your root file system (disconnecting the disk).

At other times, the cause of the crash might not be so obvious. In extreme cases, you might want or need to analyze a snapshot of the computer's memory at the time of the crash, or have HP do it for you, in order to determine the cause of the crash.

Overview of the Dump / Save Cycle

When the system crashes, in order to preserve the evidence of what caused the crash, HP-UX tries to save the image of physical memory, or certain portions of it, to predefined locations called dump devices. When the system is subsequently rebooted, a special utility copies the memory image from the dump devices to the HP-UX file system area.

Figure 3-8 The Crash Dump Sequence
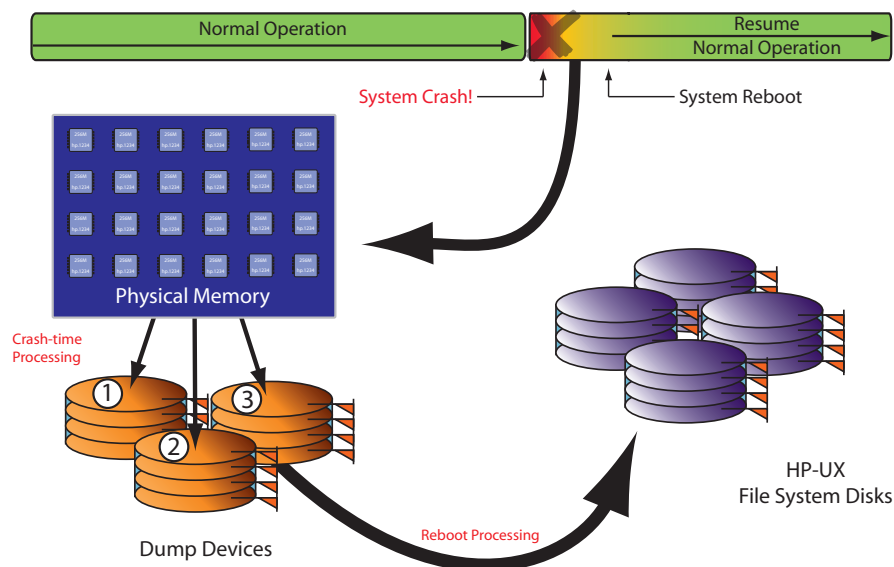
When the memory image is in the HP-UX file system, you can analyze it with a debugger or save it to removable media for shipment to someone else for analysis.

There are multiple ways that dump devices can be configured:

- In the kernel
- During system initialization when the initialization script for `crashconf` runs (and reads entries from the `/etc/fstab` file)
- During run time, by an operator or administrator manually running the `/sbin/crashconf` command

Preparing for a System Crash

The dump process exists so that you have a way of capturing what your system was doing at the time of a crash. This is not for recovery purposes; processes cannot resume where they left off following a system crash. Rather, this is for analysis purposes in order to help you determine why the system crashed and hopefully prevent it from happening again.

If you want to be able to capture the memory image of your system when a crash occurs (for later analysis), you need to define in advance the location(s) where HP-UX will put that image at the time of the crash. This location can be on local disk devices or logical volumes.

Wherever you decide that HP-UX should put the dump, it is important to have enough space at the dump location (see "How Much Dump Space You Need" (page 79)). If you do not have enough space, not every page selected to be dumped will be saved and you might not capture the part of memory that contains the instruction or data that caused the crash.

If necessary, you can define more than one dump device so that if the first one fills up, the next one is used to continue the dumping process until the dump is complete or no more defined space is available. Beginning with HP-UX 11i version 3 you can even configure multiple dump devices to be written to in parallel (rather than one after the other), significantly cutting down dump times.

How Much Dump Space You Need

To guarantee that you have enough dump space, define a dump area that is at least as big as your computer's physical memory, plus one megabyte. If you are doing a selective dump (which is the default dump mode in most cases), much less dump space will actually be needed. Full dumps require dump space equal to the size of your computer's memory plus a little extra for header information.

In HP-UX Release 11i compressed dumps are enabled by default, however, dump compression will only occur if conditions in the crash environment are favorable. Do not plan your dump storage space based on potential compression; allow enough space for an uncompressed full or selective dump. For more information on compressed dumps, see "Compressed Dumps" (page 81).

Dump Configuration Decisions

As computers continue to grow in speed and processing power, they also tend to grow in physical memory size. Where once a system with 256MB of memory was considered to be a huge system, today it is barely adequate for most tasks. Some of today's HP-UX systems can have terabytes of memory. This is important to consider because the larger the size of your computer's physical memory the longer it will take to dump its contents following a system crash (and the more disk space the dump will consume).

Usually, when your system crashes it is important to get it back up and running as fast as possible. If your computer has a very large amount of memory, the time it takes to dump that memory to disk might be unacceptably long when you are trying to get the system back up quickly. And, if you happen to already know why the computer crashed (for example, if somebody accidently disconnected the wrong cable), there's little or no need for a dump anyway.

With HP-UX, a runtime dump subsystem gives you a lot more control over the dump process. With it you can override dump definitions configured into the kernel while the system is running. An operator at the system console can even override the runtime configuration as the system is crashing.

You have control over the following crash dump features:

- Which classes of memory get dumped.
- Run-time crash dump configuration. It is no longer necessary to build your dump configuration into the kernel file or to reboot the system to change the crash dump configuration.
- Whether or not a dump is compressed.

These capabilities give you a lot of flexibility, but you need to make some important decisions regarding how you will configure your system dumps.

There are three main criteria to consider. Select which of these is most important to you and read the corresponding section. The criteria are:

- System Recovery Time
- Crash Information Integrity
- Disk Space Needs

System Recovery Time

Use this section if your *most important* criterion is to get your system back up and running as soon as possible. The factors you have to consider here are:

- Dump Level: Full Dump, Selective Dump, or No Dump
- Concurrent Dumps
- Compressed Save versus Noncompressed Save
- Using a Device for Both Paging and Dumping (System Recovery Time)
- Partial Saves

Dump Level: Full Dump, Selective Dump, or No Dump

In addition to being able to choose "dump everything" or "dump nothing," you have the ability to determine which classes of memory pages get dumped, allowing you to capture important memory structures without having to dump the whole contents of memory.

You are reading this section because system recovery time is critical to you. Obviously, the fewer pages your system needs to dump to disk (and on reboot copy to the HP-UX file system area), the faster your system can be returned to service. Therefore, when system recovery time is critical avoid using the *full dump* option.

When you define dump devices, whether in a kernel build or at run time, you can list which classes of memory must always get dumped and which classes of memory should not be dumped. If you leave both of these lists empty HP-UX will decide for you which parts of memory should be dumped based on what type of error occurred. In nearly all cases it is best to let HP-UX determine which pages to dump.

IMPORTANT: You can interrupt the dump at any time by pressing the **ESC** (escape) key. It can take as long as 15 seconds to abort.

If you interrupt a dump, it will be as though a dump never occurred; that is, you will *not* get a partial dump.

Even if you have defined that you do not want a full dump to be performed, an operator at the system console at the time of a crash can override those definitions and request a full dump.

Likewise, if at the time of a crash you know what caused it (and therefore do not need the system dump) but have previously defined a full or selective dump an operator at the system console at the time of a crash can override those definitions and request that no dump be performed.

Concurrent Dumps

On servers with very large amounts of memory, the process of writing memory contents to disk can take a very long time. If you have multiple devices configured to receive the memory dump you can configure HP-UX to split the task of dumping memory and write to the multiple devices in parallel. This process is called *dump concurrency* and is configured using either the kernel tunable dump_concurrent_on (see *dump_concurrent_on*(5)), or the crash-processing configuration command crashconf (see *crashconf*(1M)).

> **NOTE:** Concurrent dump performance improvements are not likely to occur on systems with only one instance of any of the crash dump resources (for example, only one dump device or only one core). And, concurrent dump performance improvements are currently supported only on HP Integrity servers.

Compressed Dumps

Following a system crash, the HP-UX operating system can use this feature to compress data from memory *before it writes the data to the dump device*. Compression decreases the volume of crash data, making the dump times faster.

By reducing the time required to store the entire dump the recovery period is shorter and your system can be returned to service much sooner. Dump compression provides a greater time saving on systems that have large amounts of memory.

- Dump compression is not forced, it is only a user request that will be honored *if possible*.

  At the time of a system crash the dump subsystem examines the state of the system and its resources to determine whether it is possible to use compression. Depending on the resources available, HP-UX determines dynamically whether to dump using compressed or uncompressed format.

  (For example if the processor that is processing the crash fails to assign a sufficient number of processors to do the compression, the dump will not be compressed. A recursive crash, such as a panic during the processing of a previous dump, also causes the system to dump using uncompressed format.)

- For selective dumps that exclude *unused* pages, you can expect the dump to take about one-third the time of uncompressed dumps on the same server. This interval includes the time required to run the savecrash program and write the dump to its final storage location on the HP-UX file system. A dump that previously took 3 hours to complete should now take only about an hour.

- You can use the crashconf command (see *crashconf*(1M)) to disable or enable compressed dumps. (Compression is configured into the kernel by default.) During a crash event you can also choose to override the previously defined dump compression setting.

  Normally, there is no benefit in disabling compression unless the initial (compressed) dump is corrupt and you want to attempt an uncompressed dump on a subsequent crash event. Compressed saves (to the HP-UX file system area) are only possible with sequential dumps.

- Using the command crashutil, you can convert the compressed dump file to any of several dump formats for storage and analysis. See the manpage *crashutil*(1M) for detailed information on how to do this and what dump formats are available.

- A compressed dump file requires less disk storage space and creates a smaller tar file that takes less time to copy to tape or transmit for analysis, for example via ftp)

- If your server uses virtual partitions (vPars), the dump might not be compressed but the dump process will proceed.
- If more than one crash occurs in close succession, it might not be possible for HP-UX to compress the dump.

Compressed Save versus Noncompressed Save

System dumps can be very large, so large that your ability to store them in your HP-UX file system area can be taxed.

The boot time utility called `savecrash` can be configured (by editing the file /etc/ rc.config.d/savecrash) to compress or not compress the data *as it copies the memory image from the dump devices to the HP-UX file system area* during the reboot process. This has system recovery time implications in that compressing the data can take longer if the saving occurs as foreground processing (for example, when HP-UX is trying to quickly evacuate a dump device that is also used for paging). So, if you have the disk space and require that your system be back up and running as quickly as possible, configure `savecrash` to not compress the data.

Using a Device for Both Paging and Dumping (System Recovery Time)

It is possible to use a specific device for both paging (swap space) and as a dump device. However, if system recovery time is critical to you do *not* configure the primary paging device as a dump device. From the *savecrash*(1M) manpage:

- "By default, when the primary paging device is not used as one of the dump devices or after the crash image on the primary paging device has been saved, savecrash runs in the background. This reduces system boot-up time by allowing the system to be run with only the primary paging device."

Another advantage to keeping your paging and dump devices separate is that paging will not overwrite information stored on a dump device, no matter how long the system has been up or how much activity has taken place. Therefore, you can prevent `savecrash` processing at boot time (by editing the file /etc/rc.config.d/savecrash). This can save you a lot of time at boot time by allowing you to save the memory image after the server has been returned to service. After the server is up and running you can run `savecrash` manually to copy the memory image from the dump area to the HP-UX file system area.

