HP System Partitions Guide

Administration for nPartitions

Sixth Edition

Revision 6.0

Manufacturing Part Number: 5187-3603

April 2003

Legal Notices

The information in this document is subject to change without notice.

Hewlett-Packard makes no warranty of any kind with regard to this manual, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. Hewlett-Packard shall not be held liable for errors contained herein or direct, indirect, special, incidental or consequential damages in connection with the furnishing, performance, or use of this material.

Warranty

A copy of the specific warranty terms applicable to your Hewlett-Packard product and replacement parts can be obtained from your local Sales and Service Office.

Restricted Rights Legend

Use, duplication or disclosure by the U.S. Government is subject to restrictions as set forth in subparagraph (c) (1) (ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013 for DOD agencies, and subparagraphs (c) (1) and (c) (2) of the Commercial Computer Software Restricted Rights clause at FAR 52.227-19 for other agencies.

HEWLETT-PACKARD COMPANY
3000 Hanover Street
Palo Alto, California 94304 U.S.A.

Use of this manual and flexible disk(s) or tape cartridge(s) supplied for this pack is restricted to this product only. Additional copies of the programs may be made for security and back-up purposes only. Resale of the programs, in their present form or with alterations, is expressly prohibited.

Copyright Notice

Preface

The *HP System Partitions Guide* describes nPartition system administration procedures, concepts, and principles for HP rp7405/rp7410 servers, HP rp8400 servers, and HP Superdome servers running the HP-UX 11i operating system.

This preface has the following sections:

Reader Comments and Feedback

Hewlett-Packard welcomes your feedback on this publication. Please address your comments to `edit@presskit.rsn.hp.com` and note that you will not receive an immediate reply. All comments are appreciated.

About This Book: Overview of Chapters

9. *Processor Instant Capacity on Demand (iCOD)* on page 397

   This chapter covers using Hewlett-Packard's processor iCOD (Instant Capacity on Demand) product on nPartitions.

   iCOD is an *optional* software product that enables you to instantly increase or adjust processing power within nPartitions. As you need more or fewer processors, you use iCOD tools to adjust the number of activated processors in the nPartition.

10. *Processor Sets (Psets) on nPartitions* on page 419

    This chapter describes how to use and manage processor sets (Psets) on nPartition systems.

    Using Psets, you can create multiple independent processor groups in an nPartition. Each Pset has its own processors, schedules, and attributes. Because Psets are dynamic, you can create, modify, and destroy Psets instantly as your system needs demand.

11. *Virtual Partitions (vPars) Management on nPartitions* on page 441

    This chapter describes how to create, configure, and manage HP's virtual partitions within an HP Superdome nPartition (hard partition) system environment. Each virtual partition can boot a single instance of the HP-UX B.11.11 operating system.

    The HP virtual partitions (vPars) software is an *optional* feature that you can use to further subdivide a server's resources into multiple, smaller virtual machines through software partitioning.

    By configuring multiple virtual partitions within an nPartition, you can boot multiple instances of HP-UX B.11.11 in a single nPartition.

How to Buy This Book

You can purchase a printed copy of the *HP System Partitions Guide* from Hewlett-Packard's **http://software.hp.com** Web site. When at this Web site, click the **Documentation** link for a list of current publications.

The **http://www.software.hp.com/BOOKS_products_list.html** Web site lists technical books currently available for sale, including this book.

You also can find this book by searching for "Partitions Guide" using the HP software depot search facility.

Related Information

For the most current HP-UX 11i nPartition administration details refer to this publication, the *HP System Partitions Guide*.

You also can find other information on general HP-UX 11i administration, HP nPartition server hardware management, and diagnostic support tools in the following publications.

Web Site for HP Technical Documentation: http://docs.hp.com

The main Web site for Hewlett-Packard technical documentation is the **`http://docs.hp.com`** site, which has complete information available for free.

HP-UX 11i Information

The following Web site and publications are available for info about the HP-UX 11i operating system.

- **`http://docs.hp.com/hpux/os/11i/`** — This is the portion of the docs.hp.com Web site that has complete HP-UX 11i information.
- *Configuring HP-UX for Peripherals*
- *HP-UX 11i June 2003 Release Notes*
- *HP-UX 11i Installation and Update Guide*
- *HP-UX Workload Manager User's Guide*
- *HP Process Resource Manager User's Guide*
- *Installing and Managing HP-UX Virtual Partitions (vPars)*
- *Instant Capacity on Demand (iCOD) and Pay Per Use (PPU) User's Guide for Version B.04.x*
- *Instant Capacity on Demand (iCOD) User's Guide for Version B.05.00*
- *Managing Systems and Workgroups: A Guide for HP-UX System Administrators*

Server Hardware Information

The following Web sites and publications describe HP nPartition server hardware management, including site prep, installation, and other details.

- **http://docs.hp.com/hpux/hw/** — This is the systems hardware portion of the docs.hp.com Web site. The following Superdome, rp8400, and rp7405/rp7410 server Web sites are available from this systems hardware page.
- **HP Superdome — http://docs.hp.com/hpux/hw/index.html#Superdome%20Server**

  This is the Web site for hardware info about the HP Superdome server.
- **HP rp8400 — http://docs.hp.com/hpux/hw/index.html#rp8400%20Server**

  This is the Web site for hardware info about the HP rp8400 server.
- **HP rp7405/rp7410 — http://docs.hp.com/hpux/hw/index.html#rp7405/rp7410%20Server**

  This is the Web site for hardware info about the HP rp7405/7410 server.

Diagnostics and Event Monitoring: Hardware Support Tools

Complete information about HP's hardware support tools, including online and offline diagnostics and event monitoring tools, is at the **http://docs.hp.com/hpux/diag/** Web site. This site has manuals, tutorials, FAQs, and other reference material.

Web Site for HP Technical Support: http://us-support2.external.hp.com

Hewlett-Packard's IT resource center Web site at **http://us-support2.external.hp.com/** provides comprehensive support information for IT professionals on a wide variety of topics, including software, hardware, and networking.

Books about HP-UX Published by Prentice Hall

The **http://www.hp.com/hpbooks/** Web site lists the HP books that Prentice Hall currently publishes, such as HP-UX books including:

- *HP-UX 11i System Administration Handbook*
  http://www.hp.com/hpbooks/prentice/ptr_0130600814.html
- *HP-UX Virtual Partitions*
  http://www.hp.com/hpbooks/prentice/ptr_0130352128.html

HP Books are available worldwide through bookstores, online booksellers, and office and computer stores.

Publication History

The publication history for the *HP System Partitions Guide* includes the following editions.

Sixth Edition	April 2003, 5187-3603. CD-ROM, Web (**http://docs.hp.com/**), and print delivery. You can order this book in print from the **http://software.hp.com** Web site. Updates include HP rp7405 server details, vPars A.02.02 information, and other changes throughout.
Fifth Edition	August 2002, B2355-90762. CD-ROM, EPSS, Web (**http://docs.hp.com/**), and print delivery.
Fourth Edition	June 2002, B2355-90752. CD-ROM, EPSS, and Web (**http://docs.hp.com/**) delivery.
Third Edition	March 2002, B2355-90746. CD-ROM, EPSS, and Web (**http://docs.hp.com/**) delivery.
Second Edition	December 2001, B2355-90744. CD-ROM, EPSS, and Web (**http://docs.hp.com/**) delivery.
First Edition	September 2001, B2355-90736. CD-ROM, EPSS, and Web (**http://docs.hp.com/**) delivery.

Notational Conventions

The following notational conventions are used in this publication.

WARNING **A warning lists requirements that you must meet to avoid personal injury.**

CAUTION A caution provides information required to avoid losing data or avoid losing system functionality.

NOTE A note highlights useful information such as restrictions, recommendations, or important details about HP product features.

- `Commands and options` are represented using `this font`.
- **`Text that you type exactly as shown`** is represented using **`this font`**.
- *`Text to be replaced with text that you supply`* is represented using *`this font`*.

  Example:
  "Enter the `ls -l` *`filename`* command" means you must replace *`filename`* with your own text.
- **Keyboard keys and graphical interface items (such as buttons, tabs, and menu items)** are represented using **this font**.

  Examples:
  The **Control** key, the **OK** button, the **General** tab, the **Options** menu.
- **Menu —> Submenu** represents a menu selection you can perform.

  Example:
  "Select the **Partition —> Create Partition** action" means you must select the **Create Partition** menu item from the **Partition** menu.
- `Example screen output` is represented using `this font`.

Contents

Figures

Tables

Procedures

1 nPartition System Overviews

This chapter introduces Hewlett-Packard's nPartition system features, server models, nPartition concepts, administration tools, and HP-UX 11i support for nPartitions.

- The first sections of this chapter introduce the nPartition system environment and the HP servers that support nPartitions.
- Details about nPartition concepts are given starting with the section *Overview of nPartitions* on page 60.
- Descriptions of tools and HP-UX features for using nPartitions are given starting with the section *Tools for Managing nPartitions* on page 70.

For nPartition configuration requirements and related HP recommendations, refer to the chapter *Planning nPartition Configurations* on page 109.

For procedures to manage nPartitions, refer to the chapter *Managing nPartitions* on page 243.

Introduction

Hewlett-Packard's nPartition system capabilities enable you to configure a single server complex as one large system or as multiple smaller systems.

Each nPartition definition establishes a subset of the server hardware resources that are used as an independent system environment. An nPartition includes: one or more *cells* (containing processors and memory) that are assigned to the nPartition as well as all *I/O chassis* connected to those cells.

All processors, memory, and I/O in an nPartition are used exclusively by software running in the nPartition. Thus, each nPartition runs its own instance of the Boot Console Handler (BCH) interface and independently boots and reboots instances of HP-UX 11i.

By defining multiple nPartitions within an nPartition server, you establish hardware partitioning that enables a single server complex to run multiple instances of the HP-UX 11i operating system.

You also can establish *virtual partitions* within an nPartition on HP Superdome servers. The HP virtual partitions software enables you to further subdivide an nPartition's active hardware resources by using software partitioning to create one or more virtual partitions (vPars). Each virtual partition can load/boot HP-UX B.11.11 independently. Refer to the chapter *Virtual Partitions (vPars) Management on nPartitions* on page 441 for details.

The HP-UX 11i June 2003 release supports nPartitions on the following servers:

- HP rp7405/rp7410 server (`model` string: 9000/800/rp7410)
- HP rp8400 server (`model` string: 9000/800/S16K-A)
- HP Superdome 16-way (`model` string: 9000/800/SD16000)
- HP Superdome 32-way (`model` string: 9000/800/SD32000)
- HP Superdome 64-way (`model` string: 9000/800/SD64000)

For server hardware details see *Supported HP Server Models* on page 34.

You can reconfigure a server's nPartition definitions without physically modifying the server's hardware configuration by using HP's software-based nPartition management tools.

You can reconfigure any nPartition to include more, fewer, and/or different hardware resources. Doing this requires shutting down the operating system running in the nPartition and resetting the nPartition to reconfigure it; this *reboot for reconfig* operation is performed using the `shutdown -R` HP-UX command (using the `-R` option, not `-r`).

With HP's nPartition servers, you can start with a system that meets your needs now and add more components (cells and I/O) as your needs increase.

For example, with a Superdome server you can add cells, I/O chassis, and/or upgrade to larger-capacity systems as needed. A Superdome 16-way server can be upgraded to a Superdome 32-way server, and likewise the Superdome 32-way server can be upgraded to Superdome 64-way server. You also can add I/O expansion cabinets to a Superdome server at any time.

Administration Tools for nPartitions

You can use several administration tools to manage nPartitions in a server complex, including the service processor, consoles, Boot Console Handler (BCH) interfaces, HP-UX commands, and Partition Manager.

- Server complex's **service processor (GSP or MP)**, which includes the Command menu, partition consoles, partition Virtual Front Panels, partition Console Logs, and the Chassis Log viewer.
- **nPartition console and BCH interface**, which provides console access (through the service processor) as well as interactive control before HP-UX has booted on an nPartition.
- **HP-UX nPartition commands** (including `parstatus`, `parcreate`, `parmodify`, and others) enable you to list, monitor, configure, and manage nPartitions from HP-UX.
- The **Partition Manager** utility (`/opt/parmgr/bin/parmgr`) provides a graphical interface for listing and managing nPartitions.

See *Tools for Managing nPartitions* on page 70 for more details.

Supported HP Server Models

The HP servers that support nPartitions include the following models:

- HP rp7405/rp7410 server—See *rp7405/rp7410 Server Model* on page 36.
- HP rp8400 server—See *rp8400 Server Model* on page 38.
- Three models of HP Superdome servers—See *Superdome Server Models* on page 40.

These nPartition servers have different hardware configurations and limits, as described in the following sections, and all include support for nPartitions.

Within each HP nPartition **server cabinet** are multiple **cells**, each of which contains processors and memory. The nPartition cabinets are shown in Figure 1-1 on page 35; cells are discussed in *Cells* on page 47.

Each nPartition server cabinet also may have multiple **I/O chassis** that provide PCI slots for I/O cards. Each I/O chassis connects to one of the cells in the server. See *nPartition I/O Chassis and PCI Card Slots* on page 48.

HP Superdome servers also support optional **I/O expansion cabinets** to provide additional I/O chassis. See *I/O Chassis in HP Superdome IOX Cabinets* on page 51.

All hardware within a server—including all cells, I/O chassis, cables, cabinet hardware, and power and utilities components—is considered to be a **server complex**.

An HP Superdome complex can consist of one cabinet or two server cabinets, and can also include one or two I/O expansion cabinets (to provide additional I/O chassis).

Each HP rp7405/rp7410 or HP rp8400 server complex consists of a single server cabinet only.

Hardware Models: Superdome, rp8400, and rp7405/rp7410 nPartition Servers

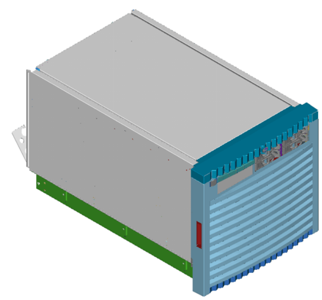
HP Superdome, HP rp8400, and HP rp7405/rp7410 server cabinets are shown in Figure 1-1.

Figure 1-1 HP nPartition Server Hardware

HP Superdome Server Cabinet

HP rp8400 Server Hardware

HP rp7405/rp7410 Server Hardware



rp7405/rp7410 Server Model

HP rp7405/rp7410 servers scale from one to two cells and include complete support for hard partitions (nPartitions).

Figure 1-2 on page 37 shows an overview of the HP rp7405/rp7410 server hardware architecture.

You can configure a single nPartition using one or both cells, or can configure up to two separate nPartitions within an HP rp7405/rp7410 server complex.

In a two-partition HP rp7405/rp7410 complex, you would use cell 0 and its core I/O in one nPartition, and use cell 1 and its core I/O in the other nPartition.

The HP rp7405/rp7410 server model includes these features:

- *A single server cabinet* that includes all cells, I/O chassis, processors, memory, PCI cards, and core I/O.
- *Either one or two cells*. Each cell has up to four PA-RISC processors and up to 16 DIMMs.
- *Two PCI I/O chassis* that share the same chassis hardware.

  One I/O chassis is connected to cell 0, the other is connected to cell 1.

  Each I/O chassis has 8 PCI card slots, numbered from 1 to 8.

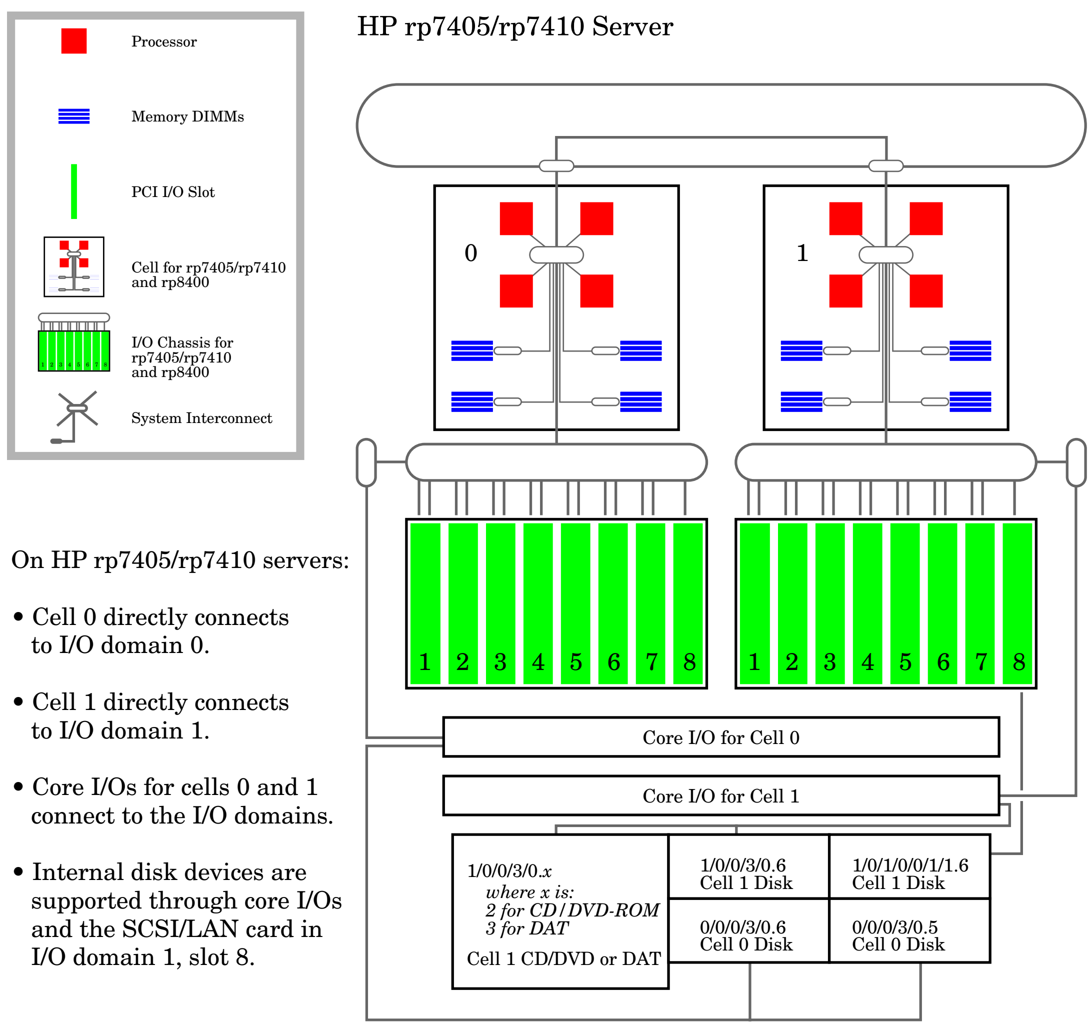
NOTE On HP rp7405/rp7410 servers, two PCI slots by convention are dedicated for use by a combination LAN/SCSI card: PCI domain 0 slot 1 (the first slot on the left) and PCI domain 1 slot 8 (the last slot on the right).

- *Up to two core I/O devices*, one connected to cell 0, and the other connected to cell 1.
- *A total server complex capacity of*: 2 cells, 8 processors, 32 DIMMs, and 16 PCI card slots.
- The model string for HP rp7405/rp7410 servers is `9000/800/rp7410`.

HP rp7405/rp7410 servers currently include a single server cabinet that is rack-mounted only. In the future HP also will support a stand-alone HP rp7405/rp7410 server configuration.

Also see *nPartition System Hardware Details* on page 47 for more information about HP rp7405/rp7410 server features.

Figure 1-2 HP rp7405/rp7410 Server Architecture Overview



On HP rp7405/rp7410 servers:

- Cell 0 directly connects to I/O domain 0.
- Cell 1 directly connects to I/O domain 1.
- Core I/Os for cells 0 and 1 connect to the I/O domains.
- Internal disk devices are supported through core I/Os and the SCSI/LAN card in I/O domain 1, slot 8.

rp8400 Server Model

HP rp8400 servers scale from one to four cells and include complete support for hard partitions (nPartitions).

Figure 1-3 on page 39 shows an overview of the HP rp8400 server hardware architecture.

You can configure a single nPartition using some or all cells, or can configure up to two separate nPartitions within an HP rp8400 server complex.

In a multiple-partition HP rp8400 complex, you would use cell 0 and its core I/O in one nPartition, and use cell 1 and its core I/O in the other nPartition. Any other cells (cells 2 and 3) could be assigned to either of the two nPartitions, or could be unassigned.

The HP rp8400 server model includes these features:

- *A single server cabinet* that includes all cells, I/O chassis, processors, memory, PCI cards, and core I/O.
- *From one to four cells*. Each cell has up to four PA-RISC processors and up to 16 DIMMs.
- *Two PCI I/O chassis* that share the same chassis hardware.

  One I/O chassis is connected to cell 0, the other is connected to cell 1.

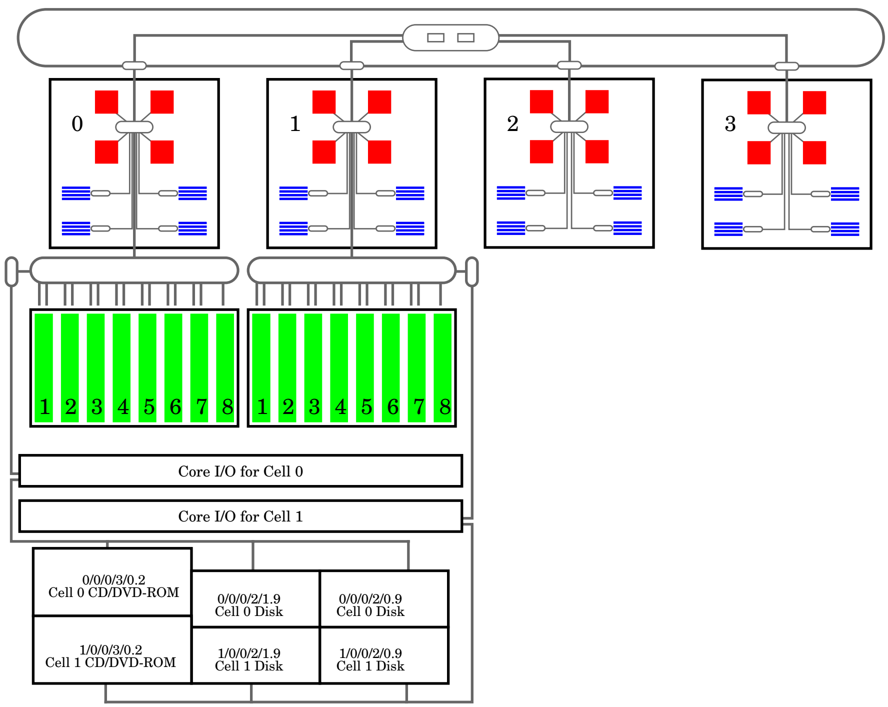
  Each I/O chassis has 8 PCI card slots, numbered from 1 to 8.
- *Up to two core I/O devices*, one connected to cell 0, and the other connected to cell 1.
- *A total server complex capacity of*: 4 cells, 16 processors, 64 DIMMs, and 16 PCI card slots.
- The model string for HP rp8400 servers is `9000/800/S16K-A`.

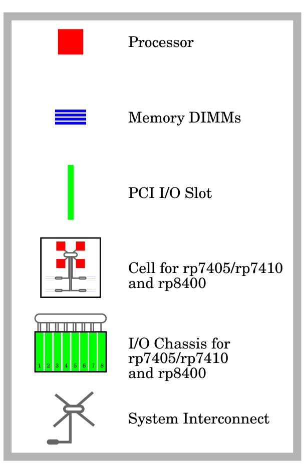
HP rp8400 servers include a single server cabinet that can be rack-mounted or stand-alone.

Also see *nPartition System Hardware Details* on page 47 for more information about HP rp8400 server features.

Figure 1-3 HP rp8400 Server Architecture Overview

HP rp8400 Server





On HP rp8400 servers:

- Cell 0 directly connects to I/O domain 0.
- Cell 1 directly connects to I/O domain 1.
- Cell 2 and cell 3 do not connect to I/O.
- Core I/Os for cells 0 and 1 connect to the I/O domains.
- Internal disk devices are supported through core I/Os.

Superdome Server Models

HP Superdome servers scale up to 16 cells and include complete support for hard partitions (nPartitions). You can configure a single nPartition using some or all cells, or can configure multiple nPartitions within the same Superdome server complex (up to one nPartition for each cell that has core I/O attached).

You can add up to two Superdome I/O expansion cabinets to the Superdome 32-way and 64-way models. Each I/O expansion cabinet has up to six additional 12-slot I/O chassis.

The three Superdome models include: *HP Superdome 16-Way (SD16000) Server*, *HP Superdome 32-Way (SD32000) Server*, and *HP Superdome 64-Way (SD64000) Server*.

Details on these models are given in the following sections.

Also see *nPartition System Hardware Details* on page 47 for more information about HP Superdome hardware features.

HP Superdome 16-Way (SD16000) Server

The HP Superdome 16-way server is a single-cabinet server that has from two to four cells, each with four HP PA-RISC processors and up to 32 DIMMs.

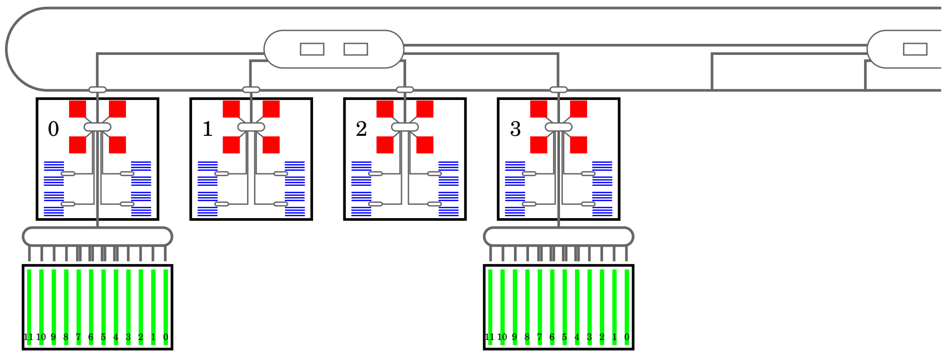
Figure 1-4 on page 41 shows an overview of the Superdome 16-way server hardware architecture.

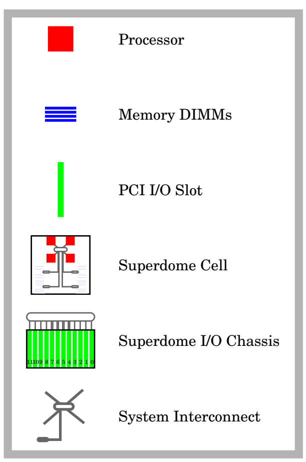
The Superdome 16-way server can have up to 16 processors, 128 DIMMs, and up to four 12-slot PCI I/O chassis.

The model string for Superdome 16-way servers is `9000/800/SD16000`.

Figure 1-4 HP Superdome 16-Way Architecture Overview

HP Superdome 16-Way Server (SD16000)





On HP Superdome 16-Way servers:

- Each cell (0–3) can connect to any one of the available I/O chassis in the cabinet.
- PCI card slot 0 in each I/O chassis is for use by a Superdome core I/O card.

HP Superdome 32-Way (SD32000) Server

The Superdome 32-way server is a single-cabinet server that has from two to eight cells, each with four HP PA-RISC processors and up to 32 DIMMs.

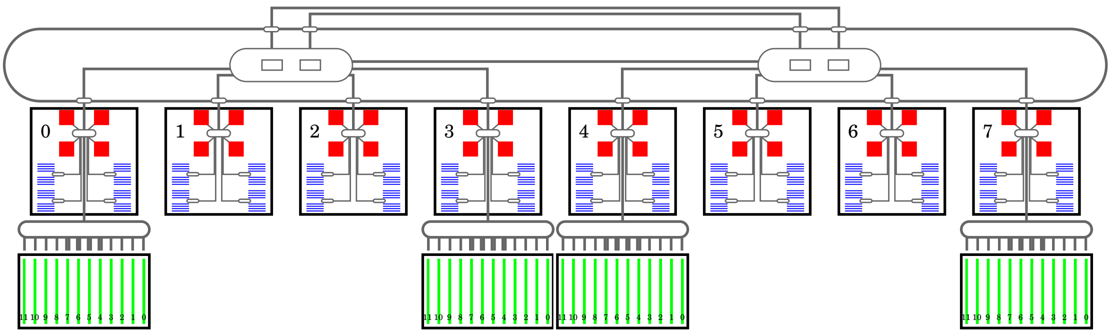
Figure 1-5 on page 43 shows an overview of the Superdome 32-way server hardware architecture.

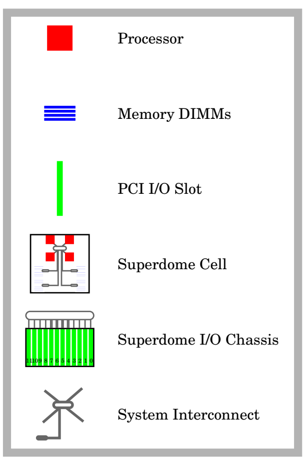
The Superdome 32-way server can have up to 32 processors, 256 DIMMs, up to four internal 12-slot PCI I/O chassis, plus optional I/O expansion cabinet hardware.

The model string for Superdome 32-way servers is `9000/800/SD32000`.

Figure 1-5 HP Superdome 32-Way Architecture Overview

HP Superdome 32-Way Server (SD32000)





On HP Superdome 32-Way servers:

- Each cell (0–7) can connect to any one of the available I/O chassis.
- Additional I/O chassis can be provided in a connected I/O expansion cabinet.
- PCI card slot 0 in each I/O chassis is for use by a Superdome core I/O card.

HP Superdome 64-Way (SD64000) Server

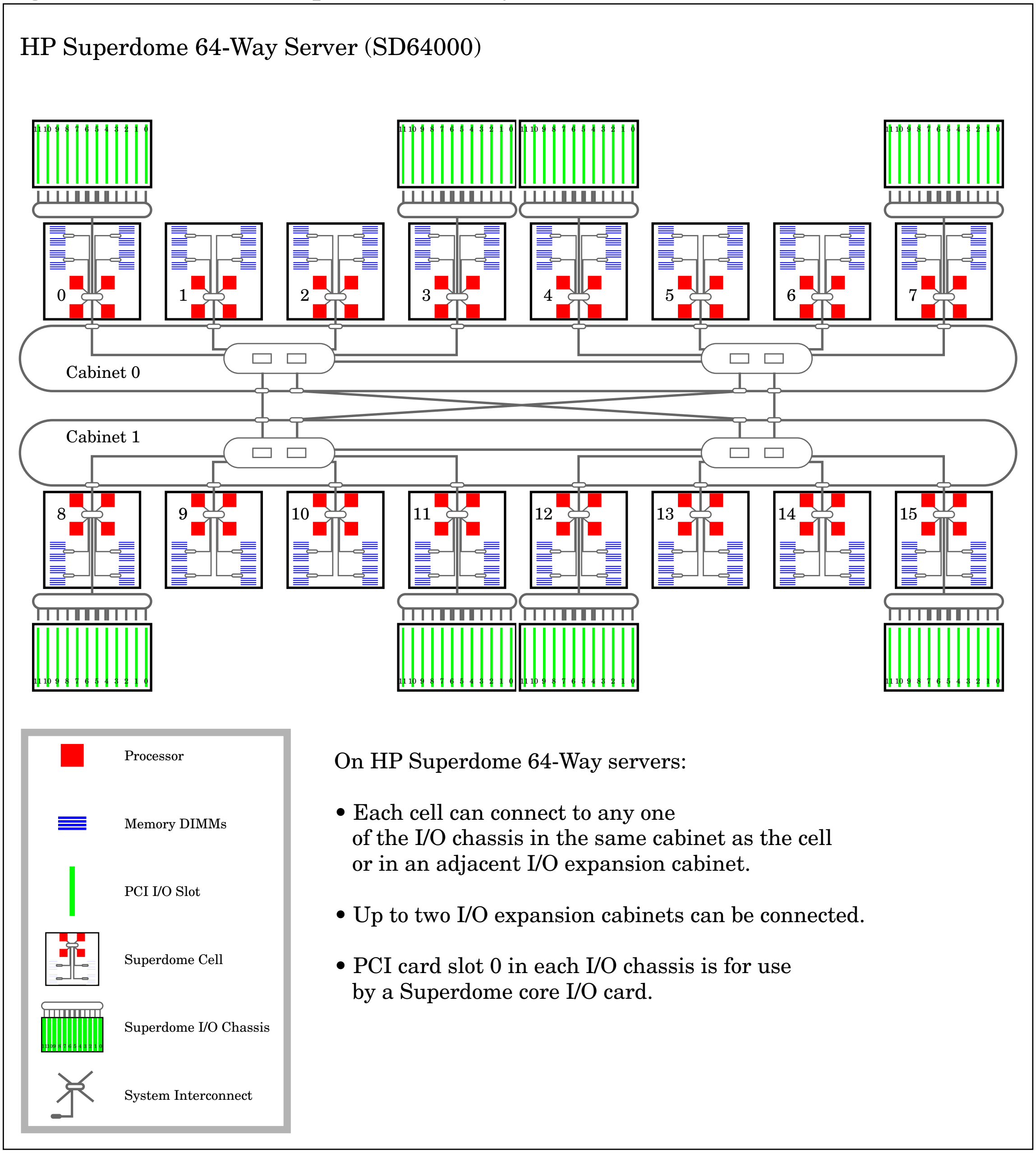
The Superdome 64-way server is a tightly interconnected dual-cabinet server that has from 4 to 16 cells, each with four HP PA-RISC processors and up to 32 DIMMs.

Figure 1-6 on page 45 shows an overview of the Superdome 64-way server hardware architecture.

The Superdome 64-way server can have up to 64 processors, 512 DIMMs, and up to eight internal 12-slot PCI I/O chassis. (Each of the two cabinets in a Superdome 64-way server provides up to 32 processors, 256 DIMMs, and up to four 12-slot PCI I/O chassis.) HP Superdome 64-way servers also can have optional I/O expansion cabinet hardware.

The model string for Superdome 64-way servers is `9000/800/SD64000`.

Figure 1-6 HP Superdome 64-Way Architecture Overview



HP Superdome I/O Expansion Cabinet

HP Superdome 32-way and Superdome 64-way servers can include I/O expansion cabinets in addition to the server cabinet(s) in the complex.

Each I/O expansion cabinet has a cabinet number of either 8 or 9.

A Superdome I/O expansion cabinet includes up to 3 I/O bays, with two 12-slot I/O chassis in each bay. This provides for up to 6 chassis with a total of 72 PCI card slots in each I/O expansion cabinet.

The Superdome I/O expansion cabinet is a standard-size cabinet that, space permitting, you can mount peripherals in as well as I/O chassis.

See the section *I/O Chassis in HP Superdome IOX Cabinets* on page 51 for more details.

Also refer to the book *I/O Expansion Cabinet Guide for Superdome Servers*.

nPartition System Hardware Details

This section gives physical details about the Hewlett-Packard servers that support nPartitions, including HP Superdome, rp8400, and rp7405/rp7410 servers.

The following nPartition server hardware topics are covered here:

- *Cells* on page 47
- *Processors: HP PA-RISC CPUs* on page 48
- *nPartition I/O Chassis and PCI Card Slots* on page 48
- *Internal Disk Devices for HP rp7405/rp7410 and rp8400 Servers* on page 55
- *nPartition Service Processor (GSP or MP) Hardware* on page 56

Also see *Supported HP Server Models* on page 34 for an introduction to the HP nPartition-capable server models, including architectural overviews.

Cells

This section briefly describes cell hardware details for HP's nPartition servers.

Each cell in an HP nPartition server contains HP PA-RISC processors, memory DIMMs, and provides the connection to any I/O chassis attached to the cell.

For details about cell ID formats, see *Specifying Cells and I/O Chassis to Commands* on page 87. For details about configurable cell attributes, see *Cell Properties* on page 61.

All cells assigned to an nPartition **must** have the same firmware revisions and the same type/speed of processors and **should** have identical memory configurations. On HP servers that have multiple nPartitions, each nPartition can have different types of cells.

All processors in a cell **must** be of the same type and speed. All memory DIMMs in a cell **should** be identical for best performance.

In *HP Superdome* servers, each cell can support up to 32 memory DIMMs and 1–4 processors.

In *HP rp7405/rp7410 and rp8400* servers, each cell can support up to 16 memory DIMMs and 1–4 processors.

Each HP Superdome cell can be connected to an I/O chassis that resides either in the same cabinet as the cell or in an I/O expansion cabinet.

In HP rp7405/rp7410 and rp8400 servers, cell 0 connects to I/O chassis 0, and cell 1 connects to I/O chassis 1.

Processors: HP PA-RISC CPUs

This section describes the supported processor (CPU) types for HP nPartition servers.

Within each cell in an nPartition server, *all processors* must operate at the same speed. If multiple cells reside in a server, each cell can run a set of processor whose operating speed is different from the processors in the other cell(s) in the server.

In HP Superdome cells, the following processors types are supported: PA8600 (552 MHz) or PA8700 (650, 750, or 875 MHz).

In HP rp7405/rp7410 and rp8400 servers, the supported processor types are: PA8700 (650, 750, or 875 MHz).

To list the operating speed of processors in a cell, you can use the `parstatus` HP-UX command or the `PR` command from an nPartition's BCH Information menu.

For example, `parstatus -V -c 2` lists hardware details about cell 2, including the operating speed and processor type for the cell.

Refer to the chapter *Listing and Managing Server Hardware* on page 305 for details on listing cell processor info, including a reference chart of cell processor frequencies.

nPartition I/O Chassis and PCI Card Slots

This section has details about the I/O chassis and PCI card slot locations in various models of HP nPartition servers and I/O expansion cabinets, and details about slot frequencies and power capabilities.

The following I/O chassis and slots are discussed here:

- *I/O Chassis in HP Superdome Compute Cabinets* on page 49

- *I/O Chassis in HP Superdome IOX Cabinets* on page 51
- *I/O Chassis for HP rp7405/rp7410 and rp8400 Servers* on page 54

I/O Chassis in HP Superdome Compute Cabinets

Each HP Superdome I/O chassis can connect to one cell in the same compute cabinet.

A Superdome I/O chassis has 12 slots, numbered from 11 to 0. The HP Superdome core I/O card *fits only in slot 0*.

Card slot details for Superdome I/O chassis are in Table 1-1.

Table 1-1 HP Superdome I/O Chassis: Card Slot Details

Slot Number	11	10	9	8	7	6	5	4	3	2	1	0
Signaling (Volts)	5.0 / 3.3				3.3				5.0 / 3.3			
Hardware Fabric	single rope				dual rope				single rope			
Cards Accepted	PCI cards keyed as 5-volt or universal.				PCI cards keyed as 3.3-volt or universal.				PCI cards keyed as 5-volt or universal.			

In HP Superdome I/O chassis, all slots can operate as 64-bit slots and can accept 66 MHz PCI cards.

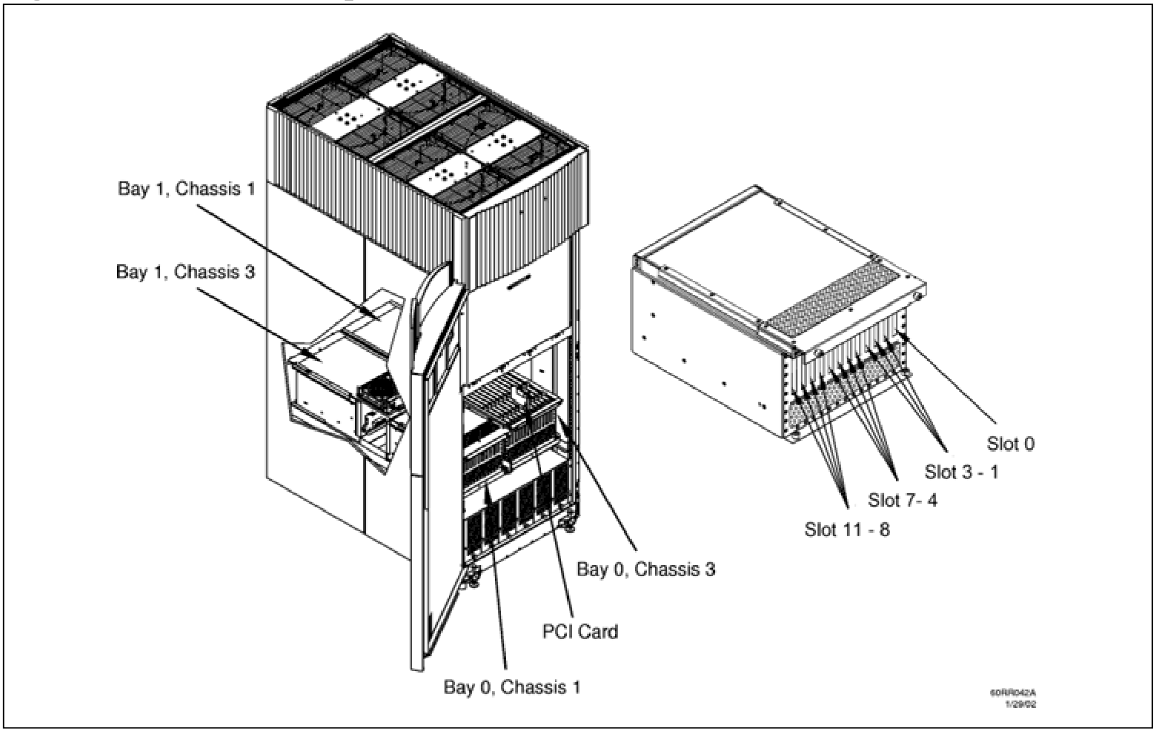
HP Superdome I/O chassis slots 0–3 and 8–11 can accept cards that are physically keyed as 5-volt cards or are keyed as universal cards. These slots normally operate at 33 MHz with 5-volt signaling, but when a 66 MHz capable card is detected they are switched by software to operate at 66 MHz with 3.3-volt signaling.

Slots 4–7 in a Superdome I/O chassis can accept cards that are physically keyed as 3.3-volt cards or are keyed as universal cards. These slots can operate at 66 MHz or 33 MHz with 3.3-volt signaling only.

Slots 4–7 also are **dual-rope slots**, which have two connections between the slot's local bus adapter (LBA) and the I/O chassis system bus adapter (SBA). All other slots are **single-rope slots**, which have a single connection between the slot LBA and the I/O chassis SBA. The dual-rope slots can have greater sustained bandwidth than single-rope slots.

The Superdome I/O chassis and slot locations are shown in Figure 1-7.

Figure 1-7 HP Superdome I/O Chassis and PCI Card Slot Locations



As Figure 1-7 shows, I/O chassis in HP Superdome compute cabinets reside both in the cabinet's front (I/O bay 0) and its rear (I/O bay 1). When you face each I/O bay, the left I/O chassis is chassis 1 and the right chassis is I/O chassis 3. In I/O chassis in a Superdome compute cabinet, PCI slot 11 is to the left and slot 0 is to the right.

In HP Superdome I/O expansion cabinets, the I/O chassis are identical but are positioned sideways, with either slot 0 or slot 11 at the bottom.

Accessing Superdome Compute Cabinet I/O Chassis and PCI Slots

Step 1. To access the I/O chassis in an HP Superdome compute cabinet, you must open either the cabinet's front door (to access I/O bay 0) or its rear door (to access I/O bay 1). In each I/O bay—when facing the bay—I/O chassis 1 is on the left and chassis 3 is on the right.

See Figure 1-7 on page 50 for details.

Step 2. To access the PCI card slots in an HP Superdome compute cabinet's I/O chassis, you must remove the cover from the top of the I/O chassis.

In each Superdome compute cabinet I/O chassis—when facing the chassis—PCI slot 0 is on the right and PCI slot 11 is on the left.

I/O Chassis in HP Superdome IOX Cabinets

Up to two I/O expansion (**IOX**) cabinets can reside in an HP Superdome complex.

I/O expansion cabinets are numbered cabinets 8 and 9.

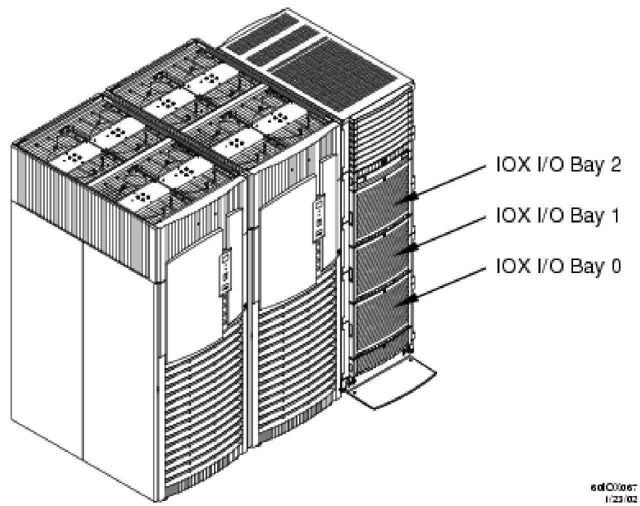
The IOX cabinet uses the same I/O chassis as the Superdome compute cabinet. Each IOX I/O chassis has 12 slots, numbered from 11 to 0. See Table 1-1 on page 49 for details about the card slots.

Each I/O expansion cabinet has its own power supplies, fans, and utilities (which are connected to the Superdome server's service processor bus).

The I/O chassis (and PCI card slots) within each I/O expansion cabinet are made available to nPartitions through direct I/O chassis-to-cell connections—exactly as internal Superdome server cabinet I/O chassis are connected to cells.

Three I/O bays can be housed in each IOX. These bays are numbered from bottom to top: I/O bay 0, bay 1, and bay 2, as shown in Figure 1-8.

Figure 1-8 **I/O Expansion Cabinet (IOX) for HP Superdome**



Each I/O bay in an IOX houses two I/O chassis: the left chassis is I/O chassis 1, right is chassis 3. Thus, an IOX can have up to six I/O chassis that can connect to the cells in an attached Superdome compute cabinet.

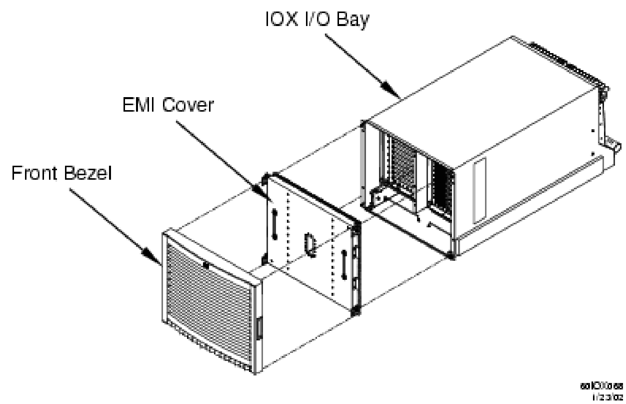
Accessing Superdome IOX I/O Chassis and PCI Card Slots

This procedure describes how to access the I/O chassis and PCI card slots in an HP Superdome I/O expansion cabinet (IOX).

Step 1. To access the I/O chassis in an IOX bay, you must remove the front bezel from the bay, and also remove the EMI cover, as shown in Figure 1-9.

The two I/O chassis in each IOX bay are accessible when the I/O bay slides out from the IOX cabinet.

Figure 1-9 IOX Bezel, Cover, and Bay



Step 2. To access the PCI card slots in an IOX I/O chassis, remove the I/O chassis cover.

To access slots in chassis 1 of the bay remove the cover from the *left* side of the I/O bay, or remove the cover from the *right* side of the bay to access I/O chassis 3's PCI card slots. See Figure 1-9 for details.

I/O Chassis for HP rp7405/rp7410 and rp8400 Servers

HP rp7405/rp7410 and rp8400 servers have two I/O chassis, each with 8 slots numbered left to right from 1 to 8.

Both HP rp7405/rp7410 and rp8400 server cabinets have a single I/O bay on the cabinet's rear that houses the two I/O chassis or "I/O domains". When you face the I/O bay, viewing the rear of the cabinet, the chassis on the left is I/O chassis 0, and right is I/O chassis 1.

Cell 0 connects to chassis 0 and cell 1 connects to chassis 1.

In HP rp7405/rp7410 servers, two PCI card slots *are reserved* for use by a SCSI/LAN card: chassis 0, slot 1 and chassis 1, slot 8. This is a 64-bit card that operates at 66 MHz and 3.3-volt signaling.

HP rp7405/rp7410 and rp8400 I/O chassis card slot details are listed in Table 1-2.

Table 1-2 HP rp7405/rp7410 and rp8400 I/O Chassis: Card Slot Details

<table>
<tr><td>Slot Number</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td></tr>
<tr><td>Signaling (Volts)</td><td>3.3</td><td>3.3</td><td>3.3</td><td>3.3</td><td>3.3</td><td>3.3</td><td>5.0 / 3.3</td><td>5.0 / 3.3</td></tr>
<tr><td>Hardware Fabric</td><td colspan="7">dual rope</td><td>single rope</td></tr>
<tr><td>Cards Accepted</td><td colspan="6">PCI cards keyed as 3.3-volt or universal.</td><td colspan="2">PCI cards keyed as 5-volt or universal.</td></tr>
</table>

All slots in HP rp7405/rp7410 and rp8400 I/O chassis can operate as 64-bit slots and can accept 66 MHz PCI cards.

HP rp7405/rp7410 and rp8400 I/O chassis slots 1–6 accept cards physically keyed as 3.3-volt cards or keyed as universal cards. These slots can operate at 66 MHz or 33 MHz with 3.3-volt signaling only.

Slots 7 and 8 in rp7405/rp7410 and rp8400 I/O chassis accept cards keyed as 5-volt cards or keyed as universal cards. These slots normally operate at 33 MHz with 5-volt signaling, but when a 66 MHz capable card is detected they are switched by software to operate at 66 MHz with 3.3-volt signaling.

On both HP rp7405/rp7410 and rp8400 servers, slots 1–7 are **dual-rope slots**, which have two connections between the slot's local bus adapter (LBA) and the I/O chassis system bus adapter (SBA). The core I/O connections (slot 0) and slot 8 are **single-rope slots**, which have a single connection between the slot LBA and the I/O chassis SBA. The dual-rope slots can have greater sustained bandwidth than single-rope slots.

Internal Disk Devices for HP rp7405/rp7410 and rp8400 Servers

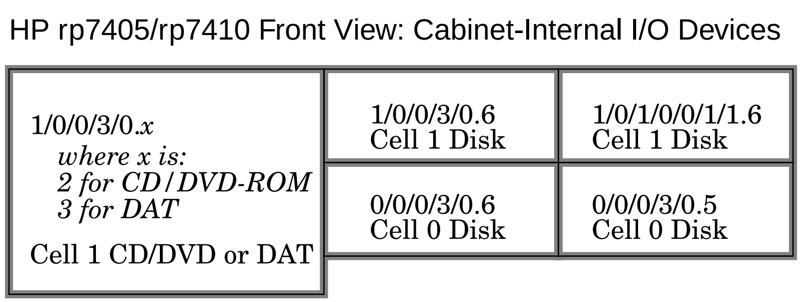
This section covers hardware paths and locations for the internal disk devices in rp7405/rp7410 and rp8400 servers:

- *Disk Devices in HP rp7405/rp7410 Cabinets* on page 55
- *Disk Devices in HP rp8400 Cabinets* on page 56

Disk Devices in HP rp7405/rp7410 Cabinets

The hardware paths for internal drives in an HP rp7405/rp7410 cabinet are shown in Figure 1-10.

Figure 1-10 HP rp7405/rp7410 Internal Storage Hardware Paths



As Figure 1-10 shows, in an HP rp7405/rp7410 cabinet the top internal disk drives connect to cell 1 through the core I/O for cell 1 (for 1/0/0/3/0.6) and the LAN/SCSI card in slot 1_8 (I/O chassis 1 slot 8, for 1/0/1/0/0/1/1.6).

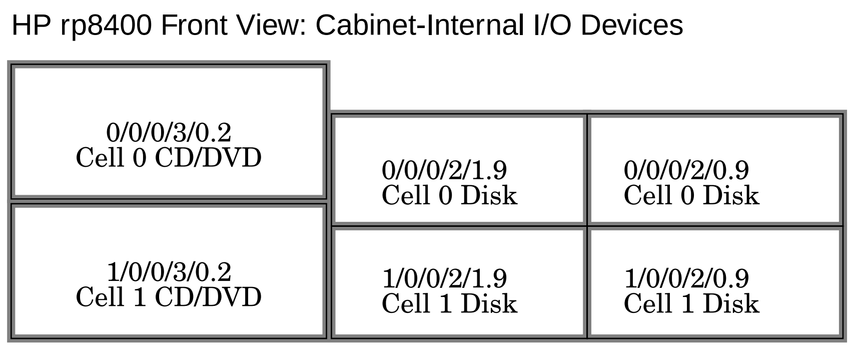
Both of the bottom disk drives (0/0/0/3/0.6 and 0/0/0/3/0.5) connect to cell 0 through the core I/O for cell 0.

A CD/DVD-ROM drive or DAT drive connects to cell 1 through the core I/O card for cell 1, thus it can be accessed through cell 1's nPartition only.

Disk Devices in HP rp8400 Cabinets

The hardware paths for internal drives in an HP rp8400 cabinet are shown in Figure 1-11.

Figure 1-11 HP rp8400 Internal Storage Hardware Paths



As Figure 1-11 shows, in an HP rp8400 cabinet the top internal drives connect to cell 0 through its core I/O card, and the bottom internal drives connect to cell 1 through the cell 1 core I/O card.

nPartition Service Processor (GSP or MP) Hardware

This section introduces the service processor (GSP or MP) hardware in HP's nPartition servers:

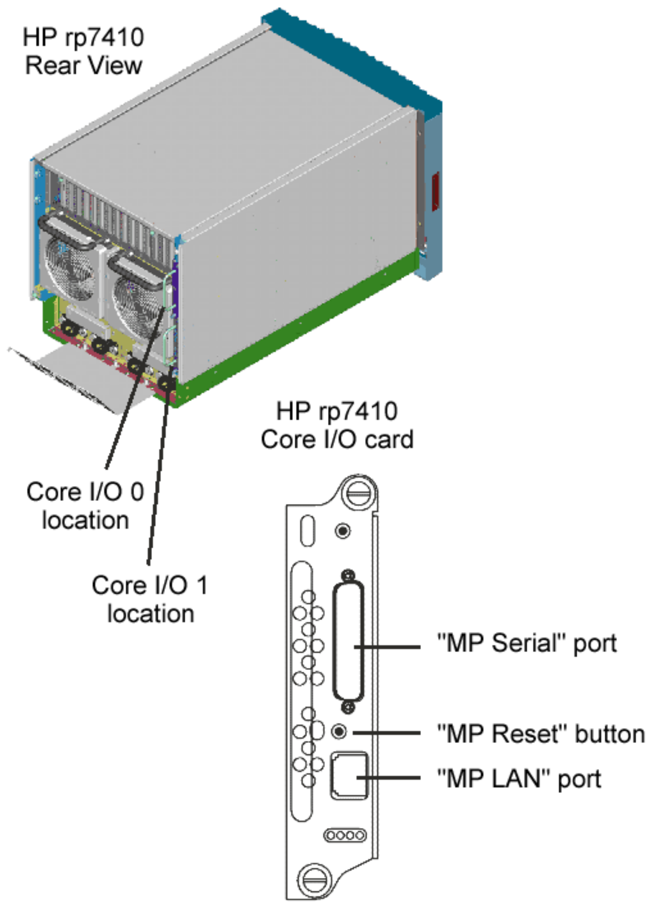
- *Service Processor for HP rp7405/rp7410* on page 57
- *Service Processor for HP rp8400* on page 58
- *Service Processor for HP Superdome* on page 59

For further details on connecting to and using an nPartition server's service processor, refer to the chapter *Using Console and Service Processor Interfaces* on page 125.

Service Processor for HP rp7405/rp7410

On HP rp7405/rp7410 servers, service processor functionality is provided in the core I/O card, shown in Figure 1-12. The rp7405/rp7410 core I/O card's customer LAN port, which permits remote telnet access to the service processor, is labeled "MP LAN". The "MP Serial" port is single DB25 serial port from which three DB9 serial connectors are available (a DB25-to-3xDB9 dongle must be connected). A direct RS-232 serial connection to the service processor is available through the DB9 connector labeled "Console". Remote modem access to the service processor can be provided through the DB9 connector labeled "Remote".

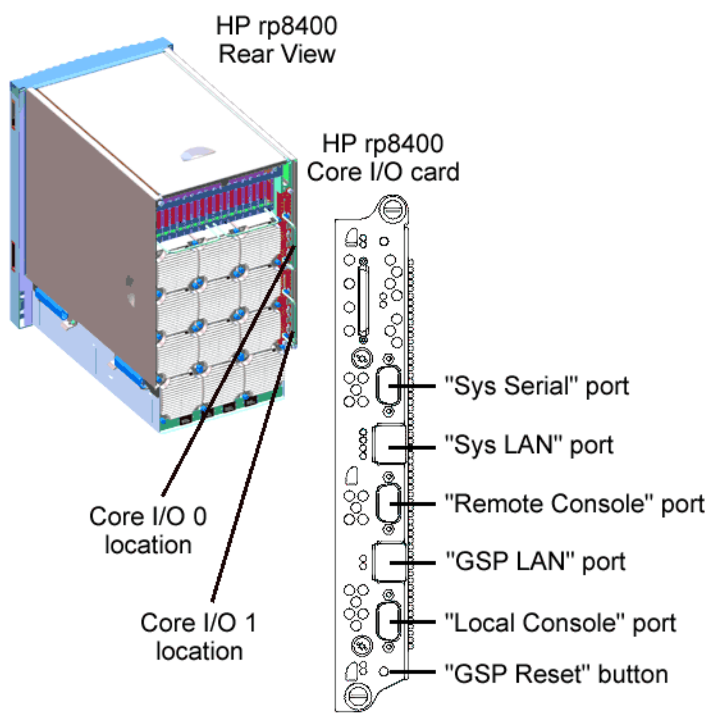
Figure 1-12 HP rp7405/rp7410 Service Processor LAN and Serial Ports



Service Processor for HP rp8400

On HP rp8400 servers, service processor functionality is provided in the core I/O card, shown in Figure 1-13. The rp8400 core I/O card's customer LAN port, which permits remote telnet access to the service processor, is labeled "GSP LAN". A direct RS-232 serial connection to the service processor is available through the "Local Console" port. The "Remote Console" port is for external, remote modem access to the service processor.

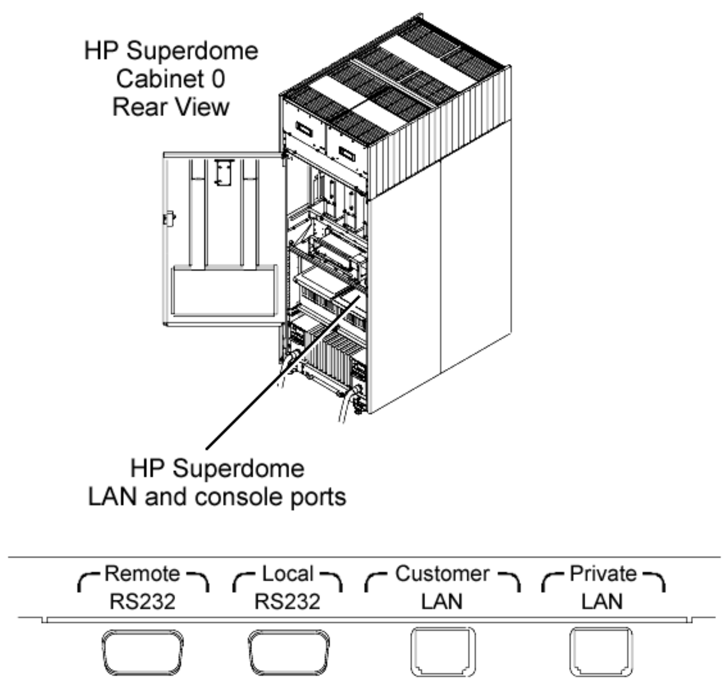
Figure 1-13 HP rp8400 Service Processor LAN and Serial Ports



Service Processor for HP Superdome

On HP Superdome servers, service processor functionality is provided by the “GSP UGUY and SUB” cabinet hardware, which can be seen in the rear of the cabinet above the LAN and console ports (see Figure 1-14). Use the *cabinet 0 (not cabinet 1) LAN and RS-232 connections* to access the service processor. The cabinet 0 “Customer LAN” port provides remote telnet access to the service processor. A direct RS-232 serial connection to the service processor is available through the “Local RS232” port. The “Remote RS232” port is for external, remote modem access to the service processor.

Figure 1-14 HP Superdome Service Processor LAN and Serial Ports



Overview of nPartitions

On HP's nPartition servers, each **nPartition** is a "logical system" that has its own dedicated portion of the server hardware that can run a single instance of the HP-UX 11i operating system. Each nPartition can boot, reboot, and operate independently of any other nPartitions and hardware within the same server complex.

Each nPartition has one or more **cells** (containing processors and memory) that are assigned to the nPartition for its exclusive use. Any **I/O chassis** that is attached to a cell belonging to an nPartition also is assigned to the nPartition. (Each chassis has PCI card slots plus any I/O cards and attached devices, and may also have **core I/O**.)

The **server complex** includes all hardware within an nPartition server: all cabinets, cells, I/O chassis, I/O devices and racks, management and interconnecting hardware, power supplies, and fans.

You can configure one or more nPartitions within a server complex, allowing the hardware to function as a single HP-UX 11i system or as many systems.

The following concepts and issues related to nPartitions are introduced in the rest of this section:

- *Cell Properties* on page 61
- *Genesis Partition* on page 63
- *Partition Numbers* on page 64
- *nPartition Local and Remote Access* on page 64
- *nPartition Active and Inactive States* on page 66

Cell Properties

Cells in an HP nPartition server have various properties that determine how the cells can be used and managed.

The cell properties discussed here include: *Assigned and Unassigned Cells*, *Base Cells*, *Core Cells*, and *Active and Inactive Cells*.

To list details about all cells in a server complex, you can use the `parstatus -C` HP-UX command or Partition Manager.

The `parstatus -C` command output includes the current nPartition assignments, usage, and I/O details for the cells.

```
# parstatus -C
[Cell]
                         CPU     Memory                                Use
                         OK/     (GB)                          Core    On
Hardware   Actual        Deconf/ OK/                           Cell    Next Par
Location   Usage         Max     Deconf    Connected To        Capable Boot Num
========== ============ ======= ========= =================== ======= ==== ===
cab0,cell0 active core  4/0/4    8.0/ 0.0 cab 0,bay0,chassis1 yes     yes  0
cab0,cell1 active base  4/0/4    8.0/ 0.0 -                   no      yes  0
cab0,cell2 active base  4/0/4    8.0/ 0.0 cab 0,bay1,chassis3 yes     yes  0
cab0,cell3 absent       -       -         -                   -       -    -
cab0,cell4 active core  2/0/4    4.0/ 0.0 cab 0,bay0,chassis3 yes     yes  1
cab0,cell5 active base  2/0/4    4.0/ 0.0 -                   no      yes  1
cab0,cell6 active base  2/0/4    4.0/ 0.0 cab 0,bay1,chassis1 yes     yes  1
cab0,cell7 absent       -       -         -                   -       -    -

#
```

Assigned and Unassigned Cells

Each cell in an nPartition server complex either is *assigned* to one of the nPartitions in the complex, or it is *unassigned* and thus is not used by any of the nPartitions. If an I/O chassis is attached to an unassigned cell, then the chassis likewise is not assigned to an nPartition.

Cells that are unassigned are considered to be available resources; they are on the server complex's “free cell list” and are free to be assigned to any of the existing nPartitions, or can be used to create new nPartitions.

Base Cells

For the HP-UX 11i release, all cells within an Partition are **base cells**.

The HP-UX 11i utilities for managing nPartitions automatically set the cell type to base cell, if you do not specify the cell type.

Core Cells

One cell in each nPartition must serve as the **active core cell**. The core cell is a cell that is connected to an I/O chassis that has **core I/O**. The core cell controls the nPartition until HP-UX has booted, and it provides console access for the nPartition.

The core cell's core I/O provides console access for the nPartition through the service processor (GSP or MP).

The monarch processor on the core cell runs the Boot Console Handler (BCH) code while all other processors are idle until HP-UX is booted.

Although an nPartition can have multiple **core-capable cells** (any assigned cell that has an I/O chassis with core I/O), only one core I/O is actively used in an nPartition (the one belonging to the active core cell).

To be *eligible* as a core cell, a cell must be assigned to the nPartition, it must be active, and it must be attached to an I/O chassis containing functional core I/O.

The core cell is selected by system firmware in the early stages of the nPartition boot process.

By default—on HP Superdome and HP rp8400 servers—the lowest numbered eligible cell in an nPartition is selected as the core cell.

By default on HP rp7405/rp7410 servers only, cell 1 is selected as the core cell if it is eligible.