Partial Saves

If a memory dump resides partially on dedicated dump devices and partially on devices that are also used for paging, you can choose to save (to the HP-UX file system) only those pages that are endangered by paging activity. Pages residing on the dedicated dump devices can remain there. If you know how to analyze memory dumps, it is even possible to analyze them directly from the dedicated dump devices using a debugger that supports this feature.

Before sending your memory dump to someone else for analysis you must move the dumped pages from the dedicated dump devices to the HP-UX file system. You can then use a utility such as `pax` or `tar` to bundle them up for shipment.

Crash Information Integrity

Use this section if your *most important* criterion is to make sure you capture the part of memory that contains the instruction or piece of data that caused crash. The factors you have to consider here are:

- Full Dump versus Selective Dump
- Dump Definitions Built into the Kernel
- Using a Device for Both Paging and Dumping (Crash Integrity)

Full Dump versus Selective Dump

You have chosen this section because it is critical to you to capture the specific instruction or piece of data that caused your system crash. The only way to *guarantee* that you have it is to capture everything. This means selecting to do a *full* dump of memory.

Be aware, however, that this can be costly from both a time and a disk space perspective. From the time perspective, it can take quite a while to dump the entire contents of memory from an HP-UX instance using very large amounts of memory. It can take an additional large amount of time to copy that memory image to the HP-UX file system area during the reboot process.

From the disk space perspective, if you have large amounts of memory (some HP-UX servers can have terabytes of memory), you will need an amount of dump area at least equal to the amount of memory in your system; and, depending on a number of factors, you will need additional disk space in your HP-UX file system area equaling the amount of physical memory in your system, in the worst case.

Dump Definitions Built into the Kernel

You can configure HP-UX dump devices using one or more of the following methods:

- **Preferred Method:** At run time using the `/sbin/crashconf` command
- At boot time (entries defined in the `/etc/fstab` file)
- During kernel configuration (put the definitions in the `/stand/system` file). *This method is obsolescent and should no longer be used!*

Definitions at each of these places add to or replace any previous definitions from the other sources. However, consider the following situation:

Example 3-5 Example of a Crash During the Very Early Stages of the Boot Process

Consider a server that has ten gigabytes (10 GB) of physical memory. If you were to define system dump devices with a total of two gigabytes (2 GB) of space in the kernel file, and then define an additional nine gigabytes (9 GB) of disk space in the `/etc/fstab` file, you would have enough dump space to hold the entire memory image (a full dump) by the time the system was fully up and running.

But, what if a crash occurs before `/etc/fstab` is processed? Only the amount of dump space already configured will be available at the time of the crash; in this example, two gigabytes of space.

If it is *critical* to you to capture every byte of memory *in all instances*, including the early stages of the boot process, use `crashconf` with the `-s` option (which tells `crashconf` to retain dump device definitions across reboots) to define enough dump space in advance to account for this. `crashconf` is the preferred method for defining dump devices in HP-UX 11i version 3.

NOTE: The preceding example is presented for completeness. The actual amount of time between the point where kernel dump devices are activated, and the point where runtime dump devices are activated is very small (a few seconds), so the window of vulnerability for this situation is very small.

Using a Device for Both Paging and Dumping (Crash Integrity)

It is possible to use a specific device for both paging purposes and as a dump device. But, if crash dump integrity is critical to you, this is not recommended. From the *savecrash*(1M) manpage:

- "If `savecrash` determines that a dump device is already enabled for paging, and that paging activity has already taken place on that device, a warning message will indicate that the dump may be invalid. If a dump device has not already been enabled for paging, `savecrash` prevents paging from being enabled to the device by creating the file `/etc/savecore.LCK`. `swapon` does not enable the device for paging if the device is locked in `/etc/savecore.LCK`."

So, if possible, avoid using a given device for both paging and dumping: particularly the primary paging device!

HP-UX systems configured with small amounts of memory and using only the primary swap device as a dump device are in danger of not being able to preserve the dump (copy it to the HP-UX file system area) before paging activity destroys the data in the dump area. HP-UX systems configured with larger amounts of memory are less likely to need paging (swap) space during startup, and are therefore less likely to destroy a memory dump on the primary paging device before it can be copied.

Disk Space Needs

Use this section if you have *very limited disk resources* for the post-crash dump and/or the post-reboot save of the memory image to the HP-UX file system area. The factors you have to consider here are:

- Dump Level
- Compressed Save versus Non-compressed Save
- Partial Save (`savecrash -p`)

Dump Level

You are reading this section because disk space is a limited resource on your server. Obviously, the fewer pages that you have to dump, the less space is required to hold them. Therefore, unless your server also has a small amount of physical memory, a full dump is not recommended. If disk space is very limited, you can always choose no dump at all.

However, there is a happy medium, and it happens to be the default dump behavior, which is called a *selective dump*. HP-UX does a pretty good job of determining which pages of memory are the most critical for a given type of crash, and saves only those. By choosing this option, you can save a lot of disk space on your dump devices, and again later, in your HP-UX file system area. For instructions on how to do this, see *HP-UX System Administrator's Guide: Routine Management Tasks*.

Compressed Save versus Non-compressed Save

Regardless of whether you choose to do a full or selective save, whatever is saved on the dump devices usually needs to be copied to your HP-UX file system area before you can use it.

If the disk space shortage on your system is in the HP-UX file system area (not in the dump devices), you can choose to have `savecrash` (the boot time utility that does the copy) compress your data as it makes the copy.

Partial Save (`savecrash -p`**)**

If you have plenty of dump device space but are limited on space in your HP-UX file system, you can use the `-p` option to the `savecrash` command. This command copies only those pages on dump devices that are endangered by paging activity (the pages residing on devices that are being used for both paging and dumping). Pages that are on dedicated dump devices are not copied.

> **NOTE:** It is possible to analyze a crash dump directly from dump devices using a debugger that supports this feature[7]. But, if you need to save it to removable media or send it to someone you will first need to copy the memory image to the HP-UX file system area.

For More Information on Defining Dump Devices

The following resources have additional information on defining dump devices:

- The manpage *crashconf*(1M) describes the primary command used to configure crash dumps.
- The manpage *savecrash*(1M) describes the various options for saving crash dumps to a file system area for later analysis or archiving.
- The manpage *crashutil*(1M) describes the utility for converting crash dumps into various formats for later analysis. Similar to `savecrash`, `crashutil` can also be used to retrieve crash dump information from raw dump devices into the HP-UX file system area.
- The *HP-UX System Administrator's Guide: Routine Management Tasks* (Chapter 2: Booting and Shutdown) has a list of additional relevant manpages in the Configuring Dump Devices section.

What Happens When the System Crashes
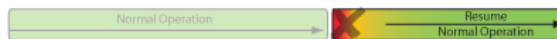
An HP-UX system crash (system panic) is an unusual event. When a panic occurs, it means that HP-UX encountered a condition that it did not know how to handle (or could not handle). Sometimes you know right away what caused the crash. Other times the cause is not readily apparent. It is for this reason that HP-UX is equipped with a dump procedure to capture the contents of memory at the time of the crash for later analysis.

You define in advance:

- Where you want memory contents dumped (**dump devices**)
- Whether or not you want the dump to be compressed to save space on your dump devices (**dump compression**)
- Whether or not to dump to multiple devices in parallel to save time, allowing (*dump concurrency*)

Use the `/sbin/crashconf` command to configure these options. See the *crashconf*(1M) for details on how to configure the various options.

Operator Override Options

When a HP-UX panics, the current dump control option settings are displayed at the system console during a crash. You then have 10 seconds to interact with the system console before the current settings are used to proceed with dump processing.

If you choose to interact with the system during the 10-second override period, follow the on-screen prompts.

---

7. Analyzing crash dumps is not a trivial task. It requires intimate knowledge of HP-UX internal structures and the use of debuggers. It is beyond the scope of this document to cover the actual analysis process. If you need help analyzing a crash dump, contact your HP representative.

You can choose to do the following:

C option   **[CURRENT]** Proceed with the current settings. Use this to immediately proceed with the current settings, not waiting for the 10-second override period to automatically expire.

S option   **[SELECTIVE]** Proceed with a selective dump with both compression and concurrency turned off, regardless of what was previously configured.

F option   **[FULL DUMP]** This option is available if enough dump space has been configured to hold the contents of the entire physical memory. Select this option to dump the contents of all physical memory. With this option, compression and concurrency are off.

P option   **[PARTIAL DUMP]** This option is available instead of the full dump option if there is not enough dump space configured to hold a full dump. The amount of memory that will be dumped is displayed on the console. With this option, compression and concurrency are off.

N option   **[NO DUMP]** Do not perform a dump. Immediately reboot the system. Use this option if you know the cause of the panic and do not need a dump.

The Dump

After the operator is given a chance to override the current dump level, or the 10-second override period expires, HP-UX writes the physical memory contents to the dump devices until one of the following conditions is true:

- The entire contents of memory are dumped (if a full dump was configured or requested by the operator)
- The entire contents of selected memory pages are dumped (if a selective dump was configured or requested by the operator)
- Configured dump device space is exhausted

Depending on the amount of memory being dumped, and a number of other factors, this process can take from a few seconds to hours.

While the dump is in occurring, status messages on the system console indicate the dump's progress.

IMPORTANT: You can interrupt the dump at any time by pressing the **ESC** (escape) key. It can take as long as 15 seconds to abort.

If you interrupt a dump, it will be as though a dump never occurred; that is, you will *not* get a partial dump.

Following the dump, the system attempts to reboot.

The Reboot

After the dumping of physical memory pages is complete, the system attempts to reboot (if the AUTOBOOT flag is set). For information on the AUTOBOOT flag, see *HP-UX System Administrator's Guide: Routine Management Tasks*.

The savecrash Processing Option

You can define whether or not you want a process called savecrash to run as your system boots. This process copies (and optionally compresses) the memory image stored on the dump devices to the HP-UX file system area. Space permitting, you can store multiple crash dumps in the *file system area* in case there is more than one panic event. If you do not run savecrash during or shortly after boot, you risk only having the latest dump available, *on the dump devices*.

Dual-Mode Devices (dump / swap)

By default, `savecrash` is enabled and performs its copy during the boot process. You can disable this operation by editing the `/etc/rc.config.d/savecrash` file, setting the `SAVECRASH` environment variable to a value of `0`. This is generally safe to do if your dump devices are not also being used as paging devices.

From the *savecrash*(1M) manpage:

- "If there is insufficient space in the file system for the portions of the crash dump that need to be saved, `savecrash` will save as much as will fit in the available space. (Priority is given to the index file, then to the kernel module files, and then to the physical memory image.) The dump will be considered saved, and `savecrash` will not attempt to save it again, unless there was insufficient space for any of the physical memory image. (See the description of option `-r`.)"

The `-r` option to `savecrash` allows you to resave a crash that has already been marked as saved. If a save fails (or if only a partial save was made) due to lack of file system space, you have a chance, once the system is running again, to clean up the file system in order to gain the space you need for the `savecrash` operation; or you can run the `savecrash` command manually, specifying an alternate destination for the saved data.

> △ **CAUTION:** If you are using your devices for both paging and dumping, *do not disable the savecrash boot processing* or you will lose the dumped memory image to subsequent system paging activity.

What to Do After the System Has Rebooted

After your system is rebooted, one of the first things you need to do is to be sure that the physical memory image that was dumped to the dump devices is copied to the HP-UX file system area so that you can either package it up and send it to an expert for analysis, or analyze it yourself using a debugger.

> 📝 **NOTE:** It is possible to analyze a crash dump directly from dump devices using a debugger that supports this feature. But if you need to save it to removable media, or send it to someone, you first need to copy the memory image to the HP-UX file system area.

Unless you specifically disable `savecrash` processing during reboot, the `savecrash` utility will copy the memory image for you during the reboot process. The default HP-UX directory that it will put the memory image in is `/var/adm/crash`. You can specify a different location by editing the file `/etc/rc.config.d/savecrash` and setting the environment variable called `SAVECRASH_DIR` to the name of the directory where you would like the dumps to be located. Just be sure the destination has enough disk space to hold the copied memory image.

LiveDump (Memory Dumps of Running Systems)

Beginning with HP-UX 11i version 3, it is possible to perform dump processing on HP Integrity servers *without crashing the system* and *without causing instability in the system*.

Live dumps (memory dumps of systems that have not crashed and are still running) can be useful:

- For analyzing what caused some recoverable operating system faults
- In the event of failures with dynamically loadable kernel modules that do not destabilize the kernel (for example, troubleshooting certain problems with I/O drivers)
- For debugging problems with system performance degradation
- For analyzing a snapshot of a running kernel off line.

Live dumps can be user initiated or kernel initiated. Users with appropriate privileges can initiate a live dump using the `livedump` command (see *livedump*(1M) for details on how to do this).

Limitations of Live Dumps

There are some key differences between live dumps and true crash dumps, some of which represent limitations:

- The key difference between a live dump versus a crash dump is, of course, that HP-UX continues to run while the dump is in progress.

  The good news is that users of the system continue to work, and are unaffected by the `livedump` process.

  The bad news is that, because HP-UX continues to run, data structures within the operating system continue to change during the dump process yielding a less accurate picture of the system at the time the dump was initiated. Therefore, the dump saved by `livedump` can contain some data structures in an inconsistent state. Also, the cause of an event you are troubleshooting might be overwritten (and therefore lost) by ongoing system operations.

- Do not use `livedump` on systems that have experienced operating system faults that destabilize the whole system. If a system has become unstable, it is probably better to let it crash on its own or force a crash using a **transfer of control** (TOC)

Operating System and Software (Installation, Modification, and Removal)

Software is distributed in many forms (tar files, zip archives, and so on). HP-UX supports many of the utilities that are used to distribute software; however, in the world of HP-UX, one format is preferred above all others: Software Distributor (SD).

Software Distributor

Software Distributor is a collection of tools used for installing, maintaining, and distributing applications and other software on HP-UX servers. It is the format HP uses to distribute HP-UX and many other HP-UX related applications.

In addition to being a distribution format — which can reside on top of `tar`, `cpio`, ISO_9660, HFS, VxFS, and most other physical file storage methods — Software Distributor maintains an *Installed Product Database* that contains information about what applications and software are currently installed on servers, version information about that software, and other important attributes about the installed software. The installed product database is used by the SD utilities to maintain the applications on a server.

Software Distributor is multi-server aware. Software packages (known as *software depots*) can be maintained on one server and used by another server to install the associated packages.

The section "Software Distributor (SD)" (page 100) describes the individual components that make up Software Distributor. The document *Software Distributor Administration Guide* fully describes this technology.

Other Software Operating System Installation Technologies

There are several other pieces of the software installation picture:

Ignite/UX
: If you need to install a common set of HP-UX software on multiple systems and would like to do it at one time, from one location, you can use Ignite/UX. See "Ignite-UX" (page 93).

update-ux
: The `update-ux` command updates the HP-UX operating system from new HP-UX media. See the manpage *update-ux*(1M) for details.

Dynamic Root Disk
: Dynamic Root Disk is an HP-UX system administration tool set used to clone an HP-UX system image to a disk other than that from which the system is currently booted, for purposes of software maintenance and recovery. In this way, you can install software and patches to a clone of your current system *without affecting the running system*. Then,

when an opportunity to reboot the system is available, the patched or updated clone can be booted. This reduces system downtime to just the time it takes to reboot to the clone. Also, if necessary, the changes can be quickly backed out by simply rebooting to the original clone.

Software Manager
Software Manager is used by Ignite-UX and Update-UX to perform software installation. Software Manager improves software selection, provides support for OEs, and provides update support for preview and a terminal user interface (TUI). See the manpages *swm*(1M) and *swm-oeupdate*(1M) for more details.

Further Reading about Software Maintenance on HP-UX

See also the following manpages for additional information about various software packaging utilities supported by HP-UX:

ar(1)
The `ar` command maintains groups of files combined into a single archive file. Its main use is to create and update library files as used by the link editor (see *ld*(1)). It can be used, however, for any similar purpose.

cpio(1)
The `cpio` command saves and restores archives of files on magnetic tape, other devices, or a regular file, and copies files from one directory to another while replicating the directory tree structure.

gzip(1)
The `gzip` command compresses files to save disk space. It can compress single files or whole directory structures, packaging the files of the directory structure into a single archive in the process.

pax(1)
The `pax` command extracts, writes, and lists archive files and copies files and directory hierarchies. A more contemporary utility, `pax` performs basically the same functions as the older (still available) utilities `cpio` and `tar`.

sd(5)
sd — overview of Software Distributor: commands to create, distribute, install, monitor, and manage software.

shar(1)
The `shar` command bundles the named files and directories into a distribution package suitable for mailing or moving to a new location. Use the posix shell (`/usr/bin/sh`) to unpack the archive. If any named files contain unusual data, `shar` uses uuencode to protect the data from certain mailers that do not properly handle the file's native format. See *uuencode*(1). If uuencode is used to encode the data, a `uudecode` script is included in the shar package so that the shell can still be used to unpack the archive.

tar(1)
The `tar` command saves and restores archives of files on a magnetic tape or in a disk file.

Networking Services

HP-UX provides a rich and robust set of networking services covering:
- Electronic Mail
- Remote Logins / Terminal Emulation
- File Transfers
- Web Access
- Remotely Mounted File Systems

Electronic Mail

Electronic mail (E-mail) systems generally have two functional layers:

- A transport and routing layer
- An E-mail client for reading, composing, and sending E-mail

For transporting and routing E-mail, HP-UX supports sendmail, a highly configurable and widely used E-mail transport service. For information on configuring sendmail on an HP-UX based server see the networking document: *HP-UX Mailing Services Administrator's Guide*.

For reading, composing, and sending E-mail HP-UX offers the traditional UNIX E-mail clients:

mail

mailx

elm

Though not formally supported by HP, you can also use the following commonly used utilities for reading, composing, and sending electronic mail:

Mozilla   Mozilla is shipped with HP-UX as it is needed as a default browser for accessing the System Management Homepage. Mozilla includes a built-in E-mail client.

rmail   If you have installed and use the GNU Emacs editor, you can use its built in E-mail client, rmail.

Remote Logins / Terminal Emulation

From an HP-UX shell you can log into a remote system using any of the following protocols if they have been made available to you:

rlogin   rlogin, a utility that runs on your local server, communicates with a daemon called rlogind on the remote server, if it is running, to allow you to log into the remote server if you have a valid account on that system.

> **IMPORTANT:** The protocol supporting rlogin is not a secure protocol. Your login information, including passwords and other information you enter during your login session is sent over the network *unencrypted*! That information is at risk of being intercepted and misused.

ssh   ssh (Secure Shell) provides much more secure remote logins than does the rlogin protocol. It authenticates the remote server (and allows the remote server to authenticate your local server) by using public-key encryption. It uses encryption in all communications with the remote server during your login session.

telnet   As with rlogin, telnet generally uses unencrypted communications making your login sessions to remote servers vulnerable to interception. If you are working on a closed network, among servers and users you trust, telnet is available on HP-UX for your use. If secure communications between your local server and the remote server you are trying to login to are important, use ssh.

File Transfers

There are several protocols available on HP-UX to transfer files between computers. Depending on the protocol used, files can be transferred between HP-UX based systems, between an HP-UX based system and a Linux based system, or between an HP-UX based system and a Microsoft

Windows based system. In addition to the following protocols, applications and custom programs can transfer files using inter process communication (via system calls):

FTP FTP (file transfer protocol) is an open standard for transferring files between computers. HP-UX, Linux, and Microsoft Windows based computers all support `ftp` server and client software.

FTP is generally used in interactive mode. The user establishes a connection between their local system (the FTP client) and the remote system (running an FTP server). Once the connection is established, files can be transferred in either direction between the two computers, and the directory structure of either computer can be traversed (within the limits of the user's access authority) to establish a source and a destination directory for file transfers.

> **IMPORTANT:** FTP is generally an insecure protocol and files are transferred in the clear (unencrypted). Passwords to establish the connection between the two computers are also sent unencrypted making these transmissions vulnerable to interception and misuse.

SSH FTP `sftp` (the HP-UX command that implements the SSH file transfer protocol) uses `ssh` to provide more secure file transfers than does the `ftp` command. Files are encrypted during transfer and, with some SSH FTP clients, remote file system operations are available as well (for example, removing a file from a remote system).

RCP `rcp` (remote copy) copies files between two computers.

One of the most notable `rcp` features is that it can copy an entire directory, traversing the tree as it recursively copies subdirectories and their contents.

Both computers can be remote, both can be the same system (for example, you can use `rcp` to copy files from one directory to another on your local system or on a remote system), or you can use `rcp` to copy files to or from a remote system from or to your local system.

`rcp` can use either a `.rhosts` file or the Kerberos authentication system for authenticating users.

RCP (like FTP) is generally insecure. For a more secure remote copy protocol, see SSH RCP.

SSH RCP `scp` (The HP-UX command that implements the secure shell remote copy protocol) uses `ssh` to securely copy files between two computers.

HTTP Primarily used to retrieve web pages for display, HTTP can be used to transfer files

Web Access

The HP-UX 11i version 3 operating system includes the Tomcat web server for use with web based subsystems such as the System Management Homepage. It is therefore also available for serving web pages that you create.

Remotely Mounted File Systems

HP-UX 11i version 3 implements several ways to share file systems and directories between multiple computers. These include:

NFS NFS (Network File System) allows you to export file systems or directories from a server to a pre-defined set of servers which in turn will mount the exported file system so that users and processes of those servers can access the remotely mounted file systems as if they were local file systems.

Samba   Samba is an implementation of Microsoft's SMB (Server Message Block) protocol (and other protocols) that allows for a directory and its sub-directories to be shared between multiple computers.

4 System Administration Tools

Whether you are managing a single server, multiple servers, multiple operating systems in hard or soft partitions on a multiple cell server or a multiple core server, HP-UX 11i version 3 offers a wide variety of tools to manage and control those resources.

This chapter highlights the key system administration tools that are available and what they allow you to control.

Table 4-1 System Administration Tools

Tool	Best Suited for...	How to Obtain
HP System Management Homepage (HP SMH)	Single Operating System Administration[1]	Included with HP-UX 11i Version 3
HP Systems Insight Manager (SIM)	Managing groups of systems	Included with HP-UX 11i Version 3 or available for download at: http://www.hp.com/go/hpsim
OpenView Suite	Enterprise Level Management	Contact your HP Products Sales Representative

1   The operating system can be HP-UX 11i or (if using HP SMH) can be based on Linux or Microsoft Windows

Tools for Installing and Updating HP-UX

The following tools can be used for installing and updating HP-UX. For complete details on this topic, see the appropriate version of the *HP-UX 11i v3 Installation and Update Guide*.