You can define up to four **core cell choices (or "alternates")** for an nPartition (two core-capable cells are currently supported on HP rp7405/rp7410 and HP rp8400 servers). The core cell choices are cells that you prefer to be selected as the nPartition's core cell. If your first core cell alternate cannot be used, then the second choice is checked; if the second choice fails, then any other choices are tried, in the order you specified.

When none of the core cell choices can serve as the active core cell, the nPartition then attempts to select an eligible cell using the default process.

Active and Inactive Cells

Cells that are assigned to an nPartition and have booted to form an nPartition are **active cells** whose resources (processors, memory, and any attached I/O) can be actively used by software running in the nPartition.

Cells that are **inactive** either are not assigned to an nPartition, or they have not participated in *partition rendezvous* to form an nPartition with any other cells assigned to the nPartition. (Partition rendezvous is the point during the nPartition boot process when all available cells in an nPartition join together to establish which cells are active for the current boot of the nPartition.)

For example, a cell can be inactive when it is powered off, has booted with a "n" use-on-next-boot value, or is assigned to an nPartition that has been reset to the ready for reconfig state.

The resources belonging to inactive cells are not actively used by an nPartition. For a cell's resources to be actively used the cell must boot and participate in partition rendezvous.

Genesis Partition

The **Genesis partition** is the initial, one-cell nPartition created within a server complex. The Genesis partition is *just like any other nPartition* except in how it is created.

If your server complex has its nPartitions pre-configured by HP, you do not need to create a Genesis partition.

However, you always have the option of creating a Genesis partition by using the service processor (GSP or MP) Command menu's `CC` command, `G` option, to "wipe out" any existing nPartition definitions and start a new complex configuration that includes only the Genesis partition.

You can use HP-UX utilities running on the Genesis partition as the method for configuring all nPartitions in the complex. The Genesis partition always is partition number 0.

When it is first created, the Genesis partition consists of one cell that is connected to an I/O chassis that has core I/O installed. The Genesis partition also should have a bootable disk (or a disk onto which you can install HP-UX).

If HP-UX is not installed on the Genesis partition's disk(s), you can boot the Genesis partition to the Boot Console Handler (BCH) menu and from that point install HP-UX. This installation requires either having access to an HP-UX install server, or a CD-ROM drive (or DVD-ROM drive) connected to the cell's I/O chassis.

After you boot HP-UX on the Genesis partition, you can modify the nPartition to include additional cells. You also can create other, new nPartitions and can modify them from the Genesis partition or from any other nPartition running HP-UX.

Note that—once you create additional nPartitions—you do not necessarily have to use the Genesis partition to perform your nPartition management and configuration tasks.

Partition Numbers

Each nPartition has its own unique **partition number** that the nPartition commands and utilities use for identifying the nPartition.

When you create an nPartition, the utility you use assigns the nPartition the lowest available partition number. For example, the Genesis partition always is partition number 0 because it is the first and only nPartition in the server complex when it is created, and the second nPartition to be created is partition number 1.

After you remove an nPartition, no cells are assigned to the nPartition. As a result, the nPartition tools can assign cells to the partition number when creating a new nPartition.

For example, if you remove partition number 2, then the `parcreate` command or Partition Manager tool can assign cells to partition number 2 when creating a new nPartition, if all lower-numbered nPartitions (partition numbers 0 and 1) already are defined.

nPartition Local and Remote Access

Your access to an nPartition—whether local or remote—determines your ability to configure and manage the nPartition. Some capabilities require *local* partition access while other capabilities only require that you login to *any* of the nPartitions in the server complex, including remote partitions.

Local nPartition

When you login to HP-UX running on an nPartition, or when you access an nPartition's BCH interface or console, the nPartition you are accessing is considered to be the **local nPartition**.

Remote nPartition

All nPartitions in the complex *other than the one you are accessing* are considered to be **remote nPartitions**.

You can use the `parstatus -w` command to list the partition number for the local nPartition.

```
# parstatus -w
The local partition number is 1.
# parstatus -P
[Partition]
Par              # of  # of I/O
Num Status       Cells Chassis  Core cell  Partition Name (first 30 chars)
=== ============ ===== ======== ========== ==============================
 0  active         2      2     cab0,cell0 feshd2
 1  active         1      1     cab1,cell2 feshd5
#
```

Tools Requirements and Limits for Use in Local and Remote nPartitions

The following list describes many of the administration requirements for using HP-UX tools on a local or remote nPartition. For detailed procedures, refer to these chapters: *Booting and Resetting nPartitions* on page 197, *Managing nPartitions* on page 243, and *Listing and Managing Server Hardware* on page 305.

- **Listing Information**—You can use the `parstatus` command or the Partition Manager utility from *any* nPartition to list nPartition and complex information.
- **Adding (Assigning) a Cell** to an nPartition—You can use `parmodify` or Partition Manager from *any* nPartition to assign a cell to any nPartition in the server complex.
- **Removing (Unassigning) a Cell** from an nPartition—You can unassign an *inactive cell* from its nPartition by using `parmodify` or Partition Manager on any nPartition. However, to unassign an *active cell* you must use these tools from the *local* nPartition (the nPartition to which the cell is assigned).
- **Powering On or Off a Cell**—To power on or off a cell that is *unassigned*, you can use `frupower` or Partition Manager on any nPartition. To power on or off an *assigned cell*, the cell must be *inactive* and you must use `frupower` or Partition Manager from the *local* nPartition (the nPartition to which the cell is assigned).

- **Rebooting or Shutting Down HP-UX**—To reboot or shut down HP-UX you must issue the `/usr/sbin/shutdown` command and appropriate options (such as `-r`, `-R`, `-h`, `-R -H`, or others) from the *local* nPartition.
- **Turning Attention Indicators (LEDs) On or Off**—You can use the `fruled` command or Partition Manager to control the attention indicators for all hardware in the server complex from *any* nPartition.

nPartition Active and Inactive States

Each nPartition's boot state either is *active* or *inactive*.

Active nPartition

An nPartition that is **active** has at least one cell that is active (not in a boot-is-blocked state). When an nPartition is active, the nPartition's available cells complete partition rendezvous and then the Boot Console Handler (BCH) interface is loaded and is displayed on the nPartition's console. HP-UX is loaded and run from BCH on an active partition.

Inactive nPartition

An **inactive partition** is considered to be in the **ready for reconfig** state, because all cells assigned to the nPartition either remain at a boot-is-blocked state or are powered off.

Use the `parstatus -P` HP-UX command to list all nPartitions and their boot states (active or inactive).

```
# parstatus -P
[Partition]
Par              # of  # of I/O
Num Status       Cells Chassis  Core cell  Partition Name (first 30 chars)
=== ============ ===== ======== ========== ===============================
 0  inactive       2      1     ?          feshd5a
 1  active         2      1     cab1,cell2 feshd5b
#
```

To make an inactive partition *active*, use the service processor (GSP or MP) Command menu's `BO` command. The `BO` command clears the boot-is-blocked flag for all cells assigned to the nPartition, thus allowing the cells to rendezvous and enabling the nPartition to run the BCH interface. (If all of an nPartition's cells are powered off, you must power on its cells to enable the nPartition to become active.)

To make a partition *inactive*, you can issue commands from HP-UX, the BCH interface, or the service processor (GSP or MP) Command menu.

- When HP-UX is running on an nPartition, you can make the nPartition inactive by issuing the `shutdown -R -H` command to shut down HP-UX, reboot all cells, and hold all cells at a boot-is-blocked state.
- When the BCH interface is available for an nPartition, you can make the nPartition inactive by issuing the BCH interface's `RECONFIGRESET` command. This reboots all cells assigned to the nPartition and holds all cells at a boot-is-blocked state.
- If an nPartition is active but is not responsive (that is, if you can neither login as root to issue the `shutdown -R -H` command nor access the nPartition's BCH interface from its console), then use the service processor Command menu's `RR` command to make the nPartition inactive. This reboots all cells assigned to the nPartition and holds all cells at a boot-is-blocked state.

CAUTION Issuing the service processor Command menu's `RR` command immediately halts all processing and I/O activity on the specified nPartition. Be certain to *correctly specify* which nPartition is to be reset to the ready for reconfig state.

All three methods above reboot an nPartition and hold all of its cells at boot-is-blocked; as a result the rebooted nPartition is placed in the ready for reconfig (inactive) state.

Complex Profiles

Each HP nPartition server's **Complex Profile** includes the data that determine how the server's hardware is assigned to and used by nPartitions.

When you configure nPartitions and modify nPartition settings, the commands and utilities you use lock and unlock the server's Complex Profile when revising it.

The Complex Profile consists of two parts: *Stable Complex Configuration Data* (complex-wide settings) and *Partition Configuration Data* (individual nPartition settings).

You can modify nPartition configurations (and thus revise the Complex Profile) by using the server's service processor Command menu, nPartition Boot Console Handler (BCH) interfaces, or HP-UX nPartition commands and Partition Manager.

Each Complex Profile contains the following information for the server complex.

- **Stable Complex Configuration Data**

  This portion of the Complex Profile stores complex-wide information, including the following details:

  - The name of the complex
  - Which cells are assigned to which nPartitions, and which cells are unassigned (those on the free cell list, which are available to be assigned to any nPartition)
  - The model number, model string, product numbers, and the serial number for the complex

  The server complex's service processor stores the master copy of the Stable Complex Configuration Data. Each cell also stores a copy if this data.

- **Partition Configuration Data**

  This portion of the Complex Profile stores nPartition-specific information.

The Partition Configuration Data includes the following details for each nPartition in the server complex:

— The nPartition’s name, number, and IP address

— The PRI, HAA, and ALT boot paths and boot actions (path flags)

— The use-on-next-boot setting for each cell

This determines whether the cell is allowed to become active and join (rendezvous) the rest of the cells in the nPartition.

— The core cell choices

This is a list of any cells that are preferred to be selected as the nPartition’s active core cell.

Each nPartition has its own Partition Configuration Data, a copy of which is stored on each cell in the nPartition. The server’s service processor also stores copies of this data for all nPartitions.

The server’s service processor manages all Complex Profile data and keeps all copies of the data coherent.

Complex Profile Locks

Locking and unlocking Complex Profiles is automatically managed by the commands and utilities that you use to configure and modify nPartitions. Portions of the Complex Profile data are updated when you modify nPartition configurations or server complex configurations. For more details on nPartition reconfiguration, including procedures for manually unlocking complex profiles, refer to the chapter *Managing nPartitions* on page 243.

Tools for Managing nPartitions

You can use several different software tools to create, modify, and monitor a server's nPartitions and related server complex hardware.

These tools have capabilities that overlap in some cases, but each tool also has unique features and access requirements.

The tools for managing nPartitions are:

- Service Processor (GSP or MP) menus
- Virtual Front Panel (VFP) interfaces
- Boot Console Handler (BCH) interfaces
- HP-UX nPartition Configuration Commands
- Partition Manager (`/opt/parmgr/bin/parmgr`)
- System Administration Manager (SAM, `/usr/sbin/sam`)

NOTE The service processor in HP servers is sometimes called the Management Processor (MP) and sometimes the Guardian Service Processor (GSP).

Regardless of the name, the service processor in these servers provides approximately the same features and performs essentially the same role.

Throughout this document, the term "service processor" refers to both the MP and GSP service processors.

Table 1-3 lists the nPartition management tools and describes each tool's features and capabilities.

Use Table 1-3 to select the most appropriate nPartition management tool based on the tasks you need to perform and the ways in which you can access the system.

Table 1-3 Management Tools for nPartitions

Partition Tool	Features and Restrictions
Service Processor (GSP or MP) menus	The service processor menus provide a complex-wide service interface that allows access to complex hardware and nPartitions defined within the complex. Also refer to the chapter *Using Console and Service Processor Interfaces* on page 125 for details. • Availability—Using service processor menus requires logging in to the service processor. Your service processor login account determines your level of access to the complex hardware and nPartitions. • Features—Service processor commands, access to nPartition consoles, Virtual Front Panels (VFPs) for live nPartition status details, ability to power cycle hardware, ability to reset and TOC nPartitions, ability to view live chassis codes, and access to console and chassis code log files. • Tasks Supported—Monitoring and listing status for all nPartitions and hardware within a server complex. Viewing chassis codes. nPartition console access. nPartition reset and complex hardware power control.
Virtual Front Panel (VFP) interfaces	The VFP interface provides a real-time display of nPartition and cell *boot states* and *activities*. Also refer to the chapter *Using Console and Service Processor Interfaces* on page 125 for details. • Availability—Viewing the VFP interface for an nPartition (or entire system) requires logging in to the service processor. Your service processor user account determines which nPartition VFPs you can access. • Features—Real-time text summaries of nPartition and cell boot states and activities. • Tasks Supported—Monitoring nPartition boot progress and associated cell status.

Table 1-3 Management Tools for nPartitions (Continued)

Partition Tool	Features and Restrictions
Boot Console Handler (BCH) interface	The BCH interface is the method for interacting with an nPartition before it has booted HP-UX. Each nPartition's BCH interface provides menus for configuring nPartition settings and booting HP-UX. Also refer to the chapter *Using Console and Service Processor Interfaces* on page 125 for details. • Availability—Using an nPartition's BCH interface requires accessing the nPartition's console through the service processor Console menu. • Features—Allows you to select which device and which HP-UX kernel is booted, to configure the *boot actions* for devices, and to software-deallocate CPUs, memory, and cells. • Tasks Supported—Configuring and managing the HP-UX boot process, getting nPartition-specific information, resetting the local nPartition, configuring various nPartition settings.
HP-UX nPartition Configuration Commands	The HP-UX nPartition configuration commands allow you to configure, modify, and monitor nPartitions and hardware within a server complex. See the section *Using HP-UX nPartition Configuration Commands* on page 85 for details. The commands include `parcreate`, `parmodify`, `parstatus`, `parremove`, `parunlock`, `fruled`, and `frupower`. • Availability—Using the HP-UX nPartition configuration commands requires logging in to HP-UX running on an nPartition. All users can issue the `parstatus` and `fruled` commands, but all other commands require `root` user permissions. • Features—These commands allow you to manage nPartitions and hardware when HP-UX is in single– or multi-user mode and when you are logged in with text-only terminal access. • Tasks Supported—Configuring, modifying, and getting information about nPartitions and hardware within a server complex.

Table 1-3 Management Tools for nPartitions (Continued)

Partition Tool	Features and Restrictions
Partition Manager (`parmgr`)	Partition Manager (`/opt/parmgr/bin/parmgr`) provides a graphical interface for configuring, modifying, and managing nPartitions and hardware within a server complex. See the section *Using the Partition Manager Utility* on page 106 for details. • Availability—You can use Partition Manager when HP-UX is running in multi-user mode on the nPartition. You can use Partition Manager as a stand-alone X window application (`parmgr`) and can launch it from SAM. Partition Manager also can be launched from a PC Web browser. • Features—Provides a graphical user interface and also supports Web console access. Performs additional error checking beyond what the HP-UX nPartition configuration commands support. Also supports I/O card online addition and replacement. • Tasks Supported—Configuring, modifying, and getting information about nPartitions and hardware within a hard-partitionable server complex. • Detailed Information—See the `parmgr` online help.
System Administration Manager (SAM)	When using SAM (`/usr/sbin/sam`) in graphical mode, you can launch Partition Manager from SAM. See Partition Manager, above, for details.

HP-UX 11i Release Features

The *HP-UX 11i June 2003 Release Notes* lists the latest feature additions and changes to HP-UX operating system and the various “operating environment” bundles.

Each of the HP-UX operating environment bundles includes its own collection of applications. You can install any *one* of the operating environments at a time.

Use the **`swlist -l bundle`** command to list all installed software bundles, including operating environments.

The *Read Before Installing or Updating HP-UX 11i, June 2003* booklet, which is distributed with HP-UX media, also has current details on release and operating environment features.

The release notes and “Read Before” booklet also are available on the **`http://docs.hp.com/`** Web site.

The HP-UX 11i operating environments are described in the following list.

- HP-UX 11i Operating Environment

  This is an integrated and tested software solution for servers. It contains the base HP-UX 11i operating system and selected drivers and applications.

- HP-UX 11i Enterprise Operating Environment

  This is an operating environment marketed and supported only for commercial servers. It contains everything in the basic HP-UX 11i Operating Environment plus additional applications.

- HP-UX 11i Mission Critical Operating Environment

  This is an operating environment marketed and supported only for commercial servers. It contains everything in the HP-UX 11i Enterprise Operating Environment plus additional applications.

- HP-UX 11i Technical Computing Environment

  This is an operating environment marketed and supported for technical computing servers and workstations. It contains the base HP-UX 11i operating system and selected drivers and applications.

- HP-UX 11i Minimal Technical Operating

  This is an operating environment defined for HP workstations. It contains all the base functionality. However, compared to the Technical Computing Operating Environment, the set of additional applications is greatly reduced.

HP-UX Hardware Paths for nPartitions

The HP-UX hardware path for nPartition systems is provided in the format described here.

The `/usr/sbin/ioscan` HP-UX command reports the hardware path for active components within the nPartition in which the command is issued.

You also can use the `/usr/bin/rad -q` command to list details about active I/O slots and cards in the local nPartition.

NOTE The `ioscan` and `rad` commands only report information about the *currently active* hardware components *in the local partition*.

These commands do not report details for hardware that is not assigned to the local nPartition or hardware that is inactive in the nPartition.

Hardware Paths in nPartitions

The components of nPartition hardware paths are:

`a/b/c/d/e.f.g`

where these components are as described in the following list.

- a

  Is the global cell number.
- b

  Is a processor (10–13), memory (5), or a system bus adapter (0). Each I/O chassis has a single system bus adapter.
- c

  Is a local bus adapter (the *LBA*, one for each PCI card slot in the chassis). The LBA connects its corresponding PCI card slot with the system bus adapter.

NOTE The LBA number *is not necessarily the same* as the PCI slot number.

Use the `rad -q` command to list all active PCI slots in an nPartition along with their corresponding hardware paths. See *PCI Card Slot and Hardware Path Numbering* on page 79.

- `d`

  Is the card's address on the slot's PCI bus.

  Typically this is 0 (zero), although the core I/O card has multiple devices and addresses in a single card.

- `e`

  Is the function for the I/O card. Typically this is 0 (zero) for single-function cards.

- `f`

  Is the target of the I/O device, or SCSI ID.

- `g`

  Is a device-specific address such as a SCSI controller (initiator).

See the *ioscan* (1M) manpage for details on using `ioscan` to list hardware path information.

Example 1-1 ioscan Output for a One-Cell HP Superdome nPartition

The following example shows `ioscan` output for a one-cell nPartition.

In this example, the hardware path for the cell is 12, indicating that the cell is in slot 4 in cabinet 1. See *Specifying Cells and I/O Chassis to Commands* on page 87 for details about cell path formats.

```
# ioscan
H/W Path        Class                         Description
=======================================================
                root
12              cell
12/0                ioa                       System Bus Adapter (804)
12/0/0                 ba                     Local PCI Bus Adapter (782)
12/0/0/0/0                   tty              PCI Serial (103c1048)
12/0/0/1/0                   lan              HP PCI 10/100Base-TX Core
12/0/1                 ba                     Local PCI Bus Adapter (782)
12/0/2                 ba                     Local PCI Bus Adapter (782)
12/0/3                 ba                     Local PCI Bus Adapter (782)
12/0/4                 ba                     Local PCI Bus Adapter (782)
```

```
12/0/6                 ba                    Local PCI Bus Adapter (782)
12/0/6/0/0                   ext_bus         SCSI C87x Ultra Wide Differential
12/0/6/0/0.5                    target
12/0/6/0/0.5.0                     disk      SEAGATE ST39173WC
12/0/6/0/0.6                    target
12/0/6/0/0.6.0                     disk      SEAGATE ST39173WC
12/0/6/0/0.7                    target
12/0/6/0/0.7.0                     ctl       Initiator
12/0/8                 ba                    Local PCI Bus Adapter (782)
12/0/9                 ba                    Local PCI Bus Adapter (782)
12/0/10                ba                    Local PCI Bus Adapter (782)
12/0/11                ba                    Local PCI Bus Adapter (782)
12/0/12                ba                    Local PCI Bus Adapter (782)
12/0/14                ba                    Local PCI Bus Adapter (782)
12/5                memory                   Memory
12/10               processor                Processor
12/11               processor                Processor
12/12               processor                Processor
12/13               processor                Processor
#
```

PCI Card Slot and Hardware Path Numbering

On nPartition servers, the PCI card slot numbers (within an I/O chassis) are not necessarily the same as their *local bus adapter* (LBA) number, such as is reported by the `ioscan` or `rad` HP-UX commands.

Table 1-4 shows the correlations among PCI slots and their LBA numbers.

Table 1-4 I/O Numbering: PCI slots and Busses (LBAs)

PCI Card Slot	HP Superdome LBA Number	HP rp8400 and HP rp7405/rp7410 LBA Number
0	0	0
1	1	8
2	2	10
3	3	12
4	4	14
5	6	6
6	14	4
7	12	2
8	11	1
9	10	—
10	9	—
11	8	—

The **`rad -q`** command lists each active PCI slot, its associated HP-UX hardware path, and other slot details.

The **`ioscan -C ba`** command lists the active PCI slots (“Local PCI Bus Adapter”) for your nPartition.

The order in which `ioscan` reports the PCI slots (in LBA number order) *does not* correspond to the order in which the slots physically are arranged (PCI card slot order, left-to-right or right-to-left).

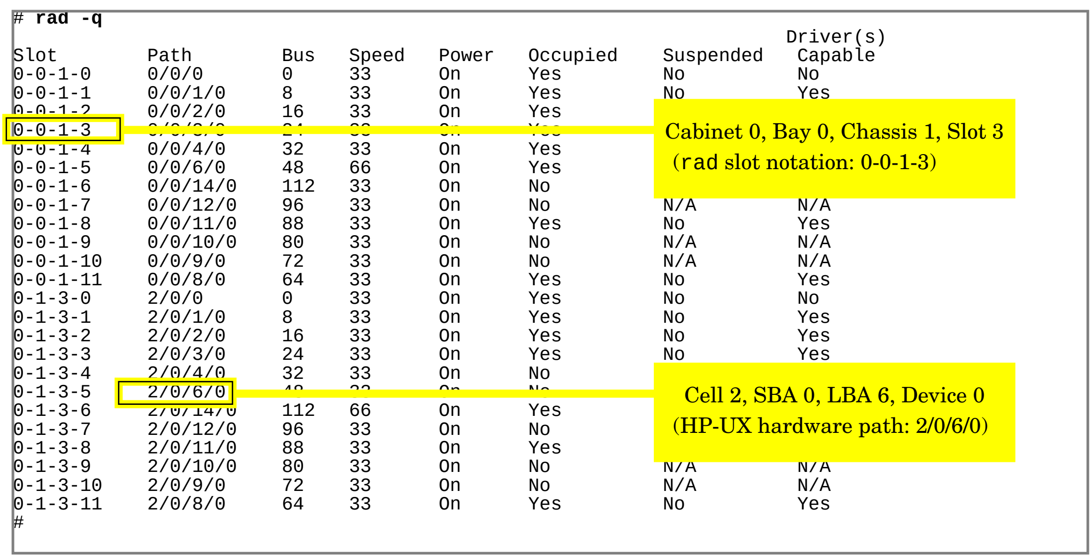
rad Command Output in nPartitions

This section covers the `rad` command's output on nPartitions. For complete details, see the *rad* (1M) manpage.

NOTE When adding or replacing I/O cards, use the SAM (`/usr/sbin/sam`) procedures when possible rather than equivalent `rad` command procedures.

See also the *sam* (1M) and *rad* (1M) manpages.

On HP nPartition servers, the `rad` command reports PCI card slot details as shown in the following example output. The `rad` command reports each available PCI slot (*`cabinet-bay-chassis-slot`*), its corresponding HP-UX hardware path (*`cell/sba/lba/device`*), and more details.



```
# rad -q
                                                                       Driver(s)
Slot          Path          Bus    Speed   Power   Occupied   Suspended  Capable
0-0-1-0       0/0/0         0      33      On      Yes        No         No
0-0-1-1       0/0/1/0       8      33      On      Yes        No         Yes
0-0-1-2       0/0/2/0       16     33      On      Yes
0-0-1-3       0/0/3/0       24     33      On      Yes
0-0-1-4       0/0/4/0       32     33      On      Yes
0-0-1-5       0/0/6/0       48     66      On      Yes
0-0-1-6       0/0/14/0      112    33      On      No
0-0-1-7       0/0/12/0      96     33      On      No         N/A        N/A
0-0-1-8       0/0/11/0      88     33      On      Yes        No         Yes
0-0-1-9       0/0/10/0      80     33      On      No         N/A        N/A
0-0-1-10      0/0/9/0       72     33      On      No         N/A        N/A
0-0-1-11      0/0/8/0       64     33      On      Yes        No         Yes
0-1-3-0       2/0/0         0      33      On      Yes        No         No
0-1-3-1       2/0/1/0       8      33      On      Yes        No         Yes
0-1-3-2       2/0/2/0       16     33      On      Yes        No         Yes
0-1-3-3       2/0/3/0       24     33      On      Yes        No         Yes
0-1-3-4       2/0/4/0       32     33      On      No
0-1-3-5       2/0/6/0       48     33      On      No
0-1-3-6       2/0/14/0      112    66      On      Yes
0-1-3-7       2/0/12/0      96     33      On      No
0-1-3-8       2/0/11/0      88     33      On      Yes
0-1-3-9       2/0/10/0      80     33      On      No         N/A        N/A
0-1-3-10      2/0/9/0       72     33      On      No         N/A        N/A
0-1-3-11      2/0/8/0       64     33      On      Yes        No         Yes
#
```


The `rad` command only lists slots in PCI chassis that are assigned to the local nPartition and are active.

Licensing Information: Getting Product Details

When you license a software product to run on an HP system, you may need to provide machine or system details to the software vendor as part of the software registration process.

This section describes how to obtain information you may need when licensing non-HP software to run on an HP nPartition server.

For complete information about software product licensing, refer to the company that manufactures or sells the software you plan to use.

To license software for use on HP-UX running on an nPartition, you may need to provide the following details about the nPartition or its server complex:

- **Unique Machine (Complex) Identifier**

  `/usr/bin/getconf _CS_MACHINE_IDENT`

- **Unique nPartition Identifier**

  `/usr/bin/getconf _CS_PARTITION_IDENT`

- **Unique Virtual Partition Identifier**

  `/usr/bin/getconf _CS_PARTITION_IDENT`

- **Machine (Complex) Serial Number**

  `/usr/bin/getconf _CS_MACHINE_SERIAL`

  `/usr/sbin/parstatus -X`

- **Server (Complex) Product Number**

  `/usr/sbin/parstatus -X`

- **Hardware (Complex) Model String**

  `/usr/bin/model`

- **HP-UX Version and Installed Bundles**

  For the HP-UX version: `/usr/bin/uname -r`

  For all bundles installed: `/usr/sbin/swlist -l bundle`

nPartition and Virtual Partition Unique Identifiers

NOTE Use the `getconf` command or the `confstr()` call to obtain unique identifiers. Do not use the `uname -i` command, which *does not report unique IDs* for nPartition systems.

In order to guarantee compatibility on current and future platforms, use the interfaces to *getconf* (1) and *confstr* (3C) to retrieve unique machine identifiers.

The interfaces include the `_CS_PARTITION_IDENT` and `_CS_MACHINE_IDENT` parameters:

- For a *nPartition-specific* or a *virtual partition-specific* unique ID use this command:

  /usr/bin/getconf _CS_PARTITION_IDENT

  The unique partition identifier value for a virtual partition environment has virtual partition-specific data added that does not appear for an equivalent non-vPars environment. See the examples that follow.

- For a *complex-specific* unique ID use this command:

  /usr/bin/getconf _CS_MACHINE_IDENT

On HP PA-RISC nPartition servers, the complex, nPartition, and virtual partition unique IDs are based in part on the machine serial number.

To retrieve the machine serial through these interfaces, specify the `_CS_MACHINE_SERIAL` parameter to them.

See the *confstr* (3C) manpage for details on these parameters and their use.

Example 1-2 **Unique IDs for an nPartition and Complex**

The following examples show nPartition-unique and complex-unique IDs returned by the `getconf` command, as well as the local nPartition number and machine serial number.

```
# parstatus -w
The local partition number is 1.
# /usr/bin/getconf _CS_PARTITION_IDENT
Z3e02955673f9f7c9_P1
```

```
# /usr/bin/getconf _CS_MACHINE_IDENT
Z3e02955673f9f7c9
# /usr/bin/getconf _CS_MACHINE_SERIAL
USR2024FP1
#
```

Example 1-3 **Unique IDs for Virtual Partitions (vPars)**

The following example shows the virtual partition-unique ID returned by the `getconf` command, as well as the local nPartition number and the current virtual partition's name.

```
# parstatus -w
The local partition number is 0.
# vparstatus -w
The current virtual partition is Shad.
# getconf _CS_PARTITION_IDENT
Z3e0ec8e078cd3c7b_P0_V00
#
```

For details on virtual partitions, refer to the chapter *Virtual Partitions (vPars) Management on nPartitions* on page 441.

Using HP-UX nPartition Configuration Commands

HP-UX 11i provides you with several HP-UX commands for configuring and managing nPartitions and related server hardware.

The nPartition commands include: `parcreate`, `parmodify`, `parremove`, `parstatus`, `parunlock`, `fruled`, and `frupower`. Table 1-5 on page 86 describes each of these commands.

Using these commands you can create, modify, monitor, and remove nPartitions; get detailed server hardware information; and manipulate attention indicators (LEDs) and power.

When using these commands, you can specify cells and I/O chassis with the notations shown in *Specifying Cells and I/O Chassis to Commands* on page 87.

NOTE The HP-UX nPartition configuration commands are supported only on HP servers that support nPartitions.

These commands are supported by HP-UX kernels built with nPartition support enabled (the `hd_fabric` driver), and they use the libfab.1 library.

Table 1-5 describes the nPartition configuration commands and lists sections where you can find each command's syntax and details.

Table 1-5 HP-UX nPartition Configuration Commands

Command	Description
`parcreate`	Create a new nPartition; `root` permission is required. See *parcreate Command* on page 93.
`parmodify`	Modify an existing nPartition; `root` permission is required. See *parmodify Command* on page 95.
`parremove`	Remove an existing nPartition; `root` permission is required. See *parremove Command* on page 98.
`parstatus`	Display nPartition information and hardware details for a server complex. See *parstatus Command* on page 99.
`parunlock`	Unlock Complex Profile data (use this command with caution); `root` permission is required. See *parunlock Command* on page 101.
`fruled`	Blink the attention indicators (LEDs) or turn them off. This command can control these indicators for cells, I/O chassis, and cabinet numbers. See *fruled Command* on page 102.
`frupower`	Display status or turn power on or off for cells and I/O chassis; `root` permission is required. See *frupower Command* on page 104.

Specifying Cells and I/O Chassis to Commands

Use the cell and I/O chassis notation described in this section when you manage, configure, and inquire about cells and I/O chassis using the HP-UX nPartition configuration commands.

Details are in the *Cell Specification Formats* and *I/O Specification Format* sections that follow.

Cell Specification Formats

Use either of the following two formats to specify cells when using the HP-UX nPartition configuration commands: *Global Cell Number Format* or *Cell Hardware Location Format*.

- **Global Cell Number Format**

  The global cell number format is identical to the cells' HP-UX hardware path, as reported by `ioscan`. In global format, each cell is given a single unique number that indicates the cell's relative location in the entire server complex.

Table 1-6 **Cell IDs in Global Cell Number Format**

Cell Slot	0	1	2	3	4	5	6	7
rp7405/rp7410 Global Format	0	1	—	—	—	—	—	—
rp8400 Global Format	0	1	2	3	—	—	—	—
Superdome Cabinet 0 Global Format	0	1	2	3	4	5	6	7
Superdome Cabinet 1 Global Format	8	9	10	11	12	13	14	15

- **Cell Hardware Location Format**

  In cell hardware location format, each cell is identified using two numbers that specify the *cabinet* and the *cell slot with the cabinet* where the cell resides: *cabinet/slot*.

Table 1-7 Cell IDs in Hardware Location Format

Cell Slot	0	1	2	3	4	5	6	7
rp7405/rp7410 HW Loc Format	0/0	0/1	—	—	—	—	—	—
rp8400 HW Loc Format	0/0	0/1	0/2	0/3	—	—	—	—
Superdome Cabinet 0 HW Loc Format	0/0	0/1	0/2	0/3	0/4	0/5	0/6	0/7
Superdome Cabinet 1 HW Loc Format	1/0	1/1	1/2	1/3	1/4	1/5	1/6	1/7

Both of these cell ID formats specify each cell's precise physical location in a server complex. For example, `parstatus -c9` and `parstatus -c1/1` specify the same cell.

```
# parstatus -c9
[Cell]
                        CPU     Memory                                Use
                        OK/     (GB)                          Core    On
Hardware   Actual       Deconf/ OK/                           Cell    Next Par
Location   Usage        Max     Deconf    Connected To        Capable Boot Num
========== ============ ======= ========= =================== ======= ==== ===
cab1,cell1 active base  4/0/4    8.2/ 0.0 -                   no      yes  1

# parstatus -c1/1
[Cell]
                        CPU     Memory                                Use
                        OK/     (GB)                          Core    On
Hardware   Actual       Deconf/ OK/                           Cell    Next Par
Location   Usage        Max     Deconf    Connected To        Capable Boot Num
========== ============ ======= ========= =================== ======= ==== ===
cab1,cell1 active base  4/0/4    8.2/ 0.0 -                   no      yes  1

#
```

I/O Specification Format

Use the following I/O hardware location format when specifying an I/O chassis to the HP-UX nPartition configuration commands:

cabinet/*bay*/*chassis*

The *cabinet*, *bay*, and *chassis* fields specify the physical location of the I/O chassis. The values of these fields are as follows.

- *cabinet*

  specifies the cabinet number where the I/O chassis resides.

  On HP rp7405/rp7410 and rp8400 servers, the cabinet number always is `0`.

  On HP Superdome servers, the cabinet number can be:

  — `0` — the left Compute cabinet.

  — `1` — the right Compute cabinet, if present.

  — `8` — an I/O Expansion cabinet, if present.

  — `9` — an I/O Expansion cabinet, if present.

- *bay*

  specifies the I/O bay (within a cabinet) where the I/O chassis resides.

  On HP rp8400 and HP rp7405/rp7410 servers, the bay number always is `0`.

  On HP Superdome servers, the bay number can be:

  — `0` — the *front* bay of a Compute cabinet, or the *bottom* bay of an I/O Expansion cabinet.

  — `1` — the *rear* bay of a Compute cabinet, or the *middle* bay of an I/O Expansion cabinet.

  — `2` — the *top* bay in an I/O Expansion cabinet.

- *chassis*

  specifies the I/O chassis (within a bay).

  On HP rp8400 and HP rp7405/rp7410 servers, the chassis number is:

  — `0` — Chassis 0, which connects to cell 0 and is the left chassis as *viewed from the cabinet rear*: the left eight PCI card slots.

— `1` — Chassis 1, which connects to cell 1 and is the right chassis as *viewed from the cabinet rear*: the right eight PCI card slots.

On HP Superdome servers, the chassis number is:

— `1` — Chassis 1, the left chassis in the bay, as viewed when facing the bay/chassis.

— `3` — Chassis 3, the right chassis in the bay, as viewed when facing the bay/chassis.

In HP Superdome servers all chassis are 12-slot I/O chassis, both in Compute cabinets and in I/O Expansion cabinets.

The example below shows the `parstatus` command listing details about two different I/O chassis (cabinet 0/bay 0/chassis 1, and cabinet 0/bay 1/chassis 3).

```
# parstatus -i0/0/1
[Chassis]
                                 Core Connected  Par
Hardware Location   Usage        IO   To         Num
=================== ============ ==== ========== ===
cab0,bay0,chassis1  absent       -    -          -

# parstatus -i0/1/3
[Chassis]
                                 Core Connected  Par
Hardware Location   Usage        IO   To         Num
=================== ============ ==== ========== ===
cab0,bay1,chassis3  active       yes  cab0,cell0 0

#
```

Use the `parstatus -I` command to list all I/O chassis within a server complex, regardless of the chassis cell connections and nPartition assignments.

Use the `rad -q` command to list the *currently available* PCI I/O slots in the local nPartition and their status.

In the following example, both the `parstatus` and `rad` commands show details for various chassis and slots, including chassis 0/1/3.

```
# parstatus -I
[Chassis]
                                 Core Connected  Par
Hardware Location   Usage        IO   To         Num
=================== ============ ==== ========== ===
cab0,bay0,chassis0  absent       -    -          -
cab0,bay0,chassis1  absent       -    -          -
cab0,bay0,chassis2  absent       -    -          -
cab0,bay0,chassis3  inactive     yes  cab0,cell4 -
cab0,bay1,chassis0  absent       -    -          -
cab0,bay1,chassis1  absent       -    -          -
cab0,bay1,chassis2  absent       -    -          -
cab0,bay1,chassis3  active       yes  cab0,cell6 0
cab1,bay0,chassis0  absent       -    -          -
cab1,bay0,chassis1  inactive     -    -          -
cab1,bay0,chassis2  absent       -    -          -
cab1,bay0,chassis3  absent       -    -          -
cab1,bay1,chassis0  absent       -    -          -
cab1,bay1,chassis1  absent       -    -          -
cab1,bay1,chassis2  absent       -    -          -
cab1,bay1,chassis3  active       yes  cab1,cell2 1
cab8,bay0,chassis1  inactive     -    -          -
cab8,bay0,chassis3  active       yes  cab0,cell2 0
cab8,bay1,chassis1  inactive     yes  cab1,cell0 -
cab8,bay1,chassis3  inactive     -    -          -
cab8,bay2,chassis1  absent       -    -          -
cab8,bay2,chassis3  absent       -    -          -
cab8,bay3,chassis1  absent       -    -          -
cab8,bay3,chassis3  absent       -    -          -

# rad -q
                                                                      Driver(s)
Slot         Path         Bus    Speed   Power   Occupied    Suspended  Capable
0-1-3-0      0/0/0        0      33      On      Yes         No         No
0-1-3-1      0/0/1/0      8      33      On      No          N/A        N/A
0-1-3-2      0/0/2/0      16     33      On      No          N/A        N/A
0-1-3-3      0/0/3/0      24     33      On      No          N/A        N/A
0-1-3-4      0/0/4/0      32     33      On      No          N/A        N/A
0-1-3-5      0/0/6/0      48     33      On      Yes         No         Yes
0-1-3-6      0/0/14/0     112    33      On      No          N/A        N/A
0-1-3-7      0/0/12/0     96     33      On      No          N/A        N/A
0-1-3-8      0/0/11/0     88     33      On      Yes         No         Yes
0-1-3-9      0/0/10/0     80     33      On      No          N/A        N/A
0-1-3-10     0/0/9/0      72     33      On      No          N/A        N/A
0-1-3-11     0/0/8/0      64     33      On      No          N/A        N/A
8-0-3-0      2/0/0        0      33      On      Yes         No         No
8-0-3-1      2/0/1/0      8      33      On      No          N/A        N/A
8-0-3-2      2/0/2/0      16     33      On      No          N/A        N/A
8-0-3-3      2/0/3/0      24     33      On      No          N/A        N/A
8-0-3-4      2/0/4/0      32     33      On      No          N/A        N/A
8-0-3-5      2/0/6/0      48     33      On      No          N/A        N/A
8-0-3-6      2/0/14/0     112    33      On      Yes         No         Yes
8-0-3-7      2/0/12/0     96     33      On      No          N/A        N/A
8-0-3-8      2/0/11/0     88     33      On      No          N/A        N/A
8-0-3-9      2/0/10/0     80     33      On      No          N/A        N/A
8-0-3-10     2/0/9/0      72     33      On      No          N/A        N/A
8-0-3-11     2/0/8/0      64     33      On      No          N/A        N/A
#
```

Cabinet 0/Bay 1/Chassis 3 (0/1/3)

nPartition Commands—Details and Syntax

This section has details and command-line syntax for the following HP-UX nPartition configuration commands:

- *parcreate Command* on page 93
- *parmodify Command* on page 95
- *parremove Command* on page 98
- *parstatus Command* on page 99
- *parunlock Command* on page 101
- *fruled Command* on page 102
- *frupower Command* on page 104

NOTE The sections that follow provide useful reference information for using the HP-UX nPartition commands.

For the most current information for these commands, see their online manpages: *parcreate* (1M), *parmodify* (1M), *parremove* (1M), *parstatus* (1), *parunlock* (1M), *fruled* (1M), and *frupower* (1M).

parcreate Command

The /usr/sbin/parcreate command creates a new nPartition.

This command assigns the specified cells (and any attached I/O chassis) to an nPartition after removing the cells from the free cell list. This command assigns a number to the new nPartition and returns the partition number of the newly created nPartition.

Root permission is required to use parcreate.

See the *parcreate* (1M) manpage for complete details. Also refer to the section *Creating a New nPartition* on page 260 for procedures and examples.

Synopsis

parcreate [-P *PartitionName*] [-I *IPaddress*]
-c *cell*:[*cell_type*]:[*use_on_next_boot*]:[*failure_usage*]
[-c...]
[-b *path*] [-t *path*] [-s *path*] [-r *cell*] [-r...] [-B] [-k *s_lock*]

Options

-P *PartitionName*

Specifies the name of the new nPartition.

-I *IPaddress*

Specifies the IP address that should be used by management tools (like SAM) to address this nPartition.

-c *cell*:[*cell_type*]:[*use_on_next_boot*]:[*failure_usage*]

Specifies the cell(s) to be assigned to the nPartition.

- The only valid *cell_type* value is:

base	Base cell (the default).

- The valid *use_on_next_boot* values for cells are:

y	Participate in reboot. (The default.)
n	Do not participate in reboot.

- The only valid *failure_usage* value is:

ri	Reactivate with interleave (the default).

`-b` *`path`*	Specifies the primary (PRI) boot path.
`-t` *`path`*	Specifies the alternate (ALT) boot path.
`-s` *`path`*	Specifies the secondary (HAA) boot path.
`-r` *`cell`*	Specifies the core cell choices. One to four cells can be specified.
`-B`	Specifies to boot the nPartition. The default is not to boot the nPartition and leave it in the ready for reconfig state.

`parmodify` Command

You can use the `/usr/sbin/parmodify` command to modify the following attributes of an existing nPartition:

Partition name
Cell assignments (add cells or remove cells)
Attributes of existing cells (such as the use-on-next-boot value)
Core cell and core alternate cells
Boot paths (the primary, alternate, and HA alternate paths)

Root permission is required to use this command.

See the *parmodify* (1M) manpage for complete details. Also refer to the chapter *Managing nPartitions* on page 243 for procedures and examples.

Synopsis

`parmodify -p` *`PartitionNumber`*
`-a` *`cell`*`:[`*`cell_type`*`]:[`*`use_on_next_boot`*`]:[`*`failure_usage`*`]`
`[-a...] |`
`-m` *`cell`*`:[`*`cell_type`*`]:[`*`use_on_next_boot`*`]:[`*`failure_usage`*`]`
`[-m...] | -I` *`IPaddress`* `| -r` *`cell`* `[-r...] | -d` *`cell`* `[-d...] |`
`-b` *`path`* `| -t` *`path`* `| -s` *`path`* `| -P` *`PartitionName`* `| -B | -k`
`s_lock``:`*`p_lock`*

The `-p` option is required.

Options

The `parmodify` command supports the following command-line options.

`-p` *`PartitionNumber`*

Specifies the nPartition to be modified. *`PartitionNumber`* specifies the unique number (integer) assigned to the nPartition. The `-p` option is required.

Note that you must also to specify any one or more of the following options.

`-a` *`cell`*`:[`*`cell_type`*`]:[`*`use_on_next_boot`*`]:[`*`failure_usage`*`]`

Specifies the cell(s) to be added to the nPartition.

- The valid *`cell_type`* value is:

  `base` Base cell. (The default.)

- The valid *`use_on_next_boot`* values for cells are:

  `y` Participate in reboot. (The default.)

`n` Do not participate in reboot.

- The only valid *failure_usage* value is:

  `ri` Reactivate with interleave (the default).

`-m` *cell*`:[`*cell_type*`]:[`*use_on_next_boot*`]:[`*failure_usage*`]`

Modify attributes of a cell already assigned the nPartition.

For details on *cell_type*, *use_on_next_boot*, and *failure_usage* see the `-a` option's descriptions (above).

`-I` *IPaddress*

Specifies the IP address that should be used by management tools (like SAM) to address this nPartition.

`-r` *cell*

Specifies the core cell and core alternate cells. One to four core cell choices can be specified.

`-d` *cell*

Remove the specified cell from the nPartition.

`-b` *path*

Specifies the primary (PRI) boot path.

`-t` *path*

Specifies the alternate (ALT) boot path.

`-s` *path*

Specifies the secondary (HAA) boot path.

`-P` *PartitionName*

Specifies the name of the nPartition.

`-B` Specifies whether to boot the nPartition. The default is not to boot.

When you modify an *inactive* nPartition and specify the `-B` option, the nPartition is booted (and becomes active) immediately after it is modified.

When you modifying an *active* nPartition and specify the `-B` option, you must perform a reboot for reconfig of the modified nPartition. You must perform this reboot for reconfig before any other cell assignments can take place in the server complex.

`parremove` Command

The `/usr/sbin/parremove` command removes an existing nPartition. This removes all cells from the nPartition and destroys the nPartition definition.

To remove the *local* nPartition (the nPartition from which you issue this command), you must specify the `-F` option.

To remove a *remote* nPartition, the remote nPartition must be inactive: it must be shut down to the ready for reconfig state or the `parremove` command will not be able to remove the nPartition.

Root permission is required to run this command.

See the *parremove* (1M) manpage for complete details. Also refer to the section *Removing (Deleting) an nPartition* on page 278 for procedures and examples.

Synopsis

`parremove -p` *`PartitionNumber`* `[-F]`

Options

`-p` *`PartitionNumber`*

Specifies the nPartition number to be removed.

`-F`

Forcibly remove the nPartition. If the nPartition is inactive, the nPartition is removed. If the nPartition is active and if it is the local nPartition, the nPartition is removed.

If the nPartition is active but is not the local nPartition, then the nPartition *will not* be removed.

parstatus Command

The /usr/sbin/parstatus command displays information about the nPartitions or hardware within a server complex. If you specify no arguments, parstatus lists information about several of the major components of the server complex.

You can specify an individual entity (cell, I/O chassis, cabinet, or nPartition) to restrict the output to information about that component.

All users can issue this command.

See the *parstatus* (1) manpage for complete details. Also refer to the chapters *Managing nPartitions* on page 243 and *Listing and Managing Server Hardware* on page 305 for procedures and examples.

Synopsis

```
parstatus -s

parstatus -w

parstatus [-X]

parstatus [-A] [-M] -C|-I

parstatus [-M] -B|-P

parstatus [-M] -i IOchassis [-i...]

parstatus [-V|-M] -c cell [-c...]

parstatus [-V|-M] -b cabinet [-b...]

parstatus [-V|-M] -p PartitionNumber [-p...]
```

Options

-s	Indicate (through parstatus exit status) whether the system is an HP server that supports nPartitions.
-w	Display the nPartition number for the local nPartition.
-X	Display the server complex's attributes.
-A	Only display the available resources in the complex.
-V	Increase the amount of information displayed.
-M	Produce output suitable for machine parsing.
-C	Show information for all the cells in the complex.
-I	Show information for all I/O chassis in the complex.

`-B` Show information for all cabinets in the complex.

`-P` Show information for all nPartitions in the complex.

`-c` *`cell`*

Show information about the specified cell.

`-i` *`IOchassis`*

Show information about the specified I/O chassis.

`-b` *`cabinet`*

Show information about the specified cabinet.

`-p` *`partition`*

Show information about the specified nPartition.

`parunlock` Command

The `/usr/sbin/parunlock` command unlocks the Stable Complex Configuration Data or Partition Configuration Data.

Use this command with caution.

Root permission is required to run this command.

See the *parunlock* (1M) manpage for details. Also refer to the section *Unlocking Complex Profiles* on page 303.

Synopsis

`parunlock [-p` *`PartitionNumber`*`] [-s]`

`parunlock -A`

Options

`-p` *`PartitionNumber`*

Unlock the Partition Configuration Data of the specified nPartition.

`-s` Unlock the Stable Complex Configuration Data.

`-A` Unlock the Stable Complex Configuration Data and the Partition Configuration Data of all the nPartitions in the complex.

`fruled` Command

The `/usr/sbin/fruled` command *blinks* hardware attention indicators (LEDs) or *turns them off*.

This command can control the cell attention LEDs in all HP nPartition servers, as well as the I/O chassis LEDs on Superdome servers. The `fruled` command also can *start and stop blinking* the cabinet number LCDs on HP Superdome compute cabinets and I/O expansion cabinets.

See the *fruled* (1) manpage for details. Also refer to the section *Turning Attention Indicators (LEDs) On and Off* on page 323 for procedures and examples.

Synopsis

```
fruled [-f|-o] [-B] -c cell [-c...]
fruled [-f|-o] [-B] -i IOchassis [-i...]
fruled [-f|-o] -b cabinet [-b...]
fruled [-f] -C [-l cabinet] [-l...]
fruled [-f] -I [-l cabinet] [-l...]
```

Options

`-f` Turn off specified attention LED(s). This is the default.

The `-f` and `-o` options are mutually exclusive.

`-o` Start blinking the specified attention LED(s). The `-o` option is unavailable with `-C` or `-I`.

`-B` Start or stop blinking the cabinet number LCD of the cabinet that contains the cell or I/O chassis.

The `-B` option is only available with `-c` and `-i`.

`-c` *`cell`*

Blink or turn off the specified *`cell`* attention LED.

`cell` can be specified either in the local (*`cabinet/slot`*) or global (*`cell_ID`*) format.

`-i` *`IOchassis`*

Blink or turn off the specified *`IOchassis`* attention LED.

`-b` *`cabinet`*

Start or stop blinking the cabinet number LCD of the specified *cabinet*.

`-C` Turn off all cell attention LEDs.

`-l` *cabinet*

Limit the scope of the `-C` or `-I` option to a given *cabinet*.

`frupower` Command

The `/usr/sbin/frupower` command turns on, turns off, or displays the current status of power for cells and I/O chassis in nPartition servers.

NOTE The `frupower` command (and Partition Manager) permits you to power on or off *inactive* cells and I/O chassis that are assigned to the current nPartition or are not assigned to any nPartition.

See the *frupower* (1M) manpage for details. Also refer to the section *Powering Cells and I/O Chassis On and Off* on page 312 for procedures and examples.

Syntax

`frupower [ -d | -o | -f ] -c` *`cell`* `[-c...]`

`frupower [ -d | -o | -f ] -i` *`IOchassis`* `[-i...]`

`frupower [-d] -C [-l` *`cabinet`*`] [-l...]`

`frupower [-d] -I [-l` *`cabinet`*`] [-l...]`

Options

`-d` Display power status of the specified cells or I/O chassis. This is the default.

`-o` Power on the specified cells or I/O chassis.

The `-o` and `-f` options are mutually exclusive. The `-o` and `-f` options are unavailable with `-C` and `-I`.

`-f` Power off the specified cells or I/O chassis.

`-c` *`cell`*

The specified *`cell`* is powered on/off or the power status is displayed.

A *`cell`* can be specified either in the local (*`cabinet/slot`*) or global (*`cell_ID`*) format.

`-i` *`IOchassis`*

The specified *`IOchassis`* is powered on/off or the power status is displayed.

`-C` Display power status of all cells. By default the scope is the entire complex if the `-l` option is not specified.

`-I` Display power status of all I/O chassis. The scope is the entire complex if the `-l` option is not specified.

`-l` *`cabinet`*

Limit the scope of the `-C` or `-I` option to the specified *`cabinet`*.

Using the Partition Manager Utility

The Partition Manager utility (`/opt/parmgr/bin/parmgr`) provides a graphical user interface for configuring nPartitions and managing resources within a server complex.

This section introduces these topics about Partition Manager: *Partition Manager Primary Window*, *Running Partition Manager*, *Requirements and Limits*, and *Partition Manager Online Help*.

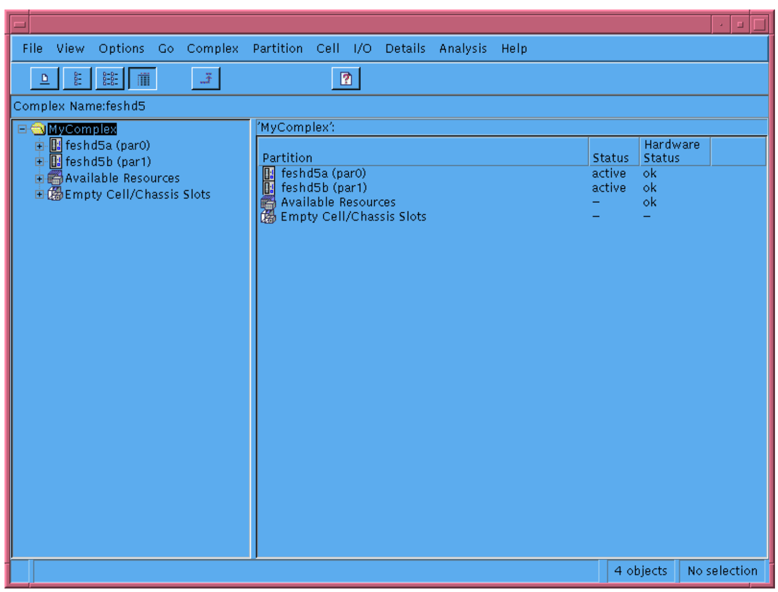
Complete information is in the online help.

Partition Manager Primary Window

The Partition Manager primary window (shown below in Figure 1-15) is the utility's main window for selecting cells, nPartitions, and tasks (menu items).

When you run Partition Manager, by default the program performs an Analyze Complex Health task. If any problems are found, a window reporting those problems is displayed. The primary window is the first window displayed after any complex health analysis results.

Figure 1-15 Partition Manager Primary Window



The left side of the primary window lists all nPartitions, available resources (installed hardware that is not assigned to an nPartition), and empty cell and I/O chassis slots. Selecting an item on the left side of the primary window displays its details on the primary window's right side.

Running Partition Manager

You can access Partition Manager using any one of the following methods.

- Run Partition Manager directly from the HP-UX command line by issuing this command: `/opt/parmgr/bin/parmgr`

  Command-line options are listed in the *parmgr* (1M) manpage.
- Run SAM (`/usr/sbin/sam`) in graphical mode and select **Partition Manager** to launch Partition Manager.
- Access Partition Manager through a PC Web browser.

  Web access requires that an Apache Web server be installed, configured, and activated on the nPartition where you will run Partition Manager. See the online help's **Starting and Exiting** section for Web configuration details.

When running Partition Manager directly or when launching it from SAM, you must set and export the nPartition system's DISPLAY environment variable. The DISPLAY variable specifies where (which X server) the system displays X windows. You also must use the `xhost` command on the X server to grant access for the nPartition system to display windows on the X server.

See the example below and the *X* (1) and *xhost* (1) manpages for details.

nPartition System

```
# hostname
feshd5a
# export DISPLAY=razmataz:0
# printenv DISPLAY
razmataz:0
#
```

X Server

```
$ hostname
razmataz
$ xhost + feshd5a
feshd5a being added to access control list
$
```

Requirements and Limits

The following are requirements and limits of Partition Manager.

See the *parmgr* (1M) manpage for other requirements.

- Partition Manager provides graphical interfaces only, and does not provide a terminal (text mode) interface.
- Using Partition Manager requires `root` permission.
- HP-UX must be running in multi-user mode to support Partition Manager.
- You can run only one instance of Partition Manager or SAM (`/usr/sbin/sam`) per user login session. To run multiple instances of Partition Manger, you must login separately to launch each.

  Both Partition Manager and SAM use the same lock file (/var/sam/lock/lock_console) to ensure that no more than one instance of either application runs at a time per user login session.
- Partition Manager uses the same driver and library as the HP-UX nPartition commands (the hd_fabric driver and libfab.1 library).
- Partition Manager also provides PCI online card add and replace functionality similar to SAM's, and uses the libolrad.1 library for this functionality.

Partition Manager Online Help

The Partition Manager online help gives complete details on using the Partition Manager utility.

Select the **Help —> Overview** menu item for an online overview.

You also can view Partition Manager help from a Web browser by issuing the following command:

```
/opt/netscape/netscape file:/opt/webadmin/parmgr/help/C/assistance.html
```

Web Site for Partition Manager Information: http://www.software.hp.com/products/PARMGR/info.html

You can find online information about Partition Manager, including manpages, help files, and an interactive demonstration version of Partition Manager, at the **`http://www.software.hp.com/products/PARMGR/info.html`** Web site.

2 Planning nPartition Configurations

This chapter describes how you can plan nPartition configurations for HP rp7405/rp7410, rp8400, and Superdome servers. Details include the configuration requirements for nPartitions and HP recommendations.

For related procedures to manage nPartitions, refer to the chapter *Managing nPartitions* on page 243.

Also, for an introduction to nPartition features, refer to the chapter *nPartition System Overviews* on page 31.

nPartition Requirements and Recommendations

The hardware *requirements* shown below determine which cells are eligible to be assigned to an nPartition.

Also consider the nPartition *recommendations*, which can improve an nPartition's performance and availability.

Configuration Requirements for nPartitions

Every nPartition you configure must meet the following hardware requirements.

- ❏ All cells in an nPartition **must** have the same *processor revision level and clock speed*. That is, the IODC_HVERSION must be identical for all processors.
- ❏ The same firmware revision **must** be present on all cells within an nPartition.
- ❏ At least one cell in every nPartition **must** be connected to an I/O chassis that has core I/O.

  Only one core I/O is active per nPartition. If an nPartition has multiple cells that are connected to I/O chassis with core I/O, only the core I/O connected to the *active core cell* is active.

Configuration Recommendations for nPartitions

You also should, as possible, configure nPartitions to meet the following configurations for better performance and availability.

- ❏ Each nPartition's size **should** be a power of two: 1, 2, 4, 8, or 16 cells.

  This provides the best memory interleaving and performance characteristics.

  You can configure nPartitions of any size, but those whose size is a power of two have best memory performance.
- ❏ The I/O chassis containing the active core I/O also **should** have an HP-UX boot disk and method of installing or recovering HP-UX (such as a CD-ROM/DVD-ROM drive, network connection to an install server, or tape drive).

This allows the nPartition to boot or recover HP-UX, even if only the nPartition's core cell is functioning.

- ❏ You **should** assign multiple core-capable cells to each nPartition.

  This allows the nPartition to boot at least to the BCH interface if a core cell fails to boot.

  (Disregard this recommendation if you are configuring multiple nPartitions in an HP rp8400 server or HP rp7405/rp7410 server, each of which has a maximum of two core cells.)

- ❏ The memory configuration of all cells in an nPartition **should** be identical to achieve best performance.

  Each of an nPartition's cells should have:

  — the same *number of DIMMs*

  — the same *capacity (size)* and the same *locations (population)* of DIMMs

  This avoids cell interconnect (crossbar) “hot spots” by distributing memory evenly across all of the nPartition's cells.

- ❏ The memory configuration of each cell **should** include a multiple of two memory ranks per cell.

  Each memory rank is 4 DIMMs. If possible, install memory in sets of 8 DIMMs: 8 DIMMs or 16 DIMMs on HP rp7405/rp7410, HP rp8400, and HP Superdome cells. On HP Superdome cells, you also can install 24 DIMMs or 32 DIMMs per cell.

  This provides a performance improvement by doubling the cell's memory bandwidth, as compared to having one memory rank installed.

  This also can provide an availability improvement, in that if one memory rank fails the cell still has at least one functional rank of memory.

  (At this time memory rank 0 must be functional for a cell to boot.)

- ❏ Each nPartition **should** have PRI (primary), HAA (high-availability alternate), and ALT (alternate) boot paths defined and configured, and their path flags appropriately configured for your purposes.

The PRI and HAA paths should be configured to reference disks that are connected to different cells, if possible, with HAA being a mirror of the root volume and PRI being the root volume. ALT should be the path of a recovery or install device.

Under this configuration, if the cell to which the PRI disk is connected fails or is otherwise inactive and the HAA disk's cell is available, the nPartition still can boot HP-UX.

Even if the PRI and HAA devices connect to the same cell (such as on a multiple-partition HP rp8400 server), the HAA device can be used to boot the nPartition to HP-UX should the PRI device fail.

Configuration Process: Selecting Cells for an nPartition

The following steps provide a basic procedure for selecting which cells to assign to the nPartitions you will create in an HP server.

Selecting Cells for an nPartition

Step 1. Determine the sizes of all nPartitions you will create in the server complex.

Before creating any nPartitions, determine how many nPartitions you plan to configure and establish each nPartition's size (the number of cells).

Step 2. Select the largest undefined nPartition.

If you will configure multiple nPartitions in the complex, assign cells to the largest nPartition first and then configure next largest, and so on, and configure the smallest nPartition last.

Step 3. Choose which cells you will assign to the nPartition by using the nPartition configuration chart for the server model on which you are configuring the nPartitions.

These charts list which cell slots HP supports for assigning to nPartitions, based on the nPartition size and server model.

For nPartition sizes for which HP recommends multiple configurations, select the first available set of cells. For example, for a two-cell nPartition select configuration 2A, if possible, before selecting 2B or 2C.

Step 4. Confirm that the cells you have selected are eligible to be assigned to the nPartition.

For the cells to be eligible, they must meet these requirements:

- The cells **must** not be assigned to another nPartition.
- The cells **must** meet the nPartition hardware requirements (the required processor, firmware, and memory configurations).

- The cells **should** be present (installed) in the server and powered on. You can assign cells that are not present or on when using `parcreate` or `parmodify`. However, you should install and power on cells *before* assigning them to nPartitions in order to allow commands to automatically check the cells' compatibility with any other cells in the nPartition. Also note that assigning a cell that is not present or on will cause the nPartition to wait 10 minutes for the cell during the nPartition boot process, if the cell has a "y" use-on-next-boot setting.

If any of the cells does not adhere to these requirements, go back to *Step 3* and select a different set of cells for the nPartition.

Step 5. Assign the cells to the nPartition.

You can either create a new nPartition that includes the selected cells, or you can modify an existing nPartition so that it conforms to the nPartition configuration recommended by the configuration chart.

For specific procedures for assigning cells, refer to the chapter *Managing nPartitions* on page 243.

Step 6. If you still have additional nPartitions for which to select and assign cells, continue with *Step 2*.

Select the largest remaining undefined nPartition, and go back to *Step 2* to choose and assign cells for it.

HP Superdome nPartition Configuration Guidelines

On HP Superdome servers, the locations of the cells you assign to each nPartition and the resulting loads on server interconnections can affect system performance within the server's nPartitions.

HP offers specific guidelines for configuring nPartitions on HP Superdome servers in order to ensure good system performance.

NOTE The guidelines in this section apply to HP Superdome servers only.

These guidelines follow two basic configuration principles:

1. Avoid sharing interconnecting hardware (crossbars and crossbar links) among multiple nPartitions.
2. *Minimize* the number of crossbar links used by each nPartition, *but do not overload* crossbar links by creating nPartitions that can generate more cell communications traffic across the links than the links can support. Overloading crossbar links degrades performance.

The above principles are incorporated into the guidelines below, and are accounted for in the charts of recommended HP Superdome nPartitions.

Also see *nPartition Requirements and Recommendations* on page 110 for other details.

Configuration Guidelines for HP Superdome nPartitions

Use these guidelines to help determine which cells to assign to the nPartitions you create on HP Superdome servers.

- ❑ **Define nPartitions in order of size.**

  Assign cells to the nPartition that has the largest cell count first. Then select cells for the next largest nPartition, and so on, and finally choose cells for the nPartition with the fewest cells last.

This provides more appropriate cell assignments for larger nPartitions (those with more cells). Any smaller nPartitions with fewer cells are more easily accommodated in the remaining, available cells.

- ❑ **Place each nPartition within an empty cabinet, if possible.**

  This applies to nPartitions in HP Superdome 64-way servers only.

  If possible, assign each nPartition cells from a cabinet whose cells have no nPartition assignments. Do this before assigning cells from a cabinet that already has cells assigned to an nPartition.

  To select cells for nPartitions that are larger than six cells, on HP Superdome 64-way servers, refer *Superdome 64-way Supported nPartition Configurations* on page 121. For such larger nPartitions, assigning some cells from both cabinet 0 and cabinet 1 provides better performance by better distributing cell communications across crossbar links.

  These guidelines can help minimize contentions for using the server's interconnecting hardware (crossbars and crossbar links).

- ❑ **Assign each nPartition cells from an unused "cell quad", if possible.**

  Each "cell quad" is a set of four cells that share the same cabinet backplane connections (crossbar chips). Within each HP Superdome cabinet, cell slots 0–3 comprise one cell quad, and cell slots 4–7 comprise the second cell quad.