Cold Installs

The term *cold install* means installing system software on a new (does not currently contain an image of HP-UX) or existing (contains an instance of HP-UX that you are willing to completely overwrite) system by completely rebuilding the root volume group, erasing the existing operating system and data on that volume, and installing the new operating system and specified software and data.

You can choose two different methods to cold-install HP-UX 11i v3 on a supported system:
*   By direct installation from the DVD media
*   By creating depots on servers using SD-UX or Ignite-UX and installing it over the network.

An advantage of cold-install is that supported software can be installed without regard for the software currently on the system, or concern for cleaning up old software.

Ignite-UX

Ignite-UX addresses the need for HP-UX system administrators to perform system installations and deployment, often on a large scale. It provides the means for creating and reusing standard system configurations. It provides the ability to archive a standard system configuration, and to use that archive to replicate systems, with the added benefit of speeding up the process. It also permits post-installation customizations, and is capable of both interactive and unattended operating modes.

*   Ignite/UX install sessions for multiple targets can be controlled from a single server in a true client/server model.
*   Installations can occur from multiple software sources in a single install session.
*   It is easy to create a system that is ready to go as soon as the install session completes.

For more information on Ignite-UX, see http://www.hp.com/go/ignite-ux.

Update-UX

The `update-ux` command is the tool to use to update one version of HP-UX to a newer version (provided that you are using a supported upgrade path). For details on supported upgrade paths and general update procedures, see the latest version of the *HP-UX 11i v3 Installation and Update Guide*. For details on the `update-ux`command and its options, see the *update-ux*(1M) manpage.

The Superdome 2 Onboard Administrator (OA)

On Superdome 2 systems, the *Onboard Administrator (OA)* is used to configure nPartitions and Virtual Partitions, install and boot operating systems, control power throughout the server, and monitor the health of your server's components. The Superdome 2 OA is firmware based and does not require any operating systems to be booted (or even installed) in order to perform its functions. It is a significant extension of the Management Processor found on non blade-based cellular systems, and performs many additional functions, including the partitioning functions formerly handled by the online Parmgr configuration tool and the various vPars commands. The Superdome 2 OA is integrated with the HP System Management Homepage and other online and network-based tools. It assists those tools in performing administrative tasks.

The Superdome 2 OA provides its extensive capabilities via three types of interface: a firmware-based graphical user interface, a command line interface, and a series of health viewers — all of which are available whether or not any operating systems are booted, or installed.

Tools for Managing Single Servers

This section describes tools used for managing single instances of HP-UX.

HP System Management Homepage (HP SMH)

> **IMPORTANT:**   As of HP-UX 11i version 3, the previous HP-UX system administration tool for *single server administration* — SAM (System Administration Manager) — has been replaced by a newer, more powerful, and more flexible tool: HP SMH (the HP System Management Homepage).

The HP System Management Homepage (HP SMH) provides three interfaces for managing HP-UX. Use the interface that best suits your needs:

- A web based graphical user interface (GUI)
- A terminal user interface (TUI)
- A command line interface (CLI)

Though HP SMH does not have an X Window native interface[8], a few applications launched by HP SMH continue to open in an ObAM based X Window or ObAM based TUI.

A majority of the applications in the Graphical User Interface (GUI) of HP SMH support multi-byte locales. However, the Text User Interface (TUI) of HP SMH does not support locales other than English. Therefore, HP recommends that you run smh(1m) TUI in C locale.

> **NOTE:**   ObAM is the technology used by SAM, the former single server administration tool.

When you run either the `/usr/sbin/sam` or `/usr/sbin/smh` command and the `DISPLAY` environment variable is set, HP SMH opens in the default web browser. If the `DISPLAY` environment variable is not set, HP SMH opens using its terminal interface.

When first launched, the terminal version of HP SMH presents you with the main menu, which is the gateway to its functional areas:

```
HP-UX System Management Homepage (Text User Interface)
                       SMH
```

8.   Although the GUI for the System Management Homepage isn't a native X Window application, HP SMH will attempt to use X Window to open a browser on a remote X display when you use the **w** menu item from the main menu.

```
   -----------------------------------------------------------------------------
   a - Auditing and Security
   c - Auditing and Security Attributes Configuration(new)
   d - Peripheral Devices
   e - Resource  Management
   f - Disks and File Systems
   g - Display
   k - Kernel Configuration
   l - Printers and Plotters(new)
   n - Networking and Communications
   p - Printers and Plotters
   s - Software Management
   u - Accounts for Users and Groups



   -----------------------------------------------------------------------------
   x-Exit smh                    w-WebLaunch         1-Help
   ENTER-Launch Functional Area  v-SAM Log Viewer
```

Launch the functional area you want to work in by typing the letter next to its entry in the main menu. For example, type **e** to launch the `Resource Manager` or **p** to launch the `Printers and Plotters` functional area.

Launching the HP SMH Web Interface

To access the HP SMH web interface from a browser (either locally or on a remote system, perhaps a PC), use the address:

http://*name_of_system_to_administer*:2301/

where *name_of_system_to_administer* is the network hostname (or IP address) of the system you want to administer. This will cause the web server on the system to start a secure web interface to the HP SMH.

IMPORTANT: For remote web access to HP SMH to be successful, the system you are trying to administer must have its web server running. By default, this means that HP-UX must be at run-level 3 or higher. For a discussion of run-levels, "Run Levels" (page 73).

If you type **w** in the TUI version of the HP SMH main menu you will launch the HP SMH web interface on the X-Window server associated with your `DISPLAY` environment variable.

Launching the HP SMH web interface from within the HP SMH terminal interface starts a web server on the system you are administering (if one is not already running) and starts a web browser to display the functional area (in HP SMH) corresponding to the functional area that was highlighted in the TUI when you typed **w**.

IMPORTANT: Read carefully any warnings that appear when you start your web browser. There are possible security implications in launching the System Management Homepage in this way and HP SMH will identify safer alternatives to launch the System Management Homepage via a *secure* http (https) URL should you require that.

NOTE: If you plan to launch the web based interface from within the terminal based version of HP SMH, be sure to set your shell's DISPLAY variable prior to starting the terminal interface so that HP SMH knows where to direct the GUI display.

The HP System Management Homepage Web Interface
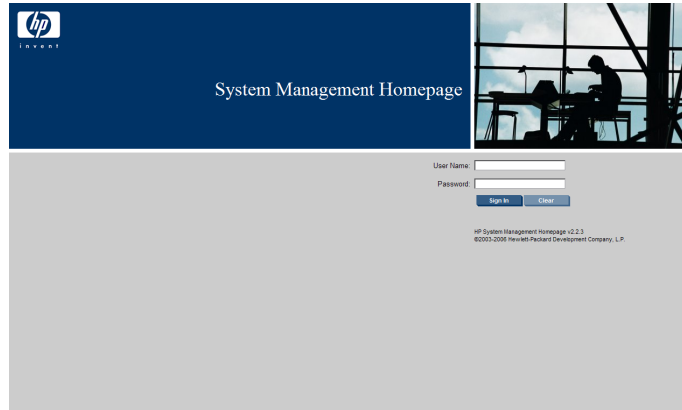
When the HP SMH GUI interface is launched in a browser, it presents the above screen and requires you to authenticate yourself (log in). Upon a successful login the GUI version of HP SMH presents a slightly different menu than the terminal version does.

Instead of showing the functional areas of HP SMH as the first screen, as the terminal version does, the graphical version of HP SMH first displays a system summary showing the general health of the system hardware and providing links to other status screens (for example, Memory Utilization).

A few links to key functional areas are displayed on the home page but to get to the functional area screen in the GUI version of HP SMH, click on the Tools menu item near the top of the display.

Using the graphical version of the HP System Management Homepage you can administer an HP-UX server from almost anywhere using a supported web browser. The GUI of the HP System Management Homepage also provides a much larger collection of tools (functional areas) than the terminal version provides.

The HP System Management Homepage, aggregates, displays, and interacts with a wide variety of web agents and other HP web-enabled system management software to give you extensive control over the configuration and operation of almost every aspect of an HP-UX server, including:

• Management of Users and Groups
• Auditing
• Volume and File System Management
• Error Management Technology (online error message lookup)
• Kernel Configuration
• Partitioning
• Peripheral Devices
• Printers
• Security

- Software Installation, Removal, and Management
- and many others

Using the HP System Management Homepage you can also:

- Launch (X-based) Applications
- Run Commands

TIP: Unlike SAM, the HP System Management Homepage (because it is based on the WEBM industry standard) can also administer Linux and Microsoft Windows based systems.

HP System Management Homepage Key Features

The HP System Management Homepage is:

- Used for *single system management* (for single point of control management of multiple systems, "HP Systems Insight Manager (HP SIM)")
- *Web based* (accessible from any supported browser)
- *Secure* (uses SSL-secure authentication)
- Capable of managing a single system running one of these OS types:
    - HP-UX 11i
    - Linux (versions of Linux supported by HP)
    - Microsoft Windows

What is HP SMH Doing on Your Behalf?

When you perform actions using the HP System Administration Homepage, the interface you interact with executes command-line commands on your behalf, behind the scenes. You could accomplish those same tasks from a shell prompt by using the same commands that HP SMH uses. But, how do you know what HP SMH is doing on your behalf? There are several ways:

- If you are using the web based interface, as you complete the web forms to prepare to execute an action, you can click on the `command preview` item at the bottom of the page to see the command that will be executed if you perform the action.
- If you are using the terminal interface, you can tab to the `[Preview]` item at the bottom of a screen to view the command that HP SMH is building for you. When `[Preview]` is highlighted, hit **Enter** to show the command.
- At any time, you can also view actions that have previously been performed (what commands were executed) regardless of which interface was used to perform the action:
    - In the GUI, click on the `Logs` menu item near the top of the page, then select the SAM Log Viewer item on "Logs" page. You can then filter the logs to locate the commands you are interested in looking at.
    - In the TUI you can get to the SAM Log Viewer from the main menu page by pressing the **v** key. You can then filter the logs to locate the commands you are interested in looking at.

Command Line Tools

The System Management Homepage and System Administration Manager, described previously, provide user interfaces and perform tasks based on the user's input, using commands and system calls.

If you require more detailed control or want to customize tasks in ways that the other system administration management tools do not allow, you can always manually enter the commands you need or execute sequences of commands in a shell script.

HP-UX provides the following commonly used shells for command entry:

- sh - the POSIX shell
- ksh - the Korn shell
- csh - the C shell

Though not officially shells, two related utilities work closely with shells to provide multiple sessions on a single terminal:

shl   The Shell Layer Manager provides a means for interacting with more than one shell from a single using shell layers. Each layer represents a virtual device. The current layer is the one that can receive keyboard input. Other layers have input blocked. Output from multiple layers is multiplexed onto a single terminal. See *shl*(1) for complete details.

tsm   The Terminal Session Manager, similar to the Shell Layer Manager, allows a user to interact with multiple shells from a single terminal. Each shell is bound to a virtual device emulating a physical terminal. `tsm` hot keys allow the user to switch between (and create new) sessions. See *tsm*(1) for complete details.

The HP System Management Homepage Command Line Interface

If you need to enter a quick command line command from within the HP SMH graphical user interface and do not want to start up a shell in a separate window, click on the `Tasks` menu item near the top of the page and then select either `Run Command` or `Run Command as Root` from the `System` section on the resulting panel.

Tools for Managing Multiple Servers

This section describes tools used for managing multiple instances of HP-UX.

HP Systems Insight Manager (HP SIM)

When you have *multiple systems* to manage, use the HP Systems Insight Manager (SIM). The HP Systems Insight Manager is:

- Used for *single point administration of multiple servers*
- Capable of managing multiple servers of varying OS types:
  - HP-UX 11i
  - Linux
  - Microsoft Windows
- *Web based* (accessible from any supported browser)
- *Secure* (uses SSL-secure authentication)
- Compatible with the OpenView Suite of products and with Serviceguard clusters.

Managing Multiple Servers from a Single Point of Control Using HP SIM

While the HP Systems Insight Manager can be used to manage a single HP-UX server, it is probably better to use the "HP System Management Homepage (HP SMH)" for that purpose.

HP Systems Insight Manager allows you to manage multiple servers called *managed systems* from a *Central Management Server* (CMS). Collectively, the CMS and the managed systems are referred to as a *management domain*. The CMS is also a managed system within its own management domain and can simultaneously be a managed system of management domains controlled from other CMSs.

The CMS is the server where the HP Systems Insight Manager software resides, and you can access this software from any network client (any computer capable of accessing the CMS via a web browser (for a graphical user interface), or via `ssh` (for a secure textual user interface). The network clients can be, but do not have to be members of the managed domain.

NOTE: HP Systems Insight Manager is capable of managing large groups of systems but for Enterprise-wide operations HP recommends the more powerful OpenView Suite of tools.

HP Systems Insight Manager is Based on the WBEM Open Standard

The HP Systems Insight Manager is based on the Web Based Enterprise Management (WBEM) standard, a set of technologies designed to consolidate the management of multiple servers of possibly varying OS types, as well as several other open standards.

Supplies Information to OpenView

As your need grow, the HP Systems Insight Manager can grow with you. You can expand its capabilities with extensions to SIM, you can configure additional systems into numerous management domains, and when you are ready for enterprise-level management you can use the HP Systems Insight Manager to feed information into the OpenView Suite of products. See "Tools for Managing an Entire Enterprise" (page 99).

Tools for Managing an Entire Enterprise

When your business grows to enterprise status, or (regardless of size) you need to allocate/control your computing and networking resources based on:

- Higher levels of automation than are provided by the System Management Homepage and HP Systems Insight Manager
- Service Level Objectives
- The need to control multi-vendor environments (for example, networks of systems that include systems from IBM, Sun Microsystems, and other vendors)

HP provides an entire group of products, collectively known as the OpenView Suite, that will address these needs.

The OpenView Suite includes, but is not limited to technologies for:

- Transaction Monitoring
- Software Deployment
- Patch Management
- Hardware Node Management (via the HP Systems Insight Manager)
- Network Node Management
- Oracle Database Management
- Help Desk Support Services
- Storage Management
- (Resource) Usage Management
- Workload Management
- Virtual Machine Management

The complete list of OpenView products is located at http://www.openview.hp.com/products

Other System Management Tools

There are also other tools to assist you with various aspects of administering an HP-UX system. These include:

EVM - Event Management

In HP-UX 11i version 3, the kernel, software applications, and the server hardware can post *events* to the *Event Management Daemon*. Users can then subscribe to these events in order to be alerted to occurrences of events that concern them.

To subscribe to events, a user (or system administrator) can use the System Management Homepage (use **Tools →Evweb →Subscription Administration**) or the evweb command (see the *evweb*(1) and *evweb_subscribe*(1) manpages).

ProviderSvcsBase

New since March, 2009 is the ProviderSvcsBase product, which notifies subscribing clients of HP-UX kernel error log records. A daemon called oserrlogd reads the records, stores them in the circular log files, and then sends EVM notification to subscribing clients.

This product is a required (always-installed) product in all OEs. For more information, see the *HP-UX 11i Version 3 March 2009 Release Notes*.

Partition Manager

Partition Manager is used to configure nPartitions on cell-based servers. It runs as a workspace in the HP System Management Homepage (HP SMH) portal. Partition Manager can be launched from the Tools menu in HP SMH or can be launched from a supported web browser by entering the URL:

```
http://yoursystemname.domainname:2301/parmgr
```

and logging in. When launching Partition Manager from the HP SMH Tools menu, choose one of the items from the list of nPartition Management tools:

- View and Manage Complex, to manage the complex in which HP System Management Homepage is running.
- View and Manage Remote Complex, to login to a remote complex or nPartition that you want to manage.

Partition Manager can be used to:

- Reconfigure nPartitions
  - Create new partitions
  - Remove partitions (to free up resources for other uses)
  - Add cell boards to a partition
  - Remove cell boards from a partition
  - Move a cell board from one partition to another
- Control power to various server components
  - Cabinets
  - I/O Chassis
  - I/O Slots[9]

Software Distributor (SD)

Software Distributor is a collection of tools used for installing and maintaining applications and other software on an HP-UX server. Software Distributor is comprised of the following utilities:

Utilities for Installing and Using Software on Servers   The following Software Distributor utilities are used for installing, removing, and inventory of applications on a server:

swinstall   Used to install software and applications onto an HP-UX operating system.

swlist   Used to identify what software is installed on a server or available in a depot.

swremove   Used to remove software and applications from an HP-UX operating system or from a depot.

swverify   Used to verify software installed on a server. This checks software states, dependency relationships, file existence and integrity, and runs vendor-supplied verification scripts.

9.   Power to I/O Slots is usually controlled during an Online Replacement and Addition (OLRAD) operation.

Utilities for Creating and Maintaining Packages of Software for use by the Above Utilities   The following Software Distributor utilities are used for creating and maintaining packages and installation bundles of software and software installation sources (known as software depots) that are used by `swinstall`, `swlist`, `swremove`, and `swverify`:

`sd`	Invoke interactive interface to display and monitor SD job information, to remove SD jobs, and to clone/recreate SD jobs. It also functions as an SD "home base" to create and launch interactive SD tasks.
`spb`	The Software Package Builder (SPB) provides a graphical user interface for building SD software packages for use by the other utilities listed here. For detailed information about SPB, see http://www.hp.com/go/sw-deployment-docs. Click **HP-UX Software Package Builder**.
`swacl`	Used to control access to software depots and installed applications.
`swagentd`	Part of a pair of processes that control the actual software operations of Software Distributor. The other process, `swagent`, is executed by `swagentd` and is not executed directly by the user.
`swask`	Used by SD control scripts to solicit information from a user about the software choices they have made. The answers given by the user are stored for later use by `swinstall` and `swconfig`.
`swconfig`	Configures, unconfigures, or reconfigures software on a server. swconfig transitions software between the `INSTALLED` and `CONFIGURED` states independently of `swinstall` and `swremove`.
`swcopy`	Copies or merges software selections from a software source to one or more software depot target selections which can then be used as installation sources by `swinstall`.
`swjob`	Display and monitor SD job information and remove SD jobs.
`swmodify`	Used to modify the attributes of software on a system. This allows you to manipulate the catalog that describes software installed on the system. For example, you can adjust the catalog to reflect software that was installed using methods other than `swinstall`.
`swpackage`	Used to construct packages of files that can be installed and maintained on a server by the Software Distributor.
`swreg`	Controls the visibility of depots and roots to users who are performing software management tasks. It can be used to register depots created by `swpackage`.

Tools for Managing Storage on HP-UX

The following tools are used to manage (or are actually pieces of) the storage stack on HP-UX – the layers of technology allowing HP-UX to write and read from mass storage devices.

Volume Managers

Volumes are logical containers that are not limited by the boundaries of the underlying physical disks on which they reside. To applications, file systems, and databases, volumes appear to be physical disks and are treated as such. HP-UX can use volumes as swap space. Volumes can contain:

- File Systems
- Swap Space (for paging operations)
- Dump Space (for memory dump operations)
- Mirror copies of other volumes
- Raw disk space managed by an application such as a database manager

Volume managers allow you to group collections of physical storage (usually disk drives) and then divide the collections into logical entities called *logical volumes* if you are using the HP Logical Volume Manager, or simply *volumes* if you are using the VERITAS Volume Manager.

HP-UX 11i version 3 supports the following volume managers:

LVM   The Logical Volume Manager (LVM) is detailed in *HP-UX System Administrator's Guide: Logical Volume Management*. LVM is the default volume manager for HP-UX 11i.

VxVM   The VERITAS Volume Manager (VxVM) has many features, some of which are not available with LVM or MirrorDisk/UX (the companion product to LVM that allows you to mirror data onto multiple physical disks).

   The version of VxVM that ships with HP-UX is a base version containing a subset of the features offered in the full version (which requires an additional license). For complete information about which features are included with the base version and the full version of VxVM see the *VERITAS Volume Manager Releases Notes* corresponding to the version of the VERITAS Volume Manager you are using.

Both volume managers can co-exist on a server. Each volume manager keeps track of which disks it is controlling and any given physical disk can only be a controlled by one volume manager at a time. The utility `vxvmconvert` can convert an LVM volume group to a VxVM disk group if you want to migrate a disk from LVM to VxVM for greater configuration flexibility.

Volume Management Tasks

The specific volume management tasks you will need to perform will vary slightly depending on which volume manager you choose. For tasks that are common to both LVM and the VERITAS Volume Manager, the specific commands or interfaces you will need to use will also vary depending on which volume manager you are using.

Common volume management tasks include:

- Volume Group / Disk Group Tasks
    - Creating volume groups (from collections of physical disks)
    - Adding physical disk drives to an existing volume group
    - Removing physical disk drives from a volume group
    - Mirroring data
- Logical Volume / Volume Tasks
    - Creating volumes
    - Removing volumes
    - Resizing volumes (and, if appropriate, the file systems within them)

In both LVM and the VERITAS Volume Manager, one volume group is treated special: the *root volume group*. This refers to the volume group that contains the kernel file that is used for booting the system. It is also where the root file system, the file system containing the root directory ("/"), resides. The specifics of how the root volume group is special (how it differs from other volume groups) varies depending on which volume manager you are using. Refer to the documentation for the volume manager you are using for specific information on the differences.

Performance Monitoring Tools

There are many tools available to help you monitor performance of HP-UX based servers, networks, and applications. Some of these tools are included with HP-UX, some are downloadable from hp.com, and some are commercial products from HP or other companies. This section discusses a few of these tools.

Tools for Monitoring the Performance of a Server

There are several applications you can use to monitor the performance of an HP-UX based server:

HP Caliper The primary purpose of HP Caliper is profiling individual applications, however Caliper is also capable of displaying overall system performance information. HP Caliper can be downloaded from: http://www.hp.com/go/caliper

HP Caliper is also supported on versions of Linux for HP Integrity servers.

> **NOTE:** HP Caliper is available *only* for HP Integrity Servers. For HP 9000 servers, consider using Prospect, a performance monitoring tool. See http://www.hp.com/go/prospect.