  Because cells in a quad share the same crossbar chips, they have the best cross-cell memory performance.

  Partitions with cells on different crossbar chips have higher memory latency (worse memory performance) than nPartitions whose cells all share the same crossbar chip.

Chart of Supported HP rp7405/rp7410 nPartition Configurations

Figure 2-1 lists the nPartition configurations that HP supports for HP rp7405/rp7410 servers.

Figure 2-1 **HP rp7405/rp7410 Supported nPartition Configurations**

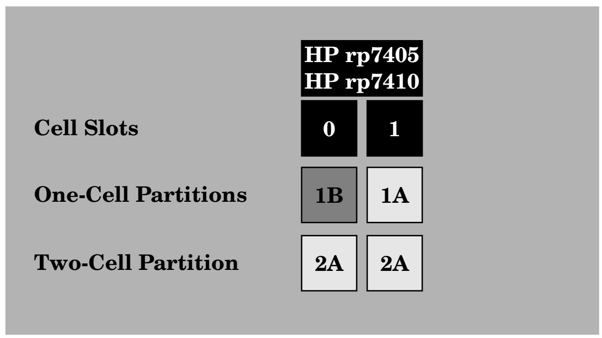


Chart of Supported HP rp8400 nPartition Configurations

Figure 2-2 lists the nPartition configurations that HP supports for HP rp8400 servers.

Figure 2-2 HP rp8400 Supported nPartition Configurations

	HP rp8400			
Cell Slots	**0**	**1**	**2**	**3**
One-Cell Partitions	1A	1B		
Two-Cell Partitions	2A		2A	
	2B			2B
	2C	2C		
		2D	2D	
		2E		2E
Three-Cell Partitions	3A	3A	3A	
	3B		3B	3B
		3C	3C	3C
Four-Cell Partition	4A	4A	4A	4A

On HP rp8400 servers, each nPartition must include either cell 0 or cell 1 because these two cells are the server's only core-capable cells.

Charts of Supported HP Superdome nPartition Configurations

Figure 2-3 lists the nPartition cell configurations that HP supports for Superdome 16-way and Superdome 32-way servers.

Figure 2-4 lists the nPartition cell configurations that HP supports for Superdome 64-way servers.

Example nPartition configurations that use these charts to determine which cells to assign to nPartitions appear in *nPartition Example Configurations for an HP Superdome Server Complex* on page 122.

Figure 2-3 Superdome 16-way and Superdome 32-way Supported nPartition Configurations

	Config Set	Superdome 16-way			
Cell Slots		**0**	**1**	**2**	**3**
One-Cell Partitions	1	1A	1C	1B	2D
Two-Cell Partitions	2	2A	2B	2A	2B
Three-Cell Partition	3	3A	3A	3A	
Four-Cell Partition	4	4A	4A	4A	4A

		Superdome 32-way							
Cell Slots		**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
One-Cell Partitions	5	1A	1E	1C	1G	1B	1F	1D	1H
Two-Cell Partitions	6	2A	2C	2A	2C	2B	2D	2B	2D
	7				2E				2E
Three-Cell Partitions	8	3A	3A	3A		3B	3B	3B	
Four-Cell Partitions	9	4A	4A	4A	4A	4B	4B	4B	4B
Five-Cell Partition	10	5A	5A	5A	5A				5A
Six-Cell Partition	11	6A	6A	6A	6A		6A		6A
Seven-Cell Partition	12	7A	7A	7A	7A		7A	7A	7A
Eight-Cell Partition	13	8A	8A	8A	8A	8A	8A	8A	8A

Figure 2-4 **Superdome 64-way Supported nPartition Configurations**

	Config Set	Superdome 64-way Cabinet 0								Superdome 64-way Cabinet 1							
Cell Slots		0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
One-Cell Partitions	14	1A	1I	1E	1M	1C	1K	1G	1O	1B	1J	1F	1N	1D	1L	1H	1P
Two-Cell Partitions	15	2A	2E	2A	2E	2C	2G	2C	2G	2B	2F	2B	2F	2D	2H	2D	2H
	16														2I	2I	
	17					2J										2J	
	18				2K				2K				2L				2L
Three-Cell Partitions	19	3A	3A	3A		3C	3C	3C		3B	3B	3B		3D	3D	3D	
	20														3E	3E	3E
Four-Cell Partitions	21	4A	4A	4A	4A	4C	4C	4C	4C	4B	4B	4B	4B	4D	4D	4D	4D
	22					4E		4E							4E		4E
Five-Cell Partitions	23	5A	5A	5A	5A				5A	5B	5B	5B	5B				5B
Six-Cell Partitions	24	6A	6A	6A	6A		6A		6A	6B	6B	6B	6B		6B		6B
Seven-Cell Partitions	25	7A	7A	7A	7A		7A	7B	7A	7B	7B	7B	7B		7B	7A	7B
Eight-Cell Partitions	26	8A	8A	8A	8A	8B	8A	8B	8A	8B	8B	8B	8B	8A	8B	8A	8B
Nine-Cell Partition	27	9A	9A	9A	9A	9A	9A		9A					9A		9A	
Ten-Cell Partition	28	10A	10A	10A	10A	10A	10A	10A	10A					10A		10A	
Eleven-Cell Partition	29	11A	11A	11A	11A	11A	11A	11A	11A					11A	11A	11A	
Twelve-Cell Partition	30	12A	12A	12A	12A	12A	12A	12A	12A					12A	12A	12A	12A
Thirteen-Cell Partition	31	13A	13A	13A	13A	13A	13A	13A	13A				13A	13A	13A	13A	13A
Fourteen-Cell Partition	32	14A	14A	14A	14A	14A	14A	14A	14A		14A		14A	14A	14A	14A	14A
Fifteen-Cell Partition	33	15A	15A	15A	15A	15A	15A	15A	15A		15A	15A	15A	15A	15A	15A	15A
Sixteen-Cell Partition	34	16A	16A	16A	16A	16A	16A	16A	16A	16A	16A	16A	16A	16A	16A	16A	16A

nPartition Example Configurations for an HP Superdome Server Complex

This section shows example cell assignments to demonstrate the procedure for selecting cells for two sample server complex configurations.

For reference in the following examples, Figure 2-3 on page 120 and Figure 2-4 on page 121 list a unique number for each nPartition *configuration set*. (For example, config set 6 shows the four two-cell nPartition configurations that HP recommends for Superdome 32-way servers.)

The following two examples are given here:

- *Example nPartition Configuration for a Superdome 32-way Server* on page 123
- *Example nPartition Configuration for a Superdome 64-way Server* on page 124

Example 2-1 **Example nPartition Configuration for a Superdome 32-way Server**

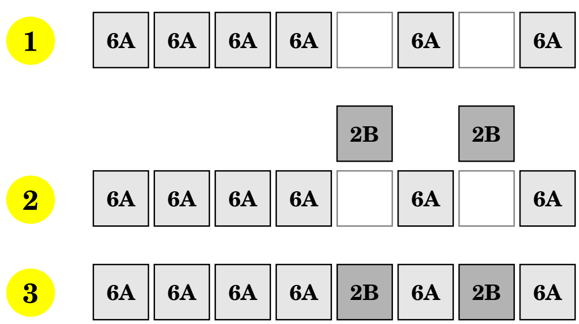
This example configures an HP Superdome 32-way server with one six-cell nPartition and one two-cell nPartition.

A Superdome 32-way server with a six-cell and two-cell nPartition would be configured with nPartitions 6A and 2B, as shown in Figure 2-3 on page 120.

In Figure 2-3, configuration sets 5–13 are eligible to be assigned on Superdome 32-way servers. The nPartition cell assignments are:

1. 6A (config set 11), the recommended six-cell nPartition.
2. 2B (config set 6), because cells 0 and 2 (2A) are assigned to 6A.
3. nPartitions 6A and 2B use all cells in the complex.

Example
Superdome 32-way complex nPartition configuration:
one six-cell nPartition and one two-cell nPartition.



Example 2-2 **Example nPartition Configuration for a Superdome 64-way Server**

This example configures an HP Superdome 64-way server with one seven-cell nPartition and two four-cell nPartitions.

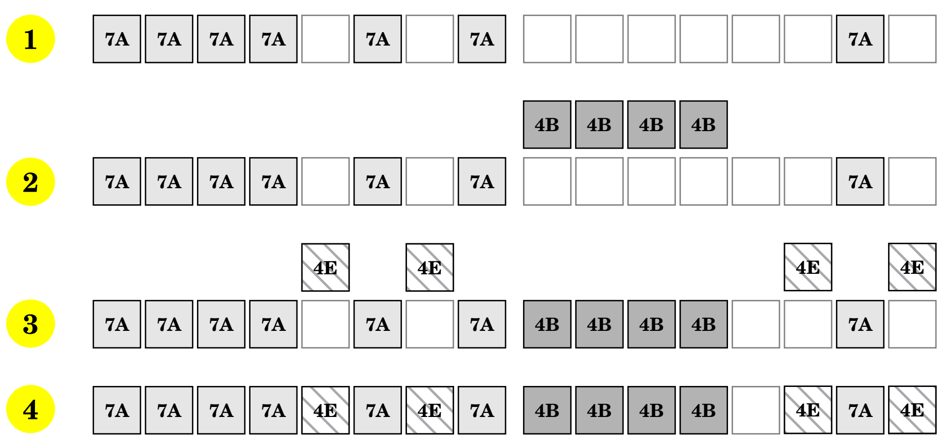
A Superdome 64-way server with a seven-cell nPartition and two four-cell nPartitions would be configured with nPartitions 7A, 4B, and 4E, as shown in Figure 2-4 on page 121.

In Figure 2-4, configuration sets 14–34 are eligible to be assigned on Superdome 64-way servers. The nPartition cell assignments are:

1. 7A (config set 25), the first recommended seven-cell nPartition.
2. 4B (config set 21), because cells in 4A are used by 7A.
3. 4E (config set 22), because some or all cells in 4A–D are assigned.
4. Partitions 7A, 4B, and 4E use all cells except one (cabinet 1, cell 4).

The following illustrations shows how the Superdome 64-way nPartition configurations would be selected, using Figure 2-4 to determine which recommended nPartitions to use.

Example
Superdome 64-way complex nPartition configuration:
one seven-cell nPartition and two four-cell nPartitions.



3 Using Console and Service Processor Interfaces

This chapter covers the service processors and nPartition console interfaces available for HP's nPartition servers.

NOTE

The **service processor** in HP servers is sometimes called the Management Processor (MP) and sometimes the Guardian Service Processor (GSP).

Regardless of the name, the service processor in these servers provides approximately the same features and performs essentially the same role.

Throughout this document, the term "service processor" refers to both the MP and GSP service processors.

Service Processor (GSP or MP) Introduction

The **service processor** (GSP or MP) utility hardware is an independent support system for nPartition servers. It provides a way for you to connect to a server complex and perform administration or monitoring tasks for the server hardware and its nPartitions.

The main features of the service processor include the Command menu, nPartition consoles, console logs, chassis code viewers, and nPartition Virtual Front Panels (live displays of nPartition and cell states).

For details, see *Service Processor Features* on page 128.

The service processor is available when its cabinet has standby power, even if the main (48-volt) cabinet power switch is turned off.

Access to the service processor is restricted by user accounts. Each user account is password protected and provides a specific level of access to the server complex and service processor commands.

Multiple users can independently interact with the service processor because each service processor login session is private. However, some output is mirrored: the Command menu and each nPartition console permit one interactive user at a time and mirror output to all users accessing those features. Likewise, the service processor mirrors live chassis codes to all users accessing the Live Chassis Logs feature.

Up to 16 users can simultaneously login to the service processor through its network (customer LAN) interface and they can independently manage nPartitions or view the server complex hardware states.

Two additional service processor login sessions can be supported by the local and remote serial ports. These allow for serial port terminal access (through the local RS-232 port) and external modem access (through the remote RS-232 port).

In general, the service processor (GSP or MP) on nPartition servers is similar to the service processor on other HP servers, while providing enhanced features necessary for managing a multiple-nPartition server.

For example, the service processor manages the complex profile, which defines nPartition configurations as well as complex-wide settings for the server.

The service processor also controls power, reset, and TOC capabilities, displays and records system events (chassis codes), and can display detailed information about the various internal subsystems.

Service Processor Features

The following list describes the primary features available through the service processor (GSP or MP) on HP rp7405/rp7410, HP rp8400, and HP Superdome servers.

- **Command Menu**

  The Command menu provides commands for system service, status, access configuration, and manufacturing tasks.

  To enter the Command menu, enter **CM** at the service processor Main menu. To exit the service processor Command menu, enter **MA** to return to the service processor Main menu.

  See *Using Service Processor Commands* on page 140 for details.

  Service processor commands are restricted based on the three levels of access: Administrator, Operator, and Single Partition User. See *Service Processor Accounts and Access Levels* on page 131 for details.

- **Consoles**

  Each nPartition in a server complex has its own console.

  Enter **CO** at the service processor Main menu to access the nPartition consoles. To exit the console, type **^b** (**Control-b**).

  See *Console Access to nPartitions* on page 150 for details.

  Each nPartition's console output is reflected to all users currently accessing the console.

  One console user can have interactive access to each nPartition's console, and all other users of the console have read-only access. To gain write access for a console, type **^e cf** (**Control-e c f**).

  Each nPartition's console provides access to:

  — Boot Console Handler (BCH) interface for the nPartition.

  The BCH interface is available if the nPartition has booted but has not yet loaded or booted the HP-UX operating system.

  — HP-UX console for the nPartition.

The nPartition console provides console login access to HP-UX and serves as `/dev/console` for the nPartition.

- **Console Logs**

  Enter **CL** from the service processor Main menu to access the console logs menu. To exit the console log, type **^b** (**Control-b**).

  Each nPartition has its own console log, which has a history of the nPartition console's output, including boot output, BCH activity, and any HP-UX console login activity.

  See *Console Log Viewing* on page 155 for details.

  The console log provides a limited history; it is a circular log file that overwrites the oldest information with the most recent.

  All console activity is recorded in the console's log, regardless of whether any service processor users are connected to the console.

- **Error Logs, Activity Logs, and Live Chassis Codes**

  Enter **SL** to access the chassis log viewer. To exit the chassis viewer type **^b** (**Control-b**).

  Three types of chassis code log views are available: activity logs, error logs, and live chassis code logs.

  See *Chassis Code Log Viewing* on page 156 for details.

  The activity log and error log provide views of past chassis codes.

  The *live chassis code* view provides:

  — Real-time view of chassis codes.

  — Options for filtering the live chassis code output to show only the chassis codes related to a specific cell (**C**), a specific nPartition (**P**), or alerts (**A**, for codes of alert level 3 and higher). Type **U** to view unfiltered codes (all chassis codes).

  All logs (activity, error, and live) can be displayed in different formats, including: keyword format, text format, hex (with keywords), and raw hex format.

  When viewing chassis logs, type **V** followed by a format selector to change the display format.

- **Virtual Front Panel (VFP) for an nPartition**

  Each nPartition's Virtual Front Panel (VFP) displays real-time status of the nPartition boot status and activity, and details about all cells assigned to the nPartition. The VFP display automatically updates as cell and nPartition status changes. A system-wide VFP also is provided.

  Enter **VFP** at the Main menu to access the View Front Panel menu. To exit a Virtual Front Panel, type **^b** (**Control-b**).

  See *Using Virtual Front Panels* on page 159 for details.

Service Processor Accounts and Access Levels

To access the service processor interface for a server complex, you must have a user account that enables you to login to the service processor.

Each server complex has its own set of service processor user accounts, which are defined for the server complex and may differ from accounts on other complexes.

Service processor user accounts have a specific login name, password, and access level.

The three user account access levels are:

- **Administrator Account**

  Provides access to all commands, and access to all nPartition consoles and Virtual Front Panels.

  Can manage user accounts (using the Command menu **SO** command) and can reconfigure various service processor settings.

- **Operator Account**

  Provides access to a subset of commands, and access to all nPartition consoles and Virtual Front Panels.

  Can reconfigure the service processor.

- **Single Partition User Account**

  Provides access to a restricted subset of commands, and access to a single nPartition's console and a single nPartition's Virtual Front Panel.

  Can only execute commands that affect the *assigned nPartition*.

  Cannot execute commands that could potentially affect multiple nPartitions or affect the service processor configuration.

Each user account can permit multiple concurrent login sessions (if it is a "multiple use" account), or restrict account access to a single login session (for "single use" accounts).

Accessing Service Processor Interfaces

This section describes how to login to the service processor (GSP or MP) for an nPartition server complex.

You can connect to a server complex's service processor using the following methods:

- Connecting through the **customer LAN port** by using `telnet`, if login access through the customer LAN is enabled for the service processor.

  On HP Superdome servers, the customer LAN hardware is labeled "Customer LAN". On HP rp8400 servers it is "GSP LAN". On HP rp7405/rp7410 servers it is the only LAN port on the core I/O.

  Use `telnet` to open a connection with the service processor, then login by entering the account name and corresponding password.

- Connecting through the **local RS-232 port** using a direct serial cable connection.

  On HP Superdome server hardware, the local RS-232 port is labeled "Local RS-232". On HP rp8400 servers it is the "Local Console" port. On HP rp7405/rp7410 servers it is the 9-pin D-shaped connector (DB9) labeled "Console".

- Connecting through the **remote RS-232 port** using external model (dial-up) access, if remote modem access is configured.

  On HP Superdome server hardware, the remote RS-232 port is labeled "Remote RS-232". On HP rp8400 servers it is the "Remote Console" port. On HP rp7405/rp7410 servers it is the DB9 connector labeled "Remote".

Example 3-1 Service Processor Login Session

The following output shows a sample login session for a server whose service processor's hostname is "hpsys-s".

```
> telnet hpsys-s
Trying...
Connected to hpsys-s.rsn.hp.com.
Escape character is '^]'.
Local flow control off

MP login: Accountname
MP password:


                           Welcome to the

                           S Class 16K-A

                        Management Processor

(c) Copyright 1995-2001 Hewlett-Packard Co., All Rights
Reserved.

                           Version 0.23



    MP MAIN MENU:

         CO: Consoles
        VFP: Virtual Front Panel
         CM: Command Menu
         CL: Console Logs
         SL: Show chassis Logs
         HE: Help
          X: Exit Connection

MP>
```

Logging in to a Service Processor

This procedure connects to and logs in to a server complex's service processor (GSP or MP) using `telnet` to access the customer LAN.

If connecting through the local or remote RS-232 port, skip *Step 1* (instead establish a direct-cable or dial-up connection) and begin with *Step 2*.

Step 1. Use the HP-UX `telnet` command on a remote system to connect to the service processor for the server complex.

You can connect directly from the command line, for example:

`telnet sdome-g`

or run `telnet` first, and then issue the `open` command (for example, `open sdome-g`) at the `telnet>` prompt.

All `telnet` commands and escape options are supported while you are connected to the service processor. See the *telnet*(1) manpage for details.

(On non-HP-UX platforms such as various PC environments you can instead use an alternate `telnet` program.)

Step 2. Login using your service processor user account name and password.

```
GSP login: Accountname
GSP password: Password
```

Step 3. Use the service processor menus and commands as needed and log out when done.

To log out, select the Exit Connection menu item from the Main menu (enter **`X`** at the `GSP>` prompt or `MP>` prompt).

You also can terminate a login session by issuing the `telnet` escape key sequence **^]** (type: **Control-right bracket**) and entering **`close`** at the `telnet>` prompt.

NOTE If possible, you should log out of any consoles and menus before terminating your telnet session.

If accessing HP-UX on an nPartition, *log out of HP-UX* before exiting the console and service processor sessions. (Otherwise an open HP-UX login session will remain available to any other service processor users.)

Using Service Processor Menus

The service processor (GSP or MP) has a set of menus that give you access to various commands, consoles, log files, and other features.

See *Navigating through Service Processor Menus* on page 138 for details on using these menus.

The following menus are available from the service processor Main menu (which is the menu you first access when logging in):

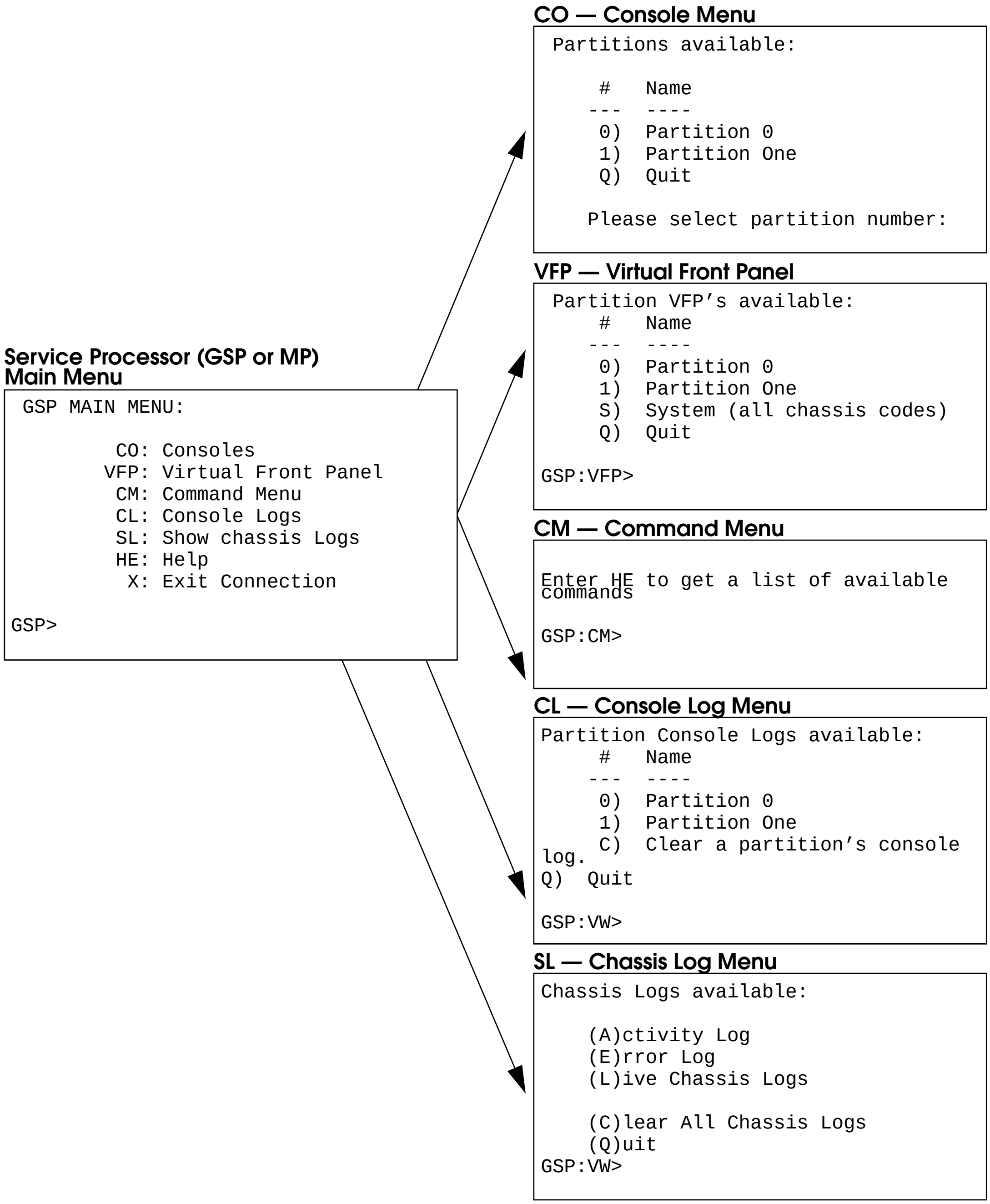
- **Console Menu**—Provides access to consoles for the server's nPartitions.
- **Virtual Front Panel Menu**—Provides a Virtual Front Panel for each nPartition (or for the entire server complex).
- **Command Menu**—Includes service, status, system access, and manufacturing commands.
- **Console Log Viewer Menu**—Allows access to the server's console logs.
- **Chassis Log Viewer Menu**—Allows access to the server's chassis code logs.
- **Help Menu**—Provides online help on a variety of service processor topics and on all service processor Command menu commands.

These menus provide a central point for managing an nPartition server complex outside of HP-UX.

The service processor menus provide many tools and details not available elsewhere. More administration features also are available from the nPartition BCH interfaces, or from HP-UX commands and utilities running on one of the server complex's nPartitions.

NOTE Some specific service processor menu options and features differ slightly on different hardware platforms and firmware revisions. However, most features are identical and behave as described here.

Figure 3-1 Overview of Service Processor (GSP or MP) Menus



Navigating through Service Processor Menus

Figure 3-2 on page 139 shows the commands and options for returning to the service processor Main menu and for ending a service processor login session.

The following list also includes tips for navigating through service processor menus and using various menu features:

- **`Control-b`**

  Exit current console, console log, chassis log, or Virtual Front Panel.

  When accessing an nPartition's console, any log files, or any Virtual Front Panel (VFP), you can exit and return to the Main menu by typing **`^b`** (**Control-b**).

- **`Q`** (or lower-case **`q`**)

  Exit or cancel current menu prompt.

  Enter **`Q`** (or lower-case q) as response to any menu prompt to exit the prompt and return to the previous sub-menu.

  You can do this throughout the service processor menus, including the console menus, various command menu prompts, and the log and VFP menus.

  Note that, from the Command menu prompt (`GSP:CM>` or `MP:CM>`) you must enter **`MA`** (not Q) to return to the Main menu. However, you can enter **`Q`** or **`q`** to cancel any command.

- **Control-]**

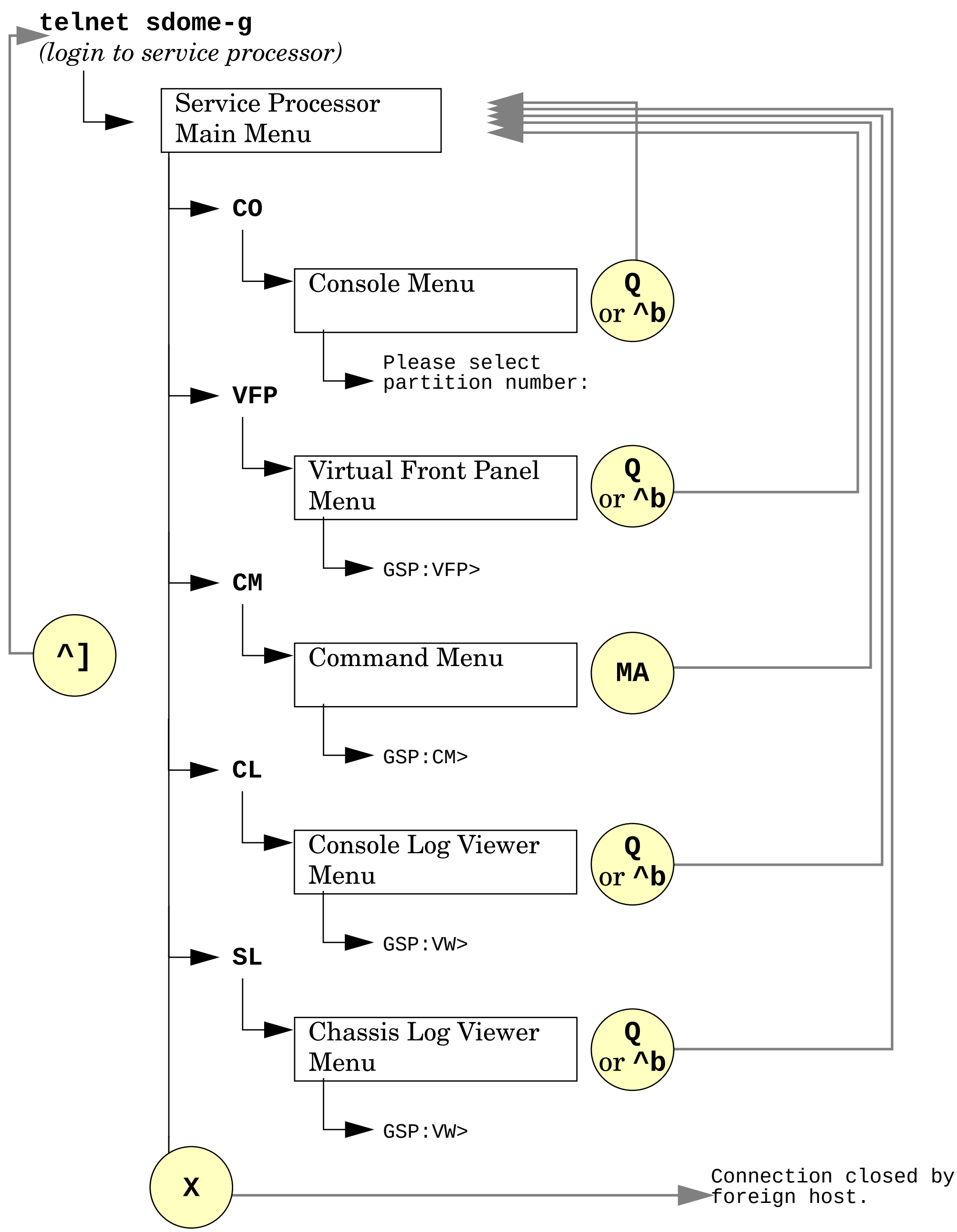
  Escape the service processor connection and return to the telnet prompt.

  At any time during your `telnet` connection to a service processor, you can type the **`^]`** (**Control-right bracket**) escape sequence.

  This key sequence escapes back to the `telnet` prompt. When at the `telnet>` prompt you can use the following commands, among others: **`?`** (print telnet command help information), **`close`** (close the current connection), and **`quit`** (exit telnet).

  To return to the service processor connection, type enter (or return) one or more times.

Figure 3-2 Navigating through Service Processor (GSP or MP) Menus



Using Service Processor Commands

You can issue commands at the service processor Command menu.

To access the service processor Command menu, enter **CM** at the service processor's Main menu. To exit the Command menu, enter the **MA** command to return to the Main menu.

All service processor users accessing the Command menu share access to the menu.

Only one command can be issued at a time. For each command issued, the command and its output are displayed to all users currently accessing the Command menu.

Some commands are restricted and are available only to users who have Administrator or Operator privileges. You can issue any command that is valid at your access level by entering the command at the Command menu prompt (`GSP:CM>` or `MP:CM>`).

When you list commands using the **HE** command, the commands are shown in the following categories:

- Service commands—Support boot, reset, TOC, and other common service activities.
- Status commands—Give command help and system status information.
- System and access configuration commands—Provide ways to configure system security and console and diagnostic settings.

The following sections give more details about the available commands.

Commands Commonly Used at the Service Processor

Table 3-1 summarizes commands that are commonly used by system administrators. These commands are available to all service processor users.

Table 3-1 Service Processor: Commonly Used Commands

Command	Description
BO	Boot an nPartition past an inactive boot-is-blocked (BIB) state to make it active.
CP	Display nPartition cell assignments.
HE	Help: list the available commands.
LS	Display LAN connected console status.
MA	Return to the service processor Main menu.
PD	Set the default nPartition for the current session.
PS	Display detailed power and hardware configuration status.
RS	Reset an nPartition.
RR	Reset an nPartition to a *ready for reconfiguration* state, which makes the nPartition inactive.
SYSREV	Display all cabinet FPGA and firmware revisions. (HP rp8400 and HP rp7405/rp7410 only.)
TC	Send a TOC signal to an nPartition.
TE	Broadcast a message to all users of the Command menu.
WHO	List all users connected to the service processor.

The above commonly used commands appear in the service, status, and the system and access configuration categories.

For additional commands, by category, see the following sections.

Service Processor Commands: Quick Reference

The following tables list commands available from the service processor Command menu:

- *Service Commands* on page 142
- *Status Commands* on page 143
- *System and Access Configuration Commands* on page 144

NOTE For a complete and current list of all service processor commands, enter the **HE** command at the service processor Command menu.

Service Commands

The service commands available provide boot, reset, power, TOC, status, and other commands for common service activities.

Table 3-2 Service Processor: Service Commands

Command	Description
BO	Boot an nPartition past an inactive boot-is-blocked (BIB) state to make it active.
DF	Display FRU information of an entity.
MA	Return to the Main menu.
MFG	Enter the manufacturing mode. (Administrator only.)
MR	Modem reset.
PE	Power entities on or off. (Administrator and operator only.)
RE	Reset entity. (Administrator and operator only.)
RR	Reset an nPartition to a *ready for reconfiguration* state, which makes the nPartition inactive.

Table 3-2 Service Processor: Service Commands

Command	Description
RS	Reset an nPartition.
SYSREV	Display all cabinet FPGA and firmware revisions. (HP rp8400 and HP rp7405/rp7410 servers only.)
TC	Send a TOC signal to an nPartition.
TE	Broadcast a message to all users of the Command menu.
VM	Margin the voltage in a cabinet. (HP Superdome servers only.)
WHO	Display a list of users connected to the service processor.

Status Commands

The status commands provide command help and system status information, such as hardware status and nPartition configurations.

Table 3-3 Service Processor: Status Commands

Command	Description
CP	Display nPartition cell assignments.
HE	Display the list of available commands.
IO	Display I/O chassis connections to cells. (HP Superdome servers only.)
LS	Display LAN connected console status.
MS	Display the status of the modem.
PS	Display detailed power and hardware configuration status.

System and Access Configuration Commands

The system and access configuration commands provide ways to configure system security and console and diagnostic settings. These commands also enable you to modify some complex configuration settings. Some of these commands are restricted (users with an “Operator” or “Single Partition User” access level can issue a subset of these commands).

Table 3-4 Service Processor: System and Access Configuration Commands

Access Level(s)	Command	Description
Administrator	AR	Configure the automatic system restart for an nPartition.
Administrator, Operator	CA	Configure asynchronous and modem parameters.
Administrator, Operator	CC	Initiate a complex configuration.
Administrator	DATE	Set the time and date.
Administrator	DC	Reset parameters to default configuration.
Administrator, Operator	DI	Disconnect remote or LAN console.
Administrator	DL	Disable LAN console access.
Administrator	EL	Enable LAN console access.
Administrator, Operator, Single Partition User	ER	Configure remote/modem port access options.
Administrator, Operator	FW	Firmware update utility. (HP rp8400 and HP rp7405/rp7410 servers only.)
Administrator, Operator, Single Partition User	ID	Display and/or change certain Stable Complex Configuration Data fields, which describe the complex identity.
Administrator	IF	Display network interface information.

Table 3-4 Service Processor: System and Access Configuration Commands

Access Level(s)	Command	Description
Administrator, Operator	IT	Modify command interface inactivity time-out.
Administrator	LC	Configure LAN connections.
Administrator, Operator, Single Partition User	LS	Display LAN connected console status.
Administrator	ND	Enable/disable network diagnostics.
Administrator, Operator	PD	Set the default nPartition for the current session.
Administrator, Operator	PWRGRD	Configure power grid settings. (HP rp8400 and HP rp7405/rp7410 servers only.)
Administrator, Operator	RL	Rekey complex profile lock.
Administrator	SO	Configure security options and access control.
Administrator, Operator	XD	Service processor diagnostics and reset options.

Network Configuration for a Service Processor

This section describes how to list and configure the network settings for service processor (GSP or MP) hardware. These settings are used for connections to the service processor and are not used for HP-UX networking.

Details on configuring service processor networking are given in the procedure *Configuring Service Processor Network Settings* on page 148.

The service processor utility hardware on HP Superdome servers has two network connections: the customer LAN and private LAN. The service processor on HP rp8400 and HP rp7405/rp7410 servers do not have a private LAN but have only customer LAN connections.

Features of service processor LANs are given in the following list.

- **Customer LAN for Service Processor**

  The **customer LAN** is the connection for login access to the service processor menus, consoles, commands, and other features.

  All HP nPartition servers have a customer LAN.

  On HP Superdome servers, the customer LAN port is labeled “Customer LAN”. On HP rp8400 servers it is “GSP LAN”. On HP rp7405/rp7410 servers it is the only LAN connection on each core I/O board.

- **Private LAN for Service Processor (Superdome Only)**

  The **private LAN** is the connection to the Superdome service support processor (SSP) workstation.

  Only Superdome servers have a private LAN.

To *configure* service processor network settings, you can use the the Command menu’s `LC` command.

To *list* the current service processor network configuration use the `LS` command.

The following examples show service processor LAN status for various HP nPartition servers.

HP rp7405/rp7410 or rp8400 Service Processor LAN Status

```
MP:CM> LS

Current configuration of MP customer LAN interface
  MAC address   : 00:30:6e:05:19:ac
  IP address    : 15.99.84.140    (0x0f63548c)
  Hostname      : redxii-c
  Subnet mask   : 255.255.255.0   (0xffffff00)
  Gateway       : 15.99.84.254    (0x0f6354fe)
  Status        : UP and RUNNING
  AutoNegotiate : Enabled
  Data Rate     : 100 Mb/s
  Duplex        : Half
  Error Count   : 0
  Last Error    : none

MP:CM>
```

HP Superdome Service Processor LAN Status

```
GSP:CM> LS

Current configuration of GSP customer LAN interface
  MAC address : 00:10:83:27:04:5a
  IP address  : 15.99.49.129    0x0f633181
  Name        : feshd5-u
  Subnet mask : 255.255.248.0   0xfffff800
  Gateway     : 15.99.49.254    0x0f6331fe
  Status      : UP and RUNNING


Current configuration of GSP private LAN interface
  MAC address : 00:a0:f0:00:83:b1
  IP address  : 192.168.2.15    0xc0a8020f
  Name        : priv-05
  Subnet mask : 255.255.255.0   0xffffff00
  Gateway     : 192.168.2.100   0xc0a80264
  Status      : UP and RUNNING

GSP:CM>
```

Default Service Processor Network Settings

Table 3-5 and Table 3-6 list the default customer LAN and private LAN network settings for nPartition servers. Only Superdome servers have a private LAN.

Table 3-5 Default Configuration for Service Processor Customer LAN (All nPartition Servers)

Customer LAN IP Address	192.168.1.1
Customer LAN Host Name	gsp0
Customer LAN Subnet Mask	255.255.255.0
Customer LAN Gateway	192.168.1.1

Table 3-6 Default Configuration for Service Processor Private LAN (HP Superdome Servers Only)

Private LAN IP Address	192.168.2.10
Private LAN Host Name	priv-00
Private LAN Subnet Mask	255.255.255.0
Private LAN Gateway	192.168.2.10

Configuring Service Processor Network Settings

This procedure (Command menu, **`LC`** command) configures the service processor's customer LAN and private LAN network settings from the service processor Command menu.

Step 1. Connect to the server complex's service processor, login as an administrator, and enter **`CM`** to access the Command menu.

Use `telnet` to connect to the service processor, if possible.

If a service processor is at its default configuration (including default network settings), you can connect to it using either of these methods:

- Establish a direct serial cable connection through the service processor's local RS-232 port, a 9-pin D-shaped connector (DB9).

  On HP Superdome servers this port is labeled "Local RS-232". On HP rp8400 servers it is the "Local Console" port. On HP rp7405/rp7410 servers use the DB9 connector that is labeled "Console".

- Access a PC or workstation on the same subnet as the service processor, modify its network routing tables to include the default customer LAN IP address, then `telnet` to the service processor. The procedure to modify networking and connect is:

  1. Access a PC or workstation on the service processor's subnet.

  2. Modify the network routing tables for the PC or workstation by using the **`route add 192.168.1.1`** ***`ClientName`*** command, where *`ClientName`* is the network name of the PC or workstation.

From a PC command prompt: **`route add 192.168.1.1 ClientName`**
On an HP-UX workstation login as `root` and use this command:

`/usr/sbin/route add 192.168.1.1 ClientName`

After you reconfigure the service processor's networking, you can remove these network routing table changes with the `route delete...` command.

3. Enter this command to confirm the new network connection to the service processor: **`ping 198.168.1.1 -n 2`**

4. Use the **`telnet 192.168.1.1`** command from the PC or workstation to connect to the service processor.

Step 2. From the service processor Command menu, enter **`LS`** to *list* the current network settings, and if needed use the **`LC`** command to *reconfigure* the network settings for the service processor.

You must be logged in as an administrator to use the `LC` command.

The `LC` command enables you to modify the customer LAN and/or the private LAN configuration.

You can cancel all changes to the service processor LAN configuration at any time by replying **`Q`** to any of the `LC` command's prompts.

Console Access to nPartitions

The service processor Console menu provides access to all nPartition consoles within the server complex.

Enter **CO** from the service processor Main menu to access an nPartition's console. To exit the nPartition console, type **^b** (**Control-b**) to return to the Main menu.

Each nPartition in a complex has a single console. However, multiple connections to the console are supported, allowing multiple users to simultaneously view the console output. Only one connection per console permits write-access.

To force (gain) console write access for an nPartition's console, type **^ecf** (**Control-e c f**).

Each nPartition console can display a variety of information about the nPartition, including:

- Partition startup, shutdown, and reset output.
- Boot Console Handler (BCH) menus, if the nPartition has not yet booted the HP-UX operating system and has completed Power-On Self Tests (POST).
- The HP-UX login prompt and "console shell access".

nPartition Console Access versus Direct HP-UX Login

You may need to consider the following factors when deciding whether to interact with an nPartition through the service processor console interface or a direct HP-UX login:

- Whether you want to log your activity to the nPartition's console log (all console activity is stored at least temporarily).
- Whether HP-UX is installed, booted, and properly configured on the nPartition.

  If HP-UX is not installed on an nPartition, you should access the nPartition's console (through the service processor) in order to install and configure HP-UX.

You should login to HP-UX running on an nPartition when you do not need to use service processor features and do not want to record a log of your activity.

Before HP-UX has booted, the service processor nPartition consoles are the primary method of interacting with an nPartition.

After an nPartition has booted HP-UX, you should be able to connect to and login to the nPartition by using `telnet` or `rlogin` to remotely login.

If the HP-UX kernel booted on the nPartition does not have networking fully configured, you may need to login using a service processor nPartition console connection to set up the nPartition's networking configuration (using `/sbin/set_parms`).

To view the /dev/console messages for HP-UX running on an nPartition, you can access the nPartition's console, view its console log, or use the `xconsole` command or `xterm -C` command and option. See the *xconsole* (1) or *xterm* (1) manpages for details.

Boot Console Handler (BCH) Access

Each nPartition in a server complex has its own Boot Console Handler (BCH) interface. When an nPartition is booted to BCH, its BCH interface is available through the nPartition's console.

The nPartition BCH interface enables you to manage and configure the HP-UX boot process for an nPartition. You also can configure some settings for the local nPartition, get some information about the nPartition and its server complex, and perform other tasks such as reboot.

Figure 3-3 shows details on accessing and using an nPartition's BCH interface, including the following points:

- To access an nPartition's console type **CO** from the service processor (GSP or MP) Main menu.
- To force console write access, type **^ecf** (**Control-e c f**).
- To exit the console, type **^b** (**Control-b**) to return to the Main menu.

The BCH interface is available after an nPartition's cells have been powered on; its hardware has completed all Power-On Self Tests (POST); and the cells have booted past boot-is-blocked, rendezvoused, and BCH has started executing. Refer to the chapter *An Overview of nPartition Boot and Reset* on page 161 for details.

Once you begin the HP-UX boot process and load ISL, the BCH interface is no longer available.

The BCH menus and commands for nPartitions differ slightly from the commands menus for BCH on other HP 9000 server systems.

To display the *current* BCH menu and commands, type **DI**.

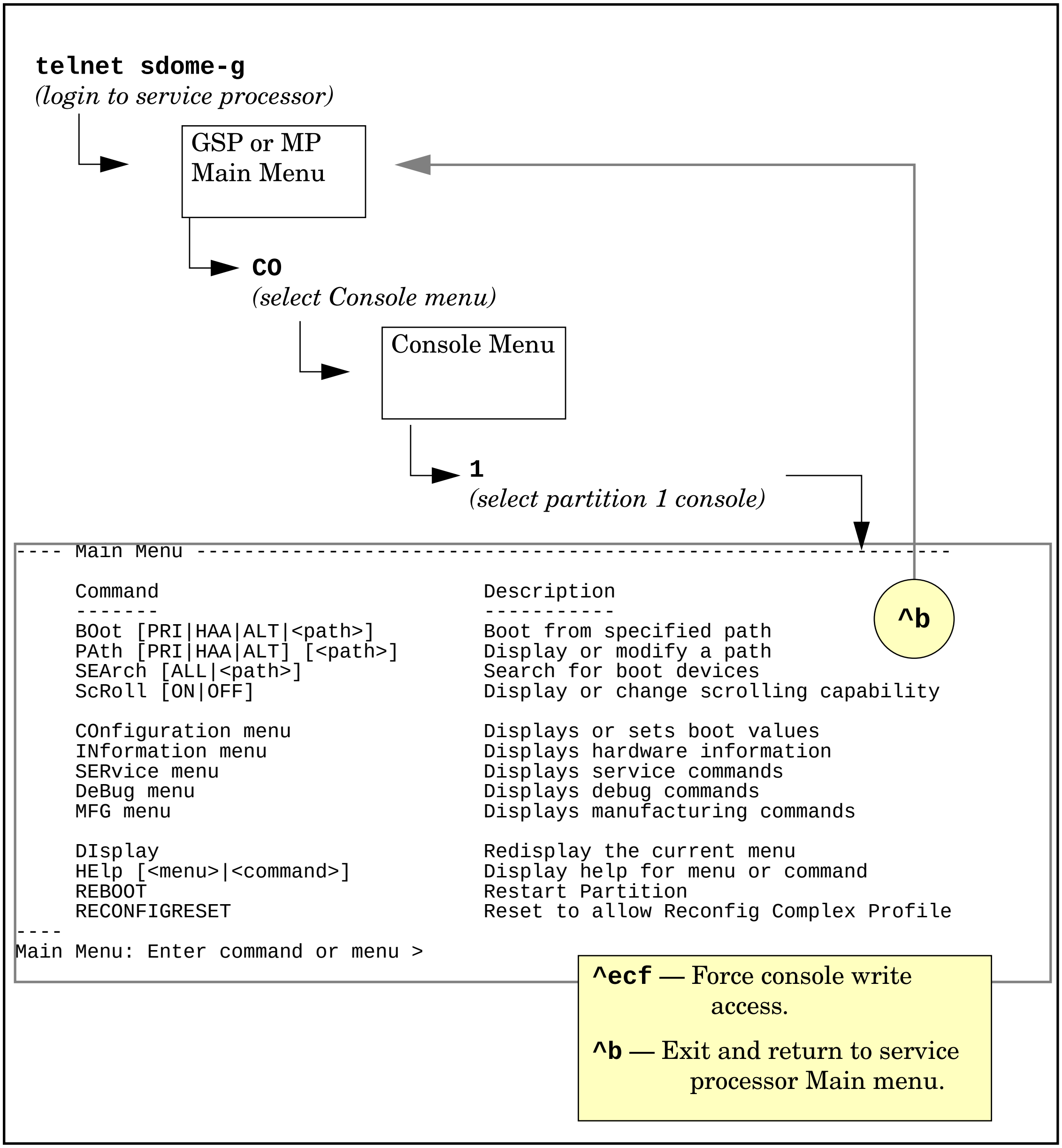
The BCH interface's **HELP** command lists BCH command or menu details.

```
Main Menu: Enter command or menu > HELP MA
---- Main Menu Help -------------------------------------------------------------

     The following submenus are available from the main menu:

     COnfiguration------------------------------------BootID
     INformation-----------------------ALL            BootTimer
     SERvice-------------BAttery       BootINfo       CEllConfig
                         CLEARPIM      CAche          COreCell
                         MemRead       ChipRevisions  CPUConfig
                         PDT           ComplexID      DataPrefetch
                         PIM           FabricInfo     DEfault
                         SCSI          FRU            FastBoot
                                       FwrVersion     KGMemory
                                       IO             PathFlag
                                       LanAddress     PD
                                       MEmory         ResTart
                                       PRocessor      TIme
...
```

Figure 3-3 Accessing an nPartition's BCH Interface



Console Log Viewing

Each nPartition in a server complex has its own console log that stores a record of the nPartition's most recent console activity.

To access an nPartition's console log, enter **CL** from the service processor Main menu and select which nPartition's console log you want to view. To exit the console log viewer, type **^b** (**Control-b**) to return to the Main menu.

When viewing an nPartition's console log, type **P** to view the previous page of the console log, or type **N** (or **Enter**) to view the next page.

When you enter an nPartition's console log viewer it displays the oldest data in the log first and allows you to page through the log to view the more recently recorded activity.

Each nPartition's console log is a circular log file that records approximately 30 to 40 pages of data. All nPartition console activity is written to this log file, regardless of whether a user is connected to the nPartition console.

As an nPartition's console log is written the oldest data in the log is overwritten by current data, as needed, so that the last 30 to 40 pages of console output always is available from the console log viewer.

Chassis Code Log Viewing

The service processor's **chassis log viewer** enables you to view chassis codes that are emitted throughout the entire server complex.

To enter the chassis log viewer enter **SL** at the service processor Main menu. To exit the viewer type **^b** (**Control-b**) to return to the Main menu.

Chassis codes are data that communicate information about system events from the source of the event to other parts of the server complex. Chassis code data indicates what event has occurred, when and where it happened, and its severity (the *alert level*).

All chassis codes pass from the event source through the service processor. The service processor takes any appropriate action and then reflects the chassis codes to all running nPartitions. If an nPartition is running event monitoring software, it may also take action based on the chassis codes (for example, sending notification e-mail).

System administrators, of course, may have interest in viewing various chassis codes—especially chassis codes that indicate failures or errors.

Hardware, software, and firmware events may emit chassis codes as a result of a failure or error, a major change in system state, or basic forward progress. For example: a fan failure, an HPMC, the start of a boot process, hardware power on or off, and test completion all result in chassis codes being emitted.

While HP-UX is running on an nPartition, it constantly emits a "heartbeat" chassis code (at alert level 0) to indicate that the operating system still is functioning and has not hung.

NOTE

Each nPartition server cabinet's **front panel attention LED** is automatically *turned on* when one or more chassis codes of *alert level 2* or higher have not yet been viewed by the administrator. When this attention LED is on, entering the chassis log viewer turns the LED off.

You can remotely check this attention LED's on/off status by using the service processor Command menu's **PS** command, **G** option.

On nPartition servers, chassis codes are recorded in the server complex **activity log** (for events of alert level 0 or alert level 1) or the **error log** (for events alert level 2 or higher).

```
GSP> SL

Chassis Logs available:

    (A)ctivity Log
    (E)rror Log
    (L)ive Chassis Logs

    (C)lear All Chassis Logs
    (Q)uit

GSP:VW> L

        Entering Live Log display

        A)lert filter
        C)ell filter
        P)artition filter
        U)nfiltered
        V)iew format selection
        ^B to Quit

Current filter: ALERTS only
```

Log Viewing Options: Activity, Error, and Live Chassis Logs

When you enter the chassis log viewer by entering **SL** at the service processor (GSP or MP) Main menu, you can select from these viewers:

- **Activity Log Viewer**

  Allows you to browse recorded chassis codes of alert level 0 or 1.

- **Error Log Viewer**

  Allows you to browse recorded chassis codes of alert level 2 or higher.

- **Live Chassis Logs Viewer**

  Displays chassis codes in real time as they are emitted.

  By default, the live chassis code viewer has the *Alert filter* enabled, which causes it to display only the events of alert level 3 or higher.

  To view all chassis codes in real-time, type **U** for the *Unfiltered* option.

You also can filter the live codes by cell (**C**) or nPartition (**P**). *Cell filter*: only display chassis codes emitted by a specific cell in the server complex. *Partition filter*: only display chassis codes emitted by hardware assigned to a specific nPartition.

When viewing chassis code logs, type **V** to change the display format. The viewers can show chassis codes in text format (**T**), keyword format (**K**), or raw hex format (**R**).

Using Virtual Front Panels

The Virtual Front Panel (VFP) provides ways to monitor the chassis codes for a particular nPartition or the entire server complex (all nPartitions).

The VFP presents a real-time display of activity on the selected nPartition(s) and it automatically updates when cell and nPartition status change.

To access the VFP feature, enter **VFP** from the service processor Main menu. To exit the VFP, type **^b** (**Control-b**) to return to the Main menu.

When you access a Virtual Front Panel, you can either select the nPartition whose VFP you want to view or select the system VFP to view summary information for all nPartitions in the server complex.

```
E indicates error since last boot
     Partition 0  state           Activity
     ------------------           --------
     Cell(s) Booting:    710 Logs

  #  Cell state                   Activity
  -  ----------                   --------
  0  Early CPU selftest           Cell firmware test                 232  Logs
  1  Early CPU selftest           Processor test                     230  Logs
  2  Memory discovery             Physical memory test               242  Logs









GSP:VFP (^B to Quit) >
```

When you access a service processor using a single-partition user account, using the VFP feature enables you to view only the VFP for the nPartition to which you have access.

4 An Overview of nPartition Boot and Reset

This chapter presents an overview of booting and reset concepts and issues for HP nPartition servers.

For procedures to boot, reboot, and configure boot options, refer to the chapter *Booting and Resetting nPartitions* on page 197.

NOTE For details on booting and rebooting virtual partitions within an nPartition, refer to the chapter *Virtual Partitions (vPars) Management on nPartitions* on page 441.

Types of Booting and Resetting for nPartitions

All standard boot and reboot methods are supported for HP nPartition servers, though some boot and reset procedures differ slightly or use different tools than on other HP servers.

HP's nPartition servers also provide two special types of reboot and reset for managing nPartitions: performing a **reboot for reconfig**, and resetting an nPartition to the **ready for reconfig** state.

The following list summarizes all types of booting, rebooting, and resetting that are supported for HP nPartition systems. See the *Reboot for Reconfig* and *Ready for Reconfig State* items for a discussion of these nPartition-specific boot processes.

NOTE When rebooting HP-UX on an nPartition under normal circumstances—such as when *not* reconfiguring or halting it—use the `shutdown -r` command.

- **Reboot**

  A **reboot** shuts down HP-UX and reboots the nPartition.

  Only the nPartition's *active* cells are rebooted.

  To perform a standard reboot of an nPartition use the `shutdown -r` command.

- **Halt**

  A **halt** shuts down HP-UX, halts all processing on the nPartition, and does not reboot.

  To perform this task use the `shutdown -h` command.

  To reboot a halted nPartition use the service processor Command menu's RS command.

- **Reset**

  A **reset** resets the nPartition immediately. Only the nPartition's *active* cells are reset.

  You can reset an nPartition using the BCH interface's `REBOOT` command or the service processor Command menu's `RS` command.

  The `RS` command *does not check* whether the specified nPartition is in use or running HP-UX—be certain to correctly specify the nPartition.

- **Boot an nPartition from the Service Processor (GSP or MP)**

  A **boot** initiated from the service processor boots an inactive nPartition past the *ready for reconfig* state.

  The nPartition's cells proceed past boot-is-blocked (BIB), rendezvous, and the nPartition boots to the BCH interface.

  To boot an inactive nPartition, use the service processor Command menu's `BO` command.

- **Boot HP-UX from the BCH Interface**

  To **boot** HP-UX on an nPartition, use the BCH interface's `BOOT` command and specify the device path from which the program loaders and HP-UX kernel .

  The BCH interface's `BOOT` command loads and boots HP-UX on an nPartition. This command also can be used to load and interact with the Initial System Loader (ISL) interface. Likewise on Superdome servers the virtual partitions monitor (`MON>` prompt) is loaded following the `BOOT` command.

- **Reboot for Reconfig**

  A **reboot for reconfig** shuts down HP-UX, resets all cells assigned to the nPartition, performs any nPartition reconfigurations, and boots the nPartition back to the BCH interface.

  To perform a reboot for reconfig of the local nPartition, use the `shutdown -R` command.

  All cells—including any inactive cells and all newly added or deleted cells—reboot and are reconfigured. All cells with a "y" *use-on-next-boot* setting participate in partition rendezvous and synchronize to boot as a single nPartition.

After you assign a cell to an nPartition, or remove an active cell from an nPartition, you can perform a reboot for reconfig of the nPartition to complete the cell addition or removal.

If an nPartition is configured to boot HP-UX automatically, it can do so immediately following a reboot for reconfig.

- **Ready for Reconfig State**

  A reboot to the **ready for reconfig** state shuts down HP-UX, resets all cells assigned to the nPartition, performs any nPartition reconfigurations, and keeps all cells at a boot-is-blocked (BIB) state, thus making the nPartition and all of its cells inactive.

  When an nPartition is at the ready for reconfig state you can add or remove cells from the nPartition from a remote nPartition within the server complex.

  To put an nPartition into the ready for reconfig state use the `shutdown -R -H` command, the BCH interface's `RECONFIGRESET` command, or the service processor Command menu's `RR` command.

  To make an nPartition boot past ready for reconfig, use the service processor Command menu's `BO` command. The `BO` command makes the nPartition active by allowing its cells to boot past BIB, rendezvous, and boot to the BCH interface (and, if configured, automatically boot HP-UX).

- **TOC: Transfer-of-Control Reset**

  When you initiate a **transfer-of-control reset**, the service processor immediately performs a TOC reset of the specified nPartition, which resets the nPartition and allows a crash dump to be saved.

  If crash dump is configured for HP-UX on an nPartition, then when you TOC the nPartition while it is running HP-UX, the nPartition performs a crash dump and lets you select the type of dump.

  To perform a TOC reset, use the service processor Command menu's `TC` command.

  HP nPartition systems do not have TOC buttons on the server cabinet hardware.

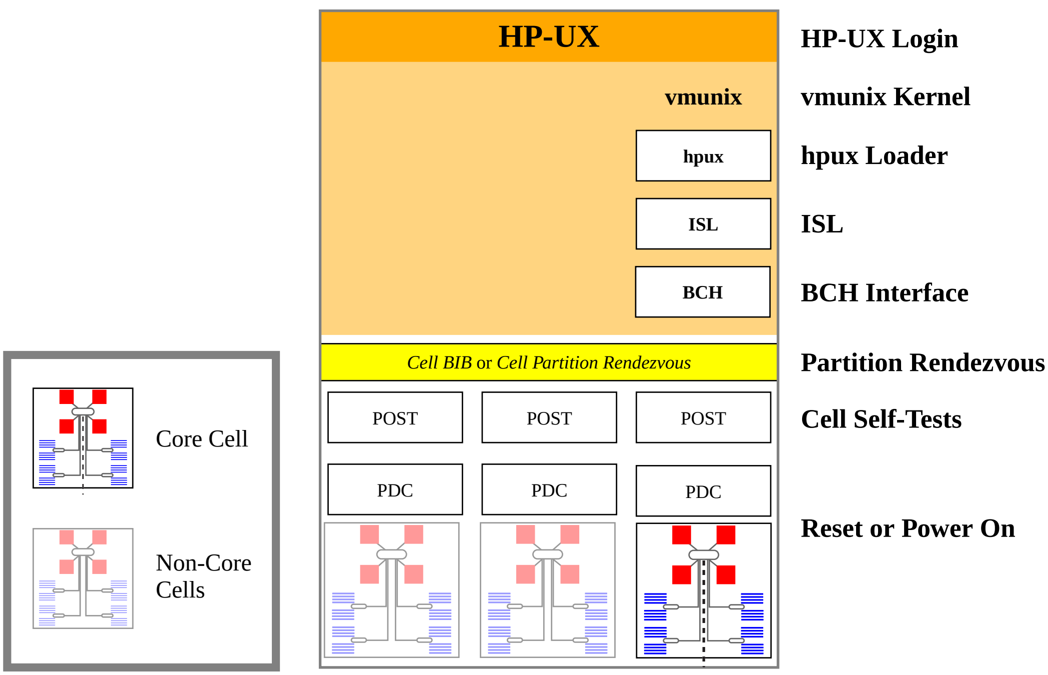
Boot Process for nPartitions, Cells, and HP-UX

The boot process for nPartitions is similar to the process on other HP servers. However, on HP nPartition servers, each cell boots and performs self tests (POST) separately, and one or more cells rendezvous to form an nPartition before providing a BCH interface for the nPartition.

NOTE This section covers nPartitions booting HP-UX in *non-vPars mode*.

For details on virtual partitions (vPars), refer to the chapter *Virtual Partitions (vPars) Management on nPartitions* on page 441.

Figure 4-1 nPartition HP-UX Boot Process (non-vPars Mode)



Each nPartition goes through the boot process shown in Figure 4-1, from power on to booting HP-UX:

1. **Power On or Reset**

   The boot process starts when any of the following events occurs:

   - An nPartition is reset or rebooted.
   - The entire server complex is powered on.
   - Power is turned on for components in the nPartition (such as cells).

2. **Processor Dependent Code (PDC)**

   The monarch processor on each cell runs its own copy of the PDC firmware.

   a. The boot-is-blocked (BIB) flag is set for the cell.

      The BIB flag remains set until the service processor (GSP or MP) clears it, allowing the cell to boot as part of an nPartition.

   b. Another flag is set for the cell, indicating that the service processor can post a new copy of the complex profile to the cell.

      The cell's complex profile is updated later in the boot process, after it completes self-tests.

3. **Power-On Self-Test (POST)**

   Each cell performs self-tests that check the processors, memory, and firmware on the cell.

   If a component fails self-tests, it is deconfigured and if possible the cell continues booting.

   Following this step, all components in the cell are known and are tested and the cell reports its hardware configuration to the service processor.

4. **I/O Discovery**

   Each cell performs I/O discovery and configures I/O busses, including: any system bus adapter (the SBA for an I/O card cage) and its local bus adapters (LBAs, one per PCI card slot in the card cage).

Following this, any I/O busses connected to the cell are known and configured by the cell.

5. **Boot-Is-Blocked (BIB) or Partition Rendezvous**

   Each cell either will remain at a boot-is-blocked state (spins at BIB) or will rendezvous with any other available cells in the nPartition.

   Cells that remain at BIB are *inactive*, and cells that rendezvous into the nPartition are *active*.

   - **Boot-Is-Blocked (BIB)**

     A cell remains at boot-is-blocked (and thus is *inactive*) in any of the following cases:

     — The cell has a “n” use-on-next-boot setting.

     — The cell boots too late to participate in nPartition rendezvous.

     — The cell’s nPartition has been reset to the ready for reconfig state.

     In this case, all of the nPartition’s cells remain at boot-is-blocked.

     — The cell fails self-tests that cause the cell to not be usable in the nPartition.

   - **Partition Rendezvous**

     Partition rendezvous of all cells occurs in the following manner:

     — Partition rendezvous begins when the first of the nPartition’s cells has completed self-tests and I/O discovery.

     — The nPartition is allowed up to ten minutes for all cells with a “y” use-on-next-boot setting to participate in partition rendezvous.

     — Once all assigned cells with a “y” use-on-next-boot setting have entered the rendezvous stage, partition rendezvous can complete.

     All cells participating in rendezvous are active cells whose resources (processors, memory, I/O) are used by the nPartition.

— If any cells with a "y" use-on-next-boot setting *do not* report to rendezvous, then ten minutes after rendezvous began the cells that have not reported become inactive cells, and all other reporting cells complete rendezvous and are active.

The inactive cells' resources are not available to be used by the nPartition, although the cells still are assigned to the nPartition.

6. **Boot Console Handler (BCH)**

   The BCH interface provides the main method for interacting with an nPartition during its boot process.

   BCH runs on top of PDC, and it provides menus for getting nPartition status, for configuring nPartition boot settings, and for booting HP-UX and rebooting the nPartition.

   One processor on the nPartition's core cell runs BCH and all other processors in the nPartition are idle while the BCH interface is available.

   An nPartition can immediately proceed past BCH to boot HP-UX when the nPartition's boot paths are set and boot actions for the paths are configured to automatically boot.

7. **Initial System Loader (ISL) and Secondary System Loader (hpux)**

   In most situations you do not need to use the ISL and hpux interfaces.

   However, when using the BCH interface's `BOOT` command you can select to stop at the ISL prompt to perform more detailed booting tasks.

   For example, you can use the ISL interface to boot HP-UX in single-user or LVM-maintenance mode, or to boot an HP-UX kernel other than /stand/vmunix.

8. **HP-UX Operating System**

   The HP-UX operating system boots on an nPartition after ISL and the Secondary System Loader (hpux) specify which kernel is to be booted.

   By default, on HP-UX boot disks, the AUTO file specifies that the /stand/vmunix kernel is booted.

For example, when you configure boot paths and boot actions to automatically boot HP-UX, the ISL and hpux loaders specify that the /stand/vmunix kernel is booted.

Overview of nPartition Boot Features

This section lists several boot issues particular to HP nPartition servers.

Each nPartition is booted, rebooted, shut down, and reset individually. In many situations you can boot and reboot nPartitions using the same basic procedures that are used on other HP servers.

The following list describes notable features related to booting, rebooting, and power cycling nPartitions:

- Each nPartition can boot and reboot independently of other nPartitions. Resetting one nPartition has no effect on the others.
- You can perform many reset and power cycling tasks remotely.

  You can reset and control power from an nPartition server's service processor Command menu, from the BCH interface for an nPartition, or from HP-UX running on an nPartition.
- In order to contribute resources to an nPartition, the cells (and I/O chassis) assigned to the nPartition must be powered on and booted in time to participate in partition rendezvous.

  Otherwise, the cells will remain inactive (though still assigned to the nPartition) and their processors, memory, and any I/O will not available for use.
- Three boot path variables—PRI, HAA, and ALT—are supported for each nPartition.

  PRI typically is the primary HP-UX boot device, HAA typically is a mirror of the root volume, and ALT is for install or recovery media such as tape or DVD-ROM devices.
- You can specify a *boot action* for each boot path variable. The boot action determines what action (for example: boot HP-UX) is taken when the nPartition boots and reaches the BCH interface.

  To set boot actions, use the BCH Configuration menu's PathFlags (`PF`) command. The `setboot` command can configure the PRI actions only.

  When an nPartition boots to BCH, it attempts to perform the PRI path's boot action. The HAA path and ALT path boot actions also can be attempted, in that order, depending on the PRI settings.

- *Before powering off* a cell, the cell should be inactive; unassigned; or assigned to an nPartition that either has been shut down and halted or has been reset to the ready for reconfig state.

  Powering on or powering off an I/O chassis *resets the cell* to which it is connected (if any). Follow the same guidelines for power cycling I/O chassis that you follow for power cycling cells.

Tools for Managing nPartition Booting

HP nPartition servers support the following software tools for booting and resetting nPartitions and for configuring and managing nPartition boot settings.

These tools overlap in some of the functionality they provide, but each has unique capabilities.

The primary tools for managing nPartition booting are shown below.

- **Service Processor (GSP or MP)**—Using a server complex's service processor menus, you can reset partitions, put partitions into the ready for reconfig state, and TOC the partitions in the complex.

  The service processor also provides power on and power off commands for power cycling server hardware components.

- **Virtual Front Panel (VFP)**—Each nPartition has its own VFP that displays current cell and partition boot states and activities.

  For each server complex you also can access a system VFP that gives a live partition boot state and activity status for all nPartitions in the complex.

- **Boot Console Handler (BCH)**—Each partition's BCH interface provides commands for booting HP-UX, rebooting the partition, and putting the partition into the ready for reconfig state.

  You also can configure boot-related settings and check the partition's hardware and boot-setting configurations using BCH menus.

- **HP-UX System Loaders (ISL and hpux)**—You can use system loaders to list files that reside on a boot device, such as kernel files in /stand, and can specify boot arguments to the hpux loader.

  You can access the ISL and hpux loaders after issuing the BCH interface's `BOOT` command, when BCH gives you the following option:

  ```
  Do you wish to stop at the ISL prompt prior to booting?
  (y/n)
  ```

  Replying "n" (no, do not stop at ISL) skips the ISL prompt and proceeds to execute the AUTO file, which by default will boot HP-UX (/stand/vmunix) on the nPartition.

ISL: Initial System Loader

Replying “y” (yes, do stop at ISL) allows you to interact directly with a boot device’s Initial System Loader (ISL) and the Secondary System Loader (hpux). Enter all ISL commands from the `ISL>` prompt.

hpux: Secondary System Loader

From the ISL prompt you also can enter commands that are executed by the Secondary System Loader (hpux). Preface your Secondary System Loader command with **hpux**. For example: `hpux ls /stand` to list the contents of the /stand directory on the booted device.

See the *isl* (1M) and *hpux* (1M) manpages for details.

- **HP-UX utilities**—Several HP-UX utilities allow you to check and set a partition’s HP-UX boot options; check the boot settings of other partitions in the server complex; and perform reboot, shutdown, and reboot for reconfig tasks.

  The `reboot`, `shutdown`, `parmodify`, `parstatus`, and `setboot` commands provide these features. For details see the command manpages.

  The Partition Manager utility (`/opt/parmgr/bin/parmgr`) also provides some boot configuration capabilities; details are available in its online help.

Configurable Boot Settings

Each nPartition has its own collection of boot-related settings that specify which hardware manages the boot process (the core cell), how the boot process proceeds (automatically boot HP-UX, or wait for BCH commands), and whether cells are configured as active cells when the nPartition boots.

nPartition boot settings are stored as part of the server Complex Profile data.

You can configure each nPartition's boot settings by using the nPartition's BCH interface or by running HP-UX utilities on the nPartition.

By using the `parmodify` HP-UX command or Partition Manager, you also can configure some boot settings for remote (non-local) nPartitions in the same server complex.

You can reconfigure boot settings at any time to change the nPartition's boot behavior, specify different boot devices, or adjust settings based on nPartition configuration changes. Some boot setting changes require rebooting to take effect.

Also see *Checklist and Guidelines for Booting nPartitions* on page 184 for details on ensuring a bootable nPartition configuration.

You can configure the following boot settings for each nPartition: boot device paths, boot actions, core cell choices, cell use-on-next-boot value.

- **Boot Device Paths**

  You can set boot device paths to reference the hardware paths where bootable devices reside within the local nPartition.

  The boot device paths include the primary boot device (PRI boot path), the high-availability alternate device (HAA boot path, such as a mirror of the root volume), and the alternate device (ALT boot path, such as an install or recovery device).

  The PRI path is the default device booted by the BCH interface's `BOOT` command.

  You can set boot paths using the BCH interface, the `parmodify` command, and Partition Manager. The `setboot` command can set the PRI and ALT paths only.

- **Boot Actions**

  Each boot device path has an associated boot action, which is established by the path's "path flag" setting. The boot actions (path flag settings) are referenced automatically when an nPartition initially boots to the BCH interface.

  Boot actions have no effect on boot behavior when you manually boot HP-UX using the BCH interface's `BOOT` command.

  The boot action for the PRI boot path establishes what the nPartition does when it boots and first reaches the BCH interface: boot the PRI device, go to the BCH Main menu, or skip the PRI path and attempt to perform the HAA path's boot action. You also can specify what action to take if an attempt to boot a device fails (either go to BCH, or try the next path).

  Depending on the PRI path flag setting, the HAA boot action may be referenced. Likewise, the HAA setting determines whether the ALT boot actions may be referenced.

  You can set boot actions using the BCH Configuration menu's PathFlags (`PF`) command. The `setboot` command can configure only the PRI actions from HP-UX.

  For details use the BCH Command menu's `HELP PF` command.

  You can stop an nPartition from automatically booting, and instead access the nPartition's BCH interface, by typing a key within ten seconds of the nPartition booting to BCH.

```
     Primary Boot Path:  0/0/1/0/0.8
          Boot Actions:  Boot from this path.
                         If unsuccessful, go to BCH.

...

Attempting to boot using the primary path.
-------------------------------------------------------------

 To discontinue, press any key within 10 seconds.
```

- **Core Cell Choices**

  The core cell is the cell that "runs" the nPartition before it boots HP-UX. A processor on the core cell serves as the monarch processor that runs Boot Console Handler (BCH). The core cell is the one whose core I/O is active for the nPartition.

One cell is selected as the **active core cell** for the nPartition when the nPartition boots. By default, the lowest numbered eligible cell in the nPartition is chosen. To be eligible the cell must: be active, have a connection to functioning core I/O, and be assigned to the nPartition.

You can designate up to four **core cell choices**, which are considered in the order you specify as candidates to be selected as the active core cell for the nPartition.

If none of the core cell choices is eligible to serve as the core cell, then the nPartition attempts to select a core cell using the default algorithm (lowest numbered eligible cell).

When no active cell in an nPartition can be selected, the nPartition will remain at the ready for reconfig state and cannot boot to BCH.

- **Use-on-Next-Boot Value for a Cell**

  Each cell in an nPartition has an associated *use-on-next-boot* value that determines whether the cell's resources are used by the nPartition.

  This setting does not affect the cell's nPartition assignment. The use-on-next-boot value only determines whether the cell is an *active* or *inactive* member of the nPartition when the nPartition boots.

  When a cell's use-on-next-boot value is "y" (use the cell), the cell can participate in nPartition rendezvous and become an *active* member of the nPartition, which enables its processors, memory, and any connected I/O to be made available for use by the nPartition.

  When a cell has a use-on-next-boot value of "n" (do not use the cell), the cell cannot participate in partition rendezvous so it will be an *inactive* member of the nPartition when the nPartition boots: all processors, memory, and I/O will not be made available.

  After changing a cell's use-on-next-boot value you might need to reset the nPartition so that all cells have a chance to either participate in partition rendezvous or remain inactive at BIB. (For example, if the nPartition is in the ready for reconfig state, just boot it using the service processor Command menu's `BO` command; but if the nPartition is active and has booted HP-UX then perform a reboot for reconfig using the `shutdown -R` command.)

  The use-on-next-boot setting does not directly affect the nPartition's boot behavior, but it will cause I/O connected to a cell to be unavailable when the cell boots with a "n" use-on-next-boot value.

For details on configure these boot-related settings for an nPartition refer to the chapter *Booting and Resetting nPartitions* on page 197.

Listing nPartition Boot Settings

You can list an nPartition's boot-related settings by using the nPartition's BCH interface or by using HP-UX commands.

To *list boot settings* for nPartitions, use the following procedures:

- *Listing nPartition Boot Settings [BCH]* on page 178
- *Listing nPartition Boot Settings [HP-UX]* on page 181
- *Listing nPartition Boot Settings [Partition Manager]* on page 183

Also see the following sections for details on *configuring boot settings* for nPartitions.

- *Configuring Boot Paths and Boot Actions* on page 227
- *Configuring Autoboot and Autostart* on page 233
- *Configuring Automatic System Restart for an nPartition* on page 235
- *Configuring Fast Boot Settings (Self Tests) for an nPartition* on page 238
- *Boot Timer Configuration for an nPartition* on page 242

Listing nPartition Boot Settings [BCH]

Use BCH commands from the **Main menu**, **Information menu**, and **Configuration menu** to list an nPartition's boot settings.

Step 1. Login to the server complex's service processor (GSP or MP), access the nPartition's console, and access the BCH Main menu.

From the nPartition console you access the nPartition's BCH interface. If the nPartition is not at the BCH interface, you must either boot the nPartition or shut down HP-UX to return to the BCH interface.

Step 2. Access the BCH menu that provides the information you want to list.

The BCH interface's **Main menu**, **Information menu**, and **Configuration menu** provide commands for listing (and setting) boot options for the nPartition.

- **BCH Main menu**

  If you are at one of the other BCH menus, enter **MA** to return to the BCH interface's Main menu.

Table 4-1 BCH Main Menu Boot Settings

PATH	Displays or sets the boot paths: primary (PRI), high-availability (HAA), and alternate (ALT).

- **BCH Configuration menu**

  From the BCH main menu, enter **CO** to access the Configuration menu.

Table 4-2 BCH Configuration Menu Boot Settings

AU	*Supported on HP Superdome servers only.* Displays or sets the auto-start flag, which determines whether the boot process proceeds following a self-test failure.
BOOTTIMER	Displays or sets the time allowed for booting.
CELLCONFIG	Displays or sets the (de)configuration of cells.
CORECELL	Displays or sets the core cell choices.
FASTBOOT	Displays or specifies whether certain self-tests are run during the boot process.
PATHFLAGS	Displays or sets the boot action for each boot path.

- **BCH Information menu**

  From the BCH Main menu, enter **IN** to access the Information menu.

Table 4-3 BCH Information Menu Boot Settings

BOOTINFO	Displays boot configuration information.

Step 3. At the appropriate BCH menu, issue the command to display the boot information of interest to you.

See the list in the previous step for commands and menus. Enter the command *with no arguments* to display (but not change) the boot setting.

The following example shows using the PATH command to list the nPartition's boot paths, then accessing the BCH Configuration menu and issuing the PATHFLAGS command to list the nPartition's boot action settings for the PRI, HAA, and ALT boot paths.

```
Main Menu: Enter command or menu > PATH

     Primary Boot Path:  4/0/2/0/0.10
                         4/0/2/0/0.a    (hex)

HA Alternate Boot Path:  4/0/1/0/0.6
                         4/0/1/0/0.6    (hex)

   Alternate Boot Path:  4/0/1/0/0.5
                         4/0/1/0/0.5    (hex)

Main Menu: Enter command or menu > CO

---- Configuration Menu ------------------------------------

....

Configuration Menu: Enter command > PATHFLAGS

     Primary Boot Path Action
          Boot Actions:  Boot from this path.
                         If unsuccessful, go to next path.

HA Alternate Boot Path Action
          Boot Actions:  Boot from this path.
                         If unsuccessful, go to BCH.

   Alternate Boot Path Action
          Boot Actions:  Skip this path.
                         Go to BCH.

Configuration Menu: Enter command >
```

Listing nPartition Boot Settings [HP-UX]

Use the **`parstatus -V -p#`** and **`setboot`** commands to list an nPartition's boot settings from HP-UX 11i.

NOTE Use the `parstatus` command to list various nPartition boot settings for *any* nPartition in a server complex.

The `setboot` command only provides information about the *local nPartition's* PRI and ALT boot paths and PRI boot actions.

Step 1. Login to HP-UX running on an nPartition.

If you want to list autoboot settings for an nPartition, you must login to the nPartition. To list other details, such as boot paths and core cell settings, you can login to any nPartition.

Step 2. Issue the **`parstatus -V -p#`** command to list detailed information about the specified nPartition (`-p#`), including boot-related details.

The boot setting information that `parstatus -V -p#` reports is equivalent to the following BCH commands: `PATH`, `CELLCONFIG`, and `CORECELL`.

The following example lists detailed information for nPartition number 0, including the nPartition's boot path settings, its core cell information, and each cell's use-on-next-boot settings.

```
# parstatus -V -p0
[Partition]
Partition Number       : 0
Partition Name         : jules00
Status                 : active
IP address             : 0.0.0.0
Primary Boot Path      : 0/0/2/0/0.13.0
Alternate Boot Path    : 0/0/2/0/0.0.0
HA Alternate Boot Path : 0/0/2/0/0.14.0
PDC Revision           : 6.0
IODCH Version          : 23664
CPU Speed              : 552 MHz
Core Cell              : cab0,cell0
Core Cell Alternate [1]: cab0,cell0
Core Cell Alternate [2]: cab0,cell2
```

```
[Cell]
                        CPU     Memory                                 Use
                        OK/     (GB)                           Core    On
Hardware   Actual       Deconf/ OK/                            Cell    Next Par
Location   Usage        Max     Deconf    Connected To         Capable Boot Num
========== ============ ======= ========= =================== ======= ==== ===
cab0,cell0 active core  4/0/4    2.0/ 0.0 cab0,bay0,chassis1  yes     yes  0
cab0,cell2 active base  4/0/4    2.0/ 0.0 cab0,bay1,chassis3  yes     yes  0

[Chassis]
                                 Core Connected  Par
Hardware Location   Usage        IO   To         Num
=================== ============ ==== ========== ===
cab0,bay0,chassis1  active       yes  cab0,cell0 0
cab0,bay1,chassis3  active       yes  cab0,cell2 0

#
```

As the above example shows, the primary (PRI) boot path is 0/0/2/0/0.13.0, the active core cell is cell 0, and the core cell choices are cell 0 (first preference) and cell 2 (second preference). Both of the nPartition's cells are set to be used ("yes") the next time the nPartition boots. Both cells are actively used ("active core" and "active base").

Step 3. Issue the **setboot** command to list the *local nPartition's* PRI and ALT (but not HAA) boot paths, and to list the boot actions for the PRI boot path.

```
# setboot
Primary bootpath : 0/0/2/0/0.13.0
Alternate bootpath : 0/0/2/0/0.0.0

Autoboot is ON (enabled)
Autosearch is OFF (disabled)

Note: The interpretation of Autoboot and Autosearch has changed
for
systems that support hardware partitions. Please refer to the
manpage.
#
```

The `setboot` command reports the local nPartition's PRI and ALT boot path values, but does not list the HAA boot path.

The `setboot` command also reports the "autoboot" and "autosearch" settings for the PRI boot path. Combined, these two settings are equivalent to the PRI path's boot actions (its "path flags" setting).

When autoboot is set to ON, the nPartition attempts to automatically boot from the PRI boot path when it first boots to BCH. Otherwise, when autoboot is OFF, the nPartition remains at the BCH interface on startup.

When autosearch is set to ON, the nPartition will attempt to perform the boot action for the HAA boot path if the PRI boot action is automatically attempted and fails to boot (when autoboot is ON). When autosearch is set to OFF, the nPartition remains at BCH if the PRI path is not automatically booted on startup.

Refer to the section *Configuring Autoboot and Autostart* on page 233 for other details and procedures.

Listing nPartition Boot Settings [Partition Manager]

This procedure (**Partition —> Show Partition Details** action, **General** tab) lists an nPartition's boot paths from Partition Manager.

Step 1. Run Partition Manager (`/opt/parmgr/bin/parmgr`) or access it from SAM or a Web browser.

Step 2. Select the nPartition whose boot path settings you want to view.

Partitions are listed on the left side of the Partition Manager primary window.

Step 3. Select the **Partition —> Show Partition Details** action and view the boot path settings in the **General** tab.

This displays the PRI, HAA, and ALT boot path values for the selected nPartition.

Checklist and Guidelines for Booting nPartitions

This section provides both a checklist to use when booting an nPartition, and a set of guidelines to consider when configuring nPartition boot settings.

Boot Checklist for nPartitions

Before you boot an nPartition, check the items listed here.

❑ All cells in the nPartition that have a “y” use-on-next-boot value should be powered on.

If any cells that are set to be used are powered off, the nPartition will take longer to boot.

During partition rendezvous, the nPartition will wait up to *10 minutes* for all cells that are designated to be used. Any cells not powered on will not be active or available in the nPartition.

❑ All I/O chassis and devices for the nPartition’s active cells should be powered on.

❑ If any cells that have a “y” use-on-next-boot value are inactive, perform a reboot for reconfig of the nPartition (`shutdown -R`) to allow them to reset and become active during partition rendezvous.

❑ All complex profile information for the nPartition must be coherent.

This means all cells assigned to the nPartition must have identical complex profile information (Partition Configuration Data).

After you add a cell or remove and active cell from an nPartition, you must perform a reboot for reconfig of the nPartition (`shutdown -R`) to synchronize the complex profile data throughout the nPartition. The reboot for reconfig also causes all the nPartition’s cells to reboot and allows cells to go through the partition rendezvous procedure.

Boot Configuration Guidelines for nPartitions

The following guidelines are points to consider when configuring boot settings for the nPartitions in your server complex.

❑ Configure HAA and ALT boot devices in addition to the PRI device.

By configuring HAA and ALT boot device paths, you establish additional bootable devices that provide redundancy in case the PRI device fails.

Also configure the path flags for the boot paths, to allow the HAA device to boot automatically if the device at the PRI path cannot boot.

- ❑ Ensure that the core I/O, PRI boot device, and network card(s) all are connected to same cell (the core cell).

  This configuration ensures that the core cell is directly connected to the I/O required for booting the nPartition and providing network connections. Having such a configuration eliminates the requirement for multiple cells to be functional to provide basic nPartition services.

- ❑ Have multiple core cells available within each nPartition, if possible.

  In order to have multiple core cell choices, the nPartition must have at least two cells, each connected to an I/O chassis and core I/O.

  Having such a configuration provides redundancy and potentially improved system availability. If one core cell has a failure or otherwise cannot serve as the active core cell, the second core-capable cell can serve as the active core cell.

  Disregard this guideline if configuring multiple nPartitions in an HP rp7405/rp7410 or HP rp8400 server complex. HP rp7405/rp7410 servers and HP rp8400 servers have up to two core-capable cells only, so following this guideline would require having only one nPartition in the server complex.

nPartition Boot Activity Monitoring

On HP nPartition servers you can monitor the nPartition boot process—from power-on or reset to HP-UX start-up—using the Virtual Front Panel (VFP) view of the nPartition.

Each nPartition has its own VFP that displays details about the nPartition's cells and the nPartition's boot state and activity.

NOTE After you add or remove cells from the nPartition, you must exit and re-enter the nPartition's VFP to update the list of cells the VFP displays.

Monitoring nPartition Boot Activity [Service Processor]

Use the following procedure (service processor Main menu, **VFP** option) to access an nPartition Virtual Front Panel for monitoring the nPartition's boot status.

Step 1. From the Main menu, enter **VFP** to select the Virtual Front Panel option.

```
 GSP MAIN MENU:

Utility Subsystem FW Revision Level: SR_XXXX_D

         CO: Consoles
        VFP: Virtual Front Panel
         CM: Command Menu
         CL: Console Logs
         SL: Show chassis Logs
         HE: Help
          X: Exit Connection

GSP> VFP
```

If you are accessing the service processor using a single-partition-user account, selecting the VFP option takes you directly to the nPartition's Virtual Front Panel.

If accessing the GSP using an operator or administrator account, you can select the VFP for any single nPartition, or can select a *system VFP* that displays the nPartition state and activity for all nPartitions within the server complex.

Step 2. Select the nPartition you wish to monitor.

Skip this step if you are accessing the service processor using a single-partition-user account.

```
 Partition VFP's available:

     #   Name
    ---  ----
     0)  jules00
     1)  jules01
     S)  System (all chassis codes)
     Q)  Quit

GSP:VFP> 1
```

Step 3. View the VFP details for information about the nPartition and its current boot state.

To exit the VFP and return to the service processor main menu, type **^b** (**Control-b**).

The VFP provides information about the nPartition state, nPartition activity, each cell's state, and each cell's activity. The VFP display updates as the cell or nPartition state and activities change.

```
E indicates error since last boot
     Partition 1  state            Activity
     ------------------            --------
     Cell(s) Booting:     57 Logs

  #  Cell state                    Activity
  -  ----------                    --------
  4  Booting                       Cell firmware test                 28  Logs
  6  Booting                       Cell firmware test                 28  Logs


GSP:VFP (^B to Quit) >
```

Hanged HP-UX and Running HP-UX Detection

This section describes how you can determine whether HP-UX still is running on an nPartition even when you are unable to login or access the nPartition console.

You also may wish to reference the information in the procedure *Configuring Automatic System Restart for an nPartition* on page 235 for details on configuring an nPartition to reboot when HP-UX has hanged on it for over three minutes.

Detecting if HP-UX is Running or Hanged on an nPartition

To determine whether HP-UX is running or has hanged on an nPartition use this procedure (first check the Virtual Front Panel, then check the Chassis Logs menu's **Live Logs** display, then the nPartition's Console).

Refer to the chapter *Using Console and Service Processor Interfaces* on page 125 for details on service processor login accounts and features.

Step 1. Access and view the nPartition's Virtual Front Panel (VFP).

Login to the service processor (GSP or MP) for the server where the nPartition resides, enter **VFP** to access the VFP menu, and select the nPartition whose boot state you want to check.

- To exit an nPartition's VFP, type **^b** (**Control-b**).
- When HP-UX has booted on the nPartition, `HPUX heartbeat` is displayed as the partition state when you view the nPartition VFP.
- If HP-UX is alive, an asterisk (`*`) blinks on and off in the partition state area of the nPartition VFP.

Also see *Boot States and Activities for nPartitions and Cells* on page 194 for details interpreting the VFP status.

Step 2. If the nPartition VFP indicates that HP-UX has booted but is not alive, exit the VFP and view the live chassis logs for the nPartition.

At the GSP main menu, enter **SL** to enter the **Show chassis Logs** menu, and enter **L** to select the **Live Chassis Logs** display from the **Chassis Logs** menu.

```
GSP> SL

Chassis Logs available:

    (A)ctivity Log
    (E)rror Log
    (L)ive Chassis Logs

    (C)lear All Chassis Logs
    (Q)uit

GSP:VW> L

        Entering Live Log display
```

Step 3. From the Live Log display, type **P** and select the nPartition whose chassis codes you want to view by typing the partition number.

By default the live chassis log viewer only displays alert codes.

When you select the partition filter, the live log's view changes to show all codes for the selected nPartition.

```
        Entering Live Log display

        A)lert filter
        C)ell filter
        P)artition filter
        U)nfiltered
        V)iew format selection
        ^B to Quit

Current filter: ALERTS only

p
Enter partition number to display

    Partitions available:
     #   Name
    ---  ----
     0)  feshd4a
     2)  feshd4b
     Q)  Quit
    Please select partition number: 0

Filtering partition 0
```

Step 4. When viewing the selected nPartition's live log display, determine whether HP-UX is emitting HEARTBEAT chassis codes.

To exit the live log display, type **^b** (**Control-b**)

Heartbeat for HP-UX

When HP-UX is running on the nPartition, the live log partition filter shows the nPartition's HP-UX HEARTBEAT chassis codes and corresponding ACTIVITY_LEVEL_TIMEOUT counter updates.

```
Filtering partition 0
Alert Level 0: No failure; Keyword: HEARTBEAT
Processor 0 ; Status: 15
Logged by HP-UX 26 during display_activity update subActivity 10
Legacy PA HEX chassis-code: f10f
0xf8e1a8001100f10f 0x000000000000f10f

HPUX 0,0,0  0  ACTIVITY_LEVEL_TIMEOUT
Alert Level 0: No failure; Keyword: ACTIVITY_LEVEL_TIMEOUT
Processor 0 timeout; Status:  0
Logged by HP-UX 0 during display_activity update subActivity 0
Activity Level/Timeout: 0% / 3 minutes
0x78e008041100f000 0x0000000200000000

HPUX 0,6,2  0  HEARTBEAT
Alert Level 0: No failure; Keyword: HEARTBEAT
Processor 0 ; Status: 15
Logged by HP-UX 26 during display_activity update subActivity 10
Legacy PA HEX chassis-code: f10f
0xf8e1a8001100f10f 0x000000000000f10f
```

Activity Timeout Counter

The nPartition activity timeout counter is reset every time HP-UX on the nPartition emits a HEARTBEAT chassis code. The timeout counter expires when no HEARTBEAT code has been emitted for three minutes in the nPartition.

See *Configuring Automatic System Restart for an nPartition* on page 235 for more details about the activity timeout counter.

Step 5. If the nPartition is not emitting HEARTBEAT chassis codes, then access the nPartition's console by entering **CO** from the service processor main menu and selecting the partition number for the nPartition.

Accessing the nPartition's console may help determine whether an HP-UX crash dump is occurring or any console or error messages were given.

Step 6. Determine whether the nPartition needs to be reset in order to restore HP-UX to a running state.

Review the findings from the previous steps in this procedure.

HP-UX on the nPartition may be considered "hanged" if you observed all of the following VFP, live log, and console behaviors:

- The VFP indicates `HPUX heartbeat` in the partition state with no asterisk (`*`) blinking to indicate activity.
- The nPartition's live log displays no `HEARTBEAT` chassis codes.
- The nPartition console is inactive with no indication of a crash dump or other error, and no console login or interactivity is possible.

If all of the above attempts to find signs of HP-UX activity on the nPartition fail, then you may need to reset the nPartition before HP-UX can be restored to a running state.

For details see the procedure *Rebooting or Resetting an nPartition* on page 214 or the procedure *Performing a Transfer-of-Control (TOC) Reset of an nPartition* on page 223.

Troubleshooting Boot Issues

On HP nPartition servers, you might encounter different boot issues than on other HP servers.

The following boot issues are possible on nPartition servers.

- **Problem:** Not all cells boot to join (rendezvous) an nPartition.

  Causes: Some cells may have their use-on-next-boot value set to "n" (do not use), or the cells may have been powered off, or the cells may have booted too late to participate in partition rendezvous, or the cells have failed self-tests and cannot be used.

  Actions: Check the cell use-on-next-boot values and change them to "y" as needed then reboot for reconfig (`shutdown -R`). Check cell power (`frupower -d -C`) and power on any cells as needed, then reboot for reconfig.

  As the nPartition's cells reboot, observe the boot progress from the nPartition's VFP and note any problems the cells have proceeding from one boot state to the next; as needed review chassis logs and error logs using the service processor Show Chassis Logs (`SL`) menu.

- **Problem:** An nPartition takes a long time to boot (over ten minutes).

  Causes: One or more cells assigned to the nPartition that have a "y" use-on-next-boot value has not booted to participate in partition rendezvous, thus causing the rest of the nPartition's cells to wait for ten minutes for the cell to report.

  For example, the cell might not be installed, might be powered off, or might have been powered on or reset too late to rendezvous with the other cells.

  Actions: You can avoid the delay by performing any of the following actions, as needed. Perform a reboot for reconfig following any changes you make.

  — Set the cell's use-on-next-boot value to "n" (do not use).

  — Power on the cell.

  — Unassign (remove) the cell from the nPartition.

- **Problem:** An nPartition does not boot to BCH and instead all cells remain at a boot-is-blocked (BIB) state.

  Causes: The nPartition has been reset to the ready for reconfig state, or no valid core cell is available to the nPartition.

  Actions: If the nPartition was reset to the ready for reconfig state, use the service processor Command menu's `BO` command to boot the nPartition base boot-is-blocked (to allow it to boot to its BCH interface).

  If no valid core cell was available to the nPartition when it booted, check the power for all core cell choices (a cell might be powered off) and power it on if needed.

  Also review the chassis logs for the nPartition to search for any core cell problems and failures.

Boot States and Activities for nPartitions and Cells

On HP nPartition servers, the cell and nPartition boot process proceeds from one boot state to the next; cells and nPartitions complete various boot activities within each boot state before proceeding to the next boot state.

You can view current details about nPartition and cells boot states and activities by viewing the nPartition's Virtual Front Panel. From the service processor (GSP or MP) Main menu enter **VFP** to access the VFPs that are available for the server complex.

Table 4-4 on page 195 presents the nPartition and cell states and activities that you can observe from an nPartition's Virtual Front Panel.

You can view a Virtual Front Panel for a specific nPartition that includes details for all cells in the nPartition, as shown below.

```
E indicates error since last boot
     Partition 0  state            Activity
     ------------------            --------
     Cell(s) Booting:    904 Logs

  #  Cell state                    Activity
  -  ----------                    --------
  0  Late CPU selftest             Processor test                     299  Logs
  2  Late CPU selftest             Processor test                     299  Logs
  4  Memory discovery              Physical memory test               304  Logs






GSP:VFP (^B to Quit) >
```

You also can view a system-wide VFP, which shows a summary of each nPartition's current state and its activity.

Inactive cells remain at a "Boot Is Blocked (BIB)" state following I/O discovery and do not participate in partition rendezvous.

Table 4-4 HP nPartition and Cell Boot States and Activities

Partition State	Partition Activity	Cell States	Cell Activities
Cell(s) Booting		Booting	Cell firmware configuration, Cell firmware test, Cell PDH controller configuration
Cell(s) Booting		Early CPU self-test	Processor test, Cell firmware test, Processor firmware slave rendezvous
Cell(s) Booting		Memory discovery	Physical memory test
Cell(s) Booting		Late CPU self-test	Processor test, Cell firmware test, Processor firmware slave rendezvous
Cell(s) Booting		I/O discovery	I/O system bus adapter configuration, I/O local bus adapter configuration
Cell(s) Booting		Remote fabric initialization	Partition rendezvous slave rendezvous
Memory Interleave	Memory controller configuration	Cell has joined partition	
At Boot Console Handler (BCH)	Partition firmware	Cell has joined partition	
ISL Menu		Cell has joined partition	
HPUX Loader Init		Cell has joined partition	

Table 4-4 HP nPartition and Cell Boot States and Activities (Continued)

Partition State	Partition Activity	Cell States	Cell Activities
HPUX Launch	Processor system initialization	Cell has joined partition	
HPUX Launch	Partition IPL launch configuration	Cell has joined partition	
HPUX Launch	Processor display_activity update	Cell has joined partition	
HPUX init process start		Cell has joined partition	
HPUX heartbeat		Cell has joined partition	

5 Booting and Resetting nPartitions

This chapter presents procedures for booting and resetting nPartitions and procedures for configuring an nPartition’s boot-related options.

For an introduction to nPartition boot issues, refer to the chapter *An Overview of nPartition Boot and Reset* on page 161.

NOTE For details on booting and rebooting virtual partitions within an nPartition, refer to the chapter *Virtual Partitions (vPars) Management on nPartitions* on page 441.

Accessing an nPartition Console and BCH Interface

Each nPartition has its own Boot Console Handler (BCH) interface that provides you a method for interacting with the nPartition before HP-UX has booted on it.

You must access an nPartition's console and BCH interfaces through the server complex's service processor (GSP or MP). See *Accessing nPartition Console and BCH Interfaces [Service Processor]* below for a detailed procedure.

On nPartition servers, each nPartition's BCH interface is available through the nPartition's console before HP-UX has booted. The BCH interface enables you to manage the nPartition's HP-UX boot process and to configure various boot-related settings.

NOTE Always login to a server complex's service processor from a tty (not console) login session. You can check your current login terminal using the **`who -m`** command.

Do not login to a service processor from an nPartition console connection. Any use of the **^b** (**Control-b**) console exit sequence would exit the original console login—not the subsequent console-based login to the service processor—thus potentially stranding the console-based login (for example, if it too were accessing a console).

Accessing nPartition Console and BCH Interfaces [Service Processor]

The following procedure (login to service processor, select Console menu, select an nPartition) accesses an nPartition's console and BCH interface using the server complex's service processor.

Step 1. Login to the service processor (the GSP or MP) for the nPartition's server complex.

You can connect to the service processor using a direct physical connection, or using `telnet` for a remote connection.

In most situations, you can `telnet` to the service processor.

```
# telnet sdome-s
Trying...
Connected to sdome-s.rsn.hp.com.
Escape character is '^]'.
Local flow control off

GSP login: Accountname
GSP password:

                              Welcome to
               Superdome's Guardian Service Processor
```

Step 2. Select the Console menu (**CO**) from the service processor's Main menu.

The Console menu is the method for accessing nPartition consoles.

```
 GSP MAIN MENU:

Utility Subsystem FW Revision Level: SR_XXXX_D

         CO: Consoles
        VFP: Virtual Front Panel
         CM: Command Menu
         CL: Console Logs
         SL: Show chassis Logs
         HE: Help
          X: Exit Connection

GSP> CO
```

If you are accessing the service processor using a single-partition-user account, selecting the `CO` (console) option takes you directly to the nPartition's console.

If using an operator or administrator account, you can access the console for any of the nPartitions within the server complex.

Step 3. At the Console menu, enter the partition number for the nPartition whose console (and BCH interface) you wish to access.

Skip this step if you are accessing the service processor using a single-partition-user account.

If using an operator or administrator account, select the nPartition whose console you wish to access.

```
GSP> CO

    Partitions available:

     #   Name
    ---  ----
     0)  jules00
     1)  jules01
     Q)  Quit

    Please select partition number: 1

        Connecting to Console: jules01

        (Use ^B to return to main menu.)

        [A few lines of context from the console log:]

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

     SERvice menu                      Displays service commands

     DIsplay                           Redisplay the current menu
     HElp [<menu>|<command>]           Display help for menu or command
     REBOOT                            Restart Partition
     RECONFIGRESET                     Reset to allow Reconfig Complex Profile
----
Main Menu: Enter command or menu >

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

The console displays the last 10 lines of console output when you connect to it. This provides you a view of the most recent console activity.

Step 4. Gain interactive access to the nPartition console.

Press **Enter** to access the nPartition console's currently available prompt, if any. You will have either interactive or non-interactive access, as described in the sections *Interactive Console Access* and *Non-Interactive Console Access* in this step.

To exit the nPartition console and return to the service processor Main menu, type **^b** (**Control-b**) at any time.

Interactive Console Access

Typically the BCH interface, ISL interface, or the HP-UX login or command prompt is available from the nPartition console.

- **When an nPartition is at the BCH interface** you can access BCH commands from the nPartition's console and can reboot BCH if needed.
- **When an nPartition has booted to ISL** you can use the `EXIT` command to exit ISL and return to the nPartition's BCH interface.
- **When an nPartition has booted HP-UX**, in order to access the BCH interface you must reboot HP-UX and if necessary interrupt the automatic boot process. (To reboot the nPartition, use the `shutdown -r` command, or use `shutdown -R` if you also are changing the nPartition's cell configuration.)

Non-Interactive Console Access

In the following situations, you cannot interact with the nPartition's console. In these cases you can wait until the console is interactive or can force interactive access.

- **When the nPartition is resetting or is booting HP-UX** you cannot interact with software running on the nPartition.

  Once the nPartition has completed resetting, or has completed booting HP-UX, you can interact with the nPartition's BCH or HP-UX prompts.

  To determine an nPartition's boot state, use the nPartition's Virtual Front Panel (the VFP menu, available from the service processor Main menu).

- **When another user already is attached to the console** you can access the nPartition's console in spy (read-only) mode or can force write access by typing **^ecf** (**Control-e c f**).

  Spy mode allows you to view console information but does not enable you to enter commands. If you type when accessing an nPartition console in spy mode, the console prints the following message.

  ```
  [Read-only - use ^Ecf to attach to console.]
  ```

  When in spy mode, you can force access to the nPartition's console by typing **^ecf** (**Control-e c f**). Doing this provides you interactive console access and forces ("bumps") the user who was using the console into spy mode.

  ```
  [Bumped user - Admin.]
  ```

Boot Device Searching and Finding

You can search for and find bootable devices for an nPartition by using the BCH interface's SEARCH command. This command searches for and reports all bootable devices connected to any of the nPartition's currently active cells.

NOTE You cannot access any I/O connected to an nPartition's *inactive* cells (cells not being used for the current nPartition boot) or cells *not assigned* to the local nPartition.

As a consequence, the BCH SEARCH command does not report any devices connected to cells that are not currently assigned and active in the local nPartition.

Finding Bootable Devices [BCH]

This procedure (BCH Main menu, SEARCH command) finds and lists the bootable devices that are available to an nPartition.

Step 1. Login to the server complex's service processor (GSP or MP) and access the nPartition's console.

From the nPartition console you access the nPartition's BCH interface to search for bootable devices.

If the nPartition is not at the BCH interface you must either boot the nPartition or shut down HP-UX to return to the BCH interface.

Step 2. From the BCH interface's Main menu, issue the SEARCH command to find and list bootable devices in the nPartition.

When accessing the nPartition's BCH interface, if you are not at the BCH Main menu then enter MA to return to the Main menu.

The SEARCH command reports all potential boot devices it locates.

```
---- Main Menu ---------------------------------------------------------------

     Command                             Description
     -------                             -----------
     BOot [PRI|HAA|ALT|<path>]           Boot from specified path
```

```
     PAth [PRI|HAA|ALT] [<path>]       Display or modify a path
     SEArch [ALL|<path>]               Search for boot devices
     ScRoll [ON|OFF]                   Display or change scrolling capability

     COnfiguration menu                Displays or sets boot values
     INformation menu                  Displays hardware information
     SERvice menu                      Displays service commands

     DIsplay                           Redisplay the current menu
     HElp [<menu>|<command>]           Display help for menu or command
     REBOOT                            Restart Partition
     RECONFIGRESET                     Reset to allow Reconfig Complex Profile
----
Main Menu: Enter command or menu > SEARCH

Searching for potential boot device(s)
This may take several minutes.

To discontinue search, press any key (termination may not be immediate).


   Path#  Device Path (dec)                      Device Type
   -----  -----------------                      -----------
   P0     0/0/1/0/0.15                           Random access media
   P1     0/0/1/0/0.12                           Random access media
   P2     0/0/1/0/0.11                           Random access media
   P3     0/0/1/0/0.9                            Random access media
   P4     0/0/1/0/0.8                            Random access media
   P5     0/0/1/0/0.6                            Random access media


Main Menu: Enter command or menu >
```

The `SEARCH` command lists up to the first 20 potential boot devices that it locates, and lists each with a path number (P0 through P19).

To boot a device that was reported by the `SEARCH` command, specify the path number or the full device path. For example, `BOOT P0` would boot the path listed as path number P0.

HP-UX Booting on an nPartition

nPartitions boot and reboot HP-UX independently from each other. This section describes how to boot a single instance of HP-UX on an nPartition.

NOTE For details on booting HP-UX in virtual partitions (vPars), refer to the section *Booting HP-UX on Virtual Partitions* on page 487.

You can boot HP-UX on an nPartition using the BCH interface's `BOOT` command.

Each nPartition's BCH interface is available through its console. All nPartition consoles are available from the complex's service processor (GSP or MP) Console menu.

An nPartition will *automatically boot HP-UX* when its boot paths (PRI, HAA, ALT) and corresponding boot actions are appropriately set. For details see *Configuring Boot Paths and Boot Actions* on page 227.

On HP Superdome servers only, if one of the nPartition's components fails self-test and AUTOSTART is OFF then the nPartition stops booting at the BCH interface.

Booting HP-UX on an nPartition [BCH]

The following procedure (BCH interface **`BOOT`** command) boots HP-UX on an nPartition using the nPartition's BCH interface.

Step 1. Login to the server complex's service processor (GSP or MP), access the nPartition's console, and access the BCH Main menu.

From the nPartition console, you access the nPartition's BCH interface.

If the nPartition is not at the BCH interface, you must either boot the nPartition or shut down HP-UX to return to the BCH interface.

When accessing the nPartition's BCH interface, if you are not at the BCH Main menu then enter `MA` to return to the Main menu.

Step 2. Choose which device you wish to boot.

From the BCH Main menu, use the `PATH` command to list any boot path variable settings. The primary (PRI) boot path normally is set to the main boot device for the nPartition. You also can use the `SEARCH` command to find and list potentially bootable devices for the nPartition.

```
Main Menu: Enter command or menu > PATH

     Primary Boot Path:  0/0/2/0/0.13
                         0/0/2/0/0.d    (hex)

HA Alternate Boot Path:  0/0/2/0/0.14
                         0/0/2/0/0.e    (hex)

   Alternate Boot Path:  0/0/2/0/0.0
                         0/0/2/0/0.0    (hex)

Main Menu: Enter command or menu >
```

Step 3. Boot the device using the BCH interface's **`BOOT`** command.

You can issue the `BOOT` command in any of the following ways:

- **`BOOT`**

  Issuing the `BOOT` command with no arguments boots the device at the primary (PRI) boot path.

`BOOT` ***`bootvariable`***

  This command boots the device indicated by the specified boot path, where *`bootvariable`* is the PRI, HAA, or ALT boot path.

  For example, `BOOT PRI` boots the primary boot path.

- **`BOOT LAN INSTALL`** or **`BOOT LAN.`*****`ip-address`*** **`INSTALL`**

  The `BOOT... INSTALL` commands boot HP-UX from the default HP-UX install server or from the server specified by *`ip-address`*.

`BOOT` ***`path`***

  This command boots the device at the specified *`path`*. You can specify the *`path`* in HP-UX hardware path notation (for example, 0/0/2/0/0.13) or in "path label" format (for example, P0 or P1) .

  If you specify the *`path`* in "path label" format then *`path`* refers to a device path reported by the last `SEARCH` command.

After you issue the `BOOT` command, the BCH interface prompts you to specify whether you want to stop at the ISL prompt.

To boot the /stand/vmunix HP-UX kernel from the device *without stopping at the ISL prompt*, enter **n** to automatically proceed past ISL and execute the contents of the AUTO file on the selected device. (By default the AUTO file is configured to load /stand/vmunix.)

```
Main Menu: Enter command or menu > BOOT PRI

     Primary Boot Path:  0/0/1/0/0.15


 Do you wish to stop at the ISL prompt prior to booting? (y/n)
 >> n


ISL booting  hpux

Boot
: disk(0/0/1/0/0.15.0.0.0.0.0;0)/stand/vmunix
```

To boot an HP-UX kernel other than `/stand/vmunix`, or to boot HP-UX in single-user or LVM-maintenance mode, stop at the ISL prompt and specify the appropriate arguments to the `hpux` loader.

Booting an nPartition to the ISL Prompt

When you issue the BCH interface's `BOOT` command, you can stop an nPartition's booting at the Initial System Loader (ISL) interface in order to interact with the ISL prompt.

To exit ISL and return to the BCH interface, enter the `EXIT` command at the ISL prompt. For help enter `HELP` at the ISL prompt.

Normally you will not need to access ISL unless you need to use the Secondary System Loader (`hpux`).

For details about ISL, see the *isl* (1M) manpage. Details on the Secondary System Loader (`hpux`) are in the *hpux* (1M) manpage.

NOTE On HP nPartition servers many of the ISL commands *are not supported*. For example, AUTOBOOT, AUTOSEARCH, and PRIMPATH are not supported at ISL.

These and other features are instead supported on HP nPartition servers by each nPartition's BCH interface.

Booting an nPartition to ISL [BCH]

This procedure (BCH BOOT command, and reply **y** to "stop at the ISL prompt") boots an nPartition to the ISL prompt.

Step 1. Login to the server complex's service processor (GSP or MP), access the nPartition's console, and access the BCH interface.

From the nPartition console, you access the nPartition's BCH interface. If the nPartition is not at the BCH interface you must either boot the nPartition or shut down HP-UX to return to the BCH interface.

If an nPartition is configured to automatically boot HP-UX, you must interrupt the boot process before HP-UX boots, then manually boot HP-UX using the `BOOT` command (in the next step) to access the ISL interface on the nPartition.

Step 2. Boot the desired device using the BCH interface's **`BOOT`** command, and specify that the nPartition stop at the ISL prompt prior to booting (reply **y** to the "stop at the ISL prompt" question).

The `EXIT` command exits ISL and returns to the nPartition BCH interface, and the `HELP` command lists all available ISL interface commands.

```
Main Menu: Enter command or menu > BOOT 0/0/2/0/0.13

 BCH Directed Boot Path: 0/0/2/0/0.13


 Do you wish to stop at the ISL prompt prior to booting? (y/n)
>> y

Initializing boot Device.


ISL Revision A.00.42  JUN 19, 1999

ISL>
```

Single-User or LVM-Maintenance Mode HP-UX Booting

On an nPartition you can boot HP-UX in single-user mode or LVM-maintenance mode by specifying options to the Secondary System Loader (hpux).

From the nPartition's console, use the BCH interface to boot the desired device and stop at the Initial System Loader (ISL) interface, then use the Secondary System Loader (hpux) to specify the options for booting HP-UX in the desired mode.

See the *hpux* (1M) manpage for details on using the Secondary System Loader (hpux).

Booting HP-UX in Single-User or LVM-Maintenance Mode [BCH, ISL, and hpux]

This procedure (BCH **BOOT** command, stop at ISL interface, use hpux loader with options) boots HP-UX in single-user mode or LVM-maintenance mode on an nPartition.

Step 1. Login to the server complex's service processor (GSP or MP), access the nPartition's console, and access the BCH interface.

From the nPartition console you access the nPartition's BCH interface. If the nPartition is not at the BCH interface you must either boot the nPartition or shut down HP-UX to return to the BCH interface.

Step 2. Boot the desired device using the BCH interface's **BOOT** command, and specify that the nPartition stop at the ISL prompt prior to booting (reply **y** to the "stop at the ISL prompt" question).

```
Main Menu: Enter command or menu > BOOT 0/0/2/0/0.13

 BCH Directed Boot Path: 0/0/2/0/0.13


 Do you wish to stop at the ISL prompt prior to booting? (y/n)
>> y

Initializing boot Device.

....
```

```
ISL Revision A.00.42  JUN 19, 1999

ISL>
```

Step 3. From the ISL prompt, issue the appropriate Secondary System Loader (hpux) command to boot the HP-UX kernel in the desired mode.

Use the hpux loader to specify the boot mode options and to specify which kernel (such as: /stand/vmunix) to boot on the nPartition.

- To boot HP-UX in single-user mode:

  `ISL> hpux -is boot /stand/vmunix`

- To boot HP-UX in LVM-maintenance mode:

  `ISL> hpux -lm boot /stand/vmunix`

- To boot HP-UX at the default run level:

  `ISL> hpux boot /stand/vmunix`

To exit the ISL prompt and return to the BCH interface, issue the EXIT command instead of specifying one of the above hpux loader commands.

See the *hpux* (1M) manpage for a detailed list of hpux loader options.

Example 5-1 Example Single-User HP-UX Boot

```
ISL Revision A.00.42  JUN 19, 1999

ISL> hpux -is /stand/vmunix

Boot
: disk(0/0/2/0/0.13.0.0.0.0.0;0)/stand/vmunix
8241152 + 1736704 + 1402336 start 0x21a0e8

....

INIT: Overriding default level with level 's'

INIT: SINGLE USER MODE

INIT: Running /sbin/sh
#
```

HP-UX Install Source Booting

You can boot an nPartition from an HP-UX installation source—such as an install CD or an Ignite server—by specifying the install source using the BCH interface's `BOOT` command.

This allows you to install HP-UX on any of the nPartition's eligible devices.

NOTE For instructions on installing HP-UX, refer to the book *HP-UX 11i Installation and Update Guide*, which is supplied with the HP-UX operating environment media.

Booting from an HP-UX Install Source [BCH]

This procedure boots an HP-UX install source on an nPartition using the nPartition's BCH interface.

Step 1. Login to the server complex's service processor (GSP or MP), access the nPartition's console, and access the BCH interface.

From the nPartition console, you access the nPartition's BCH interface. If the nPartition is not at the BCH interface you must either boot the nPartition or shut down HP-UX to return to the BCH interface.

Step 2. Select the HP-UX install source that you wish to boot.

From the BCH main menu, you can boot from an Ignite server or install CD-ROM in order to install HP-UX.

You can use the `SEARCH` command to find and list potentially bootable devices for the nPartition, including any DVD-ROM devices that may have CD-ROM install media. The ALT boot path also might be set to the DVD-ROM device's path.

Step 3. Boot the install source using the BCH interface's **`BOOT`** command.

Specify the device path where the install media resides or specify the install server.

For details on booting, see *HP-UX Booting on an nPartition* on page 204.

Shutting Down HP-UX on an nPartition

When HP-UX is running on an nPartition, you can shut down HP-UX using either the `shutdown` command or the `reset` command.

NOTE The `reboot` command does not invoke the shutdown scripts associated with subsystems. The `shutdown` command invokes the scripts and terminates all running processes in an orderly and cautious manner.

On nPartitions you have the following options when shutting down HP-UX:

- To shut down HP-UX and **reboot** an nPartition: **`shutdown -r`**
- To shut down HP-UX and **halt** an nPartition: **`shutdown -h`**
- To perform a **reboot for reconfig** of an nPartition: **`shutdown -R`**
- To hold an nPartition at a **ready for reconfig** state: **`shutdown -R -H`**

For details see the *shutdown* (1M) manpage.

Shutting Down HP-UX on an nPartition [HP-UX]

This procedure shuts down HP-UX on an nPartition.

Step 1. Login to HP-UX running on the nPartition.

You can login to HP-UX on the nPartition either by directly connecting (with the `telnet` or `rlogin` commands) or by logging in to its complex's service processor (GSP or MP) and using the Console menu to access the nPartition's console.

Accessing the console through the service processor allows you to maintain console access to the nPartition after HP-UX has shut down.

Step 2. Issue the **`shutdown`** command with the appropriate command-line options.

The command-line options you specify dictate the way in which HP-UX is shut down, whether the nPartition is rebooted, and whether any nPartition configuration changes (adding or removing cells) take place.

Use the following list to choose an HP-UX shut down option for your nPartition.

- Shut down HP-UX and halt the nPartition.

  Issue the **`shutdown -h`** command to shut down and halt the nPartition.

  This leaves the nPartition and all its cells in an *active* state (the nPartition cannot be reconfigured) after HP-UX shuts down and halts.

  To reboot the nPartition you must reset the nPartition using the GSP command menu's `RS` command.

- Shut down HP-UX and reboot the nPartition.

  Issue the **`shutdown -r`** command to shut down and reboot the nPartition.

- Perform a *reboot for reconfig* of the nPartition.

  Issue the **`shutdown -R`** command to perform a reboot for reconfig.

  This shuts down HP-UX, reconfigures the nPartition if needed, and reboots the nPartition.

- Reboot the nPartition and put it in to the *ready for reconfig* state.

  Use the **`shutdown -R -H`** command to hold the nPartition in the ready for reconfig state.

  This leaves the nPartition and all its cells in an *inactive* state (the nPartition can be reconfigured remotely).

  To reboot the nPartition you must do so manually by using the service processor Command menu's `BO` command.

If HP-UX is halted on the nPartition, thus not allowing you to use the `shutdown` command, you can reboot or reset the nPartition by issuing commands from the service processor Command menu.

See *Rebooting or Resetting an nPartition* on page 214.

Rebooting or Resetting an nPartition

When you perform a reboot or reset of an nPartition, all *active cells* in the nPartition reboot and return to BCH or HP-UX. Any *inactive* cells in the nPartition are not rebooted in this procedure.

You can reset and reboot an nPartition by using these procedures:

- *Rebooting or Resetting an nPartition [Service Processor]* on page 215
- *Rebooting or Resetting an nPartition [BCH]* on page 216
- *Rebooting or Resetting an nPartition [HP-UX]* on page 216

NOTE If possible you should down HP-UX before resetting an nPartition.

HP's nPartition servers also support other types of nPartition resetting.

See the following sections for details on these other nPartition reset methods:

- *Shutting Down HP-UX on an nPartition* on page 212
- *Performing a Reboot for Reconfig for an nPartition* on page 218
- *Holding an nPartition at the Ready for Reconfig State* on page 219
- *Performing a Transfer-of-Control (TOC) Reset of an nPartition* on page 223

Rebooting or Resetting an nPartition [Service Processor]

Use the service processor Command menu **RS** command to reset an nPartition from the service processor (GSP or MP).

Step 1. Login to the server complex's service processor (GSP or MP) and access the Command menu.

After logging in to the service processor, enter **CM** to select the Command menu.

```
GSP login: Accountname
GSP password: Password

....

GSP> CM

                Enter HE to get a list of available commands

GSP:CM>
```

Step 2. At the Command menu, enter the **RS** command, specify which nPartition is to be reset, and confirm whether to reset it.

The Command menu's RS command resets all *active cells* in the nPartition and reboots them past partition rendezvous to BCH or HP-UX.

Be certain to correctly select which nPartition to be reset.

```
GSP:CM> RS

This command resets the selected partition.

WARNING: Execution of this command irrecoverably halts all
system
         processing and I/O activity and restarts the selected
         partition.


     #   Name
    ---  ----
     0)  jules00
     1)  jules01

    Select a partition number: 1

    Do you want to reset partition number 1? (Y/[N]) y
```

```
    -> The selected partition will be reset.
GSP:CM>
```

If you are accessing the service processor using a single-partition-user account, the RS command selects which nPartition is to be reset: the nPartition that your account allows you to access.

If using an operator or administrator service processor account, you can select which of the server complex's nPartitions you want to reset.

Rebooting or Resetting an nPartition [BCH]

Use the **REBOOT** command to reset an nPartition from the BCH interface.

Step 1. Login to the server complex's service processor, access the nPartition's console, and access the BCH Main menu.

From the nPartition console you access the nPartition's BCH interface. If the nPartition is not at the BCH interface you must either boot the nPartition or shut down HP-UX to return to the BCH interface.

Step 2. From the nPartition's BCH main menu, enter the **REBOOT** command to reboot the nPartition.

The BCH interface's REBOOT command resets all *active cells* in the nPartition and reboots them past partition rendezvous to BCH or HP-UX.

```
Main Menu: Enter command or menu > REBOOT
Rebooting the partition ...
```

Rebooting or Resetting an nPartition [HP-UX]

Use the **shutdown -r** command to reset an nPartition from HP-UX running on the nPartition.

Step 1. Login to HP-UX running on the nPartition you want to reset.

You can login to HP-UX on the nPartition either by directly connecting (with the telnet or rlogin commands) or by logging in to its complex's service processor (GSP or MP) and using the Console menu to access the nPartition's console.

Step 2. Issue the **shutdown -r** command to reset the nPartition.

The `shutdown -r` command shuts down HP-UX and reboots the nPartition. All *active cells* in the nPartition are reset.

Performing a Reboot for Reconfig for an nPartition

During a **reboot for reconfig** of an nPartition, the HP-UX command that you issue (`shutdown -R`) performs the following tasks:

1. Shuts down HP-UX and resets all cells that are assigned to the nPartition, including any inactive cells.
2. Reconfigures the nPartition if necessary (adds or removes cells).
3. Boots all cells in the nPartition. Any cells with a “n” use-on-next-boot value remain inactive at BIB, and all other cells can rendezvous to form the nPartition.

You should perform a reboot for reconfig of an nPartition whenever you add or remove cells from the nPartition, and whenever you need to allow an inactive cell to join the nPartition (such as after changing a cell’s use-on-next-boot value from “n” to “y”).

Performing a Reboot for Reconfig [HP-UX]

Use the **`shutdown -R`** command to perform a reboot for reconfig for an nPartition.

Step 1. Login to HP-UX running on the nPartition.

You can login to HP-UX on the nPartition either by directly connecting (with the `telnet` or `rlogin` commands) or by logging in to its complex’s service processor (GSP or MP) and using the Console menu to access the nPartition’s console.

Step 2. Issue the **`shutdown -R`** command to perform a reboot for reconfig of the nPartition.

The `shutdown -R` command shuts down HP-UX, reboot all cells assigned to the nPartition, performs any nPartition reconfigurations, and boot all cells that have “y” use-on-next-boot values.

Holding an nPartition at the Ready for Reconfig State

Resetting an nPartition to the **ready for reconfig** state performs any changes to the nPartition's configuration and holds the nPartition and all its cells in a boot-is-blocked (*inactive*) state.

To boot an nPartition after you have reset it to the ready for reconfig state, you must use the service processor (GSP or MP) Command menu's `BO` command.

You can hold an nPartition at the ready for reconfig state by using the following procedures:

- *Holding an nPartition at the Ready for Reconfig State [Service Processor]* on page 220
- *Holding an nPartition at the Ready for Reconfig State [BCH]* on page 221
- *Holding an nPartition at the Ready for Reconfig State [HP-UX]* on page 221

When you use the above methods to hold an nPartition at the ready for reconfig state, the commands perform the following tasks:

1. Shut down HP-UX (if using the `shutdown -R -H` command) and reset all cells that are assigned to the nPartition, including any inactive cells.
2. Reconfigures the nPartition if necessary (adds or removes cells).
3. Keeps all cells at a boot-is-blocked state; the nPartition and all cells assigned to it are inactive.

You should reset an nPartition to ready for reconfig whenever you need for the nPartition and its cells to be inactive. This enables you to modify the nPartition's configuration from the GSP or from HP-UX running on a remote nPartition in the same system complex.

Holding an nPartition at the Ready for Reconfig State [Service Processor]

Use the Command menu **RR** command to reset an nPartition to the ready for reconfig state from the service processor (GSP or MP).

Step 1. Login to the server complex's service processor and enter **CM** to access the Command menu.

```
GSP> CM

                Enter HE to get a list of available commands

GSP:CM>
```

Step 2. At the service processor Command menu, enter the **RR** command, specify which nPartition is to be reset, and confirm whether to reset it to the ready for reconfig state.

The service processor's RR command resets all cells in the nPartition, performs any nPartition reconfigurations, and halts all cells at a boot-is-blocked state, thus making the nPartition and all its cells inactive.

Be certain to select the correct nPartition to be reset.

```
GSP:CM> RR

This command resets for reconfiguration the selected partition.

WARNING: Execution of this command irrecoverably halts all system
         processing and I/O activity and restarts the selected
         partition in a way that it can be reconfigured.


     #   Name
    ---  ----
     0)  jules00
     1)  jules01

    Select a partition number: 1


    Do you want to reset for reconfiguration partition number 1? (Y/[N]) y

    -> The selected partition will be reset for reconfiguration.
GSP:CM>
```

If you are accessing the service processor using a single-partition-user account, the `RR` command selects which nPartition is to be reset: the nPartition that your account allows you to access.

If using an operator or administrator GSP account, you can select which of the server complex's nPartitions you want to reset.

Holding an nPartition at the Ready for Reconfig State [BCH]

Use the **`RECONFIGRESET`** command to reset an nPartition to the ready for reconfig state from the nPartition's BCH interface.

Step 1. Login to the server complex's service processor, access the nPartition's console, and access the BCH interface.

From the nPartition console you access the nPartition's BCH interface. If the nPartition is not at the BCH interface you must either boot the nPartition or shut down HP-UX to return to the BCH interface.

Step 2. From the nPartition's BCH interface, enter the **`RECONFIGRESET`** command to reset the nPartition to the ready for reconfig state.

The `RECONFIGRESET` command resets all cells in the nPartition, performs any nPartition reconfigurations, and halts all cells at a boot-is-blocked state, thus making the nPartition and all its cells inactive.

```
Main Menu: Enter command or menu > RECONFIGRESET
Reset the partition for reconfiguration of Complex Profile ...
```

Holding an nPartition at the Ready for Reconfig State [HP-UX]

Use the **`shutdown -R -H`** command to reset an nPartition to the ready for reconfig state from HP-UX running on the nPartition.

Step 1. Login to HP-UX running on the nPartition.

You can login to HP-UX on the nPartition either by directly connecting (with the `telnet` or `rlogin` commands) or by logging in to its complex's service processor (GSP or MP) and using the Console menu to access the nPartition's console.

Step 2. Issue the **`shutdown -R -H`** command to reset the nPartition to the ready for reconfig state.

The `shutdown -R -H` command shuts down HP-UX, reset all cells in the nPartition, perform any nPartition reconfigurations, and halt all cells at a boot-is-blocked state, thus making the nPartition and all its cells inactive.

Performing a Transfer-of-Control (TOC) Reset of an nPartition

You can use the service processor Command menu's `TC` command to perform a transfer-of-control (TOC) reset of an nPartition.

If crash dump is configured for HP-UX on the nPartition, when you TOC the nPartition while it is running HP-UX the nPartition performs a crash dump and gives you an opportunity select the type of dump.

Performing a TOC Reset of an nPartition [Service Processor]

Use the Command menu **`TC`** command to perform a transfer-of-control (TOC) reset of an nPartition.

Step 1. Login to the server complex's service processor and enter **`CM`** to access the Command menu.

```
GSP> CM

                Enter HE to get a list of available commands

GSP:CM>
```

Step 2. At the Command menu, enter the **`TC`** command, specify which nPartition is to be reset, and confirm whether to TOC the nPartition.

The `TC` command performs a transfer-of-control reset on the specified nPartition.

If you are accessing the service processor using a single-partition-user account, the `TC` command selects which nPartition is to be reset: the nPartition that your account allows you to access.

If using an operator or administrator account, you can select which of the server complex's nPartitions you want to TOC.

Be certain to select the correct nPartition to be reset.

```
GSP:CM> TC

This command TOCs the selected partition.
```

```
WARNING: Execution of this command irrecoverably halts all system
         processing and I/O activity and restarts the selected
         partition.


     #   Name
    ---  ----
     0)  jules00
     1)  jules01

Select a partition number: 0

    Do you want to TOC partition number 0? (Y/[N]) y

    -> The selected partition will be TOCed.
GSP:CM>
```

Step 3. After you initiate the TOC, you can observe its progress and select the type of crash dump through the nPartition's console.

Once the nPartition completes the dump, or once you cancel it, the nPartition reboots.

```
       ******* Unexpected TOC. Processor HPA FFFFFFFF'FC07C000 *******
                              GENERAL REGISTERS:
r00/03 00000000'00000000 00000000'0099CA2C 00000000'00000000 00000000'010BB790
r04/07 00000000'00000002 00000000'010BC140 00000000'0080F000 00000000'00AA2490
r08/11 00000000'00000001 00000000'0099A800 00000000'0099A800 00000000'0099C800

....

Processor 8 TOC:  pcsq.pcoq = 0'0.0'12675c
                  isr.ior   = 0'10340004.0'2f8bfd30

Boot device reset done.
*** The dump will be a SELECTIVE dump:  457 of 4080 megabytes.
*** To change this dump type, press any key within 10 seconds.
*** Proceeding with selective dump.

*** The dump may be aborted at any time by pressing ESC.
*** Dumping:   7% complete (32 of 457 MB) (device 64:0x2)
```

Booting an Inactive nPartition past Boot-Is-Blocked (BIB)

When all cells in an nPartition are at boot-is-blocked, the nPartition is *inactive*. This is the case, for example, when an nPartition is held at the ready for reconfig state.

You can boot an nPartition past the ready for reconfig state to make it active by using the service processor Command menu's `BO` (boot) command.

To determine whether an nPartition is in a boot-is-blocked (ready for reconfig) state, use the nPartition's Virtual Front Panel to monitor the nPartition's boot activity. If all of the nPartition's cells are at boot-is-blocked, the nPartition is halted at the ready for reconfig state.

Booting an Inactive nPartition past BIB [Service Processor]

Use the service processor Command menu **BO** command to boot an nPartition past the ready for reconfig state to make the nPartition active.

If you use the Command menu's `BO` command to attempt to boot an nPartition that already is active, the command has no effect.

Step 1. Login to the server complex's service processor and enter **CM** to select the Command menu.

```
# telnet sdome-s
Trying...
Connected to sdome-s.rsn.hp.com.
Escape character is '^]'.
Local flow control off

GSP login: Accountname
GSP password: Password

....