HP GlancePlus GlancePlus is a full featured operating-system-wide performance monitoring package that provides immediate performance information about your server. It lets you easily examine system activities, identify and resolve performance bottlenecks, and tune your system for more efficient operation. GlancePlus is a part of the HP OpenView Suite of products. Information about how to obtain GlancePlus and detailed information about its features is available at: http://www.hp.com/go/openview.

HP GlancePlus supports a variety of operating systems including HP-UX and Linux.

The following HP-UX commands can also help you gather statistics about how the resources of your system are being used:

iostat iostat iteratively reports I/O statistics for each active disk on the system.

sar sar, the system activity reporter, samples and reports on cumulative activity counters in the operating system or from a previously recorded file. These values can give you a rough idea of where the HP-UX is spending its time.

top Supplied as part of HP-UX, top is a utility that lists all of the processes currently running on a server, ordered by their processing core usage. Those processes listed first in the top output are consuming the most processing time. top also shows global system load factors.

vmstat The vmstat command reports certain statistics kept about process, virtual memory, trap, and CPU activity. It also can clear the accumulators in the kernel sum structure.

Tools for Monitoring the Performance of a Network

Monitoring the performance of a network can be an involved process involving many different variables. For sophisticated network troubleshooting and performance monitoring, HP offers the OpenView Network Node Manager. For information about the Network Node Manager's features and capabilities, and how to acquire it see:

http://openview.hp.com/products/nnm/index.html

If your needs are simply to verify communications between two computers, you can use the ping command which will send packets from one computer to another and time how long it takes to receive a reply. You can do some basic tuning of network performance by tweaking various network settings and running ping to see if the response time improves or worsens. For example:

Example 4-1 Testing network performance using ping

To test a network connection between two systems called "`thissystem`" and "`thatsystem`", from a local command prompt on "`thissystem`" enter the command:

`/usr/sbin/ping thatsystem`

```
PING thatsystem.xxx.yyy.com:
64 byte packets
64 bytes from 10.17.123.456: icmp_seq=0. time=1. ms
64 bytes from 10.17.123.456: icmp_seq=1. time=0. ms
64 bytes from 10.17.123.456: icmp_seq=2. time=0. ms
64 bytes from 10.17.123.456: icmp_seq=3. time=0. ms
```

Ping will keep sending/receiving packets until you stop it by using the interrupt character (usually CTRL-C). It will then terminate the packet sending and report the final performance statistics:

```
----thatsystem.xxx.yyy.com PING Statistics----
4 packets transmitted, 4 packets received, 0% packet loss
round-trip (ms)  min/avg/max = 0/0/1
```

Tools for Monitoring the Performance of Applications

Applications running on HP Integrity Servers can be extensively profiled using the HP Caliper performance monitoring tool (Caliper). While Caliper is capable of monitoring whole-server performance, it is primarily an *application* profiling tool.

Caliper makes extensive use of hardware features on HP Integrity Servers and can run on both HP-UX 11i and Linux operating systems.

Data Protection Tools

Security of servers, networks, and data has never been more important than it is today. HP-UX has many tools for securing your servers and data.

For most HP-UX users, securing your servers and data from unauthorized access ranges from important to critical. Unauthorized access (whether malicious or accidental) is only one of many threats to the integrity and security of your data. Others include:

- Accidental destruction or removal of data by poorly behaving software
- Accidental destruction or removal of data by *authorized* users
- Storage device failures
- Other hardware failures that corrupt data
- Other hardware failures that prevent access to data
- Physical plant and equipment destruction (for example from fires, floods, and earthquakes)

There are HP-UX based tools to protect your data from all of these potential threats.

Protecting Against Unauthorized Access to Your Servers and Data

HP-UX can be configured to run in either of two modes:

Standard Mode   Offers traditional security features found in UNIX systems (accounts, groups, file access privileges, and so on). Passwords are stored (encrypted) in the `/etc/passwd` file.

In addition to the traditional security features mentioned previous, HP-UX running in standard mode has an extended set of security features (for example HP-UX Shadow Passwords) that significantly increase the security of your system without having to convert it to Trusted Mode. These additional features are fully explained in the *HP-UX 11i Security Containment Administrator's Guide*. Additional security information is located in the *HP-UX System Administrator's Guide: Security Management* document.

Trusted Mode   Offers a complete C2-level set of security features. Passwords are not stored in the `/etc/passwd` file, but are instead stored in `/tcb/files` for additional security.

Protecting Against Data Loss

The best way to protect your data against loss is to have another copy of the data somewhere when the primary copy is lost. There are many technologies that will help you make those extra copies. These include:

Backups   There are many ways in HP-UX to backup your data:

- You can backup your data to tapes, optical media, or network-based storage. Some applications and utilities that will allow you to do this include:
  - HP Storage Essentials Software: automates high performance backup and recovery, over unlimited distances, to ensure 24x7 business continuity and maximize IT resource utilization. For complete details on the HP Storage Essentials Software, see http://www.hp.com/go/storageessentials.
  - The `pax` command extracts, writes, and lists archive files and copies files and directory hierarchies. A more contemporary utility, `pax` performs basically the same functions as the older (still available) utilities `cpio` and `tar`. For details about `pax`, see *pax*(1).
  - `tar` (called the "tape archiver") is equally adept at writing to disk archive files or optical media as it is at writing to magnetic tape media. For details about `tar`, see *tar*(1).
  - `cpio` (copy in/out)
  - `vxdump` copies to magnetic tape all files in a VxFS file system that have been changed after a certain date. See *vxdump*(1M)
- You can copy important files to another system using `ftp`, `rcp`, or (for secure copies) `sftp`.
- To protect the copies from being destroyed if physical damage or theft occurs at the site of the primary data, be sure to keep at least one copy of critical data at an alternate location. Don't forget to physically protect tapes that contain unencrypted data. HP Openview Security Data Protector can encrypt backups.
- To protect your operating system files and settings, use Dynamic Root Disk to clone an HP-UX system image to a disk other than that from which the system normally boots. This allows you to quickly boot from an alternate root disk in the event your primary boot disk fails.

Disk Mirroring   Disk mirroring writes multiple copies of data to separate (physical or logical) devices simultaneously.

If you are using LVM (HP's Logical Volume Manager), you will need to install the optional product, MirrorDisk/UX to use disk mirroring. MirrorDisk/UX supports up to three copies of data if you are using LVM

with Version 1 volume groups, and up to six copies of data if you are using LVM with Version 2 volume groups.

If you are using the VERITAS Volume Manager, the ability to mirror your root volume group is built in to the base product. By licensing the full version of the VERITAS Volume Manager, you gain the ability to mirror all your volume groups, up to 32 copies of the data.

RAIDs and Surestore Disk Arrays — Data redundancy can also be accomplished at the hardware level. RAIDs (redundant arrays of inexpensive disks) and HP Surestore Disk Arrays have the capability to make multiple copies of data written to them, and some even have multiple controllers for redundancy of access should a controller fail.

> **NOTE:** RAID levels that include parity disks are able to reconstruct lost data on the fly until a failed disk is repaired or replaced. This is almost as good as multiple copies of the data, however for exceptionally important data, be exceptionally safe by having a copy of the data.

Protecting Against Hardware Failure

Depending on the specific hardware you have (server types, storage devices, and so on), HP-UX 11i version 3 offers numerous ways of protecting your computing operations against hardware failure. Here are some key features to consider:

Serviceguard — For mission critical installations, Serviceguard takes redundancy an extra step by having multiple servers connected to external disks or arrays. If one server fails, Serviceguard can switch to a stand-by server capable of carrying on the functions of the failed server while the original is repaired.

Persistent Device Special Files — HP-UX 11i version 3 introduces a new type of device special file called a *persistent device special file*. Unlike legacy device special files[10] that address devices by the *hardware path to them*, persistent device special files use *unique identifiers built into (or associated with) supported devices* to address them. This means that multiple hardware paths can be used to address the same device, preventing single points of failure in interface cards/slots.

Online Addition and Replacement — Online Addition, Replacement, and Deletion (OL*) is an HP-UX feature that allows for the addition, replacement, and deletion of PCI / PCI-X cards (adapters) while a system is running (without requiring a reboot).

This feature enhances overall high-availability since the system can remain active while an I/O adapter is being added or replaced. When combined with other high-availability products, such as Serviceguard, system availability is significantly improved.

Failed devices that support PCI OL*, if not critical to your operation, can be replaced or removed.

---

10. Legacy device special files are still supported in HP-UX 11i version 3 and can be used in conjunction with persistent device special files.

Online Replacement *suspends* the driver instance associated with the failed card and powers down the slot so the card can be replaced with a new one *of the same type*. Then power can be restored to the slot and new card, and the driver resumed.

Online Deletion *removes from the running kernel* the driver instance associated with the failed card and powers down the slot so the card can be removed. You can then (optionally) install a new card of the same or different type using Online Addition.

> **NOTE:** If the driver is not a core driver (one that is always present in the kernel), you will need to install a driver when adding in the new card, even if it is the same type as the one that was deleted.
>
> If the driver *is* a core driver, a new instance of the driver will be created when you add back the new card of the same type.

Network Administration Tools

Configuring a network can be an involved process involving many different variables. For sophisticated network troubleshooting and performance monitoring, HP offers the OpenView Network Node Manager. For information about the Network Node Manager's features and capabilities, and how to acquire it see:

http://openview.hp.com/products/nnm/index.html

The Basics of Network Administration

Networking on HP-UX can include:
- Remote logins
- Remote file system access
- File transfers
- Remote database access
- Electronic mail
- Web server
- Web client (browser)
- and much more

Configuring a Network Interface

Before you can use any of the above networking services you need to configure at least one network interface so that HP-UX can communicate with a network.

Specifically, you need to:
1. Physically connect your server to network hardware (unless you are using a wireless network connection for your server).
2. Set up one or more network interface controllers (NIC).
3. Set up a network address (usually an IP address).
4. Configure Network Services and the ports used by those services.

Use the `nwmgr` command to configure network interfaces. See the manpage *nwmgr*(1M) for details on the numerous tasks `nwmgr` can do.

> **TIP:** For users familiar with networking on previous versions of HP-UX, the `nwmgr` command performs the functions previously served by the (now deprecated) commands:
>
> - `lanadmin`
> - `lanscan`
> - `linkloop`
>
> See *nwmgr*(1M) for details on using the `nwmgr` command, and discontinue using `lanadmin`, `lanscan`, and `linkloop`.

Refer to the *HP-UX LAN Administrator's Guide* for information on these basic tasks.

There are many networking documents to assist you with the specific components of networking you want to use. These links into the Business Support Center will help you locate them:

- Network and Systems Management
- I/O Cards and Networking Software
- Internet and Security Solutions

Glossary

accept
One of four commands that control the flow of print requests through the Line Printer Spooling System (spooler). `accept` instructs the spooler to allow new print requests to be added to the print queue of a printer or class.
See also reject, enable, disable.

Agile View
A view of the I/O device tree using the more flexible and scalable persistent device special files, LUN hardware paths.
See also Legacy View.