GSP> CM

                Enter HE to get a list of available commands


GSP:CM>
```

Step 2. From the Command menu, enter the **BO** command and specify which nPartition is to be booted (released from boot-is-blocked).

As a result of the BO command, the complex's service processor releases the selected nPartition's cells from boot-is-blocked: the cells proceed to rendezvous to form an active nPartition, which no longer is in the ready for reconfig state.

```
GSP:CM> BO

This command boots the selected partition.


     #   Name
    ---  ----
     0)  jules00
     1)  jules01

    Select a partition number: 0

    Do you want to boot partition number 0? (Y/[N]) y

    -> The selected partition will be booted.
GSP:CM>
```

Any of the nPartition's cells that are not configured (those with a "n" use-on-next-boot value) remain inactive at boot-is-blocked.

When the nPartition becomes active it proceeds through the normal boot process and performs, as necessary, the boot action set for each of the boot paths (PRI, HAA, ALT).

Configuring Boot Paths and Boot Actions

You can configure each nPartition's boot *paths* (device paths for booting HP-UX) and boot *actions* (preferred automatic boot behavior) by using the following procedures:

- *Configuring Boot Paths and Actions [BCH]* on page 229
- *Configuring Boot Paths and Actions [HP-UX]* on page 230

By configuring boot paths and boot actions for an nPartition, you can set the nPartition to automatically boot from a primary source or, if the primary source fails, from backup devices.

Each nPartition's **boot device paths** list the hardware paths of devices for booting HP-UX on the nPartition.

The boot paths are:

- PRI—Primary boot path.
- HAA—High-availability alternate boot path, typically a mirror of the primary root volume.
- ALT—Alternate boot path. Typically used for install or recovery media (such as DAT or CD-ROM drive).

Each nPartition also has a set of **boot actions (path flags)**, which specify the *default actions to be automatically performed* when the nPartition boots to the BCH interface. Each of the three boot paths (PRI, HAA, and ALT) has its own path flag setting that defines its boot action.

The order in which an nPartition's boot actions are attempted is: PRI boot action, then HAA boot action (if necessary), and finally ALT boot action (if necessary).

The boot actions (path flag settings) for each boot path are:

- 0—Go to BCH.
- 1—Boot this path, if fail go to BCH.
- 2—Boot this path, if fail attempt to perform the next path's boot action.
- 3—Skip this path, attempt to perform the next path's boot action.

By default, all path flags are set to 0 ("Go to BCH").

The boot actions are performed *automatically* by the BCH interface when an nPartition boots to BCH, as possible and necessary. However, boot action settings *do not* affect the behavior of the BCH `BOOT` command.

Setting Autoboot through Boot Paths and Boot Actions

Each nPartition's **Autoboot setting** is established by the boot action (path flag) settings for the nPartition's boot paths.

For an nPartition to *automatically boot HP-UX*, it must be configured in the following way:

- The nPartition must have at least one bootable HP-UX device that is pointed to by the PRI, HAA, or ALT boot path variable.
- The path flag (boot action) setting for a bootable device's path variable must be set to "boot this path" (1 or 2).
- When the nPartition boots it must proceed to execute a bootable device's boot action that specifies to "boot this path", and it must find the device.

For example, an nPartition could automatically boot HP-UX with the following configuration: both the PRI and HAA paths point to bootable devices, and the PRI action is 2 ("boot this path, if fail attempt to perform the next path's boot action") and the HAA action is 1 ("boot this path, if fail go to BCH").

In this example configuration, the nPartition could automatically boot HP-UX even if the PRI path were not available. When the nPartition boots to BCH it first attempts to boot the PRI device. If the PRI device cannot be booted, because the PRI path flag specifies to "if fail attempt to perform the next path's boot action", it then refers to the HAA path and action. Because in this example the HAA path points to a bootable device, and because the HAA path flag specifies to attempt to boot the HAA device, the nPartition can still automatically boot HP-UX (if the HAA device is available).

Configuring Boot Paths and Actions [BCH]

Use the BCH Main menu **PATH** command and Configuration menu **PATHFLAGS** command to configure an nPartition's boot paths and boot actions (path flags) through its BCH interface.

To list all boot path and action settings for an nPartition, you also can use the BCH Information menu's `BOOTINFO` command.

Step 1. Determine which devices will be used for booting HP-UX on the nPartition, and determine the boot behaviors you desire.

You need to determine the hardware paths of all potential boot devices that you will configure as the PRI, HAA, and ALT boot paths.

You also need to determine which device you want to boot by default (if any), and which (if any) device you want to boot if the default device fails to boot.

Typically, the PRI path is set to the default boot device and the HAA path is set to the device you want to boot if PRI fails to boot.

Step 2. Login to the service processor (GSP or MP), access the nPartition's console, and access the BCH Main menu.

From the nPartition console you access the nPartition's BCH interface. If the nPartition is not at the BCH interface you must either boot the nPartition or shut down HP-UX to return to the BCH interface.

Step 3. At the BCH Main menu, set the boot path values using the **PATH** command.

To list the current boot path settings, enter **PATH** with no arguments.

To set a boot path, enter **PATH *VAR hwpath***, where *VAR* is the boot path variable (PRI, HAA, or ALT) and *hwpath* is a boot device's hardware path.

For example, to set the PRI boot path to a new value (4/0/2/0/0.10, in this case) enter `PATH PRI 4/0/2/0/0.10`, as shown below.

```
Main Menu: Enter command or menu > PATH PRI 4/0/2/0/0.10.0

     Primary Boot Path:  4/0/2/0/0.10
                         4/0/2/0/0.a    (hex)

Main Menu: Enter command or menu >
```

Step 4. Access the BCH Configuration menu by entering **`CO`** at the Main menu, and set the boot action for each boot path, as desired, by using the **`PATHFLAGS`** command.

At the BCH Configuration menu, you can list the path flags (boot actions) for all boot path variables by entering **`PATHFLAGS`** with no arguments.

To set the boot action for each of the boot paths, enter **`PATHFLAGS`** ***`VAR action`***, where *`VAR`* is the boot path variable (PRI, HAA, or ALT) and *`action`* is the boot action (0 for "go to BCH", 1 for "boot, if fail go to BCH", 2 for "boot, if fail try next path", or 3 for "skip this path, try next path").

For example, to configure an nPartition to boot from the PRI device or (if PRI fails to boot) the HAA device, use the following two BCH Configuration commands: **`PATHFLAGS PRI 2`** and **`PATHFLAGS HAA 1`**, as shown below.

```
Configuration Menu: Enter command > PATHFLAGS PRI 2

     Primary Boot Path Action
          Boot Actions:  Boot from this path.
                         If unsuccessful, go to next path.

Configuration Menu: Enter command > PATHFLAGS HAA 1

HA Alternate Boot Path Action
          Boot Actions:  Boot from this path.
                         If unsuccessful, go to BCH.

Configuration Menu: Enter command >
```

For other help in setting path flags, enter `HELP PATHFLAGS` at the BCH Configuration menu prompt.

Configuring Boot Paths and Actions [HP-UX]

Use the **`parmodify -p#...`** and **`setboot...`** commands to set nPartition boot path variables from HP-UX and to check and set the local nPartition's PRI boot action (the PRI path flag).

Step 1. Determine which devices will be used for booting HP-UX on the nPartition, and determine the boot behaviors you desire for the PRI boot path.

Step 2. Login to HP-UX running on an nPartition in the complex.

You can modify the boot paths for any nPartition from any other nPartition in the complex when using the `parmodify` command.

However, when using the `setboot` command to modify the PRI and ALT paths or the PRI boot action, you can modify only the *local* nPartition's settings.

Step 3. Configure boot path settings using the **`parmodify -p#...`** command.

Use the following commands to set the boot path variables for a specified partition number (`-p#`):

- PRI path— `parmodify -p# -b` *`PRI`* where *`PRI`* is the hardware path.
- HAA path—`parmodify -p# -s` *`HAA`* where *`HAA`* is the hardware path.
- ALT path—`parmodify -p# -t` *`ALT`* where *`ALT`* is the hardware path.

If using the `setboot` command to set boot paths for the *local nPartition*, you can specify `setboot -p` *`PRI`* or `setboot -a` *`ALT`* but cannot set the HAA path variable.

You can list an nPartition's current boot path settings by issuing the `parstatus -V -p# | grep Path` command and specifying the partition number (`-p#`). The `setboot` command with no arguments lists the PRI and ALT settings for the local nPartition as well as the local nPartition's PRI path flags (boot actions).

For example, to set the PRI boot path to 0/0/4/0/0.8.0 and the HAA boot path to 0/0/4/0/0.9.0 for partition number 0, issue the `parmodify -p0 -b 0/0/4/0/0.8.0 -s 0/0/4/0/0.9.0` command, as shown below.

```
# parmodify -p0 -b 0/0/4/0/0.8.0 -s 0/0/4/0/0.9.0
Command succeeded.
#
```

Step 4. As needed, configure the PRI boot action for the *local nPartition* by using the **`setboot -b`** ***`Autoboot`*** **`-s`** ***`Autosearch`*** command.

The `setboot` command supports the following options for setting local nPartition boot actions:

`-b` **Autoboot setting** for the local nPartition:

`-b on` to automatically boot the PRI path.

`-b off` to not boot PRI.

`-s` **Autosearch setting** for the local nPartition:

`-s on` to attempt to perform the HAA path's boot action when PRI is not booted (either when `-b` is `off`, or when PRI fails to boot when `-b` is `on`).

`-s off` to never attempt to perform the HAA action.

For example, to always stop the local nPartition at BCH when booting, issue the `setboot -b off -s off` command.

See *Setting Autoboot through Boot Paths and Boot Actions* on page 228 or the *setboot* (1M) manpage for details.

Configuring Autoboot and Autostart

The **Autoboot setting** specifies whether an nPartition automatically boots HP-UX. You can configure each nPartition's Autoboot setting by modifying the nPartition's boot actions for its boot paths. See the *Autoboot Configuration* section.

On HP Superdome servers only, you can configure an **Autostart setting** for each nPartition to specify the nPartition's boot behavior when one or more self tests fails. See the *Autostart Configuration* section that follows.

Autoboot Configuration

Each nPartition's Autoboot setting is established by a combination of its boot path variable settings and the settings for each path's boot actions (determined by its path flags).

You can use the BCH Main menu's `PATH` command and the BCH Configuration menu's `PATHFLAGS` command to set boot paths and boot actions for an nPartition. You also can use the `parmodify` and `setboot` HP-UX commands to configure some of the boot path and action settings.

See *Configuring Boot Paths and Boot Actions* on page 227 for details on configuring these settings to enable Autoboot.

Autostart Configuration

On HP Superdome servers only, the BCH interface's Autostart setting for each nPartition determines the boot behavior when one of the nPartition's components (processors or memory) fails self test.

By default Autostart is set to OFF, and the nPartition stops at the BCH interface when a processor or DIMM fails self-test.

When Autostart is ON, the nPartition proceeds with the normal boot process and performs the boot actions for its boot paths as necessary.

Configuring Superdome nPartition Autostart [BCH]

Use the BCH Configuration menu's **AU** command to configure Autostart for an nPartition on an HP Superdome server.

NOTE This procedure applies to nPartitions on HP Superdome servers only.

Step 1. Login to the Superdome complex's service processor (GSP), access the nPartition's console, and access the BCH Configuration menu.

From the nPartition console, you can access the nPartition's BCH interface. If the nPartition is not at the BCH interface you must either boot the nPartition or shut down HP-UX to return to the BCH interface.

From the BCH Main menu, enter **CO** to access the Configuration menu.

Step 2. From the BCH Configuration menu, use the **AU** command to list or set Autostart for the nPartition.

Enter **AU** with no arguments to list the Autostart setting.

Enter **AU ON** to set Autostart to ON, or **AU OFF** to set it to OFF.

Configuring Automatic System Restart for an nPartition

The **automatic system restart** feature on nPartition servers enables you to configure an nPartition to be automatically rebooted when HP-UX hangs on the nPartition.

By default, automatic system restart *is disabled* for nPartitions.

To enable or disable automatic system restart, use the service processor Command menu's AR command, as described in *Configuring nPartition Automatic System Restart [Service Processor]* on page 236.

To use the AR command, you must login to the server complex's service processor using an account that has administrator authority.

CAUTION When automatic system restart is enabled for an nPartition, all cells in the nPartition automatically will be reset—and the nPartition will reboot—if HP-UX running on the nPartition is hung for three (3) minutes.

When HP-UX is booted on an nPartition, it indicates that it is "alive" by emitting a HEARTBEAT chassis code and an ACTIVITY_LEVEL_TIMEOUT chassis code approximately every four seconds.

The service processor manages automatic system restart for each nPartition through a timer that tracks the time since the nPartition was active. This timer is reset every time an ACTIVITY_LEVEL_TIMEOUT chassis code is emitted by HP-UX on the nPartition. If HP-UX does not emit this chassis code for three minutes then it emits an "Alert Level 13: System hang detected" chassis code. If the nPartition has automatic system restart enabled then the service processor issues a PARTITION_TIMEOUT_RESET chassis code, resets all cells assigned to the nPartition, and the nPartition reboots.

The following output shows the chassis codes (with keywords) for an HP-UX timeout and automatic reset.

```
129  GSP  0     *13 0x591008d1a000205f 0x000065060c0f1611 PARTITION_TIMEOUT_RESET
128  HPUX 0,0,0 *13 0x78e004d41100f000 0x0000000300000009
128  HPUX 0,0,0 *13 0x58e00c000000f000 0x000065060c0f1610 07/12/2001 15:22:16
```

Monitoring HP-UX Activity and Chassis Logs

You can monitor whether HP-UX is active on an nPartition through the nPartition's Virtual Front Panel and through the Chassis Logs viewer.

- You can track an nPartition's HP-UX activity through its Virtual Front Panel (VFP) display, which is available through the service processor. When HP-UX has booted on an nPartition, the nPartition's VFP blinks an HP-UX heartbeat indicator based on the HEARTBEAT chassis code.
- You also can track HP-UX activity though the service processor's Chassis Logs viewer, which enables you to view live (real-time) chassis codes as well as previously recorded error and activity chassis codes.

  For example, to monitor an nPartition's chassis codes in real time: from the service processor Main menu select `SL` for the Chassis Logs viewer, select the live chassis logs option, then type `P` and select which nPartition's chassis codes you want to monitor (to exit to the Main menu type **`^b`**).

Configuring nPartition Automatic System Restart [Service Processor]

Use the service processor Command menu's **`AR`** command to enable or disable automatic system restart for an nPartition.

Step 1. Login to the server complex's service processor (GSP or MP) and enter **`CM`** to access the Command menu.

Step 2. Issue the service processor Command menu's **`AR`** command to enable or disable automatic system restart for an nPartition.

To use the `AR` command, you must be logged in using an account that has administrator authority.

```
GSP:CM> AR

This command modifies the automatic system restart configuration of
the selected partition.
```

```
     #   Name
    ---  ----
     0)  feshd5a
     1)  feshd5b

    Select a partition number: 0

    Automatic system restart for partition 0 is currently enabled.
    Do you want to disable automatic system restart? (Y/[N]) y

    -> Automatic system restart is disabled.
GSP:CM>
```

Configuring Fast Boot Settings (Self Tests) for an nPartition

The **fast boot settings** for an nPartition determine which self tests the nPartition performs during the power on or nPartition boot process.

You can configure nPartition fast boot settings by enabling and disabling various self tests using these procedures:

- *Configuring Fast Boot for an nPartition [BCH]* on page 239
- *Configuring Fast Boot for an nPartition [HP-UX]* on page 239

NOTE

HP recommends that all self tests be performed for nPartitions.

When an nPartition reboots due to a system panic, HPMC, or TOC, all self tests are performed when the nPartition reboots.

On HP nPartition servers you can configure the following self tests:

- **PDH tests**—Processor-dependent hardware tests that test a checksum of read-only memory.

  Can be configured from BCH and HP-UX `setboot` as “PDH”.

- **Early CPU tests**—Firmware, cache, and CPU-specific tests that are performed out of firmware.

  Can be configured from BCH (as “EARLY”) and HP-UX `setboot` (as “early_cpu”).

- **Late CPU tests**—Firmware, cache, and CPU-specific tests that are performed out of memory and thus are faster than early CPU tests.

  Can be configured from BCH (as “LATE”) and HP-UX `setboot` (as “late_cpu”).

Configuring Fast Boot for an nPartition [BCH]

Use the Configuration menu's `FASTBOOT` command to configure an nPartition's fast boot settings using its BCH interface.

Step 1. Login to the server complex's service processor (GSP or MP), access the nPartition's console, and access the BCH Configuration menu.

From the nPartition console you access the nPartition's BCH interface. If the nPartition is not at the BCH interface you must either boot the nPartition or shut down HP-UX to return to the BCH interface.

From the BCH Main menu, enter `CO` to access the Configuration menu.

Step 2. At the BCH Configuration menu use the **`FASTBOOT`** command to list or set the nPartition's fast boot settings.

Enter `FASTBOOT` with no arguments to display the current fast boot settings. This lists which self tests are set to be performed or skipped.

NOTE HP recommends that *all self tests* be performed for all nPartitions.

To *enable all tests* for an nPartition, use the **`FASTBOOT RUN`** command at the nPartition's BCH Configuration menu.

To *disable* an individual test, enter `FASTBOOT` *`test`* `SKIP`, where *`test`* is the name of the self test ("PDH", "EARLY", or "LATE").

To *enable* an individual test, enter `FASTBOOT` *`test`* `RUN`.

For details on setting self tests, enter `HELP FASTBOOT` at the Configuration menu.

Configuring Fast Boot for an nPartition [HP-UX]

Use the **`setboot`** command to configure an nPartition's self test configuration from HP-UX running on the nPartition.

Step 1. Login to HP-UX running on the nPartition whose self test configuration you want to change.

From HP-UX you can configure self tests for the *local* nPartition only.

Step 2. Enter the **`setboot -v`** command to list the current self test configuration for the local nPartition.

The self test details listed by `setboot -v` include:

TEST—The keyword names of self tests that you can enable or disable.
CURRENT—The nPartition's setting for the test in stable storage: `on` means the test is normally executed on each boot, `off` means the test is normally omitted on each boot, `partial` means some subtests normally are executed on each boot. This may differ from the NEXT BOOT settings.
SUPPORTED—Whether the server supports the test completely (`yes`), partially (`partial`), or not at all (`no`).
DEFAULT—The default setting for the test, either `on`, `off`, or `partial`.
NEXT BOOT—The nPartition's self test behavior for the *next boot only*. If these settings differ from CURRENT, then the CURRENT settings are reestablished after the next boot.

The following example shows `setboot -v` output for an nPartition.

```
# setboot -v
Primary bootpath : 0/0/6/0/0.6.0
Alternate bootpath : 0/0/1/0/0.8.0

Autoboot is OFF (disabled)
Autosearch is OFF (disabled)

Note: The interpretation of Autoboot and Autosearch has changed for
systems that support hardware partitions. Please refer to the manpage.
TEST            CURRENT         SUPPORTED       DEFAULT         NEXT BOOT
----            -------         ---------       -------         ---------
all             partial         partial         partial         partial
  SELFTESTS     on              yes             on              on
    early_cpu   on              yes             on              on
    late_cpu    on              yes             on              on
  FASTBOOT      partial         partial         partial         partial
    full_memory off             no              off             off
    PDH         on              yes             on              on
  CEC           off             no              off             off
#
```

Step 3. Use the **`setboot...`** command to enable or disable boot-time self tests for the local nPartition.

You can use the following commands to configure tests:

`setboot -t` *`test_name`*`=[on|off|default]`

`setboot -T` *`test_name`*`=[on|off|default]` `test_name` is the name of the self test ("PDH", "early_cpu", "late_cpu") or is "all" (for all tests).

The `setboot` command's `-t` option changes the test setting in stable storage and affects all following boots. The `-T` option changes the test setting for the next boot only.

NOTE HP recommends that *all self tests* be performed for all nPartitions.

To *enable all tests* for an nPartition, use the following command:
`setboot -t all=on`

For example, to enable the early CPU tests and PDH tests but disable the late CPU tests issue the following command:

```
setboot -t early_cpu=on -t PDH=on -t late_cpu=off
```

This changes the local nPartition's settings for these tests in its stable storage and uses these test configurations for all following boots.

After modifying an nPartition's self test configuration, you can list the new settings with the `setboot -v` command.

For more details see the *setboot* (1M) manpage.

Boot Timer Configuration for an nPartition

The boot timer setting establishes the number of seconds an nPartition will wait for a boot device before timing out.

When a boot device does not respond to a boot request within the number of seconds defined by the boot timer setting, the boot is considered unsuccessful.

Configuring an nPartition Boot Timer [BCH]

Use the Configuration menu's **`BOOTTIMER`** command to configure an nPartition's boot timer setting from its BCH interface.

Step 1. Login to the server complex's service processor (GSP or MP), access the nPartition's console, and access the BCH Configuration menu.

From the nPartition console, you access the nPartition's BCH interface. If the nPartition is not at the BCH interface, you must either boot the nPartition or shut down HP-UX to return to the BCH interface.

From the BCH Main menu, enter `CO` to access the Configuration menu.

Step 2. From the BCH Configuration menu, use the **`BOOTTIMER`** command to list or set the boot timer setting.

Enter `BOOTTIMER` with no arguments to list the current setting.

Enter `BOOTTIMER` *seconds* to set the boot timer setting to the specified (*seconds*) number of seconds.

6 Managing nPartitions

This chapter presents the procedures for creating, configuring, and managing nPartitions on HP servers that support them.

For an introduction to nPartition features, refer to the chapter *nPartition System Overviews* on page 31.

For nPartition configuration requirements and related HP recommendations, refer to the chapter *Planning nPartition Configurations* on page 109.

Coordinating Changes to nPartitions

When you modify nPartitions, you should perform your changes so that they occur at a time when they will not conflict with other nPartition changes in the same server complex.

The HP-UX nPartition configuration commands and the Partition Manager utility coordinate their actions by using *locks* to restrict access to portions of the server's *Complex Profile* data while they modify that data.

Commands and utilities can lock Stable Complex Configuration Data and Partition Configuration Data to ensure that only the command holding the lock can modify that portion of the Complex Profile.

In most cases, the nPartition commands and utilities will not have locking conflicts because they can complete changes quickly (within about a second), usually before any other commands attempt to modify the same portion of the server's Complex Profile data.

However, some nPartition changes involve locking parts of the Complex Profile for a longer time.

- The Partition Manager utility locks all parts of the server's Complex Profile that it may potentially modify, and it keeps them locked for as long as the associated menu items are being used.

  For example, the **Partition —> Create Partition** menu item and the **Complex —> Set Complex Name** menu item lock the Stable Complex Configuration Data portion of the server's Complex Profile. You cannot use other tools to perform any tasks that modify the Stable Complex Configuration Data (such as adding or removing cells) until the task that acquired the lock completes or is canceled.

  Likewise, the **Partition —> Modify Partition** menu item locks both the Stable Complex Configuration Data as well as the selected nPartition's Partition Configuration Data. As a result, no other tool (including another instance of Partition Manager) can add or remove cells or modify the selected nPartition until this task has completed or been canceled.

  Some tasks performed through Partition Manager also require performing a reboot for reconfig to release locks (for example, removing an active cell from an nPartition).

- When *removing an active cell* from an nPartition, you must perform a reboot for reconfig (`shutdown -R`, not `-r`) of the modified nPartition to release the Complex Profile lock, regardless of whether you use `parmodify` or Partition Manager.

  You must perform the reboot for reconfig before you can add or remove other cells from nPartitions in the server complex. (The lock on the Stable Complex Configuration Data is not released in this case until the reboot for reconfig has occurred.)

- When you *add or remove cells from an active nPartition* and specify the `parmodify` command's `-B` option, the Stable Complex Configuration Data remains locked until the modified nPartition has performed a reboot for reconfig. In this situation, no further changes to cell assignments can occur until after the reboot for reconfig.

Although you can use the `parunlock` command to force-unlock any parts of a server's Complex Profile, you *should not use* this command unless the program that established the lock has abnormally terminated. Instead, if possible, you should allow the Complex Profile to be unlocked as part of the normal procedures described above. See the *parunlock* (1M) manpage for details.

Rebooting to Implement nPartition Changes

Once an nPartition has booted and is active, the nPartition has a fixed set of active hardware resources. In order to establish a different set of active hardware resources for an nPartition you must reboot the nPartition, as described below.

You can add and remove cells from an active, booted nPartition; however, you only can add or remove *inactive* cells without having to reboot the nPartition.

To remove an active cell from an nPartition, or to make a newly added cell or inactive cell *active*, you must perform a reboot for reconfig of the nPartition.

The following list describes situations where you may need to reboot an nPartition to implement changes.

- Perform a reboot for reconfig (**`shutdown -R`**) of an nPartition in the following situations.

  — When you want to add one or more cells to an nPartition.

  Newly added cells initially are inactive when assigned to an nPartition. To allow the new cells to rendezvous (join the nPartition as active members), perform a reboot for reconfig.

  — When you remove one or more cells from an nPartition.

  Removing an active cell requires an nPartition reboot for reconfig, but removing an *inactive* cell *does not* require an nPartition reboot for reconfig. Inactive cells are removed immediately.

  — When you change a cell's use-on-next-boot value from "n" (no, do not use) to "y" (yes, use the cell).

  A reboot for reconfig permits the cell to rendezvous into the nPartition and become active; see below.

  — When you want to allow a currently inactive cell to become active.

  A reboot for reconfig reboots all cells, allowing them an opportunity to join (rendezvous) the nPartition as active members.

- Reset an nPartition to the ready for reconfig state (**`shutdown -R -H`**) to make the nPartition inactive.

  All cells in an nPartition remain inactive when the nPartition is in the ready for reconfig state; the cells do not perform a partition rendezvous.

- Perform a standard reboot (**`shutdown -r`**) of an nPartition in most other situations where you do not need to add or remove cells from the nPartition.

  A standard reboot causes only the *currently active* cells in an nPartition to reboot, and it does not allow any pending complex configuration changes to complete (the changes remain pending, still requiring a reboot for reconfig for them to be in effect).

  Pending changes that require a reboot for reconfig (`shutdown -R`, not a `shutdown -r`) include removing an active cell from an nPartition. The cell cannot be unassigned until its nPartition has a reboot for reconfig performed.

  Other changes, such as adding a cell to an nPartition or changing a cell's use-on-next-boot value from "n" to "y", also require performing a reboot for reconfig (`shutdown -R`, nor `-r`) to enable the inactive cell to become active.

Listing the Local (Current) Partition Number

Each nPartition within a server complex has a unique number assigned to it. This **partition number** identifies the nPartition in various menus, commands, and utilities. You also can specify the partition number when performing operations on an nPartition, such as adding or removing cells or resetting an nPartition.

You can list the local partition number by using the following procedures:

- *Listing the Local nPartition Number [BCH]* on page 248
- *Listing the Local nPartition Number [HP-UX]* on page 248

Listing the Local nPartition Number [BCH]

Use the Configuration menu **PD** command to list the local partition number from the BCH interface.

Step 1. Access the Boot Console Handler (BCH) interface for the nPartition, and access the BCH Configuration menu.

Enter `CO` from the BCH Main menu to access the Configuration menu. If you are at a BCH menu other than the Main menu, enter `MA` to access the Main menu.

Step 2. From the BCH Configuration menu, enter the **PD** command to list the local nPartition's name and partition number.

```
Configuration Menu: Enter command > PD

Partition Number: 1
Partition Name: jules01

Configuration Menu: Enter command >
```

Listing the Local nPartition Number [HP-UX]

Use the **`parstatus -w`** command to list the partition number of the local (current) nPartition from HP-UX.

Step 1. Login to HP-UX running on the nPartition.

Step 2. Issue the **parstatus -w** command to list the partition number for the local nPartition.

```
# parstatus -w
The local partition number is 0.
#
```

The `parstatus -P` command lists all nPartitions within the server complex, including the local nPartition.

```
# parstatus -P
[Partition]
Par              # of  # of I/O
Num Status       Cells Chassis  Core cell  Partition Name (first 30 chars)
=== ============ ===== ======== ========== ===============================
 0  active         2      2     cab0,cell0 jules00
 1  active         2      2     cab0,cell4 jules01
#
```

Listing All Configured nPartitions

You can configure each server complex to have multiple nPartitions, which are composed of cells in the complex.

You can list all configured nPartitions in the server complex by using the following procedures:

- *Listing All nPartitions [Service Processor]* on page 250
- *Listing All nPartitions [HP-UX]* on page 251
- *Listing All nPartitions [Partition Manager]* on page 252

Listing All nPartitions [Service Processor]

Use the Command menu **CP** command to list all nPartitions in a server complex from the complex's service processor.

Step 1. Login to the service processor for the complex and enter **CM** to access the Command menu.

```
# telnet sdome-s
Trying...
Connected to sdome-s.rsn.hp.com.
Escape character is '^]'.
Local flow control off

GSP login: Accountname
GSP password: Password

....

GSP> CM
                Enter HE to get a list of available commands
GSP:CM>
```

Step 2. From the service processor Command menu, enter the **CP** command to list all configured nPartitions within the server complex.

The CP command lists each nPartition (by partition number) and indicates which cells from each cabinet are assigned to the nPartition.

In the following example the complex has two nPartitions: partition number 0 has cells 0 and 2, and partition number 1 has cells 4 and 6.

```
GSP:CM> CP

-------------------------------------------------------------------------------
Cabinet |   0    |   1    |   2    |   3    |   4    |   5    |   6    |   7
--------+--------+--------+--------+--------+--------+--------+--------+--------
 Slot   |01234567|01234567|01234567|01234567|01234567|01234567|01234567|01234567
--------+--------+--------+--------+--------+--------+--------+--------+--------
Part  0 |X.X.....|........|........|........|........|........|........|........
Part  1 |....X.X.|........|........|........|........|........|........|........

GSP:CM>
```

To the right of each partition number is a list of cells assigned to the nPartition. Assigned cells are marked with an "X". The cell's slot (0 to 7) and its cabinet number (0 or above) are listed above each cell.

You also can use the **DU** command to list *all* cells in a server complex (and other complex hardware details), including unassigned cells.

Listing All nPartitions [HP-UX]

Use the **parstatus -P** command (and **parstatus -C**, for more details) to list information about all nPartitions in a server complex from HP-UX.

From any nPartition in a complex, you can list details about all cells and nPartitions within the complex.

Step 1. Login to HP-UX running on any of the server complex's nPartitions.

You can login to HP-UX on the nPartition either by connecting with `telnet` or `rlogin`, or by logging in to its complex's service processor and accessing the nPartition's console.

Step 2. Issue the **parstatus -P** command to list brief details about all nPartitions in the server complex.

The `parstatus -P` command lists all nPartitions and shows each nPartition's number and name, the number of cells assigned to it, the number of active I/O chassis, and the nPartition's active core cell.

```
# parstatus -P
[Partition]
Par                # of  # of I/O
Num Status         Cells Chassis  Core cell  Partition Name (first 30 chars)
=== ============ ===== ======== ========== ==============================
```

```
 0  active         2      2     cab0,cell0 jules00
 1  active         2      2     cab0,cell4 jules01
#
```

While an nPartition is booting, the `parstatus` command cannot determine the nPartition's I/O chassis and core cell information. When this is the case `parstatus` does not count the I/O chassis and reports a question mark (`?`) for the core cell. When the nPartition has completed booting, `parstatus` reports all details.

Step 3. To list detailed information about all cells and nPartitions in the server complex, issue the **`parstatus -C`** command.

The `parstatus -C` command presents more detailed information about all cells and nPartitions. These details include each cell's status (active, inactive), its processor and memory configuration, its I/O chassis connections (if any), the cell's use-on-next-boot setting, and nPartition assignment.

```
# parstatus -C
[Cell]
                        CPU     Memory                                Use
                        OK/     (GB)                          Core    On
Hardware   Actual       Deconf/ OK/                           Cell    Next Par
Location   Usage        Max     Deconf    Connected To        Capable Boot Num
========== ============ ======= ========= =================== ======= ==== ===
cab0,cell0 active core  4/0/4    2.0/ 0.0 cab0,bay0,chassis1  yes     yes  0
cab0,cell1 absent       -       -         -                   -       -    -
cab0,cell2 active base  4/0/4    2.0/ 0.0 cab0,bay1,chassis3  yes     yes  0
cab0,cell3 absent       -       -         -                   -       -    -
cab0,cell4 active core  4/0/4    2.0/ 0.0 cab0,bay0,chassis3  yes     yes  1
cab0,cell5 absent       -       -         -                   -       -    -
cab0,cell6 active base  4/0/4    2.0/ 0.0 cab0,bay1,chassis1  no      yes  1
cab0,cell7 absent       -       -         -                   -       -    -

#
```

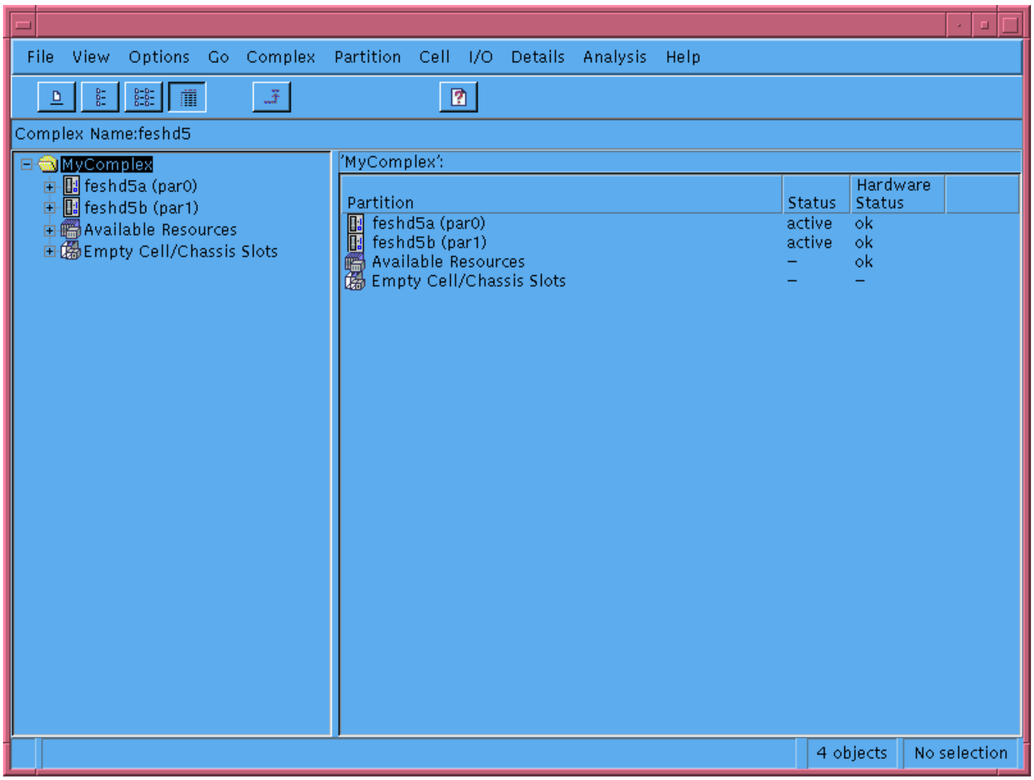
For cells and nPartitions that have not finished booting, the `parstatus` command cannot determine processor, memory, or I/O details and instead reports a question mark (`?`) for these details.

Listing All nPartitions [Partition Manager]

View the left side of Partition Manager primary window to see a list of all nPartitions in a server complex using Partition Manager.

Step 1. Run Partition Manager (`/opt/parmgr/bin/parmgr`) or access it from SAM or a Web browser.

Step 2. When the Partition Manager starts up, the left side of its primary display lists the nPartitions in the complex.



The right side of the primary display also lists the complex's nPartitions when nothing is selected on the display's left side, or when **My Complex** is selected.

Listing Cell nPartition Assignments

Each cell in an nPartition server complex either is assigned to an nPartition, or it is unassigned (if it is an available resource).

You can list all cells and their nPartition assignments by using these procedures:

- *Listing Cell nPartition Assignments [Service Processor]* on page 254
- *Listing Cell nPartition Assignments [HP-UX]* on page 254
- *Listing Cell nPartition Assignments [Partition Manager]* on page 255

Listing Cell nPartition Assignments [Service Processor]

Use the Command menu's **CP** and **DU** commands to list all cell nPartition assignments (and other details) from the server complex's service processor.

Step 1. Login to the service processor for the complex and enter **CM** to access the Command menu.

Step 2. Enter the **CP** command to list all configured nPartitions in the server complex.

Step 3. Enter the **DU** command to list additional details (such as available core I/O) for the cells assigned to the various nPartitions in the server complex.

On HP Superdome servers, you also can use the service processor Command menu's **IO** command to list cell-to-I/O chassis connections.

Listing Cell nPartition Assignments [HP-UX]

Use the **parstatus -C** command to list all cells in a server complex and their nPartition assignments.

Step 1. Login to HP-UX running on one of the server complex's nPartitions.

Step 2. Issue the **parstatus -C** command to list all cells, any I/O chassis connections, and any nPartition assignments for the cells.

In addition to reporting the cell nPartition assignments (listed in the "Par Num" column), the `parstatus -C` command reports each cell's current status (absent, inactive, active core, active base) in the "Actual Usage" column.

```
# parstatus -C
[Cell]
                        CPU     Memory                                Use
                        OK/     (GB)                          Core    On
Hardware   Actual       Deconf/ OK/                           Cell    Next Par
Location   Usage        Max     Deconf    Connected To        Capable Boot Num
========== ============ ======= ========= =================== ======= ==== ===
cab0,cell0 active core  4/0/4    2.0/ 0.0 cab0,bay0,chassis1  yes     yes  0
cab0,cell1 absent       -       -         -                   -       -    -
cab0,cell2 active base  4/0/4    2.0/ 0.0 cab0,bay1,chassis3  yes     yes  0
cab0,cell3 absent       -       -         -                   -       -    -
cab0,cell4 active core  4/0/4    2.0/ 0.0 cab0,bay0,chassis3  yes     yes  1
cab0,cell5 absent       -       -         -                   -       -    -
cab0,cell6 inactive     4/0/4    2.0/ 0.0 cab0,bay1,chassis1  no      -    -
cab0,cell7 absent       -       -         -                   -       -    -

#
```

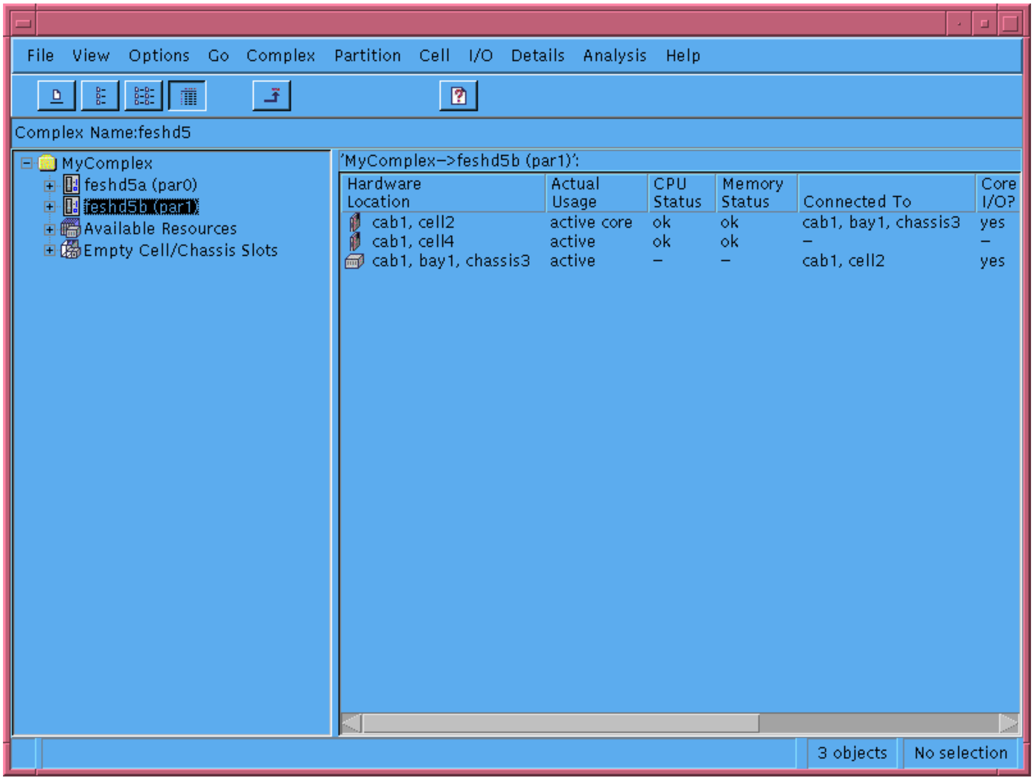
For cells that are not assigned to an nPartition, `parstatus -C` lists a hyphen (`-`) in the "Par Num" column instead of the cell's partition number. The "Connected To" column lists any I/O chassis connections for the cells, and "Core Cell Capable" lists whether core I/O is available through each the cell's I/O chassis.

Listing Cell nPartition Assignments [Partition Manager]

Select each nPartition and Available Resources on the left side of the primary window to view all cell nPartition assignments in a server complex from Partition Manager.

Step 1. Run Partition Manager (`/opt/parmgr/bin/parmgr`) or access it from SAM or a Web browser.

Step 2. At the Partition Manager primary window, select each nPartition on the left side of the window to list the nPartition's cell assignments on the window's right side, or select Available Resources to list all unassigned cells.



On the right side of the primary window, for each selected nPartition, Partition Manager lists the cell assignments and any I/O chassis connected to the cells.

Creating a Genesis Partition

When you create a Genesis Partition, you establish a one-cell nPartition on the server complex. The Genesis Partition replaces all other nPartitions, and once created it is the only nPartition in the server.

The only way to create a Genesis Partition is to use the service processor Command menu's **CC** command on the server complex.

Genesis Partition Creation [Service Processor]

Use the service processor Command menu's **CC** command and **G** option to create a Genesis Partition on an HP nPartition server.

As a result of this procedure, *all existing nPartitions are destroyed* and are replaced with a single, one-cell nPartition (the Genesis Partition).

You can revert to the previous nPartition configuration—if any existed before you created the Genesis Partition—by using the `CC` command's `L` option to restore the last configuration.

Step 1. Save all current nPartition configuration details, if any nPartitions are configured in the complex.

Saving the current nPartition information provides you the details you would need to re-create all nPartitions as they currently exist.

Use the `parstatus -V -p`*#* HP-UX command (or an equivalent `parmgr` procedure) to save configuration details about each nPartition.

For each nPartition, enter the `parstatus -V -p`*#* command to display detailed information about the partition number (`-p`*#*) specified.

Step 2. Determine which cell will be configured as the Genesis Partition.

The cell must be connected to an I/O chassis. The I/O chassis must have a core I/O card installed, and it should have a bootable HP-UX disk (or a method for installing HP-UX and a disk onto which it can be installed).

Step 3. Ensure that all nPartitions within the complex are in the ready for reconfig (inactive) state.

If an nPartition is running HP-UX, you can shut down the nPartition to the ready for reconfig state by using the `shutdown -R -H` command.

Or, you can put an nPartition into the ready for reconfig state by using the BCH interface's `RECONFIGRESET` command or using the service processor Command menu's `RR` command.

Step 4. Login to the server complex's service processor (GSP or MP).

Login as a user with administrator privileges, which are required for creating a Genesis Partition.

Step 5. Enter **`CM`** to access the service processor Command menu.

Step 6. Issue the **`CC`** command, select **`G`** for Genesis Complex Profile, and specify the cabinet and cell slot for the cell that will comprise the Genesis Partition.

```
GSP:CM> CC


This command allows you to change the complex profile.

WARNING: You must shut down all Protection Domains before
executing
         this command.


G - Genesis Complex Profile
L - Last Complex Profile
    Select Profile: g

    Enter Cabinet number: 0

    Enter Slot number: 0

    Do you want to modify the complex profile? (Y/[N]) y

    -> The complex profile will be modified.
GSP:CM>
```

You can confirm that the Genesis Partition was successfully created if the `CC` command reports that the "complex profile will be modified".

If the `CC` command reports "Sorry, command failed", then the Genesis Partition was not created, possibly because one or more nPartitions are not at the ready for reconfig state. If this is the case, go back to *Step 3* and ensure all nPartitions are inactive at the ready for reconfig state.

Step 7. Issue the **BO** command to boot the Genesis Partition past its ready for reconfig state and make it an active nPartition.

When a Genesis Partition is created, it remains at boot-is-blocked (in an inactive, ready for reconfig state), so you must boot it manually.

The Genesis Partition always is assigned partition number 0, because when it is created it is the first and only nPartition in the server complex.

Using the BO command to boot partition 0 will boot the Genesis Partition to its Boot Console Handler (BCH) interface.

```
GSP:CM> BO

This command boots the selected partition.

     #   Name
    ---  ----
     0)  Partition 0

    Select a partition number : 0

    Do you want to boot partition number 0,
    named Partition 0 ? (Y/[N]) y

    -> The selected partition will be booted.
GSP:CM>
```

Step 8. Access the Genesis Partition's console and configure the nPartition as appropriate and necessary.

From the service processor Command menu, enter **MA** to return to the Main menu, then enter **CO** to access the Console menu. The Genesis Partition is partition 0 and by default is named "Partition 0".

You will need to set the boot paths (PRI, ALT, and HAA), any core cell choices, the nPartition name, and other settings as appropriate. You also may need to add cells to the Genesis Partition if you want it to have more than one cell.

Creating a New nPartition

In a server complex, you can create multiple nPartitions if the server has enough cells and core I/O to support the nPartitions.

You can create a new nPartition by using the following procedures:

- *Creating a New nPartition [HP-UX]* on page 260
- *Creating a New nPartition [Partition Manager]* on page 264

At least one cell in each nPartition must be connected to an I/O chassis that has core I/O attached. To boot HP-UX the nPartition also must have a boot device and any required PCI cards and devices installed.

When creating an nPartition, you should adhere to the HP nPartition requirements and guidelines. HP recommends only specific sets of nPartition configurations.

If no nPartitions exist in a server complex, you must first establish a Genesis Partition before creating other nPartitions.

Creating a New nPartition [HP-UX]

Use the **`parstatus`**, **`parcreate`**, and **`parmodify`** commands to create and configures a new nPartition from HP-UX.

This procedure uses `parstatus` to find available (unassigned) cells, uses `parcreate` to create an nPartition using the cells, and uses `parmodify` to modify the nPartition's settings and configure it for use.

One alternative to using this complete procedure is to replace steps 2–5 with a single `parcreate` command.

For example, the commands performed in steps 2–5 could be replaced with the following `parcreate` command line.

```
# parcreate -c4:base:y:ri -c6:base:y:ri -P "hostname05" -r0/4 \
> -r0/6 -b 4/0/1/0/0.9 -B
Partition Created. The partition number is : 1
#
```

In the above alternative command line, the `-B` option is specified and causes the nPartition to be booted past boot-is-blocked immediately, thus making the new nPartition active. (It is booted to its BCH interface.)

Step 1. Login to HP-UX running on an existing nPartition in the server complex, and plan your nPartition configuration by selecting which cells will comprise the new nPartition.

Use the **`parstatus -AC`** command to list all unassigned (available) cells in the server complex.

```
# parstatus -AC
[Cell]
                        CPU     Memory                                Use
                        OK/     (GB)                          Core    On
Hardware   Actual       Failed/ OK/                           cell    Next Par
Location   Usage        Max     Failed  Connected To          Capable Boot Num
========== ============ ======= ======= =================== ======= ==== ===
cab0,cell1 absent       -       -       -                     -       -    -
cab0,cell3 absent       -       -       -                     -       -    -
cab0,cell4 power on     4/0/4   2.0/0.0 cab 0,bay0,chassis3 yes     -    -
cab0,cell5 absent       -       -       -                     -       -    -
cab0,cell6 power on     4/0/4   2.0/0.0 cab 0,bay1,chassis1 yes     -    -
cab0,cell7 absent       -       -       -                     -       -    -

#
```

You can select any of the cells listed to create the new nPartition; only the cells that are *not "absent"* are present within the server complex.

All cells that you choose **must** meet the hardware requirements for nPartitions (for example, they all must have the same processor revision and firmware) and **should** form an HP-recommended nPartition configuration. At least one cell must have an I/O chassis with core I/O.

Step 2. After confirming that cells you have chosen would establish a valid nPartition configuration, use the **`parcreate -c`**... command to create a new nPartition with the cells.

When using the `parcreate` command, *do not* specify the `-B` option for this procedure.

(The `-B` option causes `parcreate` to immediately boot the newly-created nPartition past the default ready for reconfig state, thus making the nPartition active and preventing you from further modifying it.)

By *not* specifying `-B`, the new nPartition can be further modified because it will remain inactive at the ready for reconfig state (until you boot it using the service processor Command menu's `BO` command).

If creating a single-cell nPartition, just use one `-c` option.

To create a multiple-cell nPartition, you should specify the `-c` option multiple times (once for each cell) issuing a single command line.

```
# parcreate -c4:base:y:ri -c6:base:y:ri
Partition Created. The partition number is : 1
#
```

When `parcreate` successfully creates a new nPartition, it reports "Partition Created" and reports the nPartition number ("partition number is...").

If parcreate detects any problems or issues when creating an nPartition, it lists them in its output. If it cannot create the nPartition, `parcreate` reports "Command failed" along with more details.

The `parcreate` command's `-c` option is as follows:

`-c` *`cell`*`:[`*`cell_type`*`]:[`*`use_on_next_boot`*`]:[`*`failure_usage`*`]`

This option specifies the cell ID (*`cell`*) to be assigned to the nPartition.

- The only valid *`cell_type`* value is: `base` (base cell, the default).
- The valid *`use_on_next_boot`* values for cells are:

`y`	Participate in reboot (the default).
`n`	Do not participate in reboot.

- The only valid *`failure_usage`* value is: `ri` (reactivate with interleave, the default).

For details, see the *parcreate* (1M) manpage.

Step 3. Use the **`parmodify`** command to modify the new nPartition's configuration and set the nPartition name (`-P`), boot paths (`-b`, `-s`, and `-t`), and any core cell choices (`-r`).

When using the `parmodify` command, you must use the `-p`*#* option to specify the partition number for the nPartition. Use the partition number that the `parcreate` command reported in *Step 2*.

```
# parmodify -p1 -P "hostname05"
Command succeeded.
# parmodify -p1 -r0/4 -r0/6
Command succeeded.
# parmodify -p1 -b 4/0/1/0/0.9
Command succeeded.
#
```

When each modification takes place, `parmodify` reports “Command succeeded”. Otherwise it reports any problems.

You can specify each configuration option on a separate command line or can combine all options into a single, longer command line.

For details on the various options for modifying nPartition settings, see the *parmodify* (1M) manpage.

Step 4. Use the **`parstatus -V -p#`** command to list all details about your newly created and configured nPartition.

If any configuration details should be modified, use the `parmodify` command before you boot the nPartition in the next step.

```
# parstatus -V -p1
[Partition]
Partition Number       :  1
Partition Name         : hostname05
Status                 : inactive
IP address             :
Prmary Boot Path       : 4/0/1/0/0.9
ALternate Boot Path    : 0/0/0/0/0/0/0/0.0.0
HA Alternate Boot Path : 0/0/0/0/0/0/0/0.0.0
PDC Revision           : 104.1
IODCH Version          : 23664
CPU Speed              : 552 MHz
Core Cell              : ?
Core Cell Alternate    :
                0. cab0,cell4
                1. cab0,cell6
[Cell]
                        CPU     Memory                                      Use
                        OK/     (GB)                                Core    On
Hardware   Actual       Failed/ OK/                                 cell    Next Par
Location   Usage        Max     Failed    Connected To              Capable Boot Num
========== ============ ======= ========= =================== ======= ==== ===
cab0,cell4 inactive     4/0/4     2.0/ 0.0 cab 0,bay0,chassis3 yes     yes  1
cab0,cell6 inactive     4/0/4     2.0/ 0.0 cab 0,bay1,chassis1 yes     yes  1
....
```

Step 5. Boot your newly-created nPartition past boot-is-blocked to make it active and make its BCH interface available.

Use the service processor Command menu's **BO** command to boot the nPartition.

Once the nPartition is booted, you can access its BCH interface through its console. Use the service processor Console menu (enter **CO** at the service processor Main menu).

Creating a New nPartition [Partition Manager]

Use the **Partition —> Create Partition** action to create a new nPartition using Partition Manager.

Step 1. Plan your nPartition configuration by selecting which cells will comprise the new nPartition.

All cells that you choose **must** meet the hardware requirements for nPartitions (for example, they all must have the same processor revision and firmware) and **should** form an HP-recommended nPartition configuration. At least one cell must have an I/O chassis with core I/O.

Step 2. Run Partition Manager (`/opt/parmgr/bin/parmgr`) or access it from SAM or a Web browser.

You optionally can specify the `parmgr -t create` command and options to automatically launch the **Partition —> Create Partition** action. See the *parmgr* (1M) manpage for command option details.

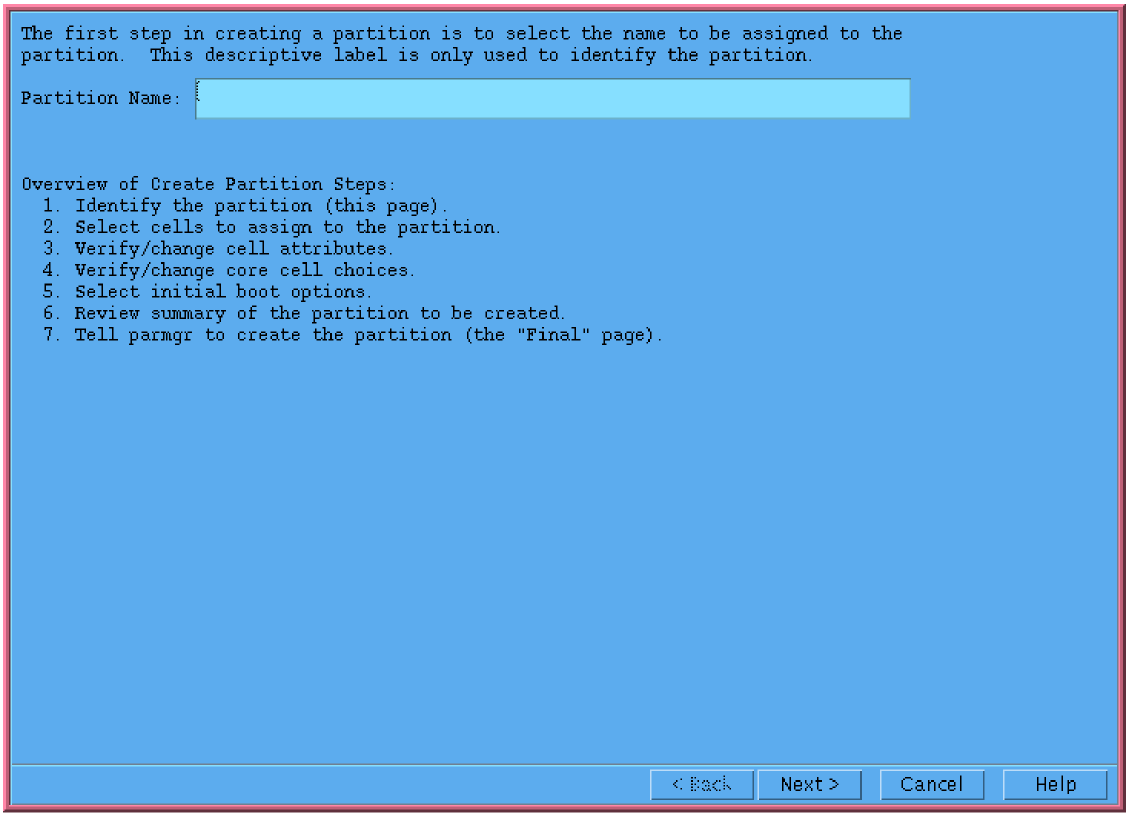
Step 3. Select the **Partition —> Create Partition** action to run the Partition Manager task wizard for creating a new nPartition.

Partition Manager guides you through the steps it requires for creating a new nPartition.

You can move backward and forward through the steps by using **Next** and **Back** buttons. At the final steps, you can verify the settings you have established for the new nPartition and, if they are correct for your purposes, click the **Finish** button to create the new nPartition.

You can *cancel the new nPartition* creation at any time by clicking the **Cancel** button.

The following window shows the first step and overview for Partition Manager’s create **Partition —> Create Partition** action.



If you specify for Partition Manager to automatically boot the new nPartition, you can access the new nPartition’s BCH interface from its console when you finish using the create partition task wizard.

Otherwise, if you *do not* specify to automatically boot the new nPartition, you must use the service processor Command menu’s **BO** command to boot the nPartition past boot-is-blocked (inactive, ready for reconfig state) and make its BCH interface available.

Assigning (Adding) Cells to an nPartition

You can add cells to the local nPartition or to any remote nPartitions in the same server complex.

Adding cells to an nPartition involves selecting available cells (those not currently assigned to an nPartition) and assigning them to an existing nPartition. Both the selected cells and any I/O chassis connected to the cells are assigned to the designated nPartition.

You can add cells to an nPartition by using the following procedures:

- *Adding Cells to an nPartition [HP-UX]* on page 267
- *Adding Cells to an nPartition [Partition Manager]* on page 268

When adding cells to an nPartition, you should refer to the guidelines in the chapter *Planning nPartition Configurations* on page 109.

Reboot for Reconfig Guidelines for Adding Cells

In some situations, you must immediately perform a reboot for reconfig of a modified nPartition after adding cells to it.

- You **must** immediately perform a reboot for reconfig (`shutdown -R`) of an nPartition when you have added a cell to an active nPartition and you specified the `-B` option to the `parmodify` command.
- You **should** perform a reboot for reconfig of an nPartition *as soon as possible* after you have added a cell to an active nPartition and have specified a “y” use-on-next-boot value for the new cell.
- You need not perform a reboot for reconfig of an nPartition in these situations:
  - — When you have added a cell to an inactive nPartition.
  - — When you have added a cell with a “n” use-on-next-boot value and you did not specify the `-B` option to the `parmodify` command.

Adding Cells to an nPartition [HP-UX]

Use the **`parstatus`** and **`parmodify`** commands to add cells to an nPartition using HP-UX commands.

Step 1. Use the **`parstatus -A -C`** command to list all available cells (the unassigned cells) in the server complex.

Step 2. Choose one or more eligible cells from the list to add to the nPartition.

Adding the cell(s) to the nPartition should create a configuration that adheres to the hardware requirements and performance guidelines.

Step 3. Modify the nPartition by issuing the **`parmodify -p# -a#...`** command to add the cell.

The `-p`*#* option specifies the partition number (*#*) for the nPartition being modified.

The -a *`cell`*:*`type`*:*`use`*:*`fail`* option specifies the cell ID and other details for the cell to be added to the nPartition.

To add multiple cells, you can specify the `-a` option multiple times in the same command.

For example: `parmodify -p1 -a0:base:y:ri -a2:base:y:ri` adds two cells (cell ID 0 and cell ID 2) to nPartition number 1.

The `-a` option (`-a` *`cell`*:*`type`*:*`use`*:*`fail`*) specifies the following details for each cell that you add to the nPartition.

`cell` — The cell to be added to the nPartition. You can specify the cell in global (*`cell`*) format or in hardware location (*`cabinet`*/*`slot`*) format.

`type` — The cell type: `base` is the only supported cell type and it is the default.

`use` — The cell's use-on-next-boot value: `y` or `n`. Use `y` (the default) if the cell is to be an active member of the nPartition, or use `n` if the cell is to remain an inactive member.

`fail` — The cell's failure usage: `ri` (reactivate with interleave) is the only supported failure usage policy and it is the default.

You can optionally specify the `parmodify` command's `-B` option to require that the modified nPartition be rebooted.

- When you specify `-B` to modify an *inactive* nPartition, the inactive nPartition completes partition rendezvous and becomes active if possible.
- When you specify `-B` to modify an *active* nPartition, you must perform a reboot for reconfig of the nPartition before any other cell assignment changes can be made within the server complex.

The `parmodify -p1 -a0:base:y:ri -a2:base:y:ri` command adds cell 0 and cell 2 to partition number 1. This command also sets a "y" use-on-next-boot value for both cells, meaning that they will be active members of the nPartition following the next time all cells boot (for example, when reboot for reconfig is performed on the nPartition.

Because this example command does not include the `-B` option, if partition 1 were an *inactive nPartition*, it would remain inactive; if partition 1 were an *active nPartition* the new cells would be assigned, but they would remain inactive cells until a reboot for reconfig is performed.

See the *parmodify* (1M) manpage for details on all options.

Step 4. As needed, perform a reboot for reconfig (**`shutdown -R`**) on the modified nPartition.

See the *Reboot for Reconfig Guidelines for Adding Cells* on page 266 for details on when to perform a reboot for reconfig.

Adding Cells to an nPartition [Partition Manager]

Use the **Partition —> Modify Partition** action, **Add/Remove Cells** tab to add cells to an nPartition from Partition Manager.

Step 1. Run Partition Manager (`/opt/parmgr/bin/parmgr`) or access it from SAM or a Web browser.

Step 2. In the Partition Manager primary window, select the nPartition to which you want to add cells, then select the **Partition —> Modify Partition** menu item.

Step 3. In the Modify Partition window, click the **Add/Remove Cells** tab.

Step 4. Select the cells that you want to add to the nPartition from the Available Cells list, then click the **Add** button to add them to the nPartition's cell list.

If adding multiple cells, you can select multiple cells by pressing the **Control** key while clicking on the cells.

Adding the cell(s) to the nPartition should create a configuration that adheres to the hardware requirements and performance guidelines.

Step 5. After you add the new cells to the nPartition's cell list, click the **OK** button.

The cells are not actually *assigned* to the nPartition until after the next step.

Step 6. Review the information shown in the **Notes and Warnings**, the **Summary of Changes**, and the **HA Checks** tabs.

Partition Manager generates this information when it checks details of the new nPartition configuration, such as cell compatibility and various high-availability guidelines.

To cancel all nPartition changes, click the **Cancel** button in the Notes and Warnings window and then click **Cancel** in the Modify Partition window.

To proceed with adding the cell(s) to the nPartition, click **OK**.

Step 7. Once Partition Manager confirms that the "partition has been successfully modified" click the **OK** button.

The cells are assigned to the nPartition after clicking **OK** in the previous step. However, any cells you have added to an active nPartition will remain *inactive* until you perform a reboot for reconfig of the modified nPartition.

Step 8. As needed, perform a reboot for reconfig (**`shutdown -R`**) of the modified nPartition.

See the *Reboot for Reconfig Guidelines for Adding Cells* on page 266 for details on when to perform a reboot for reconfig.

Unassigning (Removing) Cells from an nPartition

Removing a cell from an nPartition involves unassigning the cell from the nPartition to which it is assigned and, if necessary, performing a reboot for reconfig of the nPartition.

You can remove *any cell* from the local nPartition and can remove *inactive cells* from remote nPartitions in the same server complex. However, at least one core-capable cell must remain in each nPartition.

You can remove (unassign) cells from nPartitions by using these procedures:

- *Removing Cells from an nPartition [HP-UX]* on page 271
- *Removing Cells from an nPartition [Partition Manager]* on page 274

When removing cells from an nPartition, you should ensure that the modified nPartition still adheres to the hardware requirements and performance guidelines for nPartitions. Refer to the chapter *Planning nPartition Configurations* on page 109 for details.

After you remove a cell from an nPartition, the cell's I/O chassis also is removed from the nPartition. As a result, any I/O devices associated with the cell are made unavailable to the nPartition after the cell is removed.

If you want to remove the last cell in an nPartition, you must instead remove the nPartition using the `parremove` command or Partition Manager.

Once a cell is unassigned, the cell (and any I/O resources connected to the cell) is considered to be an available resource that is on the "free cell list" and can be assigned to any nPartition in the server complex.

Reboot for Reconfig Guidelines for Removing Cells

In some situations, you must immediately perform a reboot for reconfig (`shutdown -R`) of a modified nPartition after removing cells from it. Performing a required reboot for reconfig completes cell assignment changes and unlocks the server's Complex Profile.

- You **must** immediately perform a reboot for reconfig of an nPartition when you have removed an *active cell* from the nPartition.
- You **must** immediately perform a reboot for reconfig of an nPartition when you have removed a cell from an active nPartition and specified the `-B` option to the `parmodify` command.
- You need not perform a reboot for reconfig of an nPartition when you have removed an *inactive* cell from an nPartition and did not specify the `-B` option to the `parmodify` command.

In the cases where you must immediately perform a reboot for reconfig after removing a cell, *not doing so* will leave the Complex Profile locked and thus will prevent any other changes to the server complex configuration. In these cases, the reboot for reconfig is required to complete the cell assignment changes and permit other changes to occur.

Removing Cells from an nPartition [HP-UX]

Use the **`parstatus`** and **`parmodify`** commands to remove cells from an nPartition using HP-UX commands.

Step 1. List the current nPartition assignments and status for the cells you plan to remove from their assigned nPartition by issuing the **`parstatus -c#...`** HP-UX command.

Specify each cell you plan to remove with a separate `-c` option.

For example, to list details on cells 0, 1, and 2, issue the `parstatus -c0 -c1 -c2` command.

The cells must all be assigned to the same nPartition in order to remove them using a single procedure. Otherwise, if the cells are assigned to different nPartitions, you must perform this procedure separately for each nPartition.

In order to remove cells that are not assigned to the local nPartition, the cells must be *inactive* (their "Actual Usage" must be "inactive"). You can list the local nPartition by issuing the `parstatus -w` command.

To remove an *active* cell from its nPartition, you must do so when logged in to HP-UX running on the cell's nPartition.

Step 2. Remove the cell from the nPartition to which it is assigned by using the **`parmodify -p# -d#...`** command.

Specify the partition number (`-p#`) and each cell (`-d#`) that you want to remove from the nPartition.

If removing *multiple cells* from an nPartition, specify each cell with a separate `-d#` option on the same command line (such as: `parmodify -p1 -d0 -d2...` to remove cells 0 and 2 from partition number 1).

Slightly different procedures are required for removing active cells and inactive cells. See the following information for details (*Guidelines for Removing an Active Cell* and *Guidelines for Removing an Inactive Cell*).

When you are removing multiple cells from the local nPartition, if at least one of the cells you plan to remove is currently active, then you should follow the guidelines for removing active cells.

- **Guidelines for Removing an Active Cell**

  You **should** specify the `-B` option to `parmodify` when removing an active cell from the local nPartition if you want the nPartition to become active following its reboot for reconfig.

  For example, the following command removes cell 4 from partition 0 and the `-B` option ensures that the nPartition will be active following its reboot for reconfig.