ASCII
American Standards Committee on Information Interchange. A standard used by computers for interpreting binary numbers as characters.

block special files
See device special files.

board
A printed circuit assembly (PCA). Also called a card or adapter.

boot loader
A software program, used in the boot sequence, to load the HP-UX kernel from disk and start it running.

bus
A common data path over which data is transported.

cell
A circuit board that contains processors and memory, all under the control of a cell controller (CC) chip. Cell boards can be combined into nPartitions to create electrically isolated environments that can be used to install an operating system, or further subdivided into Virtual Partitions.
See also Superdome 2 blade.

cell board
See cell.

Central Management Server
A system in the management domain that executes the HP Systems Insight Manager software. All central operations within HP Systems Insight Manager are initiated from this system.

character special files
See device special files.

cold install
Installation of a fresh copy of HP-UX to either a blank disk device or disk volume or completely overwriting any previous contents on the device or volume — specifically, not an update.

compartments
Compartments are a method for isolating components of the system from one another. When configured properly, they can be effective in safeguarding HP-UX, its processes, and its resources.

components without usage rights
See iCAP components.

continental cluster
A group of clusters that use routed networks or common carrier networks for data replication and cluster communication to support package failover between separate clusters in different data centers. Continental clusters are often located in different cities or different countries and can span 100s or 1000s of miles.

core
Formerly referred to as a "CPU". An individual processing unit on a processor chip. Sometimes referred to as a "processing core".

device multipathing
Used with agile addressing and persistent device special files, device multipathing allows multiple hardware paths to a device to use a single device special file. Using device multipathing, you can balance traffic loads between the various hardware paths to a device. You also have redundancy should one of the paths fail.

device special files
Associated with physical and pseudodevices, device special files are used by the operating system and applications to write to and read from their associated devices.

Device file types:

- *Legacy Device Special Files* — the traditional, hardware path dependent style of device special file. Each path to a device has its own device special file. Moving a device means using different device special files to access the device.
- *Persistent device special files* — the newer, hardware path independent style of device special files. Because of the hardware path independence, moving a device to a different hardware path does not require you to use a new device special file, or even change the current one.

Device file access:

- *Block Special Files* – Device files associated with block devices. Block devices transfer data in multi-byte blocks by means of the system's normal buffering mechanism.
- *Character Special Files* – Device files associated with character-mode devices, such as printers, most terminals, and modems. Character-mode devices transfer data in an unbuffered stream.

device swap
Swap space located in a dedicated disk volume or disk partition. Device swap is not part of a file system and is not memory based. Device swap is generally faster than file system swap because it is more direct (does not have the file system layer to traverse).
See also file system swap, and pseudo swap.

directory path
A string of characters representing the sequence of directories that must be traversed from the root directory ("/") to the directory represented by the path. Directory names in the path string are delimited by the slash character ("/").

Example: `/usr/share/man/man1.Z/cat.1`

disable
One of four commands that control the flow of print requests through the Line Printer Spooling System (spooler). `disable` instructs the spooler to prevent queued requests from flowing *out of* a print queue; in other words, *to prevent the printing of requests from a queue*.
See also accept, reject, enable.

disk group
(**1**) A VxVM disk group is conceptually similar to an LVM volume group. It is a collection of VM disks that share a common configuration.
(**2**) A collection of individual disks that share a common role in disk array operation. All disks on the disk array will be a member of one of the following disk groups: a LUN, hot spare, or unassigned.

disk striping
A method of writing to a logical disk device (for example, an LVM logical volume or VxVM Volume) comprised of multiple physical devices such that consecutive chunks of data (such as bits, bytes, or blocks, or extents) are distributed to different physical devices. This can speed up logical device access as multiple data chunks can be written or read simultaneously to or from the different physical devices.

dump concurrency
The ability to dump to multiple devices in parallel, speeding up memory dumps and shortening system downtime.

Dynamic Root Disk (DRD)
An HP-UX based tool set that enables you to create and modify an inactive system image (a clone of your running system), on a separate disk, without having to shut down the system. Once created, the cloned system image can be used for system recovery, can be updated offline to a newer version of HP-UX to be booted at a more convenient time, or it can be used to boot another system (real or virtual).

enable
One of four commands that control the flow of print requests through the Line Printer Spooling System (spooler). `enable` instructs the spooler to allow queued requests to flow *out of* a print queue; in other words, *to actually be printed*.
See also accept, reject, disable.

event
An indication, provided by any system component or application that something noteworthy has occurred. Events are made available to any interested entity by being posted to the Event Manager subsystem.
See also Event Manager (EVM).

Event Management Daemon
Part of the Event Manager subsystem, the event management daemon (`/usr/sbin/evmd`) provides posting and notification services for system and application clients running on the local system.
See also Event Manager (EVM) and the evmd1M manpage.

Event Manager (EVM)	An HP-UX subsystem that provides a mechanism for the posting and retreiving of events. *See also* events and the evm5 manpage.
extended campus cluster	*See* extended distance cluster.
extended distance cluster	A cluster with alternate nodes located in different data centers separated by some distance. Extended distance clusters are connected using a high speed cable that guarantees network access between the nodes as long as all guidelines for disaster tolerant architecture are followed. The maximum distance between nodes in an extended distance cluster is set by the limits of the data replication technology and networking limits.
extent	A fixed size chunk of disk space used by the Logical Volume Manager to allocate space in a volume group. Logical volumes are managed as a collection of extents and extents are the unit of measure by which logical volumes are created or resized.
file system	A disk based mechanism – residing in a disk partition, logical volume, or on optical media – used for organizing files, directories, links, and occasionally for paging operations. In HP-UX, file systems are hierarchical and can be joined via the mount process (see *mount*(1M)) to form larger directory hierarchies.
file system swap	Swap space that resides within a file system. File system swap is slower than device swap because the file system is required to allocate the space in order to prevent files from being overwritten. This adds an extra layer of access that the kernel must traverse to write or read paged out memory. *See also* device swap and pseudo swap.
fine-grained privilege	A permission to perform a specific, low-level operation (for example, permission to execute a specific system call).
FTP	Stands for File Transfer Protocol. A method for transferring files over a computer network from one computer (a server) to another (a client). FTP also enables limited file operations (for example directory listings) on the remote computer. The two computers involved are a server and a client. • FTP Server – listens on the network for connection requests from clients. • FTP Client – initiates the connection to an FTP server and enables the user to transfer files and perform other file manipulation tasks on the server.
guest operating system	*See* Integrity VM guest.
hard partition	*See* nPartitions.
hardware partitioning	*See* nPartitions.
hardware threading	A hardware technique used in Itanium processors to enhance the computational performance of a core. Itanium processors are those used in HP Integrity Servers.
high availability cluster	A group of servers functioning in a coordinated fashion to create a configuration that allows application services to continue in spite of a hardware or software failure.
HP Insight Dynamics–VSE	Software for HP Integrity servers that allows you to quickly adjust to dynamic business demands. Using this software you can provision your infrastructure with automated activations of servers, storage, and networking; optimize that infrastructure with built-in capacity planning and rebalancing tools, and protect continuity of services with automated failover. For additional information on HP Insight Dynamics - VSE see http://www.hp.com/go/insightdynamicsvse.
HP SMH (System Management Homepage)	The primary single system administration tool beginning with HP-UX 11i version 3. HP SMH supports HP-UX, HP supported versions of Linux, and Microsoft Windows operating systems.
iCAP	(Instant Capacity) A group of technologies that allow you to activate/deactivate pre-purchased "stand-by" hardware components, paying only for the computing resources you actually consume. This allows you to handle temporary usage demands more cost efficiently.
iCAP components	Components (for example, processing cores and memory), purchased at a reduced price, but without usage rights. These components can be activated when needed by purchasing either temporary or permanent usage rights.

Installed Product Database	A database of software currently installed on a server, created and maintained by the Software Distributor suite of utilities. See "Software Distributor (SD)" (page 100).
Integrity VM guest	An instance of HP-UX running within an Integrity virtual machine.
interleaved swap	Swap space spanning multiple physical devices for read/write efficiency, similar to disk striping. *See also* disk striping.
JBOD	An acronym meaning **J**ust a **B**unch **o**f **D**isks and representing a group of disk devices concatenated together to be treated as though they were a single large device.
kernel	The nucleus of the HP-UX operating system. Comprised of drivers and other code modules, the kernel centrally controls nearly all of the essential functions of an operating system (for example, memory management, communication between hardware and software, and process scheduling).
kernel modules	Modular chunks of code that collectively make up the kernel. Some modules require a reboot in order to be added to (or removed from) a kernel, others do not.
kernel tunables	Variables within the kernel that govern various kernel functions (for example, how many processes can simultaneously exist, or how physical memory is allocated). By altering the values of these variables the kernel's behavior can be influenced ("tuned").
lazy swap	When a process is scheduled, enough swap space is usually reserved just in case the process needs to be paged out of physical memory to make room for other running processes.

Often, swap space is reserved but never used because the process it was reserved for did not get paged out, or only portions of it did. This results in wasted swap space. When enabled, lazy swap causes swap space to be reserved *at the time memory contents for a process are actually paged out*, rather than when the process is scheduled, yielding more efficient swap space utilization. |
| **Legacy View** | A view of the I/O device tree using physical hardware path addresses and legacy device special files.
See also Agile View. |
| **line printer spooling system** | An HP-UX subsystem used to control the flow of printing in order to:
• prevent intermixed listings
• prioritize print jobs
• control who can use specific printers
• allow for printer maintenance
• group printers to associate them with a single print queue (See "printer class") |
| **local printer** | A printer that is physically attached to a server and directly controlled by the Line Printer Spooling System of an HP-UX instance running on that server. |
| **logical volume** | A subdivision of a volume group, a logical volume can span multiple physical volumes or represent only a portion of one physical volume.

The size of a logical volume is measured in units called extents. The size of logical volumes can be altered after they are initially created. Logical volumes can be extended and, if their contents permit, reduced.

Once created, logical volumes can be treated just like disk partitions. They can be used for file systems, swap space, as dump devices, or for raw disk access. |
| **LUN hardware path** | A virtual hardware path that represents the device itself, not the hardware path to it. A LUN hardware path represents all physical paths ("lunpath hardware paths") to the LUN it represents. The first path element in a LUN hardware path is `64000`, followed by a virtual bus address and virtual LUN ID. For example: `64000/0xfa00.0x22`. |
| **lunpath hardware path** | An individual physical hardware path to a mass storage device (usually a disk). Lunpath hardware paths are composed of a series of bus-nexus addresses separated by / (slash) characters leading up to the Host Bus Adapter (HBA). Beneath the HBA, additional address elements are represented in hexadecimal. The first elements represent a transport-dependent target address. The final element is a LUN address, which is a 64-bit representation of the LUN identifier reported by the targeted device.
See also LUN hardware path and persistent device special file. |

major number	The part of a device special file that determines whether the file is used for block access or character access and also used to determine which device driver to use when communicating with the device.
managed systems	Any system managed by HP Systems Insight Manager, including servers, desktop PCs, and Remote Insight Boards.
management domain	A collection of resources called managed systems that have been placed under the control of the HP SIM. Each Central Management Server is responsible for a management domain. The managed systems can belong to more than one management domain. *See also* Central Management Server.
metrocluster	*See* metropolitan cluster.
metropolitan cluster	A cluster that is geographically dispersed within the confines of a metropolitan area requiring right-of-way to lay cable for redundant network and data replication components.
minor number	The part of a device special file that identifies the location of the interface to which a device is attached and defines driver-dependent characteristics. This information is organized by specific bit assignments. *See also* device special files and major number.
modules	*See* kernel modules.
mount points	Directories in the HP-UX directory tree where file systems are logically attached. When mounted, the directory that is the root of a file system's directory tree is represented by the HP-UX directory to which it is mounted. See *mount*(1M).
network based printer	A printer connected directly to a network and having its own network interface (such as an HP JetDirect interface card) and network address. All printing to a network based printer must travel over the network. *See also* remote printer.
nodes	(Serviceguard) Individual systems in a Serviceguard cluster. Systems could be standalone servers or instances of HP-UX running in a partition on a server. (Networking) (Individual instances of HP-UX (or other operating systems) on a network, each identified by its own hostname and one or more IP addresses. (Directory Tree) In the HP-UX directory tree, each directory, file, or link represents a node. Similarly, HP-UX keeps track of I/O devices using a hierarchy where each component in a hardware path (regardless of which addressing scheme is used) represents a node on the I/O tree.
nPartitions	Available on cellular and Superdome 2 blade-based servers: nPartitions (also known as hard partitions) provide both operating system *and* electrical isolation. If an operating system crash or hardware failure occurs in one nPartition on a server, operating systems and hardware in other nPartitions on the same server continue working, unaffected by the failure. *See also* Virtual Partitions and Integrity VM guest.
OL*	Pronounced "oh ell star". Represents all of the On-Line hardware manipulations (the * represents the UNIX wild-card character):
	• OLA = On-line ADD
	• OLD = On-line DELETE
	• OLR = On-line REPLACEMENT
	• OL* = All of the above
Onboard Administrator (OA)	*See* Superdome 2 Onboard Administrator (OA).
operating environments	Operating environments are individual software products that deliver specific HP-UX 11i configurations. Each operating environment is comprised of the "base" HP-UX 11i functionality, commonly needed network drivers, and selected additional layered software products (ISU Products). There are four operating environments in HP-UX 11i version 3:
	• HP-UX 11i v3 Base Operating Environment – (BOE)
	• HP-UX 11i v3 Virtual Server Operating Environment – (VS-OE)
	• HP-UX 11i v3 High Availability Operating Environment – (HA-OE)
	• HP-UX 11i v3 Data Center Operating Environment – (DC-OE)

operating system	Software that controls the resources of a system (or partition). An instance of HP-UX running in an nPartition, vPars virtual partition, or HPVM virtual machine
	Other operating systems commonly found on HP Integrity servers include Linux and Microsoft Windows.
partition	A grouping of server resources dedicated to an instance of an operating system. *See also* nPartition, virtual partition, Integrity VM guest.
persistent device special file	A device file for mass storage devices (usually disks), that is associated with a LUN hardware path, and thus transparently supports agile addressing and multipathing. A persistent device special, therefore, remains unchanged if the LUN it is associated with is moved from one host bus adapter to another, or if a mass storage device fails and is replaced.
primary swap	The initial location made available for paging operations during the system startup sequence. Defined using the swapon command. See *swapon*(1M). *See also* secondary swap.
print destination	A queue associated with a printer or printer class. Many of the Line Printer Spooling System commands, and other applications that provide printing services, use the a print destination to identify which printer or group of printers to affect. See the manpages *lpadmin*(1M) and *lpalt*(1M). *See also* printer class, Line Printer Spooling System and print queues.
print queues	A queue within the Line Printer Spooling System associated with a printer or printer class, used to hold print requests until they are printed.
print requests	A print job submitted to the Line Printer Spooling System. *See also* Line Printer Spooling System.
printer class	A print queue representing a group of one or more printers; treated as a single print destination. Print requests submitted to a printer class (when printed) will be sent to one of the available printers defined in the class. The Line Printer Spooling System determines which printer is actually used to print any given request in the class queue.
printer interface scripts	A script, used by the Line Printer Spooling System, to output print requests to a printer. When defined in the Line Printer Spooling System, a printer interface script is created which is a copy of a printer model script. Once created, the printer interface script can then be customized to tailor it to your needs.
printer model scripts	Scripts — usually supplied as part of HP-UX or by a printer vendor — used as templates (models) from which printer interface scripts are created at the time printers are configured into the Line Printer Spooling System.
PRM groups	A collection of users and applications that are joined together and assigned certain amounts of CPU, memory, and disk bandwidth resources.
processor	A physical piece of silicon (a "chip") containing one or more cores.
processor set	A group of cores, defined by the psrset command (or indirectly higher level products such as the Workload Manager - WLM), for use as an independent scheduling domain. The default processor set consists of all the cores on the system (server or partition).
pseudo swap	System memory used for swap space that allows users to execute processes in memory without allocating physical swap. Pseudo-swap is controlled by the operating-system parameter swapmem_on which by default is set to 1, enabling pseudo-swap.
pseudodevice	A virtual device, emulated by the operating system, not corresponding to a physical device. In HP-UX 11i version 3, examples of pseudodevices include:
	/dev/null Receives and ignores all input.
	/dev/random A source of random numbers.
	There are many others.
RAID	An acronym meaning Redundant Array of Independent Disks. RAID defines various ways (known as RAID Levels) to group mass storage devices to achieve data redundancy or read/write performance.

reject	One of four commands that control the flow of print requests through the Line Printer Spooling System (spooler). `reject` instructs the spooler to prevent print requests from being added to a print queue. *See also* accept, enable, disable.
remote printer	A printer connected directly to a remote instance of HP-UX ("system"). Print requests to a remote printer must first travel over a network to the remote system. The Line Printer Spooling System on the remote system subsequently handles the printing of the requests as though they were local requests. *Note:* The remote instance of HP-UX could be on the same physical server, in an alternate nPartition for example.
remote spooling	Spooling to printers that are defined in the Line Printer Spooling System (spooler) of a different server or HP-UX instance. The local spooler accepts print requests and submits them to the remote spooler on your behalf. The remote spooler then handles the printing of the requests.
resource path	On Superdome 2 systems: A special, operating system agnostic, form of addressing; used by the Superdome 2 Onboard Administrator (OA) to reference system components. See the *resourcepath*(5) manpage for complete information.
Role-Based Access Control	Role-Based Access Control. An HP-UX mechanism to provide fine-grained access to system resources, commands, and system calls. Users are assigned to roles and users are granted access privileges according to roles.
root directory	The top most directory in the HP-UX directory tree. The root directory is represented by the path "/".
root file system	The file system containing the root directory. It is the first file system mounted during the boot sequence and contains the mount points to which other file systems are mounted. *See also* mount points.
root volume group	The LVM volume group that contains the root file system and primary swap volume. *See also* root file system and primary swap.
run-level	A configuration of system processes. Processes spawned by boot init is assigned to one or more run-levels. Only processes having an assignment corresponding to the current system run-level are processed.
SAM (System Administration Manager)	The primary single system administration tool prior to HP-UX 11i version 3. SAM is supported only the HP-UX operating system. *See also* HP SMH (HP System Management Homepage).
secondary swap	HP-UX begins by paging on a single device only (see *primary swap*). That way only one device is needed at boot time. Additional swap areas, known as secondary swap areas, can be subsequently enabled in order to provide larger amounts of space for paging operations.
selective dump	A memory dump containing only selected portions of memory. Selective dumps use less disk space and complete faster than full memory dumps.
server	Formerly referred to as a system or a computer. The physical cabinet containing cell boards, processors, memory, and power supplies.
Service Level Objectives	Generally a specific, measurable item/objective within a broader, more comprehensive service-level agreement (SLA) contract.
Serviceguard	HP's product for implementing high availability clusters, alone or together with other products to form disaster tolerant networks. *See also* continental clusters, extended distance clusters, high availability clusters and metropolitan clusters.
software depot	An SD-UX format structure that contains one or more software products that can be installed on other systems or copied to other depots.
software partitioning	*See* Virtual Partitions and virtual machine.
software threading	A parallel computing technique used by applications and operating systems to enhance processing efficiency.
spooler	*See* line printer spooling system.

stale device special file	A device special file no longer associated with a valid device. For example, a device file associated with a device that has been removed from a server.
storage stack	The various layers of hardware and software the comprise HP-UX based storage systems. Layers in the storage stack include: • Storage Devices (disks, disk arrays, DLT libraries, and so on) • Drivers to access the storage devices • Volume Managers (for example, LVM, and VxVM) • Logical Volumes, • File systems (for example, HFS, and VxFS)
Superdome 2 blade	On Superdome 2 systems: Superdome 2 blades contain processors, memory, I/O components, and on-board LANs. Similar, though not identical, in concept to cell boards (see *cell*) in non blade-based cellular systems, Superdome 2 blades can be combined to create electrically isolated partitions known as nPartitions. Multiple nPartitions can sub-divide an entire system for more efficient utilization of resources. *See also* nPartitions.
Superdome 2 Onboard Administrator (OA)	On Superdome 2 systems: A firmware-based environment used for configuring and managing the resources of the system. The Superdome 2 Onboard Administrator (OA) can be used to configure partitions (nPartitions *and* vPars), control power to various entities within the system, boot operating systems, and monitor the health of the system and its components. The Superdome 2 OA also integrates with HP Systems Insight Manager and the HP System Management Homepage. The Superdome 2 OA replaces the Management Processor (MP) used on non blade-based cellular systems. *See also* HP SIM, HP SMH.
system	Used in this document in two ways: 1. A server or subset of a server (for example, a partition) running an independent copy of HP-UX. 2. Another type of computer (for example, a PC)
system default printer	If defined, the system default printer is the print destination that will be used as a print destination if one is not otherwise specified. See *lpadmin*(1M) manpage (`-d` option) for details.
tunables	*See* kernel tunables.
Utility Meter	The software and hardware device that receives PPU utilization information from the PPU software. The utility meter is initially installed and configured by an HP service representative.
virtual LUN ID	The final element in a LUN hardware path (`0x3` in the following example): `64000/0xfa00/`**`0x3`**
virtual machine	Abstractions of real, physical machines. Multiple virtual machines can share a common set of physical resources. *See also* Integrity VM guest.
Virtual Partition	A software partitioning of a server or nPartition where each virtual partition contains an instance of an operating system. Though an nPartition can contain multiple virtual partitions, the inverse is not true — a virtual partition cannot span nPartition boundaries. *See also* nPartitions.
virtual root node	In a virtualized path to a device (in the agile view), instead of a series of bus-nexus addresses leading to the HBA, the path contains a virtual bus-nexus (with an address of `64000`). This virtual bus-nexus is called the "virtual root node". An example of a LUN hardware path (showing the virtual root node) is: `64000/0xfa00/0x3`
Virtual Server Environment (VSE)	*See* HP Insight Dynamics–VSE.
virtualization	Technologies for using your computing resources in ways not dependent on the physical characteristics of those resources. For example, a logical volume can span multiple physical disk devices.

virtualized LUN hardware path	*See* LUN hardware path.
volume group	A collection of physical volumes (physical disks), used by the Logical Volume Manager. Volume groups can be subdivided into logical volumes (flexible virtual disks that can contain file systems, swap space, or used as dump devices or raw disk access). *See also* logical volume.
vPars	*See* Virtual Partition.
WBEM (Web-Based Enterprise Management)	WBEM is a set of management and Internet standard technologies developed by the Distributed Management Task Force (DMTF) to unify the management of enterprise computing environments.

Index