  ```
  # parmodify -p0 -d4 -B
  Cell 4 is active.
  Use shutdown -R to shutdown the system to ready for
  reconfig state.
  Command succeeded.
  #
  ```

  You **must** perform a reboot for reconfig (`shutdown -R`) after you issue the `parmodify` command to remove active cell(s) from the nPartition. (This is covered in *Step 3* that follows.)

- **Guidelines for Removing an Inactive Cell**

  When removing an *inactive* cell from an nPartition you do not need to specify the `-B` option to `parmodify` and do not need to perform a reboot for reconfig of the cell's nPartition.

  When you use `parmodify` to remove an inactive cell, the cell is immediately unassigned from its nPartition.

  If you specify the `-B` option when removing an *inactive cell* from an *inactive nPartition*, then the cell is immediately removed and the modified nPartition is booted past its inactive ready for reconfig state and becomes an active nPartition.

  For example, the following command removes cell 2 from partition 0. Because cell 2 is inactive, it is immediately unassigned.

```
# parmodify -p0 -d2
Command succeeded.
#
```

Step 3. As needed, perform a reboot for reconfig (**`shutdown -R`**) of the nPartition being modified.

You **must** perform a reboot for reconfig if you have removed an *active cell* or have specified the `-B` option when modifying an *active nPartition*.

See the *Reboot for Reconfig Guidelines for Removing Cells* on page 271 for details on when to perform a reboot for reconfig.

This reboot for reconfig enables the cell removal to complete and the Complex Profile to be unlocked.

If you have removed an active cell and you did not specify the `-B` option to `parmodify`, then the nPartition will remain *inactive* in the ready for reconfig state after you perform the reboot for reconfig. To make the inactive nPartition active, use the service processor Command menu's `BO` (boot) command.

Removing Cells from an nPartition [Partition Manager]

Use the **Partition —> Modify Partition** action, **Add/Remove Cells** tab to remove cells from an nPartition using Partition Manager.

Step 1. Determine which cell(s) you want to remove from the nPartition.

The cells must all be assigned to the same nPartition in order to remove them using a single procedure. Otherwise, if the cells are assigned to different nPartitions, you must perform this procedure separately for each nPartition.

Step 2. Run Partition Manager (`/opt/parmgr/bin/parmgr`) or access it from SAM or a Web browser.

Step 3. In the Partition Manager primary window, select the nPartition from which you want to remove cells, then select the **Partition —> Modify Partition** action.

Step 4. In the Modify Partition window, click the **Add/Remove Cells** tab.

Step 5. From the “Cells in the Partition” list, select the cells that you want to remove from the nPartition. Then click the **Remove** button to move them to the Available Cells list. If removing multiple cells, you can select multiple cells by pressing the **Control** key while clicking on the cells.

Removing the cell(s) from the nPartition should create a configuration that adheres to the hardware requirements and performance guidelines.

Step 6. After you have removed the cells from the nPartition’s cell list, click the **OK** button.

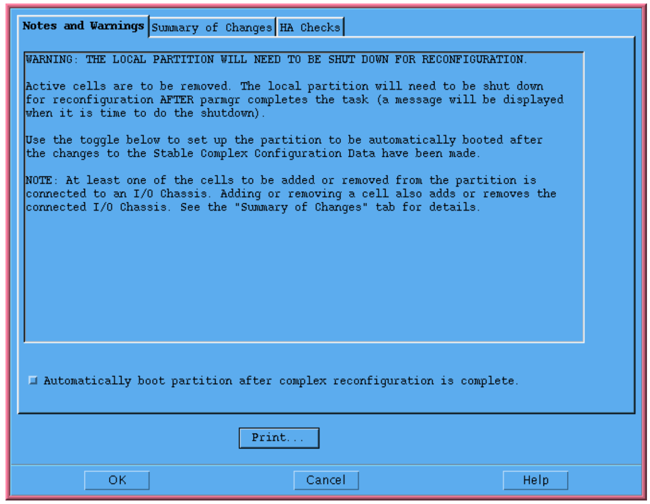
The cells are not actually *removed* from the nPartition until after the next step.

Step 7. Review the information shown in the **Notes and Warnings**, the **Summary of Changes**, and the **HA Checks** tabs.

Partition Manager generates this information when it checks details of the new nPartition configuration.

If you **must** perform a reboot for reconfig of the nPartition, such as when removing an active cell from the nPartition, then the Notes and Warnings tab provides details and options.

- If a reboot for reconfig is required, the Notes and Warnings tab has information about the reboot procedure. (See *Step 9* for details.)



- When a reboot for reconfig is required, the Notes and Warnings tab also has a check box (“Automatically boot partition”) that—when selected—enables the nPartition to rendezvous and be active after the reboot for reconfig.

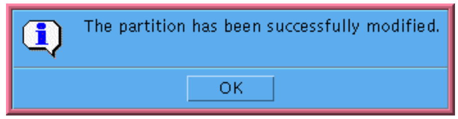
To cancel all nPartition changes, click the **Cancel** button in the Notes and Warnings window and then click **Cancel** in the Modify Partition window.

To proceed with removing the cell(s) from the nPartition, click **OK**.

Step 8. When Partition Manager confirms that the “partition has been successfully modified”, review any additional information and respond as needed to the dialog box presented.

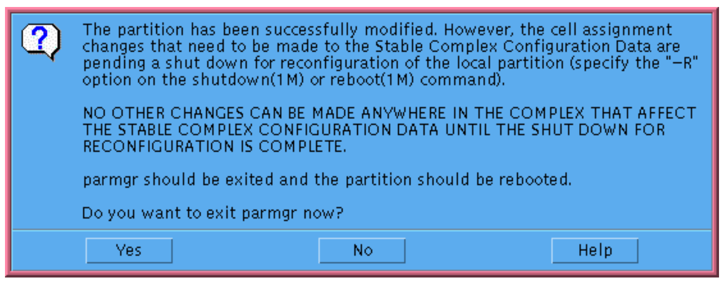
(The cells were designated to be removed from the nPartition after completing the previous step, however a reboot may be required.)

- If you have removed only *inactive* cells from the nPartition, Partition Manager provides no additional info and you can click **OK** to complete the procedure (a reboot is not needed).



- If you have removed one or more *active* cells from the nPartition, then Partition Manager provides more info about performing the required reboot for reconfig of the nPartition.

  You **must** reboot the nPartition as soon as possible, so you should click the **Yes** button to exit Partition Manager and proceed with the next step of this procedure.



Step 9. As needed, perform a reboot for reconfig (**`shutdown -R`**) of the modified nPartition.

- If you have removed only inactive cells from an nPartition, then you *do not* need to perform a reboot for reconfig of the nPartition.

- If you have removed any active cells from the nPartition, then you **must** perform a reboot for reconfig and will have seen a detailed message about rebooting from Partition Manager (see *Steps 7 and 8*).

  After you issue the `shutdown -R` command, the nPartition performs the reboot for reconfig. If you selected the “Automatically boot partition” check box earlier in this procedure (see *Step 7*), then the nPartition is active after the reboot for reconfig and you can interact with it through its console.

If you *did not* select the “Automatically boot partition” check box, then the nPartition is inactive (at the ready for reconfig state) after the reboot for reconfig occurs. In this situation, you can make the nPartition active by using the service processor Command menu’s `BO` command.

Removing (Deleting) an nPartition

You can delete (remove) any nPartition within a server complex.

The HP-UX nPartition deletion capabilities include restrictions for security reasons: you can delete only the *local nPartition* and *inactive remote* nPartitions.

You can delete an nPartition using these procedures:

- *Deleting an nPartition [HP-UX]* on page 278
- *Deleting an nPartition [Partition Manager]* on page 280

When removing the local nPartition, you must complete the procedure by issuing the **`shutdown -R -H`** command *as soon as possible* after initiating the local nPartition's removal.

Deleting an nPartition causes all of the nPartition's cells (and any I/O resources connected to the cells) to be unassigned. As a result, all of these cells become available resources that are on the "free cell list" and can be assigned to any nPartition in the server complex.

Deleting an nPartition [HP-UX]

Use the **`parremove`** command to delete an nPartition using HP-UX commands.

Step 1. Use the **`parstatus -P`** command to list all nPartitions, and check the status (active or inactive) for the nPartition you plan to remove.

To check the *local* partition number, use the `parstatus -w` command. The local nPartition always is active when it is running HP-UX.

If you are planning to remove a *remote* nPartition, check to see whether the remote nPartition is inactive.

Step 2. If a remote nPartition that you plan to remove currently is *active*, then put the nPartition into the ready for reconfig state to make it inactive.

If the remote nPartition is running HP-UX, you can shut down the nPartition to the ready for reconfig state by 1) logging in to HP-UX on the remote nPartition, 2) shutting down all applications and warning users, and 3) issuing the `shutdown -R -H` command.

You also can put the nPartition into the ready for reconfig state by using the BCH interface's `RECONFIGRESET` command or the service processor Command menu's `RR` command.

Step 3. Save all current configuration details about the nPartition you plan to remove.

Use the **`parstatus -V -p#`** command to display all current configuration information related to the nPartition you plan to remove.

Save this information, as you can use it to manually recreate the nPartition if necessary at a later time.

Step 4. Remove the nPartition.

Use one of the following procedures (*Removing an Inactive Remote nPartition* or *Removing the Local nPartition*) to remove the nPartition.

- **Removing an Inactive Remote nPartition**

  1. Issue the `parremove -p#` command to remove the inactive remote nPartition, where the `-p#` option specifies the partition number. For example:

     `#` **`parremove -p1`**

  2. Issue the `parstatus -P` command to confirm that the nPartition was removed.

     If the nPartition was removed, it no longer is listed in the `parstatus` command's output.

- **Removing the Local nPartition**

  To remove the local nPartition (the nPartition on which you currently are issuing commands), perform the following steps.

  1. Shut down all applications and warn users. Follow the same procedures you would use if you were to reboot the nPartition.

  2. Issue the `parremove -F -p#` command, which initiates the complex profile revisions that will take place when the nPartition is removed.

     When using `parremove` to remove the local nPartition, you must specify both the `-p#` option (to specify the local partition number) and the `-F` option (to force-remove the local nPartition).

Note that the local nPartition remains active following the `parremove -F -p#` command, until you perform a shutdown for reconfig (`shutdown -R -H`) to complete the removal.

As soon as possible you should proceed with the shutdown for reconfig because the server Complex Profile will remain locked—and no other changes can occur—until the pending nPartition removal is completed.

3. Perform a shutdown for reconfig (`shutdown -R -H`) of the local nPartition.

   The `shutdown -R -H` command shuts down the nPartition and all cells so that the configuration changes occur and the nPartition is deleted.

After you complete the nPartition removal, the nPartition no longer exists—its configuration information has been deleted.

All cells (and associated I/O chassis) that used to be assigned to the deleted nPartition now are unassigned and can be assigned for other uses.

Deleting an nPartition [Partition Manager]

Use the **Partition —> Delete Partition** action to remove an nPartition using Partition Manager.

Step 1. Run Partition Manager (`/opt/parmgr/bin/parmgr`) or access it from SAM or a Web browser.

Step 2. In the Partition Manager primary window, select the nPartition you want to remove.

Click the nPartition's name in the list on the left side of the primary window to select the nPartition.

If you plan to remove a *remote nPartition*, then after you select the nPartition's name, all of the nPartition's hardware (listed on the right side of the primary window) should be listed as having an Actual Usage of “inactive”. To remove a remote nPartition it must be inactive.

Step 3. Select the **Partition —> Delete Partition** action to request that the selected nPartition be removed (deleted).

Partition Manager presents the following window to confirm whether you want to delete the selected nPartition.

You can view details about the nPartition by clicking the **Show Details** button, or cancel the deletion by clicking **Cancel**.

Step 4. Proceed with the nPartition deletion by clicking the **OK** button in the Partition Manager confirmation window.

Step 5. Perform any remaining tasks (such as a `shutdown -R -H`) as needed.

You may need to perform a shutdown for reconfig, depending on the type of nPartition you are removing (local or remote) and its state (active or inactive). Review the following list for details:

- If you are removing a *remote* nPartition that was *inactive*, the nPartition was removed immediately so this removal procedure is finished and you do not need to perform any shutdowns.
- If you attempt to remove a *remote* nPartition that is *active*, Partition Manager cannot remove the remote nPartition. You first must make the remote nPartition inactive by putting it into the ready for reconfig state.

  To put the remote nPartition in the ready for reconfig state: login to the remote nPartition and issue the `shutdown -R -H` command, or issue the BCH `RECONFIGRESET` command or the service processor `RR` command for the remote nPartition.

  After the remote nPartition is inactive, you must perform this removal procedure again using Partition Manager on the local nPartition if you want to remove the remote nPartition.
- If you are removing the *local* nPartition, Partition Manager will display the following information to you after you click **OK** to remove the nPartition.

  To complete the local nPartition's removal, you must perform a shutdown for reconfig (`shutdown -R -H`) of the local nPartition as soon as possible.

Because the Complex Profile will remain locked until the local nPartition's removal is completed, no other changes can occur in the server complex until you perform the shutdown for reconfig.

After Partition Manager removes an nPartition, the nPartition no longer exists—its configuration information has been deleted.

All cells (and associated I/O chassis) that used to be assigned to the deleted nPartition now are unassigned and are available resources that can be assigned for other uses.

Naming and Renaming nPartitions

Each nPartition has both a partition *number* and an nPartition *name*.

The **partition name** for each nPartition can have from 1 to 64 characters, including upper- and lowercase letters; numbers; and dashes, underscores, and spaces ("-" "_" and " ").

You can customize each nPartition's name to help you distinguish among the nPartition in a server complex. (You cannot change the partition number, which is a permanent unique identifier that is automatically assigned for each nPartition in a server complex.)

You can name and rename nPartitions using these procedures:

- *Renaming an nPartition [BCH]* on page 283
- *Renaming an nPartition [HP-UX]* on page 284
- *Renaming an nPartition [Partition Manager]* on page 285

Partition names are displayed (along with partition numbers) in various reports and menus provided by the service processor, Boot Console Handler (BCH), and the HP-UX nPartition tools. Note that some utilities display only the first 30 characters of nPartition names.

Renaming an nPartition [BCH]

Use the Configuration menu **PD** command to check and sets the local nPartition's name from the BCH interface.

Step 1. Login to the service processor for the server complex in which the nPartition resides.

Step 2. Access the nPartition's console.

From the service processor Main menu, enter **CO** to access the console menu and select the nPartition.

If necessary, type **^ecf** (**Control-e c f**) to get write access for the console.

Note that if the nPartition is booted to HP-UX, you should instead use the HP-UX command method of modifying the nPartition name.

Step 3. Access the BCH interface's Configuration menu.

From the Main menu, enter **CO** to access the Configuration menu.

If at another BCH menu, enter `MA` to access the Main menu, then enter `CO` for the Configuration menu.

Step 4. At the BCH Configuration menu, use the **PD** command to check and set the local nPartition's name.

Enter `PD` to check the current name, or enter `PD` *New Name* to set the nPartition's name to the new name. No quotation marks are needed when specifying the new name.

```
Configuration Menu: Enter command > PD

Partition Number: 1
  Partition Name: Partition 1
Configuration Menu: Enter command >

Configuration Menu: Enter command > PD My New Name

  Partition Name: My New Name
Configuration Menu: Enter command > PD

Partition Number: 1
  Partition Name: My New Name
Configuration Menu: Enter command >
```

Renaming an nPartition [HP-UX]

Use the **parmodify -p# -P** ***name*** command to set the nPartition name for nPartitions using HP-UX commands.

Step 1. List the current nPartition states and names using the `parstatus -P` command.

This shows all nPartitions, their current status (active or inactive), and their partition numbers and nPartition names.

Step 2. Use the `parmodify -p# -P` *name* command to set the nPartition name for any of the nPartitions in the server complex.

Specify both the partition number (`-p#`) and the new name for the nPartition (`-P` *name*). If the nPartition name contains spaces then quotation marks must surround the name.

```
# parmodify -p1 -P "New Name"
Command succeeded.
#
```

You can list the nPartition's new name by using the `parstatus -p#` command or `parstatus -P`.

Renaming an nPartition [Partition Manager]

Use the **Partition —> Modify Partition** action, **General** tab to name and rename nPartitions using Partition Manager.

Step 1. Run Partition Manager (`/opt/parmgr/bin/parmgr`) or access it from SAM or a Web browser.

Step 2. In the Partition Manager primary window, select the nPartition whose name you want to change.

Click the nPartition's name in the list on the left side of the primary window to select the nPartition.

Step 3. Select the **Partition —> Modify Partition** action, and access the **General** tab.

The nPartition name is listed—and can be edited—in the Partition Name field in the **General** tab.

Step 4. Edit the nPartition's name in the Partition Name field, and click **OK** when done editing the name (or click **Cancel** to cancel any changes).

Step 5. Review any Notes and Warnings that Partition Manager presents, and click **OK** to implement the name change (or click **Cancel** to cancel the change).

If there are any important notes or warnings, Partition Manager presents them in the window before completing the changes.

When the name change is complete, Partition Manager presents a final dialog box confirming that the nPartition was successfully modified.

Setting and Checking Cell Attributes

Each cell assigned to an nPartition has *use-on-next-boot* and *failure usage* attributes that determine how the cell is used within the nPartition.

You can list and set cell attributes by using these procedures:

- *Setting Cell Attributes [BCH]* on page 287
- *Setting Cell Attributes [HP-UX]* on page 289
- *Setting Cell Attributes [Partition Manager]* on page 292

Each cell's use-on-next-boot and failure usage attribute settings establish the following behaviors for the cell:

- **Use-on-Next-Boot**

  The use-on-next-boot setting for each cell indicates whether the cell will be used (active) the next time the cell's nPartition is booted.

- **Failure Usage**

  The failure usage setting (called the "Failure Mode" in Partition Manager) for each cell indicates whether the cell will be used, if possible, if any processors or memory fail during the cell's self-tests.

NOTE

Currently, only one failure usage setting is supported: *reactivate with interleave* (`ri`).

The reactivate-with-interleave setting allows a cell to actively join its nPartition following processor or memory failures during the cell's self tests. The cell joins its nPartition if at least one processor and any valid amount of memory passes self tests. Any of the cell's components that fail (processors or memory) are not available to the nPartition.

After changing a cell's attributes, the new attribute settings are used starting the next time the nPartition and cells are rebooted.

Setting Cell Attributes [BCH]

Use the Configuration menu **`CELLCONFIG`** command to list and set a cell's use-on-next-boot setting from the BCH interface.

From the BCH interface you can modify only cell use-on-next-boot settings.

Step 1. Login to the complex's service processor, access the nPartition's console, and access the BCH interface.

From the nPartition console, you access the nPartition's BCH interface. If the nPartition is not at the BCH interface, you must either boot the nPartition or shut down HP-UX to return to the BCH interface.

Step 2. Access the BCH Configuration menu by entering **`CO`** from the BCH Main menu.

If you are at a BCH menu other than the Main menu, enter `MA` to return to the Main menu and then enter `CO` to access the Configuration menu.

Step 3. From the BCH Configuration menu, use the **`CELLCONFIG`** command to list or set each cell's use-on-next-boot setting.

To list the use-on-next-boot settings for all cells in the nPartition, issue the `CELLCONFIG` command with no arguments.

```
Configuration Menu: Enter command > CELLCONFIG

  Cell Configuration Data for Partition
 ---------------------------------------
  Configured Set  : 0x0000000000000050
  Deconfigured Set: 0x0000000000000000
  Free Cell Set   : 0xffffffffffffffaa


       Cab/
 Cell  Slot  Cell State  Configuration Status
 ----  ----  ----------  --------------------
   4   0/4       Alive       Configured
   6   0/6       Alive       Configured

Configuration Menu: Enter command >
```

To change the use-on-next-boot setting for a cell, issue the `CELLCONFIG` command with arguments: `CELLCONFIG` *`cell`* `[ON|OFF]`

For example, `CELLCONFIG 6 OFF` sets the use-on-next-boot setting for cell 6 to OFF. This causes the cell to be inactive (not rendezvous and thus not be used) the next time the nPartition boots.

```
Configuration Menu: Enter command > CELLCONFIG 6 OFF

 Are you sure you want to DECONFIGURE cell 6 for next boot?
(y/[n]) >> y
 Cell 6 will be disabled during next reboot.

Configuration Menu: Enter command >
```

Step 4. Reboot the nPartition to use the cells' new use-on-next-boot settings.

If you have changed any cell use-on-next-boot settings for the nPartition, you should reboot the nPartition in either of two ways:

- Use the BCH interface's `REBOOT` command to perform a reboot.

  If you have only changed cell configurations from ON to OFF, then perform a reboot using the `REBOOT` command. Any cells set to not be used will still be assigned to the nPartition but will not be used (will not rendezvous) in the nPartition.

- Use the BCH interface's `RECONFIGRESET` command to put the nPartition in the ready for reconfig state, then use the service processor Command menu's `BO` command to boot the nPartition.

  If you have changed any cell from OFF ("n", do not use on next boot) to ON ("y", use the cell on next boot), then you must perform these two tasks; this resets and reconfigures the nPartition and boots it.

BCH

```
Configuration Menu: Enter command > RECONFIGRESET
Reset the partition for reconfiguration of Complex Profile ...
```

Service Processor (GSP or MP)

```
GSP:CM> BO

This command boots the selected partition.

     #   Name
    ---  ----
     0)  jules00
     1)  jules01

    Select a partition number: 1

    Do you want to boot partition number 1? (Y/[N]) y

    -> The selected partition will be booted.
GSP:CM>
```

Setting Cell Attributes [HP-UX]

Use the **parstatus** and **parmodify -p# -m#...** commands to list and set the use-on-next-boot and failure usage settings for cells using HP-UX commands.

Step 1. Login to HP-UX running on the nPartition.

You can login to HP-UX on the nPartition either by connecting with `telnet` or `rlogin`, or by logging in to its complex's service processor and accessing the nPartition's console.

Connecting through the service processor allows you to maintain nPartition console access after HP-UX has shut down.

Step 2. From the HP-UX command line, use the `parstatus` command to list the use-on-next-boot and failure usage attribute settings for cells in the server complex.

You can list and modify any cell's settings from HP-UX running on any nPartition in the server complex.

Use either `parstatus -C` or `parstatus -V -c#` to list the cell attribute settings. The following examples and text describe both these commands.

- A use-on-next-boot value of "yes" means the cell will be active as part of the nPartition the next time the nPartition boots.

"Yes" is equivalent to a BCH cell configuration value of ON and "no" is equivalent to OFF.

- A failure usage setting of "activate" (equivalent to "ri") indicates that the cell is set to reactivate with interleave in the event of any failure during the cell's self test.

Use the `parstatus -C` command to list the use-on-next-boot setting for all cells, which is shown in the "Use On Next Boot" column.

```
# parstatus -C
[Cell]
                        CPU     Memory                                Use
                        OK/     (GB)                          Core    On
Hardware   Actual       Deconf/ OK/                           Cell    Next Par
Location   Usage        Max     Deconf    Connected To        Capable Boot Num
========== ============ ======= ========= =================== ======= ==== ===
cab0,cell0 active core  4/0/4    2.0/ 0.0 cab0,bay0,chassis1  yes     yes  0
cab0,cell1 absent       -       -         -                   -       -    -
cab0,cell2 active base  4/0/4    2.0/ 0.0 cab0,bay1,chassis3  yes     yes  0
cab0,cell3 absent       -       -         -                   -       -    -
cab0,cell4 active core  4/0/4    2.0/ 0.0 cab0,bay0,chassis3  yes     yes  1
cab0,cell5 absent       -       -         -                   -       -    -
cab0,cell6 active base  4/0/4    2.0/ 0.0 cab0,bay1,chassis1  no      yes  1
cab0,cell7 absent       -       -         -                   -       -    -

#
```

To list a specific cell's failure-usage and use-on-next boot settings, issue the `parstatus -V -c#` command and specify the cell number.

```
# parstatus -V -c2
[Cell]
Hardware Location       : cab0,cell2
Global Cell Number      : 2
Actual Usage            : active base
Normal Usage            : base
Connected To            : cab0,bay1,chassis3
Core Cell Capable       : yes
Firmware Revision       : 6.0
Failure Usage           : activate
Use On Next Boot        : yes
Partition Number        : 0

....

Memory OK     : 2.00 GB
Memory Deconf : 0.00 GB

#
```

Step 3. To modify a cell's use-on-next-boot and failure usage attribute settings, use the **`parmodify -p# -m#...`** command and specify the cell's new settings.

Specify both the `-p` (partition number) and `-m` (modify cell) options when using `parmodify`. The following example modifies cell 2 to not be used the next time its nPartition (partition number 0) boots.

```
# parmodify -p0 -m2:base:n:ri
Command succeeded.
#
```

The `parmodify` command's `-m` option is as follows:

`-m` *`cell`*`:[`*`cell_type`*`]:[`*`use_on_next_boot`*`]:[`*`failure_usage`*`]`

This option specifies the cell ID (*`cell`*) whose settings are modified using the following arguments.

- The only valid *`cell_type`* value is `base` (base cell).
- The valid *`use_on_next_boot`* values for cells are:

  `y` Participate in reboot (the default).

  `n` Do not participate in reboot.

- The only valid *`failure_usage`* value for cells is `ri` (reactivate and interleave).

For details, see the *parmodify* (1M) manpage.

Step 4. If you have modified a cell's attribute settings, you must reboot the nPartition to which the cell is assigned for the settings to be used.

Rebooting the cell's nPartition allows the nPartition to use each cell's new attribute settings.

- If a cell's use-on-next-boot setting is changed from "n" (do not use) to "y" (use), you must perform a reboot for reconfig of the cell's nPartition by using the `shutdown -R` command.
- Otherwise, if the cell use-on-next-boot settings are only changed from "y" to "n" then you can perform a standard reboot using the `shutdown -r` command.

Setting Cell Attributes [Partition Manager]

Use the **Partition —> Modify Partition** action, **Change Cell Attributes** tab to list and set the configurable cell attributes using Partition Manager.

Step 1. Run Partition Manager (`/opt/parmgr/bin/parmgr`) or access it from SAM or a Web browser.

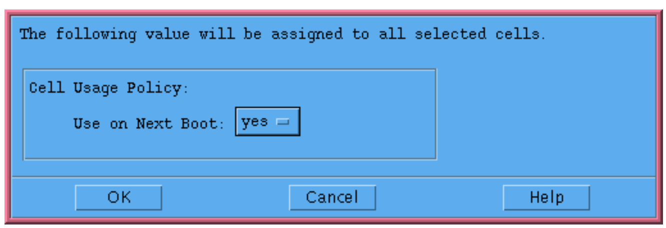
Step 2. In the Partition Manager primary window, select the nPartition whose cell attributes you want to change.

Click the nPartition's name in the list on the left side of the primary window to select the nPartition.

Step 3. Select the **Partition —> Modify Partition** action, and click the **Change Cell Attributes** tab.

Step 4. Highlight the cell whose attributes you want to modify, click the **Modify Cell** button, and configure the cell attributes as desired.

You can modify the settings for multiple cells at once by selecting all desired cells (press **Control** while clicking on the cells) before clicking the **Modify Cell** button.



Configure the cell attributes in the window, and then click **OK** to apply the modified attributes or **Cancel** to cancel any changes.

Step 5. If you have modified any cell attributes, when you return to the **Change Cell Attributes** tab you can click **OK** to apply the changes or click **Cancel** to cancel them.

Review any Notes and Warnings that Partition Manager presents, then click **OK** to proceed or **Cancel** to cancel the changes.

If the cell attribute changes are implemented, Partition Manager presents a final confirmation that the nPartition was successfully modified.

Step 6. If you have modified a cell's attribute settings, you must reboot the nPartition to which the cell is assigned for the settings to be used.

Rebooting the cell's nPartition allows the nPartition to use each cell's new attribute settings.

- If a cell's use-on-next-boot setting is changed from "no" (do not use) to "yes" (use), you must perform a reboot for reconfig of the cell's nPartition by using the `shutdown -R` command.
- Otherwise, if the cell use-on-next-boot settings are only changed from "yes" to "no" then you can perform a standard reboot using the `shutdown -r` command.

Setting and Checking nPartition Core Cell Choices

The **core cell choice** settings for an nPartition are optional preferences that establish which cells in the nPartition are preferred to be selected as the core cell for the nPartition.

You can list and set an nPartition's core cell choices by using these procedures:

- *Setting nPartition Core Cell Choices [BCH]* on page 295
- *Setting nPartition Core Cell Choices [HP-UX]* on page 296
- *Setting nPartition Core Cell Choices [Partition Manager]* on page 297

NOTE You do not need to specify core cell choices for a valid core cell to be chosen.

By default on HP Superdome and HP rp8400 server, system firmware selects the lowest numbered eligible cell as an nPartition's active core cell. By default on HP rp7405/rp7410 servers, cell 1 is selected as the core cell.

NOTE You should specify only *core-capable* cells as core cell choices. A cell must have an I/O chassis with core I/O attached to be eligible to be chosen as the core cell.

Setting nPartition Core Cell Choices [BCH]

Use the Configuration menu **`COC`** command to set the core cell choices for an nPartition using the nPartition's BCH interface.

Step 1. Access the BCH menu for the nPartition whose core cell choices you wish to set.

Step 2. Access the BCH Configuration menu for the nPartition.

From the BCH Main menu, enter **`CO`** to enter the Configuration menu.

Step 3. Issue the **`COC`** command to check current core cell choice preferences.

Entering `COC` with no arguments lists all core cell choice preferences.

Step 4. Issue the **`COC`** command *with arguments* to set or change the nPartition's core cell choice preferences.

The `COC` command syntax is: `COC` *`choice cell`*, where *`choice`* is 0–3 (with 0 being the highest-priority choice) and where *`cell`* is the cell ID.

For example, **`COC 0 2`** sets the most preferred core cell choice to be cell ID 2. Likewise, **`COC 1 4`** sets the next (second-highest priority) core preference to be cell ID 4.

Use the `HELP COC` command for other details about the `COC` command.

Step 5. [*Optional*] If you have changed the setting for the highest-priority core cell choice (choice 0) and you want the cell you have specified to become the active core cell, then issue the BCH menu's `REBOOT` command.

Even if you do not perform this step, the new core cell choice settings will be used the next time the nPartition is rebooted.

Setting nPartition Core Cell Choices [HP-UX]

Use the **parstatus** and **parmodify** commands to list and set the core cell choices for an nPartition using HP-UX commands.

Step 1. Issue the **parstatus -V -p*#*** command to list the nPartition's current core cell choices and core cell use.

The parstatus -V -p*#* command list detailed status, including the current active core cell ("Core Cell"), and any core cell choice settings (the "Core Cell Alternate" listings, if any).

```
# parstatus -V -p0
[Partition]
Partition Number       : 0
Partition Name         : jules00
Status                 : active
IP address             : 0.0.0.0
Primary Boot Path      : 0/0/2/0/0.13.0
Alternate Boot Path    : 0/0/2/0/0.0.0
HA Alternate Boot Path : 0/0/2/0/0.14.0
PDC Revision           : 6.0
IODCH Version          : 23664
CPU Speed              : 552 MHz
Core Cell              : cab0,cell0
Core Cell Alternate [1]: cab0,cell0
Core Cell Alternate [2]: cab0,cell2

....

                                 Core Connected  Par
Hardware Location   Usage        IO   To         Num
=================== ============ ==== ========== ===
cab0,bay0,chassis1  active       yes  cab0,cell0 0
cab0,bay1,chassis3  active       yes  cab0,cell2 0

#
```

The core cell choice preferences are listed by parstatus as the "Core Cell Alternate" settings with "1" being the highest priority and "2" through "4" as the lower priority core cell choices.

The parstatus core cell choice listings (1 through 4) directly correspond to the BCH core cell choice listings (0 through 3).

Step 2. Modify the nPartition's core cell choices using the **parmodify -p*#* -r*#*...** command.

You can modify the core cell choices for the local nPartition or any remote nPartition in the server complex.

Use the following command: `parmodify -p# -r# -r#...`

Specify the partition number (`-p#`) and the cell ID (`-r#`) for all cells you wish to designate as core cell choices.

```
# parmodify -p0 -r2 -r0
Command succeeded.
#
```

The order in which you list the cells is the order in which the nPartition's core cell choices are established; the first cell listed is the first preferred core cell (choice 1), and the subsequent cells are lower-priority core cell choices (choices 2 through 4, if specified).

Step 3. [*Optional*] If you wish to immediately use the new core cell choice settings, reboot the nPartition whose core cell choices you have changed.

Even if you do not reboot now, the new core cell choices will be used the next time the nPartition is rebooted.

You can issue the `shutdown` command with the `-r` option to reboot the nPartition and use the new core cell choice settings. (You *do not* need to perform a reboot for reconfig of the nPartition.)

If you have modified an *inactive remote* nPartition, use the service processor Command menu's `BO` command to boot the remote nPartition; the designated core cell choices will be used to select the active core cell.

Setting nPartition Core Cell Choices [Partition Manager]

Use the **Partition —> Modify Partition** action, **Cell Cell Choices** tab to set the core cell choices for an nPartition using Partition Manager.

Step 1. Run Partition Manager (`/opt/parmgr/bin/parmgr`) or access it from SAM or a Web browser.

Step 2. In the Partition Manager primary window, select the nPartition whose core cell choices you want to change.

Click the nPartition's name in the list on the left side of the primary window to select the nPartition.

Step 3. Select the **Partition —> Modify Partition** action, and click the **Cell Cell Choices** tab.

Step 4. Modify the core choice setting for each cell whose core choice priority you want to revise.

Highlight the cell whose core cell choice you want to configure, select the desired choice priority (1st, 2nd, none, etc.) from the Core Cell Choice pop-up list, and click the **Modify** button to modify the priority.

Step 5. Click the **OK** button when done changing the core choice priorities (or click **Cancel** to not revise any priorities).

Review any Notes and Warnings that Partition Manager presents, then click **OK** to proceed or **Cancel** to cancel the changes.

If the cell choice priority changes are implemented, Partition Manager presents a final confirmation that the nPartition was successfully modified.

The new core cell choice priorities will be used the next time the nPartition is rebooted.

Reassigning (Moving) a Cell to a Different nPartition

To move a cell from one nPartition to another nPartition in the same server complex, use the high-level procedure described here.

This high-level procedure involves:

1. Removing (unassigning) the cell from its original nPartition.
2. Performing a reboot for reconfig (`shutdown -R`) of the original nPartition, if needed.
3. Adding (assigning) the cell to the new nPartition.
4. Performing a reboot for reconfig (`shutdown -R`) of the cell's new nPartition.

Be aware that this procedure modifies the hardware configurations of both nPartitions involved.

You should *adhere to the hardware requirements and performance guidelines* when removing the cell from its original nPartition *and* when adding it to its new nPartition.

When you remove a cell from an nPartition, any I/O connected to the cell also is removed from the nPartition. As a result, any I/O devices associated with the cell are made unavailable to the nPartition.

CAUTION

Moving a cell that has an attached I/O chassis from one nPartition to another is effectively the same as moving the associated I/O devices from one computer to another.

All precautions you would take when moving I/O devices from one computer to another must be taken in this situation.

For example, LVM volume groups that are being moved from one nPartition to another must be exported from their original nPartition *before the cell or I/O chassis is moved*. For details see the *vgexport* (1M) manpage and *Managing Systems and Workgroups*.

Reassigning (Moving) a Cell to a Different nPartition

The following high-level procedure moves a cell to a different nPartition.

You can perform these steps using either HP-UX commands or Partition Manager.

Step 1. See the procedure *Unassigning (Removing) Cells from an nPartition* on page 270 to remove the cell that you want to move to the other nPartition.

As part of this step, you perform a reboot for reconfig (`shutdown -R`) of the nPartition to which the cell is originally assigned.

As a result of this step, the cell is unassigned (on the "free cell list" of available resources) so is is available to be assigned to the other nPartition in the next step.

Step 2. See the procedure *Assigning (Adding) Cells to an nPartition* on page 266 to assign the cell you made available in *Step 1* to its new nPartition.

This step also requires that you perform a reboot for reconfig (`shutdown -R`) of the nPartition to which the cell is being added.

Following the reboot for reconfig, the cell is available (assigned and active) in the new location.

Restoring a Complex Profile

You can restore a previous Complex Profile configuration, which effectively allows you to undo your last nPartition configuration change.

Restoring the previous Complex Profile allows you to revert to the previous complex configuration—including the nPartition configurations—that existed before you made your last change.

NOTE Because restoring a prior Complex Profile requires shutting down all nPartitions to an inactive ready for reconfig state, you should use this procedure only when absolutely necessary.

Restoring a Complex Profile [Service Processor]

Use the Command menu **`CC`** command and **`L`** option to restore the prior complex configuration that existed before you made your last nPartition configuration change.

Step 1. Ensure that all nPartitions within the server complex are inactive at the ready for reconfig state.

You can put each nPartition into the ready for reconfig state by using the `shutdown -R -H` command, the BCH interface's `RECONFIGRESET` command, or using the service processor Command menu's `RR` command.

Step 2. Login to the server complex's service processor, and enter **`CM`** to access the Command menu.

Login as a user with administrator privileges, which are required for restoring the previous complex profile configuration.

Step 3. Issue the **`CC`** command, select Last Complex Profile (**`L`**), and confirm that you want to modify the Complex Profile configuration.

```
GSP:CM> CC

This command allows you to change the complex profile.

WARNING: You must either shut down the OSs for reconfiguration or
         execute the RR (reset for reconfiguration) command for all
         partitions before executing this command.
```

```
    G - Build genesis complex profile
    L - Restore last complex profile
        Select profile to build or restore: L

    Do you want to modify the complex profile? (Y/[N]) y

    -> The complex profile will be modified.
GSP:CM>
```

Step 4. Confirm that the nPartition cell assignments are what you intended to establish.

Use the service processor Command menu's **CP** command to display the current complex profile configuration.

```
GSP:CM> CP

--------------------------------------------------------------------------------
Cabinet |   0    |   1    |   2    |   3    |   4    |   5    |   6    |   7
--------+--------+--------+--------+--------+--------+--------+--------+--------
 Slot   |01234567|01234567|01234567|01234567|01234567|01234567|01234567|01234567
--------+--------+--------+--------+--------+--------+--------+--------+--------
 PD  0  |X.X.....|........|........|........|........|........|........|........
 PD  1  |....X.X.|........|........|........|........|........|........|........
 PD  2  |.X.X....|........|........|........|........|........|........|........
 PD  3  |.....X.X|........|........|........|........|........|........|........

GSP:CM>
```

If the nPartition cell assignments are not what you intended—that is, if you prefer the nPartition configuration you had before you restored the existing nPartition configuration—you can repeat this procedure to restore the configuration you had before beginning the procedure.

One level of undo is provided by the service processor Command menu's `CC` command. This allows you to undo your last nPartition change, and undo your undo.

Step 5. Issue the **BO** command to boot any nPartitions you want to make active.

After you use the `CC` command, all nPartitions still are in a boot-is-blocked ready for reconfig state and thus are *inactive* nPartitions.

You can use the Command menu's `BO` command to boot the nPartitions past boot-is-blocked to make the nPartitions active.

Unlocking Complex Profiles

This section describes how you can force-unlock portions of the nPartition Complex Profile data

CAUTION Do not force-unlock complex profile data except in extremely rare cases following nPartition or server crashes.

Improperly force-unlocking complex profiles can result in the loss of pending configuration changes to nPartitions and the server complex.

Under normal circumstances you do not need to manually unlock the Complex Profile. The commands, utilities, and related procedures handle all locking and unlocking.

In some situations, you must perform a reboot for reconfig (`shutdown -R`) of a modified nPartition in order to complete an nPartition reconfiguration and unlock the changed portion of the Complex Profile. (For example, when removing an active cell from an nPartition you must perform a reboot for reconfig.)

HP-UX provides the `parunlock` command to force-unlock parts of a server's Complex Profile in situations where the normal procedures and utilities have failed.

The service processor (GSP or MP) also provides the `RL` command for resetting Complex Profile locks.

7 Listing and Managing Server Hardware

This chapter covers the tools and procedures for listing details about the hardware assigned to nPartitions. This chapter also covers getting information about server hardware, and managing the hardware resources in nPartitions and their server complexes.

For an introduction to nPartition servers and hardware features, refer to the chapter *nPartition System Overviews* on page 31.

Tools for Listing and Managing Hardware

You can use several software tools to list server hardware details and manage the hardware in a server complex. These tools have features that overlap for some tasks, but each tool also has unique features.

The tools for listing and managing system hardware are:

- **Service Processor (GSP or MP) menus**

  Service processor menus provide a complex-wide service interface that can allow access to all hardware and nPartitions.

NOTE

The service processor in HP servers is sometimes called the Management Processor (MP) and sometimes the Guardian Service Processor (GSP).

Regardless of the name, the service processor in these servers provides approximately the same features and performs essentially the same role.

Throughout this document, the term “service processor” refers to both the MP and GSP service processors.

  Hardware management features include the service processor Command menu’s `DU`, `ID`, `PE`, `PS`, and `SYSREV` commands.

- **Boot Console Handler (BCH) interfaces**

  The BCH interface is the method for interacting with an nPartition before it has booted HP-UX.

  Hardware management features include the BCH interface’s Configuration menu, Interface menu, and Service menu.

- **HP-UX Commands**

  HP-UX commands allow you to manage and monitor nPartitions and hardware within a server complex from HP-UX running on any of the server’s nPartitions.

  Hardware management features include the `parstatus`, `frupower`, `fruled`, and `rad` commands, among many others.

- **Partition Manager** (`/opt/parmgr/bin/parmgr`)

  Partition Manager provides a graphical interface for managing and monitoring nPartitions and hardware within a server complex.

  Hardware management features include menus and windows that list details about cells, I/O chassis, and PCI I/O card slots in the server complex.

- **System Administration Manager** (SAM, `/usr/sbin/sam`)

  The SAM graphical interface (GUI) provides an alternate way to launch Partition Manager as a SAM area.

  SAM also provides a Peripheral Devices area, which has a Cards subarea that is the recommended method for managing PCI I/O cards and PCI slots.

  The **Peripheral Devices —> Cards** area includes error checking and resource analysis not available from the HP-UX command line.

Powering Server Cabinets On and Off

You can power on and power off the cabinets within a server complex either by using the main power switch on the front of the cabinet, or by using the service processor Command menu.

You can use the following procedures:

- *Powering Server Cabinets On and Off [Power Switch]* on page 309
- *Powering Server Cabinets On and Off [Service Processor]* on page 310

When powering off a cabinet, you turn off 48-volt power to the cabinet thus causing all cells and all I/O chassis to power off, and causing most fans to turn off.

CAUTION When you power on or off HP Superdome 64-way compute cabinets, you must power off and power on cabinet 0 and cabinet 1 in such a way that *both cabinets are off* for an overlapping interval.

If either Superdome 64-way cabinet is powered off then powered on while the other cabinet remains on, then communications between the two cabinets is lost.

CAUTION Before powering off system hardware, you first must check whether it is being used.

The cabinet power switch and the service processor Command menu's PE command *do not check* whether system hardware is in use before powering it off.

Changes in cabinet power status *do not* affect the standby power that supplies system utilities such as the service processor (GSP or MP) and keeps some fans running. These utilities and fans can receive power as long as standby power is enabled.

The way in which standby power is enabled and disabled differs for various HP server models. On HP Superdome servers, standby cabinet power is switched using the power breakers on the rear of the cabinet. On HP rp7405/rp7410 and HP rp8400 servers, standby power is enabled through the power cords connecting to the inputs on the rear of the cabinet.

Powering Server Cabinets On and Off [Power Switch]

Use the Virtual Front Panel to check status, and then use the cabinet power switch to manage a cabinet's 48-volt power with the cabinet hardware.

Step 1. Login to the system's service processor and access the Virtual Front Panel for the system.

From the service processor Main menu, enter **VFP** to access the Virtual Front Panel menu, then enter **S** to access the "system VFP" that displays the current status for all nPartitions.

Step 2. Check the VFP status to see whether any cabinet hardware is running HP-UX.

Any nPartition whose state is "HP-UX heartbeat" is running HP-UX and thus should not have its hardware powered off until after HP-UX is shut down.

Type **^b** (**Control-b**) to exit the VFP.

Step 3. Shut down HP-UX running on any cabinet hardware that you plan to power off.

Step 4. Confirm that nobody else is using or servicing the cabinet hardware you plan to power on or off.

You should both physically inspect the hardware, and check whether others are remotely accessing the system's service processor (using the Command menu's **WHO** command).

Step 5. Access the cabinet hardware and flip the power switch (located on the cabinet's front) to the on or off position in order to power the cabinet on or off.

Powering Server Cabinets On and Off [Service Processor]

Use the Virtual Front Panel, and the use the Command menu **PE** command to turn a cabinet's 48-volt power on or off from the service processor (GSP or MP).

Step 1. Login to the system's service processor and access the Virtual Front Panel for the system.

From the service processor Main menu, enter **VFP** to access the Virtual Front Panel menu, then enter **S** to access the "system VFP" that displays the current status for all nPartitions.

Step 2. Check the VFP status to see whether any cabinet hardware is running HP-UX.

Any nPartition whose state is "HP-UX heartbeat" is running HP-UX and thus should not have its hardware powered off until after HP-UX is shut down.

Step 3. Shut down HP-UX running on any cabinet hardware that you plan to power off.

Step 4. Confirm that nobody else is using or servicing the cabinet hardware you plan to power on or off.

You should both physically inspect the hardware, and check whether others are remotely accessing the system's service processor (using the Command menu's **WHO** command).

Step 5. Access the system's service processor Command menu, issue the **PE** command, then select the cabinet to power on or power off.

From the service processor Main menu, enter **CM** to access the Command menu. To exit the Command menu enter **MA**.

When using the PE command enter **B** to power on or off a cabinet; specify the cabinet number; and then enter **ON** (power on), **OFF** (power off), or **Q** (quit without changing the power status).

```
GSP:CM> PE

This command controls power enable to a hardware device.

    B - Cabinet
    C - Cell
    I - IO Chassis
        Select Device: b
```

```
Enter cabinet number: 1

The power state is ON for Cabinet 1.
In what state do you want the power? (ON/OFF)
```

Powering Cells and I/O Chassis On and Off

This section covers *cell* and *I/O chassis* power management procedures, which allow you to control power for cells and I/O chassis from remote locations, without physically accessing the system hardware.

You can use the following procedures:

- *Powering Cells and I/O Chassis On and Off [Service Processor]* on page 313
- *Powering Cells and I/O Chassis On and Off [HP-UX]* on page 314
- *Powering Cells and I/O Chassis On and Off [Partition Manager]* on page 316

NOTE

On HP nPartition systems, *powering on* a cell also powers on any I/O chassis attached to the cell, and *powering off* a cell also powers off any I/O chassis attached to the cell.

Powering on or off an I/O chassis connected to a powered-on cell *causes the cell to reset* if the cell located and mapped the I/O chassis during its cell boot process.

The `frupower` command and Partition Manager permit you to power on or off *inactive* cells and I/O chassis that are assigned to the current nPartition or are not assigned to any nPartition.

The service processor Command menu's `PE` command permits you to power on or off any hardware in the complex, including active cells and I/O chassis. The `PE` command does not check the current usage of components.

Powering Cells and I/O Chassis On and Off [Service Processor]

Use the Command menu **PE** command to power on and power off cells, I/O chassis, and cabinets from the service processor interface (GSP or MP).

CAUTION When using the service processor Command menu's PE command to power on or off hardware, you should be certain to *specify the correct component* to power on or off.

The PE command *does not check* whether the hardware is actively being used.

You can manage the power for all components within the system complex using the service processor Command menu's PE command, regardless of any nPartition assignment or the status (active or inactive) for the hardware components.

Step 1. Login to the system's service processor and access the Command menu.

From the service processor Main menu, enter **CM** to access the Command menu. To exit the Command menu enter **MA**.

Step 2. Issue the **PE** command and specify the type of hardware whose power you want to turn on or turn off.

You can manage power to cells, I/O chassis, and cabinets.

Step 3. Specify the hardware device to power on or power off.

The service processor *does not check* whether the specified component is currently being used.

- **Cabinets**—When you power on or off a cabinet, the firmware also powers on or off all cells and I/O chassis in the cabinet.
- **Cells**—When you power on or off a cell, the firmware also powers on or off any I/O chassis attached to the cell.

  When specifying a cell, you indicate both the *cabinet number* and the *slot* in which the cell resides.

- **I/O Chassis**—When you power off an I/O chassis from the service processor Command menu, the system firmware *resets the cell* attached to the I/O chassis (if the cell located and mapped the I/O chassis during its cell boot process).

  When specifying an I/O chassis, you indicate the cabinet, bay, and chassis numbers to identify it.

In the following example, the service processor powers off cell 2 in cabinet 0.

```
GSP:CM> PE

This command controls power enable to a hardware device.

    B - Cabinet
    C - Cell
    I - IO Chassis
        Select Device: c

    Enter cabinet number: 0
    Enter slot number: 2

    The power is ON for the Cell in Cabinet 0, Slot 2.
    In what state do you want the power for the
    Cell in Cabinet 0, Slot 2? (ON/OFF) OFF


GSP:CM>
```

Powering Cells and I/O Chassis On and Off [HP-UX]

Use the **`frupower -o -c`***#* and **`frupower -f -c`***#* commands to power on and power off cells (and their associated I/O chassis) from HP-UX.

NOTE You can use the `frupower` command to power on or off *inactive cells* that are either assigned to the local nPartition or are not assigned to an nPartition.

You cannot power off active cells or power on or off cells assigned to a remote nPartition when using `frupower`.

To power on or off an I/O chassis using `frupower`, do so by power cycling the cell to which it is connected.

Step 1. Login to HP-UX running on one of the system's nPartitions.

To manage a cell's power, you must login to the nPartition to which the cell is assigned. If the cell is not assigned to an nPartition, you can manage its power from *any* nPartition.

Step 2. Use the `frupower` command to turn on or turn off the cell's power.

Specify the **`frupower -f -c#`** command to *power off* a cell. (`-c#`). This also powers off any I/O chassis connected to the cell.

Specify the **`frupower -o -c#`** command to *power on* a cell (`-c#`). This also powers on any I/O chassis connected to the cell.

The following example shows several sample `frupower` commands and their results.

```
# frupower -f -c0
Error: Can not power off active cell 0.
# frupower -f -c2
# frupower -o -c2
# frupower -f -c6
Error: Cell 6 belongs to partition 1.  Can not power off cell.
#
# frupower -f -i0/1/1
Error: I/O chassis 0/1/1 is attached to a powered-on free
cell 4.  Please power off the free cell.
#
```

In the above example, cell 0 is active and thus cannot be powered off using `frupower`. Cell 2 is inactive and is powered off (`frupower -f -c2`) and then powered back on (`frupower -o -c2`). Cell 6 is assigned to a remote nPartition (partition number 1) and thus cannot be powered off. I/O chassis 0/1/1 is attached to cell 4, so to power it off cell 4 must be powered off.

Powering Cells and I/O Chassis On and Off [Partition Manager]

Use the **Cell —> Power On Cell** action, or **Cell —> Power Off Cell** action, to power on and power off cells (and their associated I/O chassis) from Partition Manager.

NOTE You can use Partition Manager to power on or off *inactive cells* that are assigned to the local nPartition.

You cannot power off active cells or power on or off cells assigned to a remote nPartition when using Partition Manager.

To power on or off an I/O chassis using Partition Manager, do so by power cycling the cell to which it is connected.

Step 1. Run Partition Manager (`/opt/parmgr/bin/parmgr`) or access it from SAM or a Web browser.

Step 2. Select the nPartition that contains the cell you want to power on or off.

Partitions are listed on the left side of the Partition Manager primary window.

The cells and I/O chassis assigned to the nPartition are listed on the right side of the primary window once the nPartition is selected.

Step 3. Select the cell whose power you want to turn on or off.

Step 4. Select the **Cell —> Power On Cell** menu item, or select the **Cell —> Power Off Cell** menu item.

Power Status for Hardware Components

You can use system software to check power status for the following components from remote locations:

- Cabinets
- Bulk Power Supplies and Power Boards
- Cell Boards
- I/O Chassis
- Individual PCI Slots

NOTE Cabinet power details and power supply details are specific to each server model. For example, HP Superdome servers and HP rp8400 server have different power configurations and requirements.

You can use the following procedures:

- *Determining Hardware Power Status [Service Processor]* on page 318
- *Determining Hardware Power Status [HP-UX]* on page 320
- *Determining Hardware Power Status [Partition Manager]* on page 322

Determining Hardware Power Status [Service Processor]

Use the Command menu **PS** command to check power status for cabinets, bulk power supplies and power boards, cells, and core I/O from the service processor.

Step 1. Login to the system's service processor and enter **CM** to access the Command menu.

Step 2. Issue the service processor Command menu's **PS** command.

The PS command can list detailed information—including power status—for components within the system complex.

The PS command summarizes all cabinets, cells, and core I/O cards, and prompts you to specify which hardware device you want information about.

```
GSP:CM> PS

This command displays detailed power and hardware configuration status.

The following GSP bus devices were found:
+----+-----+-----------+----------------+-----------------------------------+
|    |     |           |                |             Core IOs              |
|    |     |           |                | IO Bay | IO Bay | IO Bay | IO Bay |
|    |     |    UGUY   |     Cells      |   0    |   1    |   2    |   3    |
|Cab.|     |           |                |IO Chas.|IO Chas.|IO Chas.|IO Chas.|
| #  | GSP | CLU | PM  |0 1 2 3 4 5 6 7 |0 1 2 3 |0 1 2 3 |0 1 2 3 |0 1 2 3 |
+----+-----+-----+-----+----------------+--------+--------+--------+--------+
|  0 |  *  |  *  |  *  |* * * * * * * * |  *   * |        |        |        |
|  1 |     |  *  |  *  |* * * * * * * * |  *   * |        |        |        |
|  8 |     |  *  |  *  |                |  *   * |  *   * |        |        |
You may display detailed power and hardware status for the following items:

    B - Cabinet (UGUY)
    C - Cell
    G - GSP
    I - Core IO
        Select Device:
```

Step 3. Specify the cabinet whose hardware and power status you want to check.

For each cabinet (B), the PS command reports detailed information that includes the power status for *all components* within the cabinet, including:

- **Cabinet Power**—Whether the 48-volt cabinet power switch is on or off, whether cabinet power is enabled, and details about power boards and bulk power supplies.
- **Cell Power**—Whether power is enabled and on for all cells within the cabinet.
- **Core I/O Card Power**—Whether power is enabled and on for all core I/O cards within the cabinet.

For system complexes that have multiple cabinets, you must check details for each cabinet separately.

You also can use the PS command to check individual cell (C) or core I/O (I) hardware and power status.

The following example shows cabinet power details for cabinet 0 of an SD64000 model Superdome server.

```
 Select Device: b

    Enter cabinet number: 0

HW status for SD64000 compute cabinet #0: NO FAILURE DETECTED
Power switch: on;  Power: enabled, good;  Door: closed
Fan speed: normal;  Temperature state: normal
Redundancy state: fans or blowers redundant, BPSs redundant

                      | Main BP|                  |   IO Backplanes   |
                      | Power  |                  | IO Bay 0 | IO Bay 1 |
               | Main | Boards |      Cells       | Chassis  | Chassis  |
               | BP   | 0 1 2  | 0 1 2 3 4 5 6 7  | 0 1 2 3  | 0 1 2 3  |
+--------------+------+--------+------------------+----------+----------+
 Populated     |  *   | * * *  | * * * * * * * *  |   *   *  |          |
 Power Enabled |  *   | * * *  | * * * * * * * *  |   *   *  |          |
 Powered On    |  *   | * * *  | * * * * * * * *  |   *   *  |          |
 Power Fault   |      |        |                  |          |          |
 Attention LED |      |        | *                |   *      |          |

                          | Cabinet |     IO     |
            |     BPS     | Blowers |    Fans    |
            | 0 1 2 3 4 5 | 0 1 2 3 | 0 1 2 3 4  |
+-----------+-------------+---------+------------+
  Populated | * * * * *   | * * * * | * * * * *  |
  Failed    |             |         |            |
```

Determining Hardware Power Status [HP-UX]

Use the **`parstatus -B`**, **`parstatus -V -b`***#*, **`frupower -d -C`**, **`frupower -d -I`**, or **`rad -q`** command to check the power status for system hardware from HP-UX.

For details on these HP-UX commands, see the online manpages for *parstatus* (1M), *frupower* (1M), and *rad* (1M).

Step 1. Login to HP-UX running on one of the system's nPartitions.

To check the power status for PCI card slots, you must login to the *local nPartition* where their PCI card cage resides.

You can check the power status for cabinets, cells, and I/O chassis from *any nPartition*.

Step 2. Issue the HP-UX commands to check the power status for the system components of interest to you.

- **Cabinet Power**—Use the **`parstatus -V -b`***#* command to check cabinet power status for the specified cabinet (`-b`*#*), or use the **`parstatus -B`** command for brief power status for all cabinets.

  The `parstatus` command gives details about each cabinet's bulk power supplies and power boards, as well as details about cabinet fans and blowers.

- **Cell Power**—Use the **`frupower -d -C`** command to list cell power status for all cells, or use the **`frupower -d -c`***#* command to list power status for a specific cell (`-c`*#*).

- **I/O Chassis Power**—Use the **`frupower -d -I`** command for power status for all I/O chassis, or use the use **`frupower -d -i`***#/#/#* command to list details for a specific I/O chassis (*`cabinet/bay/chassis`*)

- **PCI Card Slot Power**—Use the **`rad -q`** command and option to list details including PCI card slot power for all PCI card slots within the local nPartition.

  The `rad` command lists information for the local nPartition only.

The following example output shows power details for an HP Superdome system's cabinet, cells, I/O chassis, and PCI slots, as presented by various HP-UX commands.

```
# parstatus -V -b0
[Cabinet]
                  Cabinet   I/O       Bulk Power  Backplane
                  Blowers   Fans      Supplies    Power Boards
                  OK/       OK/       OK/         OK/
Cab               Failed/   Failed/   Failed/     Failed/
Num Cabinet Type N Status  N Status  N Status    N Status       GSP
=== ============ ========= ========= ==========  ============   ======
 0  SD32000       4/ 0/ N+  5/ 0/ ?   5/ 0/ N+    3/ 0/ N+      active
```

Cabinet Power

```
Bulk Power Supplies(BPS)
========================
Power Supply  0  ok
Power Supply  1  ok
Power Supply  2  ok
Power Supply  3  ok
Power Supply  4  ok

Backplane Power Boards
======================
Power Supply  0  ok
Power Supply  1  ok
Power Supply  2  ok


Notes: N+ = There are one or more spare items (fans/power supplies).
       N  = The number of items meets but does not exceed the need.
       N- = There are insufficient items to meet the need.
       ?  = The adequacy of the cooling system/power supplies is unknown.

# frupower -d -C
Global cell 0; cabinet 0, cell 0 is powered on.
Global cell 2; cabinet 0, cell 2 is powered on.
Global cell 4; cabinet 0, cell 4 is powered on.
Global cell 6; cabinet 0, cell 6 is powered off.
# frupower -d -c4
Global cell 4; cabinet 0, cell 4 is powered on.
```

Cell Power

```
# frupower -d -I
Cabinet 0, bay 0, chassis 1 is powered on.
Cabinet 0, bay 0, chassis 3 is powered on.
Cabinet 0, bay 1, chassis 1 is powered off.
Cabinet 0, bay 1, chassis 3 is powered on.
# frupower -d -i0/1/3
Cabinet 0, bay 1, chassis 3 is powered on.
```

I/O Chassis Power

```
# rad -q
                                                                  Driver(s)
Slot        Path        Bus   Speed   Power   Occupied    Suspended   Capable
0-0-1-0      0/0/0       0     33      On      Yes         No          No
0-0-1-1      0/0/1/0     8     33      On      No          N/A         N/A
0-0-1-2      0/0/2/0     16    33      On      Yes         No          Yes
0-0-1-3      0/0/3/0     24    33      On      No          N/A         N/A
0-0-1-4      0/0/4/0     32    33      On      No          N/A         N/A
0-0-1-5      0/0/6/0     48    33      On      No          N/A         N/A
0-0-1-6      0/0/14/0    112   33      On      Yes         No          Yes
```

PCI Slot Power

Determining Hardware Power Status [Partition Manager]

Use the **Complex —> Show Complex Details** menu to list system hardware power status using Partition Manager.

Step 1. Run Partition Manager (`/opt/parmgr/bin/parmgr`) or access it from SAM or a Web browser.

Step 2. Select the **Complex —> Show Complex Details** menu item.

This displays the Complex Details window, which has tabs providing info for cabinets (the General tab), Cells, and I/O Chassis.

To update the information in the Complex Details window, click the the **Rescan Complex** button.

Step 3. Select and view the power status information for the components of interest to you.

- **Cabinet Power**—Click the **Cabinet Info** tab for details on the system complex's cabinet, including cabinet power status.
- **Cell Power**—Click the **Cells** tab for details on cells including their power status.
- **I/O Chassis Power**—Click the **I/O Chassis** tab for details on I/O chassis including their power status.
- **PCI Card Slot Power**—Click the **I/O Chassis** tab, then select the I/O Chassis whose PCI slots you want to list, and then click the **Show Details** button.

Turning Attention Indicators (LEDs) On and Off

HP's nPartition systems provide **attention indicators (LEDs)** to help you to visually select and confirm which hardware you want to perform operations on. Attention indicators are amber (orange) lights.

Table 7-1 on page 324 lists attention indicator meanings and LED states (off, blinking, and on). These states and meanings adhere to the PCI Hot-Plug specification.

You can control the attention indicators for various hardware components by using the following procedures:

- *Turning Attention LEDs On and Off [HP-UX]* on page 325
- *Turning Attention LEDs On and Off [Partition Manager]* on page 327

NOTE

On HP Superdome servers, the attention indicator behavior has changed since the original HP-UX 11i release.

Starting with the June 2001 HP-UX 11i release, the HP-UX command and utility behavior is to blink attention indicators (rather than light them to a steady-on state, as was the original behavior).

As a result, HP Superdome servers upgrading to the June 2001 or later release will exhibit the new, blinking behavior (see Table 7-1 on page 324) when cell and I/O chassis attention indicators are enabled.

In cases where some nPartitions are running the original HP-UX 11i release and others are running a more recent release, *both behaviors* (the original "steady-on" behavior and the "PCI Hot-Plug" behavior) may be exhibited—possibly within the same server complex.

Table 7-1 lists the meanings for the various attention indicator states. When all of a system's components are functioning and no service operations are occurring, every attention indicator should be turned off. (HP Superdome cabinet number LCDs remain lit or blinking when cabinet power is on.)

Table 7-1 Attention Indicator (LED) States and Meanings

Attention Indicator State	Meaning
OFF	Not selected.
BLINKING	Selected to be used in a service operation.
ON	*Supported for PCI card slot LEDs only*. Service required, problems have been identified with the component.

LEDs for Hardware Components

You can control (turn off, blink, and/or turn on) attention indicators for the following hardware components.

- **Cell LEDs**

  Each cell or cell slot has its own attention indicators.

  — On HP *Superdome servers*, each cell's attention indicator is located on the server cabinet hardware below the cell's slot, just to the right of the cell slot's power LED.

  — On HP *rp7405/rp7410 and rp8400 servers*, each cell's attention indicator is located on the cell hardware, to the outside of the cell's power LEDs.

- **I/O Chassis LEDs**

  On HP *Superdome servers* only, each I/O chassis has a attention indicator, which is located on the cabinet above the I/O chassis.

- **PCI Card Slot LEDs**

  On *all* HP nPartition systems, each PCI card slot has an attention indicator that you can use to select the card slot. You can view a PCI card slot's attention indicator when accessing the card cage.

On HP *rp7405/rp7410 and rp8400 servers* only, you also can view each PCI slot's attention indicator beneath the corresponding slot, on the cabinet's external chassis at the rear of the server cabinet.

- **Cabinet Number LCDs**

  On HP *Superdome servers* only, each cabinet has a cabinet number LCD that serves as the cabinet's attention indicator.

NOTE The **cabinet "attention" light** on HP Superdome, HP rp8400, and HP rp7405/rp7410 servers is not user-controllable.

The cabinet "attention" light automatically turns on when one or more alert level 6 (or higher) chassis codes has been recorded in the error logs and has not yet been read. This light automatically turns off when a user enters the service processor (GSP or MP) chassis logs viewer.

Turning Attention LEDs On and Off [HP-UX]

Use the **`fruled -o...`**, **`fruled -f...`**, or **`rad -f...`** command to manage attention indicators by turning them on, off, or blinking them from HP-UX.

Step 1. Login to HP-UX running on one of the system's nPartitions.

You can manage cabinet, cell, and I/O chassis attention indicators from HP-UX on *any nPartition*. To manage PCI slot attention indicators, you must access HP-UX in the *local nPartition* to which the PCI slot's chassis is assigned.

Step 2. Use the `fruled` command or the `rad` command to manage (turn on, off, or blink) the attention indicator for a system hardware component.

From HP-UX you can manage LEDs for the following hardware components:

- **Cells**—Use `fruled` to blink or turn off a cell's attention indicator.

  — Turn Off: The **`fruled -f -c`***#* command *turns off* the attention indicator for the specified cell (`-c`*#*). To turn off all cell attention indicators use the **`fruled -f -C`** command.

  - Blink: The **`fruled -o -c`***`#`* command *blinks* the attention indicator for the specified cell (`-c`*#*).

- **I/O Chassis**—Use `fruled` to blink or turn off HP Superdome I/O chassis attention indicators.

  Specify the I/O chassis using *`cabinet/bay/chassis`* notation (*#/#/#*).

  - Turn Off: The **`fruled -f -i`***`#/#/#`* command *turns off* the attention indicator for the specified I/O chassis (`-i`*#/#/#*). To turn off all I/O chassis attention indicators use the **`fruled -f -I`** command.
  - Blink: The **`fruled -o -i`***`#/#/#`* command *blinks* the attention indicator for the specified I/O chassis (`-i`*#/#/#*).

- **Cabinet Numbers**—Use `fruled` to blink or not-blink (keep lit) the cabinet number LCD for an HP Superdome cabinet.

  - Not-Blink: The **`fruled -f -b`***`#`* command *stops blinking* (keeps it lit) the cabinet number LCD for the specified cabinet (`-b`*#*).
  - Blink: The **`fruled -o -b`***`#`* command *blinks* the cabinet number LED for the specified cabinet (`-b`*#*).

- **PCI Card Slots**—Use `rad` to turn on, off, or blink the attention indicator for a PCI card slot.

  Specify the PCI slot using *`cabinet-bay-chassis-slot`* (*#-#-#-#*) notation.

  - Turn Off: The **`rad -f off`** ***`slot`*** command *turns off* the attention indicator for the specified PCI card slot (*`slot`*).
  - Blink: The **`rad -f attention`** ***`slot`*** command *blinks* the attention indicator for the specified PCI card slot (*`slot`*).
  - Turn On: The **`rad -f on`** ***`slot`*** command *turns on* the attention indicator for the specified PCI card slot (*`slot`*).

For details see the *fruled* (1) manpage or the *rad* (1M) manpage.

The following example turns off and blinks various attention indicators on an HP Superdome system, including cell, I/O chassis, PCI slot, and cabinet LEDs.

`# fruled -f -C` `# fruled -f -I`	*Turn off* all cell and I/O chassis attention

`# fruled -o -c0 -c2 -c4`
`# fruled -o -i0/0/1 -i0/0/3`
`# fruled -o -b0`
	Blink attention indicators for cells 0, 2, and 4 and I/O chassis 0/0/1 and 0/0/3. *Blink* the cabinet number LCD for

`# fruled -f -C`
`# fruled -f -I`
`# fruled -f -b0`
	Turn off all cell and I/O chassis attention indicators and *stop blinking* the cabinet number LED.

`# rad -f attention 0-0-1-2`
`# rad -f off 0-0-1-2`
`#`
	Blink the attention indicator for PCI slot 2 in cabinet 0, bay 0, chassis 1. Then *turn off* the same PCI slot's attention

Turning Attention LEDs On and Off [Partition Manager]

Use the **Cell —> Light Cell LED** action, the **I/O —> Light I/O Chassis LED** action, or the **I/O —> Light Chassis and Slot LEDs** action to manage a hardware component's attention indicator by blinking it and turning it off from Partition Manager.

Step 1. Run Partition Manager (`/opt/parmgr/bin/parmgr`) or access it from SAM or a Web browser.

Step 2. In Partition Manager's primary window, select the nPartition to which the hardware component (cell, I/O chassis, or PCI slot) is assigned, or select Available Resources if the component is not assigned.

Step 3. Select the hardware component whose attention indicator you want to blink, then select the appropriate menu item to blink the LED.

You can manage LEDs for the following hardware components:

- **Cells**—Select the cell in Partition Manager's primary window, then select the **Cell —> Light Cell LED** menu item.

This menu item *blinks* the selected cell's attention indicator. On HP Superdome servers this *also blinks* the cabinet number LCD for the cabinet in which the cell resides.

- **I/O Chassis**—Select the I/O chassis in Partition Manager's primary window, then select the **I/O —> Light I/O Chassis LED** menu item.

  On HP Superdome servers this menu item *blinks* the attention indicator for the selected I/O chassis, and *also blinks* the cabinet number LCD for the cabinet in which the I/O chassis resides.

- **PCI Card Slots**—Double-click the PCI slot's I/O chassis in Partition Manager's primary window, then select the PCI slot listed in the I/O chassis window, and then select the **I/O —> Light Chassis and Slot LEDs** menu item.

  This menu item blinks the selected PCI card slot's attention indicator.

  On HP Superdome servers, this also blinks the I/O chassis attention indicator and blinks the cabinet number LCD.

Step 4. Click the **OK** button in the window to *turn off* the attention indicator for the hardware component you selected.

On HP Superdome servers, this *also turns off* any I/O chassis attention indicator and *stops blinking* any cabinet number LCD changed by this procedure.

Listing Cell Processor and Memory Configurations

You can determine the processor and memory configurations for cells in a server complex by using software tools and utilities.

Table 7-2 on page 330 lists the processor version info (HVERSIONs) that is reported by the procedures given in this section.

You can list processor and memory details using the following procedures:

- *Listing Cell Processors and Memory [Service Processor]* on page 331
- *Listing Cell Processors and Memory [BCH]* on page 333
- *Listing Cell Processors and Memory [HP-UX]* on page 334
- *Listing Cell Processors and Memory [Partition Manager]* on page 335

Table 7-2 Processor (CPU) Versions for Cells

Cell's Operating CPU Frequency	HVERSION for HP rp7405/rp7410 and rp8400 Servers	HVERSION for HP Superdome Servers
PA8600 — 552 MHz	—	`0x5c70`
PA8700 — 650 MHz	`0x5e60`	`0x5d70`
PA8700 — 750 MHz	`0x5e40`	`0x5e70`
PA8700 — 875 MHz	`0x5eb0`	`0x5ea0`

PA-RISC Processor HVERSIONs

Table 7-2 lists the processor HVERSION numbers that are reported for nPartitions and cells. These are hexadecimal numbers. HP Superdome processor HVERSIONs differ from rp7405/rp7410 and rp8400 HVERSIONs. See the procedures that follow for info on listing HVERSIONs.

NOTE

The HVERSION indicates the current operating frequency for processors in cells, but does not necessarily indicate the processor hardware revisions.

For a cell that is assigned to an nPartition, the processor HVERSION is based on the *operating frequency of the monarch cell* in the nPartition to which the cell is assigned.

Likewise, for a cell not assigned to an nPartition, the reported HVERSION refers to the *operating frequency of the cell*.

All processors in a cell operate at the same frequency, and all cells in an nPartition must operate at the same frequency. Different nPartitions in a server can operate at different frequencies.

Listing Cell Processors and Memory [Service Processor]

Use the Command menu **PS** command to list cell processor and memory configurations using the service processor Command menu.

Step 1. Login to the system's service processor and enter **CM** to access the Command menu.

You can check processor and memory details for any cell in the complex from the service processor.

Step 2. Issue the **PS** command and specify the cell whose processor and memory details you want to view.

The PS command reports details for the cell including its processor configuration (CPU population) and its memory configuration (DIMM population).

For the cell *memory configuration* details, the PS command displays each populated DIMM and identifies it using its rank notation (0A–0D, 1A–1D, and so on).

The following example shows details for cell 0 in cabinet 0, which has four processors (0–3) and four DIMMs installed (0A–0D).

```
GSP:CM> PS

This command displays detailed power and hardware configuration status.

You may display detailed power and hardware status for the following items:

    B - Cabinet (UGUY)
    C - Cell
    G - GSP
    I - Core IO
        Select Device: c

    Enter cabinet number: 0
    Enter slot number: 0

HW status for Cell 0 in cabinet 0: NO FAILURE DETECTED

Power status: on, no fault
Boot is not blocked; PDH memory is shared
Cell Attention LED is off
RIO cable status: connected
RIO cable connection physical location: cabinet 0, IO bay 1, IO chassis 3
Core cell is cabinet 0, cell 0

PDH status LEDs:  ****
                              CPUs
                            0 1 2 3
          Populated         * * * *
          Over temperature

DIMMs populated:
+----- A -----+ +----- B -----+ +----- C -----+ +----- D -----+
0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7
*               *               *               *

PDC firmware rev 10.0
PDH controller firmware rev 7.6, time stamp: TUE MAY 08 20:42:26 2001

GSP:CM>
```

Listing Cell Processors and Memory [BCH]

Use the Information menu **PR** and **ME** commands to list cell processor and memory configurations from an nPartition's BCH interface.

Using the BCH interface, you can check these details only for the active cells in the local nPartition.

Step 1. Access the BCH interface for the nPartition to which the cell is assigned, and access the BCH Information menu.

From the BCH Main menu, enter **IN** to access the BCH Information menu. (If you are not at the BCH Main menu, enter **MA** to go to the Main menu.)

Step 2. From the BCH Information menu, list the processor or memory status for all cells by issuing the **PR** and **ME** commands.

- **Processor status**—Use the **PR** command to report details about all processors on all active cells in the nPartition.
- **Memory status**—Use the **ME** command to report details about all active cells' memory configurations.

  The ME command summarizes memory (DIMM) details for each ranks of memory. Each rank is a set of 4 DIMMs.

These BCH commands do not report details for inactive cells.

```
Information Menu: Enter command > PR

PROCESSOR INFORMATION

        Cab/                                                           Processor
 Cell   Slot   CPU    Speed      HVERSION   SVERSION   CVERSION      State
 ----   ----   ---   --------   --------   --------   --------    -------------
   4    0/4     0     552 MHz    0x5c70     0x0491     0x0301      Active
                1     552 MHz    0x5c70     0x0491     0x0301      Idle
                2     552 MHz    0x5c70     0x0491     0x0301      Idle
                3     552 MHz    0x5c70     0x0491     0x0301      Idle
   6    0/6     0     552 MHz    0x5c70     0x0491     0x0301      Idle
                1     552 MHz    0x5c70     0x0491     0x0301      Idle
                2     552 MHz    0x5c70     0x0491     0x0301      Idle
                3     552 MHz    0x5c70     0x0491     0x0301      Idle

Information Menu: Enter command > ME

Partition Memory Information
```

```
Cell   DIMM Rank 0/1     DIMM Rank 2/3     DIMM Rank 4/5     DIMM Rank 6/7
      Size   Status     Size   Status     Size   Status     Size   Status
---- ------ ---------  ------ ---------  ------ ---------  ------ ---------
  4  2048MB Active       ---               ---               ---
       ---               ---               ---               ---
  6  2048MB Active       ---               ---               ---
       ---               ---               ---               ---

       Partition Total Memory:    4096
      Partition Active Memory:    4096
Partition Deconfigured Memory:       0

* status is scheduled to change on next boot.

Information Menu: Enter command >
```

Listing Cell Processors and Memory [HP-UX]

Use the **parstatus** command with various options to list cell processor and memory configurations from HP-UX.

You can check these details for any cell in the complex.

Step 1. Login to HP-UX running on any of the system's nPartitions.

You can list processor and memory details from any nPartition.

Step 2. Issue the `parstatus` command to view cell hardware details including processor and memory configurations.

Use any of the following `parstatus` commands to view cell hardware information:

- **parstatus -V -c*#***

  List detailed processor and memory configuration information for the specified cell.

- **parstatus -C**

  List brief processor and memory information for all cells in the entire complex.

- **parstatus -V -p*#***

  List brief processor and memory information for all cells assigned to the specified nPartition.

The following example shows the `parstatus -V -c0` command's output. This presents detailed processor and memory info for cell 0 in cabinet 0.

```
# parstatus -V -c0
[Cell]
Hardware Location      : cab0,cell0
Global Cell Number     : 0
Actual Usage           : active core
Normal Usage           : base
Connected To           : cab0,bay1,chassis3
Core Cell Capable      : yes
Firmware Revision      : 10.0
Failure Usage          : activate
Use On Next Boot       : yes
Partition Number       : 0
Partition Name         : feshd5a

[CPU Details]
Type  : 5C70
Speed : 552 MHz
CPU  Status
===  ======
 0   ok
 1   ok
 2   ok
 3   ok
CPUs
===========
OK     : 4
Deconf : 0
Max    : 4

[Memory Details]
DIMM Size (MB) Status
==== ========= =========
0A   512       ok
0B   512       ok
0C   512       ok
0D   512       ok
Memory
=========================
DIMM OK       : 4
DIMM Deconf   : 0
Max DIMMs     : 32
Memory OK     : 2.00 GB
Memory Deconf : 0.00 GB

#
```

Listing Cell Processors and Memory [Partition Manager]

Use the **Cell —> Show Cell Details** action, **CPUs/Memory** tab to list cell processor and memory details from Partition Manager.

Step 1. Run Partition Manager (`/opt/parmgr/bin/parmgr`) or access it from SAM or a Web browser.

Step 2. On the left of the primary window, select the nPartition to which the cell is assigned, or select Available Resources if the cell is unassigned.

Step 3. On the right of the primary window, select the cell whose processor and memory details you want to list, then select the **Cell —> Show Cell Details** menu item.

Step 4. Click the **CPUs/Memory** tab to list the selected cell's processor and memory configurations.

Deconfiguring Cells, Processors, and Memory

You can **deconfigure (make inactive) a cell** that is assigned to an nPartition by setting its use-on-next-boot value to “n” (do not use). This causes the cell to remain assigned to the nPartition, but the cell will be inactive the next time its nPartition boots, meaning the cell’s resources will not be used.

You also can **deconfigure processors and memory** from any cell that is assigned to an nPartition. This causes the deconfigured processors or memory to not be available for use by the cell or its nPartition.

Whenever you configure or deconfigure cells, processors, or memory, *you must reboot the corresponding nPartition* for the configuration change to take effect.

You can use the following procedures:

- *Deconfiguring Cells, Processors, and Memory [BCH]* on page 337
- *Deconfiguring Cells, Processors, and Memory [HP-UX]* on page 339
- *Deconfiguring Cells, Processors, and Memory [Partition Manager]* on page 340

Deconfiguring Cells, Processors, and Memory [BCH]

Use the Configuration menu **CELLCONFIG** or **CPUCONFIG** command, or Service menu **DIMMDEALLOC** command, to configure and deconfigure cells, processors, and memory from the BCH interface.

Step 1. Access the BCH interface for the nPartition whose cells, processors, or memory you want to configure or deconfigure.

Step 2. To change *cell or processor* configurations, access the Configuration menu. To change *memory* configurations, access the Service menu.

To access the Configuration menu, enter **CO** at the BCH interface’s main menu. To access the Service menu enter **SER**.

Step 3. Configure or deconfigure the cell, processors, or memory.

You *cannot* deconfigure the last cell, processor, or DIMM rank. Cells must have at least one configured processor or DIMM rank, and nPartitions must have at least one configured cell.

- **Cells**

  From the Configuration menu, use the `CELLCONFIG` command to configure or deconfigure a cell in the nPartition.

  `CELLCONFIG` ***`#`*** **`OFF`** deconfigures the cell (*#*) by setting its use-on-next-boot value to “n” (do not use).

  `CELLCONFIG` ***`#`*** **`ON`** configures the specified cell (*#*) by setting its use-on-next-boot value to “y” (use the cell).

  Enter **`HELP CELLCONFIG`** for details.

- **Processors**

  From the Configuration menu, use the `CPUCONFIG` command to configure or deconfigure a processor on a cell in the nPartition.

  `CPUCONFIG` ***`cell cpu`*** **`OFF`** deconfigures the specified processor (*`cpu`*) on the specified cell (*`cell`*).

  `CPUCONFIG` ***`cell cpu`*** **`ON`** configures the specified processor on the cell

  Enter **`HELP CPUCONFIG`** for details.

- **Memory**

  DIMMs operate in ranks of four. Each rank is numbered (0, 1, 2, and so on) and the DIMMs in the rank are lettered (A to D). For example, rank 0 includes DIMMs 0A, 0B, 0C, and 0D.

  From the Service menu, use the `DIMMDEALLOC` command to configure or deconfigure memory on a cell in the nPartition.

  When you deallocate a DIMM, *all other DIMMs in the rank also will not be used* the next time the nPartition boots.

  `DIMMDEALLOC` ***`cell dimm`*** **`OFF`** deconfigures the specified DIMM (*`dimm`*) on the cell (*`cell`*) indicated.

  `DIMMDEALLOC` ***`cell dimm`*** **`ON`** configures the specified DIMM on the cell.

For example, `DIMMDEALLOC 0 1B OFF` sets DIMM 1B on cell 0 to be deallocated the next time the nPartition boots, and as a result all other DIMMs in the same rank (1A, 1C, and 1D) also will not be used.

Enter **`HELP DIMMDEALLOC`** for details.

Step 4. Reboot the nPartition using the **`REBOOT`** command.

Whenever changing cell, processor, or memory configurations you must reboot the corresponding nPartition to allow the configuration changes to take place.

Deconfiguring Cells, Processors, and Memory [HP-UX]

Use the **`parmodify -p# -m#::[y|n]:`** command to configure or deconfigure (makes inactive) cells from the HP-UX command line.

Step 1. Login to HP-UX on the nPartition whose cell you want to configure or deconfigure.

Step 2. Issue the **`parstatus -C`** command to list all cells, their nPartition assignments, their actual (current) usage, and their use-on-next-boot values.

Step 3. Issue the **`parmodify -p# -m#::[y|n]:`** command to configure or deconfigure the specified cell (`-m#`) from the nPartition (`-p#`).

The **`parmodify -p# -m#::n:`** command deconfigures the specified cell (`-m#`). This sets the cell's use-on-next-boot value to "n" (do not use).

The **`parmodify -p# -m#::y:`** command configures the specified cell to be used. This sets the cell's use-on-next-boot value to "y" (use the cell).

The partition number (`-p#`) you specify must be the local nPartition number, which you can list using the `parstatus -w` command.

Step 4. Reboot the nPartition using the **`shutdown -R`** command.

You must reboot the partition to allow the new use-on-next-boot values to take effect.

The `shutdown -R` command performs a reboot for reconfig for the nPartition, which allows *all cells* to reboot, including any currently inactive cells in the nPartition.

Deconfiguring Cells, Processors, and Memory [Partition Manager]

Use the **Partition —> Modify Partition** menu, **Change Cell Attributes** tab to configure and deconfigure (makes inactive) cells using Partition Manager.

Step 1. Run Partition Manager (`/opt/parmgr/bin/parmgr`) or access it from SAM or a Web browser.

Step 2. Select the nPartition whose cell configuration you want to modify, then select the **Partition —> Modify Partition** menu item.

Step 3. Click the **Change Cell Attributes** tab.

Step 4. Select the cell whose configuration you want to modify, then click the **Modify Cell(s)** button.

Step 5. In the Modify Cell Attributes window, set the cell's use-on-next-boot value, then click the **OK** button.

To configure the cell to be used set use-on-next-boot to "yes".

To configure the cell *to not* be used set use-on-next-boot to "no".

Step 6. Exit Partition Manager, then reboot the corresponding nPartition using the **`shutdown -R`** command.

You must reboot the nPartition whose use-on-next-boot cell values you changed to allow the new use-on-next-boot values to take effect.

Listing the Server Product Number and Serial Number

You can list the *product number* and the *serial number* for your server complex by using software commands and utilities.

You can use the following procedures:

- *Listing Product and Serial Numbers [Service Processor]* on page 341
- *Listing Product and Serial Numbers [BCH]* on page 342
- *Listing Product and Serial Numbers [HP-UX]* on page 343
- *Listing Product and Serial Numbers [Partition Manager]* on page 343

Listing Product and Serial Numbers [Service Processor]

Use the Command menu **ID** command to list the system complex's product number and serial number from the service processor.

Step 1. Login to the system's service processor and enter **CM** to access the Command menu.

Step 2. Issue the service processor Command menu's **ID** command to display the system complex information, including the product and serial numbers.

Step 3. Type **n** (or type **q**) to *not modify* the system complex information that was displayed.

```
GSP:CM> ID

This command allows you to change certain fields in the Stable complex
configuration portion of the complex profile.

Retrieving the stable complex configuration portion of the complex profile.

    GSP modifiable stable complex configuration data fields.
    Model String          : 9000/800/SD64000
    Complex System Name   : feshd5
    Complex Serial Number : USR2024FP1
    Original Product Number: A5201A
    Current Product Number : A5201A
    Enterprise Id         :
```

```
    Do you want to modify any of this information? (Y/[N]) n

    -> No fields modified.
GSP:CM>
```

Listing Product and Serial Numbers [BCH]

Use the Information menu **CID** command to list the system complex's product number and serial number from the BCH interface.

Step 1. Access the BCH interface for any nPartition in the complex.

You can list the complex's product number and serial number from any nPartition in the server.

Step 2. Access the BCH Information menu by entering **IN** from the BCH Main menu.

If you are at a BCH menu other than the Main menu, enter `MA` to go to the Main menu and then enter `IN` to access the Information menu.

Step 3. From the BCH Information menu, use the **CID** command to list the complex's ID information, including the product number and serial number.

The `CID` BCH command (also: `ComplexID`) displays information that is stored as part of the server's Stable Complex Configuration Data.

```
Information Menu: Enter command > CID

COMPLEX ID INFORMATION

                  Complex Name: feshd4
                  Model String: 9000/800/SD16000
       Original Product Number: A5201A
        Current Product Number: A5201A
                 Serial Number: USR2025FP2
                 Enterprise ID: 0x2020202020202020
     Number of Supported Cells: 32
       Complex Revision Number: 1.0

Information Menu: Enter command >
```

Listing Product and Serial Numbers [HP-UX]

Use the **`parstatus -X`** command to list a system complex's product number and its serial number from HP-UX.

Step 1. Login to HP-UX running on any of the system's nPartitions.

You can list the product and serial numbers from any nPartition.

Step 2. Issue the **`parstatus -X`** command and option to display system complex attributes, including the product and serial numbers.

```
# parstatus -X
[Complex]
   Complex Name : feshd5
   Complex Capacity
     Compute Cabinet (8 cell capable) : 2
     IO Expansion Cabinet             : 1
   Active GSP Location : cabinet 0
   Model : 9000/800/SD64000
   Serial Number : USR2024FP1
   Current Product Number : A5201A
   Original Product Number : A5201A
   Complex Profile Revision : 1.0
   The total number of Partitions Present : 2

#
```

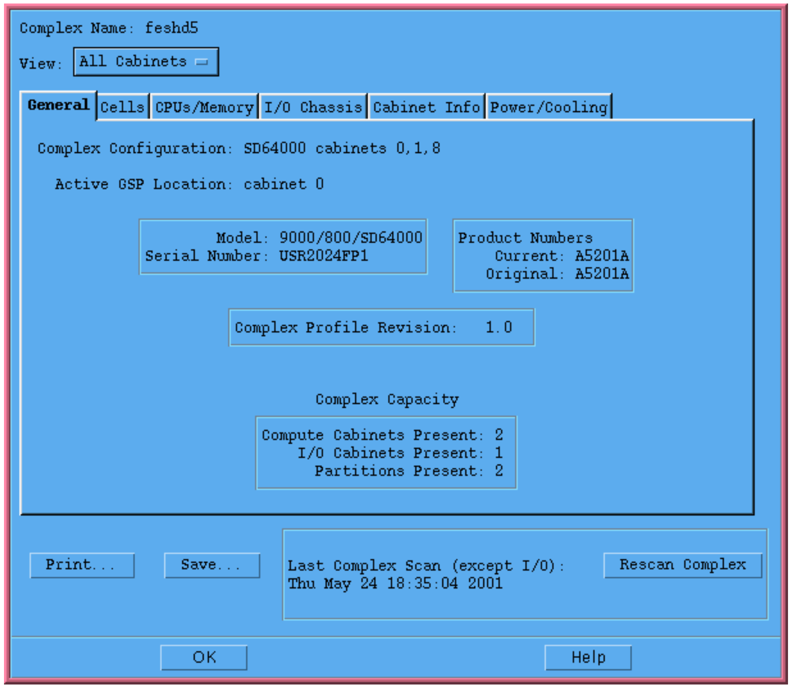
Listing Product and Serial Numbers [Partition Manager]

Use the **Complex —> Show Complex Details** menu to list the system complex's product and serial numbers using Partition Manager.

Step 1. Run Partition Manager (`/opt/parmgr/bin/parmgr`) or access it from SAM or a Web browser.

Step 2. Select the **Complex —> Show Complex Details** menu item.

The Complex Details window displays the complex's product and serial numbers through the **General** tab.


Complex Name: feshd5
View: All Cabinets
General
Cells
CPUs/Memory
I/O Chassis
Cabinet Info
Power/Cooling
Complex Configuration: SD64000 cabinets 0,1,8
Active GSP Location: cabinet 0
Model: 9000/800/SD64000
Serial Number: USR2024FP1
Product Numbers
Current: A5201A
Original: A5201A
Complex Profile Revision: 1.0
Complex Capacity
Compute Cabinets Present: 2
I/O Cabinets Present: 1
Partitions Present: 2
Print...
Save...
Last Complex Scan (except I/O):
Thu May 24 18:35:04 2001
Rescan Complex
OK
Help

Checking Blower and Fan Status

You can remotely check the operating status of a server complex's blowers and fans by using software commands and utilities.

NOTE Different HP server and cabinet models have different blower and fan configurations.

You can use the following procedures:

- *Checking Fan Status [Service Processor]* on page 345
- *Checking Fan Status [HP-UX]* on page 346
- *Checking Fan Status [Partition Manager]* on page 347

Checking Fan Status [Service Processor]

Use the **PS** command's "Cabinet" option to check fan and blower status from the service processor Command menu.

Step 1. Login to the complex's service processor and enter **CM** to access the Command menu.

Step 2. Issue the **PS** command, select the "Cabinet" option, and specify the cabinet number whose fan status you want to check.

Checking Fan Status [HP-UX]

Use the **parstatus -B** or **parstatus -V -b#** command to list the status of blowers and fans from HP-UX.

Step 1. Login to HP-UX running on any of the system's nPartitions.

You can check blower and fan details from HP-UX running on any nPartition.

Step 2. Issue the **parstatus -B** or **parstatus -V -b#** command.

The **parstatus -B** command gives a brief summary of all cabinets including fan status. The **parstatus -V -b#** command gives detailed fan status for cabinet (-b#) that you specify.

```
# parstatus -B
[Cabinet]
                  Cabinet   I/O       Bulk Power  Backplane
                  Blowers   Fans      Supplies    Power Boards
                  OK/       OK/       OK/         OK/
Cab               Failed/   Failed/   Failed/     Failed/
Num Cabinet Type  N Status  N Status  N Status    N Status     GSP
=== ============  ========= ========= ==========  ============ ======
 0  SD64000        4/ 0/ N   4/ 1/ ?   6/ 0/ N+    3/ 0/ N+    active
 1  SD64000        4/ 0/ N+  5/ 0/ ?   6/ 0/ N+    3/ 0/ N+    none
 8  IOX            2/ 0/ N+  4/ 0/ N+  2/ 0/ N+    N/A         none
                             4/ 0/ N+

Notes: N+ = There are one or more spare items (fans/power supplies).
       N  = The number of items meets but does not exceed the need.
       N- = There are insufficient items to meet the need.
       ?  = The adequacy of the cooling system/power supplies is unknown.

# parstatus -V -b0
[Cabinet]
                  Cabinet   I/O       Bulk Power  Backplane
                  Blowers   Fans      Supplies    Power Boards
                  OK/       OK/       OK/         OK/
Cab               Failed/   Failed/   Failed/     Failed/
Num Cabinet Type  N Status  N Status  N Status    N Status     GSP
=== ============  ========= ========= ==========  ============ ======
 0  SD64000        4/ 0/ N   4/ 1/ ?   6/ 0/ N+    3/ 0/ N+    active

Cabinet Blowers
===============
Fan 0  ok
Fan 1  ok
Fan 2  ok
```

```
Fan 3  ok

I/O Fans
==============
Fan 0  ok
Fan 1  failed
Fan 2  ok
Fan 3  ok
Fan 4  ok

Bulk Power Supplies(BPS)
========================
Power Supply  0  ok

....
```

Checking Fan Status [Partition Manager]

Use the **Complex —> Show Complex Details** menu, **Power/Cooling** tab to list fan status from Partition Manager.

Step 1. Run Partition Manager (`/opt/parmgr/bin/parmgr`) or access it from SAM or a Web browser.

Step 2. Select the **Complex —> Show Complex Details** menu item.

This displays the Complex Details window.

Step 3. Click the **Power/Cooling** tab to view the panel that has information about the status of the blowers and fans in the complex.

Complex Health Analysis of a Server

You can quickly check for hardware problems in an nPartition server by using Partition Manager's "Analyze Complex Health" feature.

This feature scans the server complex and uses problem detectors to check the operating status of cells, I/O chassis, fans and blowers, and power supplies.

Partition Manager automatically performs this task when you launch the application; if any problems are detected then the complex health analysis is displayed before Partition Manager's primary window.

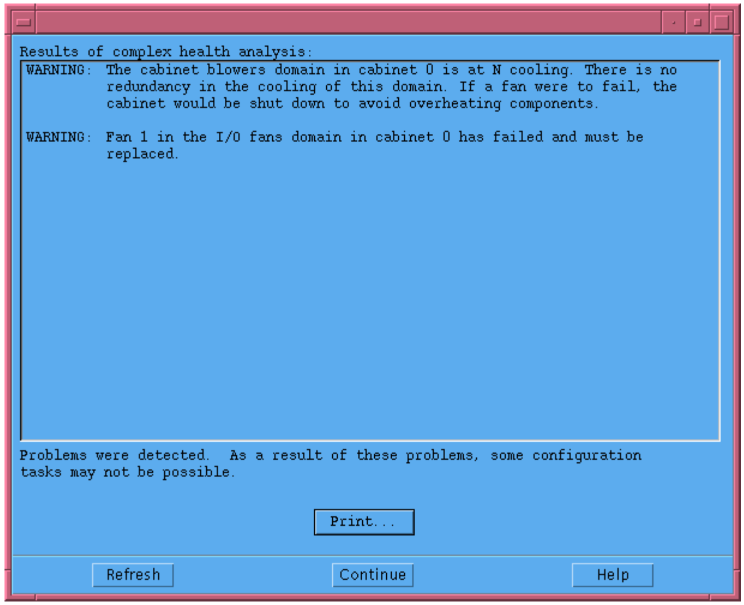
Analyzing Server Complex Health [Partition Manager]

Use the **Complex —> Analyze Complex Health** action to quickly check a server complex's operating status from Partition Manager.

Step 1. Run Partition Manager (`/opt/parmgr/bin/parmgr`) or access it from SAM or a Web browser.

Step 2. Select the **Complex —> Analyze Complex Health** action.

Partition Manager displays a window that reports the hardware analysis results. To re-analyze the server's health click the **Refresh** button.


Results of complex health analysis:
WARNING: The cabinet blowers domain in cabinet 0 is at N cooling. There is no redundancy in the cooling of this domain. If a fan were to fail, the cabinet would be shut down to avoid overheating components.
WARNING: Fan 1 in the I/O fans domain in cabinet 0 has failed and must be replaced.
Problems were detected. As a result of these problems, some configuration tasks may not be possible.
Print...
Refresh
Continue
Help

Listing the Cabinets in a Server Complex

You can list all cabinets in a server complex by using software commands and utilities, which allow you to determine the complex's cabinet configuration from a remote location.

You can use the following procedures:

- *Listing Cabinets [Service Processor]* on page 350
- *Listing Cabinets [HP-UX]* on page 351
- *Listing Cabinets [Partition Manager]* on page 351

Listing Cabinets [Service Processor]

Use the Command menu **DU** command to list all cabinets in the server complex using the service processor.

Step 1. Login to the server's service processor and enter **CM** to access the Command menu.

Step 2. Issue the **DU** command to list all cabinets in the server complex.

```
GSP:CM> DU

The following GSP bus devices were found:
+----+-----+-----------+----------------+-----------------------------------+
|    |     |           |                |              Core IOs             |
|    |     |           |                | IO Bay | IO Bay | IO Bay | IO Bay |
|    |     |   UGUY    |     Cells      |   0    |   1    |   2    |   3    |
|Cab.|     |           |                |IO Chas.|IO Chas.|IO Chas.|IO Chas.|
| #  | GSP | CLU | PM  |0 1 2 3 4 5 6 7 |0 1 2 3 |0 1 2 3 |0 1 2 3 |0 1 2 3 |
+----+-----+-----+-----+----------------+--------+--------+--------+--------+
|  0 |  *  |  *  |  *  |*   *   *   *   |      * |      * |        |        |
|  1 |     |  *  |  *  |*   *   *       |        |      * |        |        |
|  8 |     |  *  |  *  |                |      * |  *     |        |        |

GSP:CM>
```

Listing Cabinets [HP-UX]

Use the **parstatus -B** or **parstatus -V -b#** command to list cabinet details from HP-UX.

Step 1. Login to HP-UX running on any of the server's nPartitions.

You can list cabinet information from any nPartition.

Step 2. Issue the **parstatus -B** command and option to list all cabinets and their current status.

For more information, issue the **parstatus -V -b#** command for details on the specified cabinet number (-b#).

```
# parstatus -B
[Cabinet]
                 Cabinet   I/O       Bulk Power  Backplane
                 Blowers   Fans      Supplies    Power Boards
                 OK/       OK/       OK/         OK/
Cab              Failed/   Failed/   Failed/     Failed/
Num Cabinet Type N Status  N Status  N Status    N Status      GSP
=== ============ ========= ========= ==========  ============  ======
 0  SD64000       4/ 0/ N+  5/ 0/ ?   5/ 0/ N+    3/ 0/ N+     active
 1  SD64000       4/ 0/ N+  5/ 0/ ?   5/ 0/ N+    3/ 0/ N+     none

Notes: N+ = There are one or more spare items (fans/power supplies).
       N  = The number of items meets but does not exceed the need.
       N- = There are insufficient items to meet the need.
       ?  = The adequacy of the cooling system/power supplies is unknown.

#
```

Listing Cabinets [Partition Manager]

Use the **Complex —> Show Complex Details** menu, **Cabinet Info** panel to list the cabinets in a server complex from Partition Manager.

Step 1. Run Partition Manager (/opt/parmgr/bin/parmgr) or access it from SAM or a Web browser.

Step 2. Select the **Complex —> Show Complex Details** menu item, then click the **Cabinet Info** panel.

Listing the nPartitions in a Server

You can list details about the nPartitions defined in a server complex, including the partition numbers, nPartition names, and the cells assigned to the nPartitions.

You can use the following procedures:

- *Listing nPartitions [Service Processor]* on page 352
- *Listing nPartitions [HP-UX]* on page 353
- *Listing nPartitions [Partition Manager]* on page 353

Listing nPartitions [Service Processor]

Use the Command menu **CP** command to list all nPartitions from the service processor.

Step 1. Login to the server's service processor and enter **CM** to access the Command menu.

Step 2. Issue the service processor Command menu's **CP** command to list all nPartitions defined in the server complex.

The CP command lists the partition number (Part 0, Part 1, and so on) for each nPartition and lists which cells are assigned to each nPartition.

```
GSP:CM> CP

--------------
Cabinet | 0  |
--------+----+
 Slot   |0123|
--------+----+
Part  0 |*...|
Part  1 |.*..|

GSP:CM>
```

Listing nPartitions [HP-UX]

Use the **`parstatus -P`** command to list a server's nPartitions from HP-UX.

Step 1. Login to HP-UX running on any of the server's nPartitions.

You can list all nPartitions from any nPartition in the server.

Step 2. Issue the **`parstatus -P`** command and option to list all nPartitions and their current status.

```
# parstatus -P
[Partition]
Par                # of  # of I/O
Num Status         Cells Chassis  Core cell  Partition Name (first 30 chars)
=== ============ ===== ======== ========== ==============================
 0  active         2      2     cab0,cell0 jules00
 1  active         2      2     cab0,cell4 jules01
#
```

Listing nPartitions [Partition Manager]

View the left side of the primary window to see a display of all nPartitions in the server from Partition Manager.

Step 1. Run Partition Manager (`/opt/parmgr/bin/parmgr`) or access it from SAM or a Web browser.

Step 2. Access Partition Manager's primary window, and view the list of nPartitions shown on the window's left side.

Each nPartition is listed separately. Also listed here are the server's Available Resources, which are not assigned to any nPartition.

Naming or Renaming a Server Complex

You can assign a name for each server complex in order to better identify the complex as you work with it.

Several commands and utilities display the server complex name as part of their output and interfaces. For example, some nPartition commands and Partition Manager list the complex name.

The server complex name only serves as a helpful identifier; changing the name does not affect the way in which commands and utilities interact with the complex.

You can use the following procedures:

- *Renaming a Server Complex [Service Processor]* on page 354
- *Renaming a Server Complex [Partition Manager]* on page 355

The server complex name is stored as part of the server's complex profile (part of its "stable complex configuration" data).

NOTE Each server complex name has up to 20 characters, which can include upper- and lowercase letters; numbers; and dashes, underscores, periods, and spaces ("-" "_" "." and " ").

Renaming a Server Complex [Service Processor]

Use the Command menu **`ID`** command to list and modify the server complex name from the service processor.

Step 1. Login to the server's service processor and enter **`CM`** to access the Command menu.

Step 2. Issue the service processor Command menu's **`ID`** command to list the complex's name.

The `ID` command lists some of the current server complex's "stable complex configuration" data, including the complex name.

```
GSP:CM> ID

This command allows you to change certain fields in the Stable complex
configuration portion of the complex profile.

Retrieving the stable complex configuration portion of the complex profile.

    GSP modifiable stable complex configuration data fields.
    Model String          : 9000/800/SD64000
    Complex System Name    : feshd5
    Complex Serial Number  : USR2024FP1
    Original Product Number: A5201A
    Current Product Number : A5201A
    Enterprise Id          :

    Do you want to modify any of this information? (Y/[N])
```

Step 3. Specify whether you want to modify the complex profile, including its name.

You should only modify the “complex system name”. Do not change the model string, serial number, or other details used by commands, utilities, and licensing tools.

To cancel the changes at any time, enter **q** to quit the `ID` command without modifying the complex profile data.

Renaming a Server Complex [Partition Manager]

Use the **Complex —> Set Complex Name** action to rename a server complex using Partition Manager.

Step 1. Run Partition Manager (`/opt/parmgr/bin/parmgr`) or access it from SAM or a Web interface.

Step 2. Select the **Complex —> Set Complex Name** menu item.

Step 3. Enter the complex’s new name in the pop-up window.

The pop-up window shows the current complex name. If no name was previously assigned, the default name is “MyComplex”.

Step 4. Click **OK** to set the new name or click **Cancel** to keep the previously assigned name.

Checking for Power Supply Failures

From remote locations, you can check the operating status of power supplies by using software commands and utilities.

Different HP server and cabinet models have different supply requirements and configurations.

You can use the following procedures:

- *Checking Power Supply Status [Service Processor]* on page 356
- *Checking Power Supply Status [HP-UX]* on page 356
- *Checking Power Supply Status [Partition Manager]* on page 358

Checking Power Supply Status [Service Processor]

Use the Command menu **`PS`** command's "Cabinet" option to check power status from the service processor.

Step 1. Login to the complex's service processor and enter **`CM`** to access the Command menu.

Step 2. Issue the **`PS`** command, select the "Cabinet" option, and specify the cabinet number whose power status you want to check.

Checking Power Supply Status [HP-UX]

Use the **`parstatus -B`** or **`parstatus -V -b`***#* command to list the status of power supplies from HP-UX.

Step 1. Login to HP-UX running on any of the system's nPartitions.

You can check power details from HP-UX running on any nPartition.

Step 2. Issue the **`parstatus -B`** command for a brief summary of all cabinets including power status, or issue the **`parstatus -V -b`***#* command for detailed power status for a specific cabinet (`-b`*#*) whose details you want to view.

- The `parstatus -B` command summarizes the power status for all cabinets in the system complex.
- The `parstatus -V -b#` command displays a detailed status ("ok" or "failed") for all power supplies in the specified cabinet (`-b#`).

The following example shows power supply details for cabinet number 0, which has one failed bulk power supply (Power Supply 3).

```
# parstatus -V -b0
[Cabinet]
                 Cabinet   I/O       Bulk Power   Backplane
                 Blowers   Fans      Supplies     Power Boards
                 OK/       OK/       OK/          OK/
Cab              Failed/   Failed/   Failed/      Failed/
Num Cabinet Type N Status  N Status  N Status     N Status       GSP
=== ============ ========= ========= ==========   ============   ======
 0  SD32000       4/ 0/ N+  5/ 0/ ?   3/ 1/ N      3/ 0/ N+      active

Cabinet Blowers
===============
Fan 0  ok
Fan 1  ok
Fan 2  ok
Fan 3  ok

I/O Fans
==============
Fan 0  ok
Fan 1  ok
Fan 2  ok
Fan 3  ok
Fan 4  ok

Bulk Power Supplies(BPS)
========================
Power Supply  0  ok
Power Supply  1  ok
Power Supply  3  failed
Power Supply  4  ok

Backplane Power Boards
======================
Power Supply  0  ok
Power Supply  1  ok
Power Supply  2  ok


Notes: N+ = There are one or more spare items (fans/power supplies).
       N  = The number of items meets but does not exceed the need.
```

```
N- = There are insufficient items to meet the need.
?  = The adequacy of the cooling system/power supplies is unknown.

#
```

Checking Power Supply Status [Partition Manager]

Use the **Complex —> Show Complex Details** action, **Power/Cooling** tab to list power status from Partition Manager.

Step 1. Run Partition Manager (`/opt/parmgr/bin/parmgr`) or access it from SAM or a Web browser.

Step 2. Select the **Complex —> Show Complex Details** menu item.

This displays the Complex Details window.

Step 3. Click the **Power/Cooling** tab to bring up the panel that contains information about the status of the power supplies in the complex.

Checking for Memory DIMM Failures

You can list cell memory configurations and check for memory DIMM failures using software tools and utilities.

You can use the following procedures:

- *Checking Memory Status [HP-UX]* on page 359
- *Checking Memory Status [Partition Manager]* on page 361

Checking Memory Status [HP-UX]

Use the **`parstatus -V -c#`** command to list a cell's memory status, including any DIMM failures, from HP-UX.

Step 1. Login to HP-UX running on any of the system's nPartitions.

You can check memory details for any cell from any nPartition.

Step 2. Issue the **`parstatus -V -c#`** command to list detailed information about the specified cell (`-c#`).

The detailed information `parstatus` reports includes a list of all DIMMs (memory modules) installed on the cell, and the status of each DIMM.

Any memory listed as "failed" either has failed self-tests or has been software deconfigured by the Boot Console Handler (BCH) Service menu's DIMMDEALLOC command.

NOTE For any DIMM that fails or is deallocated, *all other DIMMs in the same rank also are deallocated*. All four DIMMs within the same rank must pass self-test and must be allocated for the rank to be made available for use by the cell and its nPartition.

In the following example, eight DIMMs (0A–0D and 1A–1D) are installed and are available ("ok") for use by the cell's nPartition.

```
# parstatus -V -c0
[Cell]
Hardware Location      : cab0,cell0
```

```
Global Cell Number     : 0
Actual Usage           : active core
Normal Usage           : base
Connected To           : cab0,bay0,chassis1

....

[CPU Details]
Type  : 5E70
Speed : 750 MHz
CPU  Status
===  ======
 0   ok
 1   ok
 2   ok
 3   ok
CPUs
===========
OK     : 4
Deconf : 0
Max    : 4

[Memory Details]
DIMM Size (MB) Status
==== ========= =========
0A   512       ok
0B   512       ok
0C   512       ok
0D   512       ok
1A   512       ok
1B   512       ok
1C   512       ok
1D   512       ok
Memory
=========================
DIMM OK       : 8
DIMM Deconf   : 0
Max DIMMs     : 32
Memory OK     : 4.00 GB
Memory Deconf : 0.00 GB

#
```

In the above example, if any DIMM had failed its status would be “failed” and all other DIMMs in its rank (for instance, rank 0 or rank 1) also would be listed as failed.

Any one or more of the failed DIMMs might have been software deallocated or might have failed self tests. In either case, all DIMMs in the rank automatically are deallocated when any of the rank's DIMMs fails or is deallocated.

Checking Memory Status [Partition Manager]

Use the **Cell —> Show Cell Details** action, **CPUs/Memory** tab to list a cell's memory status, including any DIMM failures, from Partition Manager.

Step 1. Run Partition Manager (`/opt/parmgr/bin/parmgr`) or access it from SAM or a Web browser.

Step 2. On the left of the primary window, select the nPartition to which the cell is assigned, or select Available Resources if the cell is unassigned.

Step 3. On the right of the primary window, select the cell whose memory details you want to list, then select the **Cell —> Show Cell Details** menu item.

Step 4. Click the **CPUs/Memory** tab to display the status of memory modules (DIMMs) for the cell.

Checking for Processor Failures

You can list cell processor configuration, including any processor failures, using software tools and utilities.

You can use the following procedures:

- *Checking Processor Status [HP-UX]* on page 362
- *Checking Processor Status [Partition Manager]* on page 363

Checking Processor Status [HP-UX]

Use the **`parstatus -V -c#`** command to list a cell's processor status, including any failures, from HP-UX.

Step 1. Login to HP-UX running on any of the system's nPartitions.

You can check processor details for any cell from any nPartition.

Step 2. Issue the **`parstatus -V -c#`** command to list detailed information about the specified cell (`-c#`).

The detailed information `parstatus` reports includes a list of all processors (CPUs) installed on the cell, and the status of each CPU.

Any processors listed as "failed" either have failed self-tests or have been software deconfigured by the Boot Console Handler (BCH) Configuration menu's `CPUCONFIG` command.

```
# parstatus -V -c5
[Cell]
Hardware location      : cab0,cell5
Global Cell Number     : 5
Actual Usage           : active base
Normal Usage           : base
Connected To           : -
Core Cell Capable      : no
Firmware Revision      : 32.5
Failure Usage          : activate
Use On Next Boot       : yes
Partition Number       : 0
Partition Name         : betty
```

```
[CPU Details]
Type  : 23664
Speed : 552 MHz
CPU  Status
===  ======
 0   ok
 1   failed
 2   ok
 3   failed
CPUs
===========
OK     : 2
Failed : 2
Max    : 4

....
```

Checking Processor Status [Partition Manager]

Use the **Cell —> Show Cell Details** menu, **CPUs/Memory** tab to list processor details and any failures using Partition Manager.

Step 1. Run Partition Manager (`/opt/parmgr/bin/parmgr`) or access it from SAM or a Web browser.

Step 2. On the left of the primary window, select the nPartition to which the cell is assigned, or select Available Resources if the cell is unassigned.

Step 3. On the right of the primary window, select the cell whose processor details you want to list, then select the **Cell —> Show Cell Details** menu item.

Step 4. Click the **CPUs/Memory** tab to display the processors installed in the selected cell and list the status for each processor.

8 Online Add and Replacement (OLAR) of PCI Cards

This chapter presents background information and procedures for performing PCI I/O card *online addition and replacement* (OLAR) on HP's nPartition servers.

The main procedures for adding and replacing PCI cards are:

- *Online Addition (OLA) for a PCI Card* on page 383
- *Online Replacement (OLR) for a PCI Card* on page 389

See the sections that follow for info on tools, requirements, limitations, and other PCI card OLAR details.

For an introduction to I/O hardware on nPartition servers, refer to the chapter *nPartition System Overviews* on page 31.

Also refer to the book *Configuring HP-UX for Peripherals* for further details on configuring PCI cards and related devices.

Overview of PCI Card Online Addition and Replacement (OLAR)

HP-UX 11i supports online addition and replacement (OLAR) of PCI I/O cards on HP nPartition systems. *Without rebooting* HP-UX running on an nPartition, you can add or replace PCI cards whose device drivers support OLAR.

The HP-UX PCI card OLAR features allow for:

- *Adding a new PCI card* without affecting other components of the system and *without* requiring a system reboot.

  This procedure is presented in *Online Addition (OLA) for a PCI Card* on page 383.

- *Replacing an existing PCI card* without affecting other components of the system and *without* requiring a system reboot.

  This procedure is presented in *Online Replacement (OLR) for a PCI Card* on page 389.

nPartition System OLAR Notes

The core I/O cards in HP nPartition servers are *not supported* for online addition or replacement.

On HP rp7405/rp7410 servers, two PCI card slots (chassis 0, slot 1 and chassis 1, slot 8) are dedicated for use by a combination SCSI/LAN card. This SCSI/LAN card is supported for online addition or replacement.

On HP rp7405/rp7410 and rp8400 servers, the PCI card slot latches must be used in conjunction with PCI card OLAR procedures when HP-UX is running. When a card slot latch is open, the slot is powered off and the slot is made accessible for card addition or replacement.

See the chapter *nPartition System Overviews* on page 31 and the section *PCI Card Slot Latches and Doorbells* on page 376 for other I/O hardware details.

OLAR Tools and Interfaces

This section discusses the tools available for performing online card addition and replacement tasks. Also given here are example uses of some of the commonly used OLAR commands and interfaces.

The primary tools for performing PCI card OLAR tasks are the System Administration Manager utility (SAM, `/usr/sbin/sam`) and Partition Manager (`/opt/parmgr/bin/parmgr`).

NOTE Always use SAM or Partition Manager when performing card addition and replacement tasks, because these tools perform critical resource analysis and properly execute all scripts and commands in the correct sequences.

The `/usr/bin/rad` command and other utilities also can provide useful OLAR-related information.

Details on the OLAR tools and interfaces are given in Table 8-1.

Table 8-1 Tools and Interfaces for PCI Card Online Addition and Replacement (OLAR)

OLAR Tool/Interface	Description
System Administration Manager (SAM)	`/usr/sbin/sam` SAM provides both a graphical user interface (GUI) and an equivalent text-based terminal interface. To perform OLAR tasks from SAM, enter the **Peripheral Devices > Cards** area, which gives a listing of all PCI cards currently available in the local nPartition. When you select a slot or path from this list, items in the **Actions** menu enable you to perform OLAR-related tasks on the selection.

Table 8-1 Tools and Interfaces for PCI Card Online Addition and Replacement (OLAR) (Continued)

OLAR Tool/Interface	Description
Partition Manager	`/opt/parmgr/bin/parmgr` Partition Manager's interface is exclusively a GUI. In addition to supporting nPartition administration tasks, Partition Manager has complete support for PCI card OLAR. To perform OLAR tasks from Partition Manager, select and "open" an I/O chassis in the primary window, which gives a listing of all PCI cards in the selected I/O chassis. When you select a slot or path from this list, items in the **I/O** menu enable you to perform OLAR-related tasks on the selection. Note that when using Partition Manager you can add or replace cards in the *local* nPartition's active I/O chassis only.
`/usr/bin/rad`	The `rad` command is a command-line interface for performing some OLAR tasks and getting system PCI card and driver status information for the local nPartition. HP recommends you perform online card add or replace tasks using SAM or Partition Manager—not `rad`. However, the `rad` command can be useful for listing status, getting additional slot or card details, and for independently managing card slot attention indicators (LEDs). The `rad` command reports *the default speed/frequency* for PCI slots when they are not occupied. When a slot is occupied with a card the `rad` command reports *the operating speed/frequency* for the card and slot. See *Example Uses of Common rad Commands* on page 370 for other details.

Table 8-1 Tools and Interfaces for PCI Card Online Addition and Replacement (OLAR) (Continued)

OLAR Tool/Interface	Description
Scripts in the directory `/usr/sbin/olrad.d/`	Each OLAR-capable card's driver(s) may have associated scripts in the `/usr/sbin/olrad.d/` directory. Each driver's script accepts the following command-line arguments: the *`action`* to perform and *`path`* for the slot for which the action is performed. The SAM and Partition Manager utilities automatically run these scripts, as needed, when performing PCI card online addition or replacement tasks. Normally, the driver OLAR scripts *are not invoked manually* by administrators. These scripts' actions include various preface–, prepare–, and post-replace tasks and post-add tasks.
I/O Chassis Hardware	Hardware in each I/O chassis includes PCI card slots, card slot dividers, and power and attention indicators (LEDs) for each slot. HP rp7405/rp7410 and rp8400 servers also have PCI card slot latches. Note that the I/O chassis locations and other features of I/O chassis hardware differ in the various HP nPartition server models. Refer to the chapter *nPartition System Overviews* on page 31 and the section *PCI Card Slot Latches and Doorbells* on page 376 for nPartition I/O hardware details.
`/usr/lib/libolrad.1`	The `libolrad` library is used by the `rad` command and other utilities such as SAM and Partition Manager to support PCI card slot inquiry and online addition and replacement tasks.

Example Uses of Common rad Commands

This section gives summaries and examples uses of common `rad` commands and options.

Table 8-2 rad Command Commonly Used Options

Command	Description
`rad -q`	Displays the status of all OLAR-capable slots in the *local* nPartition. Only displays slots in currently active I/O chassis.
`rad -N` *`path`*	Lists the slot ID for the specified hardware path (*`path`*). The `rad -N` command gives info for OLAR-capable slots only.
`rad -f` *`flag slot`*	Sets the attention indicator (LED) for the specified slot. The accepted *`flag`* arguments: `on`, `attention`, and `off`, where `attention` flashes the specified slot's LED, and `on` and `off` turn the LED steady-on or off.
`rad -c` *`slot`*	Displays the device information for all functions/interfaces at the specified slot.

See the *rad* (1M) manpage for complete details.

Example 8-1 Commonly Used rad Commands for nPartition I/O Details and Card Add/Replace Tasks

- **`rad -q`**

  To list basic slot, path, card, and driver details, use the `rad -q` command.

  The `rad -q` command lists all PCI card slots in the local nPartition, the corresponding hardware paths, and the current status of all slots and drivers.

```
# rad -q
                                                                    Driver(s)
Slot         Path         Bus    Speed   Power   Occupied     Suspended   Capable
0-0-0-1      0/0/8/0      64     33      On      No           N/A          N/A
0-0-0-2      0/0/10/0     80     33      On      No           N/A          N/A
0-0-0-3      0/0/12/0     96     33      On      Yes          No           Yes
0-0-0-4      0/0/14/0     112    33      On      No           N/A          N/A
0-0-0-5      0/0/6/0      48     33      On      Yes          Yes          Yes
0-0-0-6      0/0/4/0      32     33      On      No           N/A          N/A
```

```
0-0-0-7     0/0/2/0     16    33       On       No            N/A           N/A
0-0-0-8     0/0/1/0     8     33       On       No            N/A           N/A
#
```

- **`rad -N` *`path`***

  To determine which card slot corresponds to a hardware path, use the `rad -N` *`path`* command.

  The `rad -N` *`path`* command lists the card slot used by the device whose hardware path you specify. The slot is reported in *cabinet-bay-chassis-slot* format.

```
# rad -N 0/0/6/0/0.6.0
0-1-3-5
#
```

- **`rad -f` *`flag slot`***

  To flash, turn on, or turn off a PCI slot's attention indicator (LED) use the `rad -f` *`flag slot`* command.

```
# rad -f attention 0-1-3-1
# rad -f off 0-1-3-1
#
```

- **`rad -c` *`slot`***

  To list device information about a card use the `rad -c` *`slot`* command.

  The `rad -c` *`slot`* command lists details for all interfaces in a card, including the hardware path(s), driver name(s), and vendor and revision details.

```
# rad -c 0-1-3-5
Path                :0/0/6/0/0
Name                :c720
Device_ID           :000f
Vendor_ID           :1000
Subsystem_ID        :0000
Subsystem_Vendor_ID :0000
Revision_ID         :4
Class               :010000
Status              :0200
Command             :0156
Multi_func          :No
Bridge              :No
Capable_66Mhz       :No
Power_Consumption   :75

#
```

Requirements for OLAR Operations

To perform a card *addition* or card *removal-and-replacement* operation, the following system requirements must be met:

- The add or replace operation **must** be supported on the system hardware.

  All HP nPartition servers support PCI card OLAR.

- The replacement PCI card **must** be identical to the original card.

  When performing a card replacement task, you must use a replacement card that: uses the *same driver*, is manufactured by the *same vendor*, and is the *same hardware revision* as the original card being replaced.

  Use the **`rad -c`** ***`slot`*** command to list detailed driver, vendor, and revision information for a card in the specified slot.

- The PCI card's driver **must** support OLAR.

  Some PCI card drivers do not support OLAR.

  Use the **`rad -q`** command's output to check whether an existing PCI card's driver is capable of being suspended and resumed for card OLAR operations.

  Both SAM and Partition Manager also indicate in the "Status" column whether an existing PCI card's driver supports OLAR. All card slots except those listed as "not OLAR-able" are valid for online PCI card add or replace tasks.

- The card's driver **must** be loaded in the currently running HP-UX kernel.

  For online *addition*, the driver must be present in the kernel to support the new card.

  For online *replacement*, the replacement card must use the same revision of the driver as the original card.

  Use the SAM utility's **Kernel Configuration > Drivers** area to list all currently loaded drivers.

- The PCI slot **must** have firmware that supports OLAR.

  On all HP nPartition servers, the I/O firmware supports OLAR.

- The card **must** fit into the slot.

  On all nPartition servers, all PCI card slots can accept PCI cards keyed as universal cards.

  However, in nPartition server I/O chassis the PCI card slots also are physically keyed to accept cards that either are keyed as 5-volt cards or keyed as 3.3-volt cards.

  See the chapter *nPartition System Overviews* on page 31 for details on I/O slot capabilities.

- The resources supported by the card **must not** be critical for the server's continued operation.

  Resources that do not have a defined failover are considered to be "critical resources" that cannot be replaced online.

  For example, the following cards may be considered critical resources: cards that connect to disks for active filesystems, or a LAN card that provides the network port used by the current instance of SAM.

  You could replace a SCSI card used by a disk with an active filesystem, if the filesystem were mirrored on a different disk supported by a second SCSI card. In this case LVM could automatically failover to the mirrored disk, thus allowing you to perform an online replacement of the original SCSI card.

NOTE The core I/O cards on HP nPartition servers *are not* supported for online addition or replacement (OLAR) operations.

PCI Card OLAR Considerations

This section discusses two issues of possible concern when performing PCI card OLAR tasks: card slot power domains, and multi-function cards.

Power Domains

Each **power domain** consists of all the PCI card slots that are powered on or off together as a unit.

On HP nPartition servers *each slot is in its own power domain*, which allows each slot to power on or off without affecting any other slots.

Both SAM's and Partition Manager's OLAR procedures automatically check the effects of OLAR operations on the slots in a power domain. However, in all nPartition servers each slot's power is independent.

To list all slots in a power domain, use the **`rad -a`** ***`slot`*** command. For example, the following `rad` command output indicates that slot 0-1-3-5 is in its own power domain.

```
# rad -a 0-1-3-5
0-1-3-5
#
```

Multi-Function Cards

A **multi-function card** provides more than one function in a single PCI card that occupies one slot. For example: a dual-SCSI PCI card has two SCSI ports, and a combination SCSI/LAN PCI card has both a SCSI port and a LAN port. Such cards allow a single PCI card slot to provide services that otherwise would require two or more PCI cards.

A multi-function card has a separate hardware path for each function, and has a separate driver bound at each hardware path.

Both SAM's and Partition Manager's OLAR procedures automatically check for critical resources at all hardware paths of multi-function cards. These utilities also suspend and resume all drivers bound to multi-function cards as required for OLAR purposes.

To list all functions provided by a PCI card slot, use the **rad -h *slot*** command. For example, the following rad command lists all hardware paths associated with slot 0-1-3-8 (cabinet 0, bay 1, chassis 3, slot 8).

```
# rad -h 0-1-3-8
0/0/11/0/0
0/0/11/0/1
#
```

As the above example shows, slot 0-1-3-8 has two functions, one at each of the hardware paths listed.

To list all drivers bound to a multi-function card, use the **rad -c *slot*** command.

To list additional details about a multi-function card, use the **ioscan -H *path*** command and specify only the first three fields (*cell*/*SBA*/*LBA*) of the card's hardware path.

On HP nPartition servers, each card slot has its own local bus adapter (LBA) that is shared by all ports on the card that occupies the slot. For example, the following ioscan command lists two SCSI ports that are provided by the card at hardware path 0/0/11.

```
# ioscan -H 0/0/11
H/W Path        Class                       Description
======================================================
0/0/11                 ba                   Local PCI Bus Adapter (782)
0/0/11/0/0                   ext_bus        SCSI C896 Ultra2 Wide LVD
0/0/11/0/0.7                    target
0/0/11/0/0.7.0                     ctl      Initiator
0/0/11/0/1                   ext_bus        SCSI C896 Ultra2 Wide LVD
0/0/11/0/1.7                    target
0/0/11/0/1.7.0                     ctl      Initiator
#
```

PCI Card Slot Latches and Doorbells

NOTE This section applies only to HP rp7405/rp7410 and rp8400 servers.

This section introduces two features of HP rp7405/rp7410 and rp8400 server I/O chassis: PCI card slot latches and PCI card slot doorbell buttons.

PCI Card Slot Latches

Both HP rp7405/rp7410 and rp8400 servers have slot latches for all PCI card slots; each PCI card slot has its own latch.

Each PCI card slot latch can enable or disable power to its card slot and, when closed, the latch can secure a PCI card in place. These slot latches are accessible when you have removed the top cover from an HP rp7405/rp7410 or rp8400 server chassis.

Card slot latches are used both for *offline* PCI card procedures and for *online* PCI card procedures.

When a PCI slot latch is open, the slot is powered off.

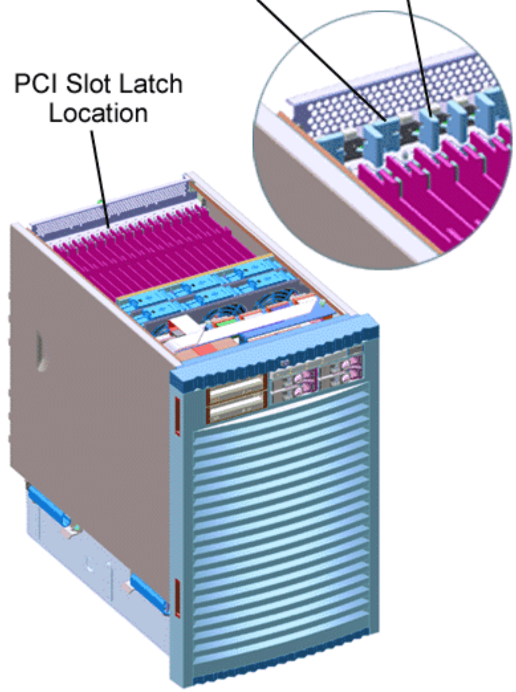
When a PCI slot latch is closed, the slot can have power enabled. However, note that when a slot's latch is closed the slot's power can be disabled as part of a PCI card OLAR procedure from SAM or Partition Manager, or as a manual operation performed from the HP-UX command-line or from a service processor (GSP or MP) command.

NOTE When HP-UX is running in the nPartition to which an I/O chassis belongs, you should use SAM or Partition Manager procedures to prepare a PCI card slot before opening or closing the slot's latch.

Otherwise you may encounter unpredictable results.

Figure 8-1 on page 377 shows positions of PCI card slot latches. While an HP rp8400 server is shown in Figure 8-1, HP rp7405/rp7410 card slot latches are available in the same location and operate identically.

Figure 8-1 **PCI Card Slot Latches (HP rp7405/rp7410 and rp8400)**



PCI Card Slot "Doorbells"

NOTE

This section applies only to HP rp7405/rp7410 and rp8400 servers.

The "doorbell" buttons currently have no functions. Pressing a card slot doorbell button has no impact on system operations.

On both HP rp7405/rp7410 and rp8400 servers, access to the doorbell buttons is prevented by a plastic covering.

The PCI card slot doorbell buttons are provided for future expanded functionality.

Determining PCI Card Slot Locations

This section describes how to determine which PCI card slot is used by a filesystem, network interface, or hardware path (such as a boot device path).

You may want to identify which PCI cards are used by critical and non-critical system resources when planning for card replacement or nPartition reconfiguration.

NOTE

While you can use the manual techniques described here to help identify which PCI cards support critical system resources, you should rely on the critical resource analysis that SAM and Partition Manager perform for a complete analysis of the services a card provides.

You can perform SAM or Partition Manager critical resource analysis for any card in an nPartition's active I/O chassis.

When performing a *card replacement* action, both SAM and Partition Manager automatically check for any critical system resources that would be affected by taking the card offline.

The manual procedures described here use the `rad -N` *`path`* command to identify which PCI card slot corresponds to the specified hardware path.

The following procedures are provided here:

- *Determining a Network Interface's PCI Card Slot* on page 379
- *Determining a Filesystem's PCI Card Slot* on page 379
- *Determining a Boot Device Path's PCI Card Slot* on page 380

To determine the actual physical location of a PCI card slot, based on the *cabinet-bay-chassis-slot* format that the `rad` command lists, refer to the chapter *nPartition System Overviews* on page 31 for an introduction to nPartition I/O hardware.

Determining a Network Interface's PCI Card Slot

This procedure determines which PCI card slot is used by a network interface.

Step 1. At the HP-UX command line, enter the **`/usr/sbin/lanscan`** command to list the local nPartition's LAN devices and status.

Step 2. Enter the **`rad -N`** ***`path`*** command, and specify the hardware path (*`path`*) for the network interface whose card slot information you want.

The `rad` command displays the hardware slot for the network interface's card, in *cabinet-bay-chassis-slot* format.

Example 8-2 Example of Determining a Network Interface's PCI Card Slot

```
# lanscan
Hardware Station        Crd Hdw   Net-Interface  NM  MAC       HP-DLPI DLPI
Path     Address        In# State NamePPA        ID  Type      Support Mjr#
0/0/0/1/0 0x0010832754E0 0   UP    lan0 snap0     1   ETHER     Yes     119
# rad -N 0/0/0/1/0
0-1-3-0
#
```

The above output indicates that the card is installed in cabinet 0, bay 1, chassis 3, slot 0 (`0-1-3-0`).

Determining a Filesystem's PCI Card Slot

This procedure determines which PCI card slot is used by a filesystem.

Step 1. At the HP-UX command line, enter the **`bdf -l`** command to list the local filesystems for the local nPartition.

Step 2. For the filesystem of interest, enter the following command:

`/usr/sbin/lvdisplay -v -k` ***`lvolume`*** **`| grep dev`**

Where *`lvolume`* is the device path of the filesystem's logical volume, as was displayed in the `bdf` command output.

Step 3. Enter the **`/usr/sbin/lssf`** ***`pvolume`*** command, and specify the physical volume path (*`pvolume`*) as was reported by the `lvdisplay` command.

For example: `lssf /dev/dsk/c0t5d0`, for a filesystem whose physical volume is `c0t5d0`.

Step 4. Enter the **`rad -N`** ***`path`*** command and specify the hardware path (*`path`*), which `lssf` reported as the "address" of the filesystem's physical volume.

The `rad` command displays the hardware slot for the filesystem's card, in *cabinet-bay-chassis-slot* format.

Example 8-3 Example of Determining a Filesystem's PCI Card Slot

This example determines which PCI card (0-1-3-5) supports the `/stand` filesystem.

```
# bdf -l
Filesystem          kbytes    used   avail %used Mounted on
/dev/vg00/lvol3     143360   42571   94743   31% /
/dev/vg00/lvol1    1025617   45445  877610    5% /stand
/dev/vg00/lvol8     512000  225212  269124   46% /var
/dev/vg00/lvol7    1015808  719619  277730   72% /usr
/dev/vg00/lvol4    1048576   77997  909975    8% /tmp
/dev/vg00/lvol6     794624  566033  214349   73% /opt
/dev/vg00/lvol5      20480    1190   18086    6% /home
# lvdisplay -vk /dev/vg00/lvol1 | grep dev
LV Name                     /dev/vg00/lvol1
VG Name                     /dev/vg00
   /dev/dsk/c0t5d0    256       256
# lssf /dev/dsk/c0t5d0
sdisk card instance 0 SCSI target 5 SCSI LUN 0 section 0 at address 0/0/6/0/0.5.0
/dev/dsk/c0t5d0
# rad -N 0/0/6/0/0.5.0
0-1-3-5
#
```

Determining a Boot Device Path's PCI Card Slot

This procedure determines which PCI card slot is used by an nPartition's boot path variable (such as PRI, HAA, or ALT).

Step 1. Issue the **`/usr/sbin/parstatus -V -p# | grep Path`** command, and specify the local nPartition's partition number (`-p`*#*).

This command displays the boot path variable settings for the nPartition.

To determine the local partition number, enter the `parstatus -w` command.

Step 2. Issue the **`rad -N`** ***`path`*** command and specify the hardware path (*`path`*) of the boot path variable of interest.

The `rad` command displays the hardware slot corresponding to the boot path, in *cabinet-bay-chassis-slot* format.

Example 8-4 Example of Determining a Boot Device Path's PCI Card Slot

```
# parstatus -w
The local partition number is 0.
# parstatus -V -p0 | grep Path
Primary Boot Path      : 0/0/6/0/0.6.0
Alternate Boot Path    : 0/0/6/0/0.5.0
HA Alternate Boot Path : 2/0/14/0/0.6.0
# rad -N 0/0/6/0/0.6.0
0-1-3-5
#
```

Checklist for Preparing for PCI Card OLAR

This section reviews the items that you must check when adding or replacing a PCI card.

Most of the items in the following checklist are incorporated into the card addition and replacement procedures that follow.

- ❑ Review and follow all server power and safety guidelines and any related guidelines for rack operation. Also follow all site safety, maintenance, and operating procedures.
- ❑ Use proper static protection and follow all site ESD procedures.
- ❑ If *adding* a PCI card:
  - — Ensure card's required driver(s) are loaded into the currently running kernel before adding the new card.
  - — Check the target slot's frequency and power capabilities, and ensure that they match new card's requirements before adding the card.
- ❑ If *replacing* a PCI card:
  - — Check to be certain the card's driver is OLAR-capable.
  - — Be certain that the replacement card uses the same driver(s) as the original card.
  - — Be certain that the replacement card is made by the same vendor and has the same revision ID as the original card.
  - — Be certain that the replacement card operates at the same voltage and same bus frequency as the original card.
  - — Label all cables connected to the original card—or at least note their connections—before removing them.

    After replacing the card, ensure the cable are connected identically to the replacement card.
- ❑ If *replacing a networking card*, check the system (user) impact of taking the card offline before beginning to replace it.
- ❑ If *replacing a defective card*, properly label/mark the card after removing it from its slot to indicate that the card is not operational.

Online Addition (OLA) for a PCI Card

This section describes the procedure for adding a new PCI card on an HP nPartition server while HP-UX remains online and running.

WARNING **When performing this procedure you must follow all server power and safety guidelines and any related guidelines for rack operation. Also follow all site safety, maintenance, and operating procedures.**

Failure to do so can result in personal injury or equipment damage.

PCI Card Online Addition with SAM or Partition Manager

This procedure adds a new PCI I/O card to an empty slot in an nPartition server while HP-UX remains running.

You can use either the System Administration Manager (SAM) or Partition Manager tool to perform the main steps of this procedure.

CAUTION You must follow all site static-protection requirements to avoid damaging equipment when using this procedure.

Step 1. Login to HP-UX running on the nPartition where the card that will be added is to be installed.

You must login as `root` to perform this procedure.

Step 2. Launch either SAM or Partition Manager, if it is not yet running.

You can use either application when performing this procedure.

To run SAM, enter **`/usr/sbin/sam`** from the command line. You can run SAM in either graphical (GUI) or text-only terminal mode.

To run Partition Manager either enter **`/opt/parmgr/bin/parmgr`** from the command line or click the **Partition Manager** icon in the main SAM area.

Step 3. Access the list of the local nPartition's PCI card slots from SAM or Partition Manager.

To list cards in SAM, access the **Peripheral Devices > Cards** area.

To list cards in Partition Manager, select the local nPartition in the primary window, then select and open the I/O chassis where the new card will be installed.

Step 4. Identify the card slot where the new PCI card will be installed, and confirm that the slot can support the card.

You can identify the card slot using SAM's **Cards** window or Partition Manager's list of cards in its primary window. The target card slot must be available (the "Description" column for the slot is "empty slot").

You also can use the `rad -q` command to confirm that the target card slot is available (the "Occupied" column for the slot is "No").

NOTE The **Show I/O Slot Details** option in SAM and Partition Manager displays the *default* bus speed for the selected slot, although all slots are capable of operating at either 33 MHz or 66 MHz.

When the slot "Power Available" listed is 65535 watts, the indication is that slot power details are unavailable to the utility.

See the chapter *nPartition System Overviews* on page 31 for details on the types of physical card keying the PCI card slots support.

Step 5. Confirm that the device driver(s) required by the card are installed in the HP-UX kernel currently running on the nPartition.

You can view currently loaded drivers using the SAM (`/usr/sbin/sam`) utility's **Kernel Configuration—>Drivers** area.

Step 6. [*Optional*] Confirm the physical location of both the server and the PCI card slot.

Perform the following tasks:

A. Set the card slot's attention indicator to flash in order to help you locate the slot.

In SAM, select (highlight) the slot, then select the **Actions —> Light I/O Slot LED** menu item. SAM flashes the selected PCI card slot's LED.

In Partition Manager, select (highlight) the slot, then select the **I/O —> Light Chassis and Slot LEDs** menu item. Partition Manager flashes the selected PCI card slot's LED, and on HP Superdome servers also flashes the corresponding I/O chassis LED and cabinet number LCD.

From the HP-UX command line, you can issue the **`rad -f attention`** ***`slot`*** command to flash the PCI card slot's attention indicator (LED).

B. Locate the server, and view the PCI card slot attention indicators. The LED for the specified card slot should be flashing.

C. After locating the server and card slot, turn off the card slot LED.

In SAM or Partition Manager, click the **OK** button to return the attention indicators to their normal state.

From the HP-UX command line you can issue the **`rad -f off`** ***`slot`*** command to turn off the PCI card slot's attention indicator.

This step is optional, but performing it is recommended to confirm that the actual location is known and accessible.

Step 7. Begin the online PCI card addition procedure.

- To initiate online card addition in SAM, select the **Actions—>Add** menu item.

  Then select (highlight) the slot where the card will be installed, and click the **OK** button.

- To initiate online card addition in Partition Manager, select (highlight) the slot where the card will be installed, and then select the **I/O—>Add Card** menu item.

Step 8. Review the results of the critical resource analysis for the slot.

The first lines of the analysis indicate whether the card addition can or cannot proceed. For example:

```
Critical Resource Analysis for slot 0-1-3-8:
No affected resources found.
```

Step 9. Click either the **Cancel** or **OK** button.

To *cancel* the card addition procedure click the **Cancel** button.

To *continue* the card addition procedure click the **OK** button. This proceeds to power off the card's slot and flash the card's attention indicator (LED).

If you are proceeding with the card addition using Partition Manager on an HP Superdome server, Partition Manager also flashes the corresponding I/O chassis LED and cabinet number LCD.

Step 10. Review the information presented in the **Insert Card** screen.

This screen lists the actions that SAM or Partition Manager has already performed, and describes how you can cancel the card addition.

CAUTION *Complete all steps* required for installing the new PCI card before clicking the **OK** button to bring the card online.

Details for installing the new card are covered in the steps that follow.

Step 11. Locate the PCI card slot that has been prepared for the card addition procedure.

As needed, open or remove any cabinet panels or bezels in order to view the I/O chassis and card slot.

The card slot's attention indicator (LED) will be *flashing* and slot power will be *off*.

Step 12. Ensure you have direct physical access to the I/O chassis.

For example, if adding a PCI card to a Superdome I/O expansion cabinet, you must remove the I/O bay's front covers and then carefully slide the I/O chassis out from its bay/rack.

Step 13. Remove the top cover from the I/O chassis.

Loosen the cover's thumb screws, pull the cover forward, then lift and remove the cover and safely set it aside.

On HP rp7405/rp7410 and rp8400 servers, the I/O chassis cover also is the server cabinet top cover.

Step 14. Confirm the location of the card slot where the new PCI card will be installed.

At this point, all attention and power indicators (LEDs) for the card slot are visible. The light bars on the card slot's divider should indicate that the slot is powered off and its attention LED is flashing.

Each slot divider corresponds to the *PCI card slot to its immediate right* (when viewing the I/O chassis by facing the card slots with the top up).

Step 15. On HP rp7405/rp7410 and rp8400 servers, flip the card slot latch to its open position.

When open, the latch is parallel to the back edge of the chassis.

Step 16. Place the new card in the slot.

Slide the card in the slot, and ensure it is properly aligned. Press firmly on the card until it is fully seated in the card slot.

Step 17. On HP rp7405/rp7410 and rp8400 servers, flip the card slot latch to its closed position.

When closed, the latch is parallel to the card slot divider.

If the latch will not close, the card might not be completely seated in its slot. In this situation, you can either press firmly on the card until it is seated, or lift the slot divider to release the card and then realign and reseat the card.

Step 18. Connect all cables to the new card to establish the desired configuration.

Step 19. Replace the top of the I/O chassis, and restore all front covers and bezels to their original locations.

Replace the I/O chassis cover and firmly push it back into place before tightening all thumb screws.

Also, as needed, carefully slide all racked equipment back into place before replacing any additional covers and bezels. Close any cabinet or rack doors.

Step 20. In the SAM or Partition Manager **Insert Card** window, click the **OK** button.

Clicking **OK** indicates that the new card has been installed.

At this point, the card slot is powered on, the slot attention indicator is turned off, and the driver(s) for the card are started to bring the card online.

If the new card is not detected in the slot, SAM or Partition Manager indicates this and presents a window indicating the problem. This gives you an opportunity to check the new card's installation and then click **Yes** to re-try the card online addition, or click **No** to cancel the operation.

Online Replacement (OLR) for a PCI Card

This section describes the procedure for replacing a PCI card on an HP nPartition server while HP-UX remains online and running.

WARNING **When performing this procedure you must follow all server power and safety guidelines and any related guidelines for rack operation. Also follow all site safety, maintenance, and operating procedures.**

Failure to do so can result in personal injury or equipment damage.

PCI Card Online Replacement with SAM or Partition Manager

This procedure replaces a PCI I/O card in an nPartition server while HP-UX remains running.

You can use either the System Administration Manager (SAM) or Partition Manager tool to perform the main steps of this procedure.

CAUTION You must follow all site static-protection requirements to avoid damaging equipment when using this procedure.

Step 1. Login to HP-UX running on the nPartition where the card that will be replaced is currently installed.

You must login as `root` to perform this procedure.

Step 2. Launch either SAM or Partition Manager, if it is not yet running.

You can use either application when performing this procedure.

To run SAM, enter **`/usr/sbin/sam`** from the command line. You can run SAM in either graphical (GUI) or text-only terminal mode.

To run Partition Manager either enter **`/opt/parmgr/bin/parmgr`** from the command line or click the **Partition Manager** icon in the main SAM area.

Step 3. Access the list of the local nPartition's PCI cards from SAM or Partition Manager.

To list cards in SAM, access the **Peripheral Devices > Cards** area.

To list cards in Partition Manager, select the local nPartition in the primary window, then select and open the I/O chassis where the card that will be replaced is currently installed.

Step 4. Identify the card's slot location and confirm that the card can be replaced online.

Perform the following tasks:

A. View the list of currently available PCI card slots in the local nPartition.

   This card list is shown in SAM's **Cards** window or in Partition Manager's primary window.

B. Identify the slot where the card that will be replaced is currently installed.

   The "Description" column indicates the type of card in the slot.

C. Confirm that the driver for the PCI card slot supports OLAR procedures.

   The "State" column indicates "not OLAR-able" if the card *cannot* be replaced online, otherwise online replacement of the card is supported.

Step 5. [*Optional*] Confirm the physical location of both the server and the PCI card slot.

Perform the following tasks:

A. Set the card slot's attention indicator to flash in order to help you locate the slot.

   In SAM, select (highlight) the slot, then select the **Actions —> Light I/O Slot LED** menu item. SAM flashes the selected PCI card slot's LED.

In Partition Manager, select (highlight) the slot, then select the **I/O —> Light Chassis and Slot LEDs** menu item. Partition Manager flashes the selected PCI card slot's LED, and on HP Superdome servers also flashes the corresponding I/O chassis LED and cabinet number LCD.

From the HP-UX command line, you can issue the **`rad -f attention`** ***`slot`*** command to flash the PCI card slot's attention indicator (LED).

B. Locate the server, and view the PCI card slot attention indicators. The LED for the specified card slot should be flashing.

C. After locating the server and card slot, turn off the card slot LED.

In SAM or Partition Manager, click the **OK** button to return the attention indicators to their normal state.

From the HP-UX command line you can issue the **`rad -f off`** ***`slot`*** command to turn off the PCI card slot's attention indicator.

This step is optional, but performing it is recommended to confirm that the actual location is known and accessible before the card's services are suspended.

Step 6. Select the PCI card to be replaced from the card list displayed by SAM or Partition Manager.

Highlight the card in the list. If you are replacing a multi-function card, you only need to highlight any one of the hardware paths in the slot.

For a multi-function (multi-port) card, SAM and Partition Manager list one entry per port, with each port having the same slot number in the "Slot" column and a unique path in the "Hardware Path" column.

Step 7. Begin the online replacement procedure for the selected card.

To initiate online card replacement in SAM, select the **Actions—>Replace** menu item.

To initiate online card replacement in Partition Manager, select the **I/O—>Replace Card** menu item.

Both of these menu items initiate the same processes, beginning with an analysis of any resources provided by the selected card.

Step 8. Review all results from the critical resource analysis of the card.

The first lines of the analysis indicate whether the card replacement can or cannot proceed. For example:

```
Critical Resource Analysis for slot 0-1-3-8:
No affected resources are in use.
```

If the card resource analysis determines that the card cannot be taken offline, this result is indicated with a “critical” message such as the following output.

```
CRITICAL: Affected resources are essential for system
operation.
The operation cannot proceed.
```

For a multi-function card, SAM and Partition Manager list the analysis results for all ports on the card.

CAUTION Review *all parts* of the critical resource analysis report to determine whether to continue or cancel an online card replacement procedure.

Even when the analysis indicates that “no affected resources are in use”, continuing the card replacement procedure will take the card offline and will halt any services the card provides.

For example, replacing a networking card causes any network connections the card provides to be suspended or terminated.

Step 9. Click either the **Cancel** or **OK** button.

To *cancel* the card replacement procedure click the **Cancel** button. This cancels the procedure and returns to the list of PCI cards. After you click **Cancel** *you must not perform the remaining steps* in this procedure.

To *continue* the card replacement procedure click the **OK** button. This proceeds to suspend the card’s driver(s), power off the card’s slot, and flash the card’s attention indicator (LED).

Step 10. Review the information presented in the **Replace Cards** screen.

This screen lists the actions that SAM or Partition Manager has already performed to take the selected card offline.

CAUTION *Complete all steps* required for replacing the selected PCI card before clicking the **OK** button to bring the card back online. Details for removing and replacing the card are covered in the steps that follow.

NOTE Clicking the **Cancel** button on the **Replace Cards** screen causes the card to *remain offline*: the card slot remains powered off and the card driver(s) remain suspended. The card slot attention indicator is turned off.

Because the card's slot power remains off, you still can replace the card.

To bring a card back online, select the card and then choose the **Actions—>Bring On-line** menu item in SAM or the **I/O—>Bring Card On-line** menu item in Partition Manager.

Step 11. Locate the PCI card slot that has been taken offline for the card removal procedure.

As needed, open or remove any cabinet panels or bezels in order to view the I/O chassis and card slot that contains the PCI card to be replaced.

The card slot's attention indicator (LED) will be *flashing* and slot power will be *off*.

Step 12. Ensure you have direct physical access to the I/O chassis.

For example, if replacing a PCI card from a Superdome I/O expansion cabinet, you must remove the I/O bay's front covers and then carefully slide the I/O chassis out from its bay/rack.

Step 13. Remove the top cover from the I/O chassis.

Loosen the cover's thumb screws, pull the cover forward, then lift and remove the cover and safely set it aside.

On HP rp7405/rp7410 and rp8400 servers, the I/O chassis cover also is the server cabinet top cover.

Step 14. Locate the PCI card that is to be replaced, note all cable connections to the card, and if needed label any unmarked cables. Save all notations for future reference.

At this point, all attention and power indicators (LEDs) for the card slot are visible. The light bars on the replacement card's slot divider should indicate that the slot is powered off and its attention LED is flashing.

Each slot divider corresponds to the *PCI card slot to its immediate right* (when viewing the I/O chassis by facing the card slots with the top up).

Step 15. Disconnect all cables from the PCI card to be replaced.

Carefully set the cable ends aside where they will not obstruct work or be damaged.

Step 16. On HP rp7405/rp7410 and rp8400 servers, flip the card slot latch to its open position.

When open, the latch is parallel to the back edge of the chassis.

Step 17. Lift the card slot divider to release the PCI card from its slot.

Pull firmly on the slot divider's "handle" until the card becomes unseated. The correct slot divider is indicated by its LED states.

Step 18. Remove the card from its slot and set the card aside.

You may need to label the removed card to indicate it is defective.

Step 19. Place the replacement card in the slot.

Slide the card in the slot, and ensure it is properly aligned. Press firmly on the card until it is fully seated in the card slot.

Step 20. On HP rp7405/rp7410 and rp8400 servers, flip the card slot latch to its closed position.

When closed, the latch is parallel to the card slot divider.

If the latch will not close, the card might not be completely seated in its slot. In this situation, you can either press firmly on the card until it is seated, or lift the slot divider to release the card and then realign and reseat the card.

Step 21. Reconnect all cables to the new card to match their prior attachments to the original card.

Step 22. Replace the top of the I/O chassis, and restore all front covers and bezels to their original locations.

Replace the I/O chassis cover and firmly push it back into place before tightening all thumb screws.

Also, as needed, carefully slide all racked equipment back into place before replacing any additional covers and bezels. Close any cabinet or rack doors.

Step 23. In the SAM or Partition Manager **Replace Card** window, click the **OK** button.

Clicking **OK** indicates that the card has been removed and replaced with a new card.

At this point, the card slot is powered back on, the slot attention indicator is turned off, and the driver(s) for the card are resumed to bring the card online.

However, if SAM or Partition Manager cannot bring the replacement card online then you may need to perform additional steps, as directed the SAM or Partition Manager user interface.

9 Processor Instant Capacity on Demand (iCOD)

This chapter covers using Hewlett-Packard's processor iCOD (Instant Capacity on Demand) product on nPartitions.

iCOD is an *optional* software product that enables you to instantly increase or adjust processing power within nPartitions. As you need more or fewer processors, you use iCOD tools to adjust the number of activated processors in the nPartition.

Two varieties of iCOD software and contracts are available from HP: iCOD Purchase and iCOD Utility (pay per use). Both are described in this chapter.

NOTE Using both Processor Sets (Psets) and iCOD simultaneously is supported for iCOD Purchase Version 5.0 and higher only.

NOTE For HP Superdome nPartitions running the HP virtual partitions software, only iCOD Purchase Version 5.0 and higher is supported.

Introduction to Processor iCOD on nPartitions

HP's iCOD product is available for both nPartition servers and non-partitionable HP servers. This document covers iCOD issues that are unique to nPartition systems.

See the *Instant Capacity on Demand (iCOD) and Pay Per Use (PPU) User's Guide for Version B.04.x* and the *Instant Capacity on Demand (iCOD) User's Guide for Version B.05.00* for complete information about iCOD on all supported platforms.

iCOD is an *optional* product that includes an iCOD software bundle and a corresponding system contract with Hewlett-Packard, which determine the type of billing for processors you activate and use.

Billing for iCOD-activated processors is calculated on a complex-wide basis: the total number of activated iCOD processors in all nPartitions.

HP nPartition systems support two varieties of iCOD:

- **iCOD Purchase**

  iCOD Purchase enables you to instantly activate and purchase *additional* processors as your needs increase.

- **iCOD Utility (PPU: Pay Per Use)**

  iCOD Utility supports instantly increasing and decreasing the number of activated processors, for billing on a *pay per use* basis.

All nPartitions in a server complex either must run the same iCOD variety (purchase or utility) or not run iCOD. If you configure iCOD Utility for one nPartition then you also must configure all other nPartitions with iCOD Utility. HP recommends (but does not require) that you configure iCOD Purchase for all nPartitions if you configure any nPartition with iCOD Purchase.

You cannot configure nPartitions with both iCOD Utility and iCOD Purchase in the same server complex.

iCOD Features for nPartitions

HP's iCOD Purchase and iCOD Utility products have some different features and behaviors on nPartition servers than iCOD on other non-partitionable systems.

The following list covers some of these unique features.

- iCOD on each nPartition is managed locally and independently.

  iCOD commands affect and list iCOD settings for the *local* nPartition (the nPartition where the commands are run).

  While there is one iCOD license for the entire server complex, iCOD processors are activated and deactivated independently for each nPartition. Each nPartition has its own settings for iCOD contact, notification, and processor configuration purposes.

- Passwords are not required for activating and deactivating iCOD processor on nPartitions.

  On nPartition servers under either iCOD contract (iCOD Purchase or iCOD Utility) you can perform all tasks that change the iCOD processor configuration for an nPartition.

  On non-partitionable systems an HP service password is required for some iCOD processor deactivation tasks.

- Both varieties of iCOD support "load balancing" processors across nPartitions in the same server complex.

  When using either iCOD Purchase or iCOD Utility, you can instantly adjust each nPartition's number of activated processors as system loads demand and maintain the same total number of activated processors in the complex.

  For example, you can deactivate processors in one underused nPartition and activate the same number of processors in another, more heavily used nPartition to load balance using the same number of processors.

  This load balancing does not change any nPartition cell assignments. Each nPartition keeps the hardware assigned to it while iCOD software in each nPartition either activates or deactivates processors.

iCOD Issues for Managing nPartitions

iCOD introduces several new issues for managing nPartitions. The following list describes some of these new management issues for nPartition systems that have iCOD configured:

- At least one iCOD processor must be activated for each active cell in an nPartition.

  For example, a three-cell nPartition with iCOD must have at least three activated iCOD processors, and the iCOD software ensures that each cell has a processor activated.

- The maximum number of activated processors in an nPartition is the iCOD “requested active processors” setting.

  However, if the number of active cells is *greater than* the number of “requested active processors” then the iCOD software activates more processors than were requested: one processor is activated for each active cell in the nPartition.

- Only processors on active cells can be activated by iCOD.

  Inactive cells in an nPartition cannot have processors activated by iCOD in the nPartition. To activate processors on inactive cells, you first must make the cells active.

- Activating and deactivating processors can potentially affect software packages that rely on certain processor IDs to be present, such as certain processor set (Pset) configurations. Refer to the chapter *Processor Sets (Psets) on nPartitions* on page 419 for details.

  Likewise, changing the number of activated processors may have implications for managing software that is licensed on a per-processor basis.

- Adding or removing cells in an nPartition with iCOD does not necessarily increase or decrease the number of activated processors in the nPartition.

  The iCOD software activates the requested number of processors for an nPartition as long as the nPartition has enough configured processors to satisfy the request.

  Adding a cell to an nPartition increases the total processors and the number of configured processors in the nPartition. However, if the requested number of processors remains the same for the nPartition then the same number of activated iCOD processors are available after performing a reboot for reconfig to make the newly added cell active.

For example, in a two-cell nPartition that has six of its eight processors activated with iCOD, adding another four-processor cell brings the total processors to 12. However, the iCOD software keeps the number of activated processors at six (no change). (Other of the new cell's resources—such as memory and I/O—are made available for use in the nPartition.)

Likewise, removing a cell from an nPartition reduces the total processors and the number of configured processors in the nPartition. If enough processors remain available then the requested number of iCOD processors are activated.

When not enough processors are configured, the iCOD software activates as many processors as possible and the number of activated processors is less than the number of "requested active processors".

The sections that follow give more details for managing iCOD on nPartitions.

Tools for Managing iCOD Processors

The HP Instant Capacity on Demand (iCOD) product includes the following commands for managing iCOD settings and processor configurations. This same set of commands is used for both the iCOD Purchase and iCOD Utility products.

For details on these commands, see the *icod_modify* (1M), *icod_notify* (1M), and *icod_stat* (1M) manpages.

- `/usr/sbin/icod_modify`

  The `icod_modify` command allows you to activate and deactivate iCOD processors. This command also lets you change system contact information and apply a software license to use iCOD.

  The `-a` option activates processors and `-d` deactivates processors.

- `/usr/sbin/icod_notify`

  The `icod_notify` command allows you to request that an iCOD asset report be delivered by e-mail, and allows you to turn on or off e-mail notification of iCOD configuration changes.

  The `-n` option turns on or off automatic change notification e-mail.

- `/usr/sbin/icod_stat`

  The `icod_stat` command displays iCOD status and configuration information as well as iCOD processor usage details.

  The `-p` option gives complex-wide iCOD Purchase details on nPartition systems. The `-u` option displays the iCOD change record, listing the changes from oldest to newest.

When using the iCOD commands to activate and deactivate processors or update contact and notification details, you affect the iCOD configuration and settings for the *local nPartition* only. While some iCOD settings are stored in complex profile data, many iCOD settings are stored on disk in the iCOD configuration file (/etc/.iCOD_data). As a result, you may need to check and adjust iCOD configuration settings when booting from different disks.

When you license iCOD (by using the `icod_modify -l...` command) you can do so from *any nPartition* in the server complex. This licenses iCOD for all nPartitions in the server, and only one license is needed for the entire complex.

iCOD Requirements for nPartition Servers

HP's iCOD software has the following requirements and restrictions for using and managing iCOD processors in an nPartition server complex.

- Each nPartition server complex can *optionally* be under either an iCOD Purchase contract or a pay per use (iCOD Utility) contract.

  In a server complex that is under one of these contracts, the appropriate software **must** be installed in the complex's nPartitions to support the contract.

  HP *does not* support mixing iCOD Purchase and iCOD Utility nPartitions in the same server complex.

- nPartition servers that are under an iCOD Utility contract (a pay per use "PPU" contract) **must** have iCOD Utility software installed and running on *every* HP-UX instance in the complex.

NOTE Where multiple devices are configured for an nPartition (for example, the PRI, HAA, and ALT boot paths), each device must have iCOD Utility software installed. This applies to nPartition servers under a PPU (iCOD Utility) contract.

- Each nPartition that is under an iCOD Purchase contract **must** have iCOD Purchase software installed to enable additional processors to be activated (or to deactivate processors).

  In an iCOD Purchase complex, you *do not* have to install iCOD Purchase software on the nPartitions that are *not* under an iCOD Purchase contract. However, in this situation HP recommends that all nPartitions have iCOD Purchase software installed to allow administrators to perform "load balancing" across nPartitions.

NOTE All potential boot disks for nPartitions with iCOD processors must have the iCOD Purchase software installed, including any alternate boot devices. This applies to nPartition servers under an iCOD Purchase contract.

- For the iCOD software to activate processors, the processors **must** be *configured* processors on *active* cells that are *assigned* to the local nPartition

  Processors that are deconfigured cannot be activated by iCOD; they first must be configured (for example, by using the BCH Configuration menu's `CPUCONFIG` command).

  Cells that are inactive cannot have processors activated by iCOD. Each cell first must boot and complete “partition rendezvous” before it can contribute resources to the nPartition to which it is assigned.

  Likewise, cells that are unassigned cannot have processors activated by iCOD. Each cell must be assigned to an nPartition and must be an active member of its nPartition before its resources can be used.

Installing and Configuring iCOD on nPartitions

NOTE This section describes iCOD software install and configuration for an nPartition server complex.

You also must establish a contract with HP for either iCOD Purchase or iCOD Utility (pay per use) to properly use this software product.

Software bundles for iCOD Purchase and iCOD Utility are available on the Support Plus media and at the **`http://software.hp.com`** Web site.

For complete details on installing and using iCOD software, refer to the *Instant Capacity on Demand (iCOD) and Pay Per Use (PPU) User's Guide for Version B.04.x* and the *Instant Capacity on Demand (iCOD) User's Guide for Version B.05.00*.

The following procedure gives an overview of initially installing and configuring iCOD Purchase or iCOD Utility on an nPartition server complex.

iCOD Installation and Configuration

Step 1. Install the appropriate iCOD software bundle on all required nPartitions in the server complex.

On nPartition server complexes that have iCOD Purchase contracts, you must install the iCOD Purchase software on those nPartitions in the complex that have iCOD processors. (For greater flexibility in load balancing iCOD processors HP recommends installing iCOD Purchase software on all nPartitions.)

On nPartition servers that have iCOD Utility (pay per use) contracts, you must install the iCOD Utility software on *every* nPartition in the complex.

If you expect to boot an nPartition from different devices—even on rare occasions—you must install the appropriate iCOD software bundle on all potential boot devices (such as the devices at the PRI, HAA, and ALT boot paths).

Step 2. [*An HP service representative must perform this step*.]

Validate the server complex as an iCOD server.

Step 3. Configure `sendmail` so that it can send e-mail to an HP mail server that is outside of your company's firewall.

HP iCOD software sends encrypted e-mail from the local nPartition running iCOD to HP for billing purposes and to request licensing information. Details on `sendmail` configuration are in the user's guide for iCOD.

You must configure `sendmail` on all nPartitions that have iCOD software installed. On nPartitions with multiple boot devices, configure `sendmail` for each boot device.

Step 4. Configure the iCOD contact information for each nPartition that has iCOD software installed, using the `icod_modify -c...` command.

This specifies the person who will receive iCOD licensing e-mail from HP and iCOD configuration change notices. For example:

```
# icod_modify -c "Joe Doe":joe@company.com:555-5555
```

On nPartitions with multiple boot devices, configure the the iCOD contact information for each boot device, in case alternate devices (such as HAA or ALT) are booted.

Step 5. [*This step needs to be performed only once for the entire server complex*.]

Request a license by issuing the `icod_notify` command (with no options) in any nPartition that has iCOD, `sendmail`, and the contact information configured.

The `icod_notify` command sends an iCOD asset report to HP and to the iCOD contact and `root` for the nPartition. After HP receives the asset report a confirmation e-mail, which contains the iCOD license key, is sent to the iCOD contact.

Apply the license for iCOD by issuing the `icod_modify -l...` command. For example:

```
# icod_modify -l AABBCCDD
```

where *AABBCCDD* is the iCOD license key given in the confirmation e-mail. You only need to apply the iCOD license once for the entire server complex.

Step 6. Use iCOD features: list iCOD statistics with `icod_stat` and, when required, activate or deactivate processors.

To list iCOD configuration details for the local nPartition, use the `icod_stat` command (with no options).

For an nPartition complex that has iCOD Purchase configured for multiple nPartitions, you also can use the `icod_stat -p` command to display iCOD processor usage statistics for all nPartitions in the server complex. (The `-p` option does not give more information for iCOD Utility configurations or for non-nPartition configurations.)

See *Procedures for Changing Processor iCOD Configurations on nPartitions* on page 409 for details on managing an nPartition's iCOD processors.

Procedures for Changing Processor iCOD Configurations on nPartitions

This section covers the following procedures for changing the iCOD configuration on nPartitions. These procedures apply for both iCOD Purchase and iCOD Utility software and contracts.

- *Activating and Deactivating Processors with iCOD* on page 410

  This procedure (using the icod_modify -a... or icod_modify -d... command) activates or deactivates processors in an nPartition with iCOD.

- *Setting the Total Number of Requested Active Processors* on page 411

  This procedure (using the icod_modify -s... command) sets the total number of requested active processors for an nPartition with iCOD software installed and configured.

- *Load Balancing Processors across nPartitions with iCOD* on page 412

  This procedure (using both the icod_modify -d... and icod_modify -a... commands) adjusts the balance of activated processors across two nPartitions in the same server complex: deactivate processors in one nPartition and activate the same number of processors in another nPartition.

- *iCOD Contract Changes for an nPartition Server Complex* on page 413

  This procedure describes how to change your existing iCOD contract to either iCOD Purchase or iCOD Utility by contacting HP sales or support representatives.

- *Removing iCOD Software and Functionality from nPartitions* on page 414

  This procedure describes how to remove iCOD software and functionality from an nPartition server complex after completing your iCOD purchasing and contract obligations.

NOTE The following nPartition changes also can cause iCOD software to activate a different number or set of the nPartition's processors: adding and removing cells from an nPartition, making cells active or inactive, or configuring or deconfiguring processors from cells in the nPartition.

Activating and Deactivating Processors with iCOD

This procedure (using the **`icod_modify -a...`** or **`icod_modify -d...`** command) activates or deactivates processors in an nPartition with iCOD.

NOTE Activating or deactivating processors can affect your billing for iCOD services.

On systems with HP processor set (Pset) software installed: *newly activated* processors are assigned to the default Pset, and deactivated processors are removed from the Pset to which they were assigned.

HP's iCOD software selects processors for activation or deactivation by following the appropriate processor installation order for the machine type. The iCOD utilities select processors based on their *physical location* in the server (not their HP-UX CPU IDs).

For example, on HP Superdome servers the processor install order for each cell is: first processor slot 0, then slots 3, 1, and 2.

As a result, all active cells in a Superdome nPartition always have processor 0 activated because a minimum of one processor must be activated per cell. Then, as needed to meet the iCOD "requested active processors" number for the nPartition, each cell's "processor 3" slot is activated, then each cell's "processor 1" slot, and finally the "processor 2" slots.

Step 1. Login to the nPartition in which you will be activating or deactivating processors.

You can activate or deactivate processors in only the *local* nPartition (the nPartition in which you issue the `icod_modify` command).

Step 2. Issue the `icod_modify` command with either the `-a #` option (to activate # processors) or `-d #` option (to deactivate # processors).

You must include the following details after the `-a` or `-d` option. This information is recorded in the nPartition's iCOD change log.

[*description*]:*user_name*:*mgr_name*:*mgr_email*:*mgr_phone*

These details provide an optional description of the change, the name of the user/person making the change, and the authorizing manager, manager's e-mail address, and manager's phone number.

For example, to activate two processors (`-a 2`):

```
# icod_modify -a 2 "two CPUs added":Ann:Joe:jdoe@comp.com:555-5555
```

In the next example, one processor is deactivated (`-d 1`):

```
# icod_modify -d 1 "one less CPU":Ann:Joe:jdoe@comp.com:555-5555
```

See also the *icod_modify* (1M) manpage for details.

Step 3. As desired, issue the `icod_stat` command to list the new processor configuration details for the local nPartition.

Setting the Total Number of Requested Active Processors

This procedure (using the **`icod_modify -s...`** command) sets the total number of requested active processors for an nPartition with iCOD software installed and configured.

Performing this procedure can increase or decrease the number of activated processors in an nPartition.

NOTE Activating or deactivating processors can affect your billing for iCOD services.

On systems with HP processor set (Pset) software installed: *newly activated* processors are assigned to the default Pset, and deactivated processors are removed from the Pset to which they were assigned.

Step 1. Login to the nPartition in which you will be activating or deactivating processors.

You can activate or deactivate processors in only the *local* nPartition (the nPartition in which you issue the `icod_modify` command).

Step 2. Issue the `icod_modify -s...` command and specify the number of processors to be activated.

You must include the following details after the `-s` option. This information is recorded in the nPartition's iCOD change log.

[*description*]:*user_name*:*mgr_name*:*mgr_email*:*mgr_phone*

For example, the following command sets the number of "requested active processors" to 10, which may increase or decrease the number of activated processors in the nPartition (depending on the number of processors available before the command is issued).

```
# icod_modify -s 10 "activate 10 CPUs total":Ann:Joe:jdoe@comp.com:555-5555
```

See also the *icod_modify* (1M) manpage for details.

Step 3. As desired, issue the `icod_stat` command to list the new processor configuration details for the local nPartition.

Load Balancing Processors across nPartitions with iCOD

This procedure (using both the **`icod_modify -d...`** and **`icod_modify -a...`** commands) adjusts the balance of activated processors across two nPartitions in the same server complex: deactivate processors in one nPartition and activate the same number of processors in another nPartition.

NOTE After "load balancing" processors across nPartitions, each nPartition still has the same cells and processors assigned to it.

However, this procedure reduces the number of activated processors in the first nPartition and increases (by the same amount) the number of activated processors in the second nPartition.

This procedure does not affect your billing for iCOD services *if*: the total number of activated processors in the complex does not change *and* the operations are not performed by HP service representatives.

Both nPartitions must have iCOD software installed and configured.

Also, both nPartitions must have enough activated or deactivated processors to accommodate the reduction or increase in processors.

Step 1. Login to the first nPartition and deactivate the number of processors you plan to activate in the second nPartition.

See the procedure *Activating and Deactivating Processors with iCOD* on page 410, and use the `icod_modify -d...` command to deactivate the processors.

If HP processor set (Pset) software is installed, deactivating processors removes the corresponding CPU IDs from the Pset to which they were assigned.

Step 2. Login to the second nPartition and activate the same number of processors you deactivated in the previous step.

See the procedure *Activating and Deactivating Processors with iCOD* on page 410 and use the `icod_modify -a...` command to activate the processors.

If HP processor set (Pset) software is installed, the newly-activated processors are assigned to the local nPartition's default Pset.

iCOD Contract Changes for an nPartition Server Complex

This procedure describes how to change your existing iCOD contract to either iCOD Purchase or iCOD Utility by contacting HP sales or support representatives.

Changing the type of iCOD contract for a server complex will affect your billing for iCOD services.

Step 1. Contact your HP sales or support representatives and request an iCOD contract and software change.

Changing the type of iCOD contract and software will require that an HP service representative alter the iCOD software configuration for all nPartitions affected by the change.

For nPartitions that have multiple boot devices (such as PRI, HAA, and ALT), HP iCOD software bundles on every boot device must be updated.

Step 2. Consider any software licensing issues or nPartition system configuration issues that you must address when changing from iCOD Purchase to iCOD Utility (pay per use) or vice versa.

For example, some nPartitions may have different sets of processors activated as a result of the change. Or, in the case of a pay per use (iCOD Utility) contract, the set of activated processors in each nPartition may change on an ongoing basis.

Such changes could potentially affect HP processor set (Pset) configurations, or the configuration of HP Process Resource Manager (PRM) or HP Workload Manager (WLM) software.

Removing iCOD Software and Functionality from nPartitions

This procedure describes how to remove iCOD software and functionality from an nPartition server complex after completing your iCOD purchasing and contract obligations.

For more details, see the *Instant Capacity on Demand (iCOD) and Pay Per Use (PPU) User's Guide for Version B.04.x* and the *Instant Capacity on Demand (iCOD) User's Guide for Version B.05.00*.

Step 1. Confirm that all processors in all nPartitions in the server complex are activated and purchased.

All processors in the complex are activated when: for every nPartition the "requested active processors" equals the "total processors". You can check this by issuing the `icod_stat` command in each nPartition.

If you have a server complex that is under an iCOD Purchase contract, you also must have paid the enablement fee for all processors. Confirm this with your HP sales or service representative.

If your server complex is under an iCOD Utility (pay per use) contract, you must check with your HP sales or service representative to determine if you have met all contract requirements.

Step 2. After confirming with HP that you have completed all requirements, use the `swremove` command to uninstall the iCOD Purchase or iCOD Utility bundle.

You must remove the bundle from every nPartition that no longer is under contract. In the case of iCOD Utility contracts this involves removing the bundles from all nPartitions.

On nPartitions where iCOD software is installed on multiple boot devices (such as PRI, HAA, and ALT) you should remove the bundle from all devices.

See the iCOD documentation and the *swlist* (1M) and *swremove* (1M) manpages for details.

Managing iCOD Utility (Pay Per Use) on nPartitions

This section describes several methods of managing processor resources for a server complex that is under a pay per use (iCOD Utility) contract.

If your server complex is under an iCOD Utility contract then you are billed for *all activated* processors in the whole nPartition server complex.

The pay per use iCOD Utility contract enables you to manage processor resources in the complex in such a way that you only pay for the amount of processor resources that you actually require.

When you have processors in an nPartition complex that you do not need, you can exclude those processors from billing by *deactivating* processors, by making processors *inactive*, or by *deconfiguring* processors.

These three methods of excluding processors from pay per use billing are discussed here. See the *Deactivated Processors*, *Inactive Processors*, and *Deconfigured Processors* sections that follow.

Tips for Pay Per Use Processor Management

- The recommended method for activating and deactivating processors on nPartitions is to use the `icod_modify` command. This command instantly increases or decreases the number of available processor resources in the nPartition without requiring a reboot.

  For example, if an nPartition is underused—as when most of the nPartition's processors are constantly idle—you could deactivate unneeded processors by using the `icod_modify -d...` command.

- When an entire nPartition in a complex is unused you can exclude that nPartition's processors from billing by making the nPartition inactive.

  For example, if an nPartition is not running HP-UX but is "just sitting at the BCH interface" you could reset the nPartition to the ready for reconfig state by using the BCH `RECONFIGRESET` command to make the nPartition inactive. (When an nPartition is running HP-UX, using the `shutdown -R -H` command makes the nPartition inactive.)

When an nPartition is inactive, all its cells and processors are inactive and cannot be used until the nPartition is booted (using the GSP or MP Command menu's `BO` command).

- Individual cells that are *inactive* are not billed for iCOD Utility purposes, because all processors on inactive cells also are inactive.

  This includes unassigned cells, as well as cells that have not participated in "partition rendezvous" for their assigned nPartition (for example: newly-added cells or cells that had a "n" use-on-next-boot value when the nPartition last booted).

Deactivated Processors

A **deactivated processor** is one that has been "turned off" by the nPartition's iCOD software, perhaps as a result of the `icod_modify -d...` command. Deactivated processors can be activated instantly by using the `icod_modify -a...` command.

The iCOD software selects which processors are activated and deactivated and chooses processors based on their physical locations.

For details, see the procedures in *Activating and Deactivating Processors with iCOD* on page 410.

Inactive Processors

An **inactive processor** is a processor that is in an inactive cell and thus is at a boot is blocked (BIB) state.

The following examples describe situations where both cells and processors are inactive (and thus are not subject to iCOD Utility billing):

- All processors on a cell that is not assigned to an nPartition are inactive.
- All processors on a cell that did not participate in "partition rendezvous" for its nPartition are inactive.

  You can make a cell inactive either by unassigning it from an nPartition, or by setting the cell's use-on-next-boot value to "n" (meaning: do not use the cell) and rebooting the cell's nPartition.
- All processors on cells that are assigned to an inactive nPartition are inactive.

  You can make all processors in an nPartition inactive by resetting the nPartition to the ready for reconfig state. In an inactive nPartition, all cells are inactive and thus all processors on cells in the nPartition are inactive.

To put an nPartition in the inactive, ready for reconfig state: if HP-UX is running use the `shutdown -R -H` command, or if at the Boot Console Handler (BCH) interface use the `RECONFIGRESET` command.

None of an inactive cell's resources (processors, memory, or any I/O connected to the cell) are available for use in an nPartition. For the cell's processors and other hardware resources to be used, the cell must be *assigned* and *active* in an nPartition.

Deconfigured Processors

A **deconfigured processor** is a processor that has been made unavailable for use by its nPartition through settings enabled by Boot Console Handler (BCH) menu commands.

You can deconfigure processors using the BCH Configuration menu's `CPUCONFIG` command. Also use this command to configure processors that have been deconfigured.

Using BCH commands to configure and deconfigure processors requires rebooting the nPartition in which the processors reside. For this reason deconfiguring processors is *not the recommended method* of making processors inactive for iCOD purposes.

Instead, the recommended method is to deactivate processors using the `icod_modify -d...` HP-UX command, which can instantly make processors deactivated and activated without rebooting.

10 Processor Sets (Psets) on nPartitions

This chapter describes how to use and manage processor sets (Psets) on nPartition systems.

Using Psets, you can create multiple independent processor groups in an nPartition. Each Pset has its own processors, schedules, and attributes. Because Psets are dynamic, you can create, modify, and destroy Psets instantly as your system needs demand.

HP's processor set software is an *optional* package that is free for all HP-UX 11i systems and is available at the **`http://software.hp.com`** Web site.

The same Pset features are available on all HP-UX 11i systems, including both non-partitionable systems and nPartitions servers.

On nPartition servers, however, you should be aware of the nPartition system configuration issues that can affect your use of processor sets. This chapter covers special configuration issues for Psets in nPartition environments.

NOTE Using both Processor Sets (Psets) and iCOD simultaneously is supported for iCOD Purchase Version 5.0 and higher only.

Introduction to Psets

HP's processor set (Pset) product is an optional software package that runs on any HP-UX 11i system, including all nPartition servers. The Pset software package is free and is available from the **`http://software.hp.com`** Web site.

Each processor set (Pset) is a group of active processors that functions as an independent *scheduling allocation domain*. When the Pset software is installed, you can establish multiple Psets in a single HP-UX system.

By dividing the active processors in an nPartition into multiple Psets, you can provide processor resource isolation for applications that run in each Pset. Each application only has access to the processors assigned to the Pset in which it runs.

You can dynamically create and reconfigure Psets using the `psrset` command or HP's Process Resource Manager (PRM). You also can launch each thread or process to run in a specific Pset and can manually migrate threads and processes to different Psets while they run.

Thread and Process Pset Bindings

In systems where Pset software is installed, every thread and process is bound to only one Pset at a time.

Applications are not migrated to different Psets unless you have configured PRM to do so, or if you manually bind a process to a different Pset using the `psrset` command.

HP-UX load balancing occurs within each Pset. Because load balancing does not occur across Psets, processors in one Pset can potentially be oversubscribed while processors in another Pset are nearly idle. This is an aspect of the processor resource isolation that Psets provide.

Both real-time and time-share schedulers are supported for processor sets and each Pset has its own schedulers. So, for example, real-time processes in one Pset only contend for processors in the Pset in which they are running.

Use of the HP-UX gang scheduler is supported only in the default Pset (processor set ID 0), as of the current Pset software release. See the *gang_sched* (7) manpage or the *mpsched* (1) manpage for details on using gang scheduling.

HP-UX Processor Numbering and Availability on nPartitions

The HP-UX operating system number processors from 0 to *n*-1, where *n* is the number of configured processors on active cells in an nPartition.

Each physical processor is not necessarily given the same logical HP-UX processor ID each time the nPartition is booted. HP-UX processor IDs are assigned on a first-come first-numbered basis. As a result, even if an nPartition's processor configuration does not change, the correlations from physical processors to logical HP-UX processor IDs may change when HP-UX is rebooted in the nPartition.

This list gives details on how processors are available and numbered by HP-UX running on nPartitions.

- The following processors are *numbered* and are *available*: processors that are configured, reside on active cells, and (if HP's iCOD software is configured) are activated by iCOD.
- Processors that are *deconfigured* are not available and are not numbered by HP-UX.

  For deconfigured processors to be available they first must be configured using the nPartition's Boot Console Handler (BCH) interface.

  For details refer to the chapter *Listing and Managing Server Hardware* on page 305.
- Processors on *inactive cells* are not available and are not numbered by HP-UX. The cells must be active and must have configured processors in order to contribute processors to the nPartition.
- Processors that have been *deactivated* by HP's Instant Capacity on Demand (iCOD) software *are* numbered by HP-UX but are not available to be used until they are activated by iCOD.

  This means that when iCOD has deactivated one or more processors, some processors were numbered by are not listed in output displayed by commands such as `mpsched -s`, `top`, or `sar`.

  For example, the `mpsched` output below shows that processors 1, 2, 5, 6 and possibly others are deactivated. (To view more iCOD details use `icod_stat`.) Refer to the chapter *Processor Instant Capacity on Demand (iCOD)* on page 397 for details.

  ```
  # mpsched -s
  System Configuration
  =====================
  Locality Domain Count: 1
  ```

```
Processor Count      : 5

Domain    Processors
------    ----------
0        0   3   4   7   8
#
```

The System Default Pset

When Pset software is installed, a *system default Pset* always exists that gives all users access to the processors assigned to it. The default Pset is Pset 0, which always has at least processor ID 0 assigned to it.

All processors are initially assigned to the default Pset until you configure processors to belong to other Psets.

When a Pset is destroyed or when a processor is removed from a Pset, the processors involved are assigned back to the default Pset.

Pset Attributes and Access Permissions

Each Pset has attributes that configure the Pset's behavior in various situations. These attributes also include "owner, group, and others" access permissions similar to traditional HP-UX file permissions. The default Pset's attributes cannot be changed, but all other Psets can have their attributes adjusted as needed.

Users who have write access for a Pset can modify *some* of the Pset's attributes, including attributes other than the access permissions. Each Pset's owner can modify the Pset's access permissions.

The following users can modify *all aspects* of all non-default Psets in a system: `root`, `superuser`, and users who belong to a group that has the PSET privileged capability. These users can modify all Pset attributes, modify all Pset processor assignments, and can create and destroy Psets.

The PSET privileged capability is established for a group by issuing the `setprivgrp` command. For example, `setprivgrp mygrp PSET` applies this privilege (and no other privileges) to the "mygrp" group. See the *setprivgrp* (1M) manpage for details.

Pset Boot-Time Configuration

When HP-UX boots on an nPartition that has Pset software installed, by default all processors are assigned to the default Pset: Pset 0.

You can have multiple Psets established at boot time either by creating HP-UX startup scripts that configure Psets, or by configuring Psets through PRM and having PRM establish configurations at boot time.

Pset Binding and Inheritance

Child threads and processes inherit the Pset bindings of their parents.

So, for example, when a process creates child processes, the children are are launched into the same Pset as the parent.

By using the Pset programming interface you can have more control over the Pset locations where threads and processes are spawned and run.

Using PRM on nPartitions with Psets

The HP Process Resource Manager product enables you to create and manage Psets through its graphical interface.

PRM provides the ability to maintain Pset configurations across system reboots. It also has the ability to assign (isolate) memory to Psets, thus giving Psets memory isolation as well as processor resource isolation.

PRM software refers to Psets that it tracks using PRM IDs or names, rather than using Pset IDs. PRM may modify Psets and cause them to be renumbered while managing Psets. Thus, if you use the `psrset -i` command while PRM has configured Psets, you may notice this renumbering of Pset IDs.

Note that if you have used PRM to assign specific processor IDs to Psets in the system, all specified processor IDs *must be present* for PRM to be able to load and establish the Pset configurations. Otherwise, when specified processors are not present, PRM cannot create the Psets.

You can help avoid this potential problem by not specifying processor IDs and instead specifying the *number of processors* for PRM to configure in each Pset.

This processor availability issue can prevent PRM from loading Pset configurations when iCOD software in an nPartition has *deactivated* one or more of the specified processors.

PRM also may be prevented from loading Pset configurations when any of the following has occurred in an nPartition: *deconfiguring* processors (at the BCH interface), *unassigning* a cell from an nPartition, or making one or more of an nPartition’s cells *inactive*.

For details on managing PRM, see the PRM online help or *HP Process Resource Manager User’s Guide*. Also see the *psrset* (1M) manpage for details on using the `-f` option while PRM is managing Psets.

Programming Interface for Psets

For details on the Pset programming interface, see the following HP-UX manpages: *pset_assign* (2), *pset_bind* (2), *pset_create* (2), *pset_ctl* (2), *pset_destroy* (2), *pset_getattr* (2), *pset_setattr* (2).

Tools for Managing and Using Psets

This section lists several tools for managing Psets.

For details, see these manpages: *psrset* (1M), *xprm* (1), *prmconfig* (1), *mpsched* (1), *rtsched* (1), *sar* (1M), *setprivgrp* (1M), and *getprivgrp* (1).

- `/usr/sbin/psrset`

  This command provides the main command-line interface for Psets.

- HP Process Resource Manager (PRM):
  `/opt/prm/bin/xprm` and
  `/opt/prm/bin/prmconfig`

  These commands provide graphical (`xprm`) and command-line (`prmconfig`) interfaces to PRM, which has built-in support for Psets.

- `/usr/bin/mpsched`

  This command provides a method for launching and managing time-share processes and threads, allowing for processor binding and unbinding, enabling gang scheduling, and inquiring about system and process attributes.

- `/usr/bin/rtsched`

  This command provides a method of launching real-time threads and processes.

- `/usr/sbin/sar`

  This command reports system activity, including Pset activity when the `-p` *`pset`* option or `-P` option is specified.

  For example, `sar -u -M -P 5` gives a snapshot of system processor use over a five second period, and because `-P` is specified Pset assignments are included.

- `/usr/sbin/setprivgrp` and
  `/usr/bin/getprivgrp`

  The `setprivgrp` command sets privileged capabilities for a specified group. When issuing this command, you must list all privileged capabilities that are to be applied for the group. For example: `setprivgrp mygrp PSET RTSCHED` grants special Pset and real-time scheduling capabilities to the members of the “mygrp” group.

To remove privileged capabilities for a group, issue the `setprivgrp` command with no capabilities specified (for example: `setprivgrp mygrp`).

The `getprivgrp` command reports privileged capabilities for the user issuing the command.

Procedures for Managing Psets

This section lists only the `psrset` command-line procedures for managing Psets.

The common Pset tasks briefly given here are:

- *Listing Pset Configurations* on page 426
- *Creating a New Pset* on page 426
- *Destroying (Deleting) a Pset* on page 427
- *Assigning (Reassigning) Processors to Psets* on page 427
- *Unassigning (Removing) Processors from Psets* on page 427
- *Configuring Pset Attribute Values* on page 427
- *Setting Pset Access Permissions* on page 428
- *Running Programs in a Pset* on page 429
- *Binding Threads and Processes to a Pset* on page 429

Also see the *psrset* (1M) manpage for details, or see *Example Uses of Psets* on page 430 for command output and examples.

For details on support for Psets in HP Process Resource Manager (PRM) refer to the book *HP Process Resource Manager User's Guide* or the PRM online help.

Listing Pset Configurations

Step 1. **`/usr/sbin/psrset -i`**

This lists all Psets defined in the system including the processors assigned to each and the owner, access permissions, and attributes for the Psets.

Creating a New Pset

Step 1. **`/usr/sbin/psrset -c [`*`processor_list`*`]`**

where *`processor_list`* is an optional list of processors that are assigned to the newly created Pset.

Destroying (Deleting) a Pset

Step 1. **`/usr/sbin/psrset -d [`*`pset_list`*` | all]`**

where you specify either `all` (to delete all Psets) or a list of the Psets to be deleted (*`pset_list`*).

When you delete a Pset, the Pset's ID no longer exists and all processors assigned to the Pset are assigned to the default Pset. Deleting all Psets (`psrset -d all`) causes all processors to be assigned to the default Pset (Pset ID 0), which then is the only Pset in the system.

The user issuing this command must have write permission for the Psets that are deleted.

Assigning (Reassigning) Processors to Psets

Step 1. **`/usr/sbin/psrset -a`** ***`pset_id processor_list`***

where *`pset_id`* is the Pset to which the processors specified in *`processor_list`* are assigned.

The user issuing this command must have write permission for both the Pset specified by *`pset_id`* and the Pset(s) to which the processors in *`processor_list`* are assigned.

Unassigning (Removing) Processors from Psets

Step 1. **`/usr/sbin/psrset -r`** ***`processor_list`***

where *`processor_list`* is the list of processors that will be removed from their current Psets and assigned to the default Pset.

The user issuing this command must have write permission for the Pset(s) to which the processors in *`processor_list`* are assigned.

Configuring Pset Attribute Values

To configure access permissions (`OWNID`, `GRPID`, `PERM`) you must have `root` or `superuser` access or membership in a group that has `PSET` privileged capabilities.

You cannot modify attributes for Pset ID 0.

Step 1. **`/usr/sbin/psrset -t`** ***`pset_id attr_name=attr_value`***

where *`pset_id`* is the Pset whose attribute(s) will be configured.

Each attribute (*`attr_name`*) is set to the corresponding value (*`attr_value`*) specified.

Attributes include `OWNID`, `GRPID`, `PERM`, and others listed in the *psrset* (1M) manpage.

Step 2. **`/usr/sbin/psrset [-n | -F]`** ***`pset_id`***

where *`pset_id`* is the Pset for which external I/O interrupts are either enabled (`-n`) or disabled (`-F`).

When configuring attributes other than access permissions, the user issuing these commands must have write permission for the Psets specified.

Setting Pset Access Permissions

Step 1. Use the `psrset` command's `-t` option, as described in *Configuring Pset Attribute Values* on page 427.

To configure access permissions (`OWNID`, `GRPID`, `PERM`), you must have `root` or `superuser` access or membership in a group that has `PSET` privileged capabilities.

Specify the `PERM` attribute and corresponding value to set access permissions.

`/usr/sbin/psrset -t` ***`pset_id`*** **`PERM=`*****`p1p2p3`***

where *`pset_id`* is the Pset and *`p1p2p3`* is the set of access permissions for the Pset owner (*`p1`*), Pset group (*`p2`*), and others (*`p3`*).

Each access permission (owner, group, and others) is a number from 0–7 to indicate execute (x), write (w), and/or read (r) permissions.

`0`=no permissions, `1`=x, `2`=w, `3`=xw, `4`=r, `5`=xr, `6`=wr, `7`=xwr

Execute allows running programs in the Pset, write allows changing the Pset configuration, and read allows reading the Pset configuration.

For example “PERM=754” gives the Pset owner execute, write, and read permissions; gives members of the Pset’s group execute and read permissions; and gives other users only read permission.

You also can specify attributes and values to change the owner (`OWNID`) and group (`GRPID`) for the Pset.

Running Programs in a Pset

Step 1. **`/usr/sbin/psrset -e`** ***`pset_id command`*** **`[`*****`arguments`*****`]`**

where *`pset_id`* is the Pset in which the specified *`command`* will be executed.

As needed, specify *`arguments`* to list any command-line options or arguments for the command.

The user issuing this command must have execute permission for the Pset in which the command is run.

Binding Threads and Processes to a Pset

Step 1. **`/usr/sbin/psrset -b`** ***`pset_id pid_list`***

where *`pset_id`* is the Pset in which the specified process IDs (*`pid_list`*) will be bound.

The user issuing this command must have execute permission for both the original and new Psets in which the process ID executes.

Example Uses of Psets

NOTE

These examples show the use of processor sets (Psets) on an HP Superdome server that also has HP Instant Capacity on Demand (iCOD) "pay per use" software installed.

Uses of the optional HP iCOD software commands are noted in the text accompanying the examples.

For iCOD management information, refer to the chapter *Processor Instant Capacity on Demand (iCOD)* on page 397.

The following Pset examples are given in this section.

- *Listing, Creating, and Using Psets* on page 430
- *Destroying a Pset and Reassigning Processors* on page 432
- *Example of Running and Binding Programs in Psets* on page 434
- *Managing Pset Permissions and Attributes* on page 436

Example 10-1 **Listing, Creating, and Using Psets**

Initially this nPartition has only one Pset: the default Pset, which is Pset 0.

```
# psrset -i
PSET         0
SPU_LIST     0    1    2    3    4    5    6    7    8    9    10   11
OWNID        0
GRPID        0
PERM         755
IOINTR       ALLOW
NONEMPTY     DFLTPSET
EMPTY        FAIL
LASTSPU      DFLTPSET

#
```

The `icod_modify` command sets the total number of processors to four. As the `psrset -i` command shows, this reduces the number of processors that are available and assigned to Psets. Note that processor IDs (listed in the SPU_LIST) are *not sequentially numbered* because several processors have been deactivated by the iCOD software.

```
# icod_modify -s 4 "set to 4":Ann:Joe:jdoe@comp.com:555-5555

4 processors are now active.

NOTE:    Verify that HP and 3rd party software licenses are upgraded
         to take into account the number of active processors.

# psrset -i
PSET         0
SPU_LIST     0    3    4    8
OWNID        0
GRPID        0
PERM         755
IOINTR       ALLOW
NONEMPTY     DFLTPSET
EMPTY        FAIL
LASTSPU      DFLTPSET

#
```

Create a new Pset using processor IDs 4 and 8, using the `psrset -c...` command. Then list all Psets using the `psrset -i` command.

```
# psrset -c 4 8
successfully created pset 2
successfully assigned processor 4 to pset 2
successfully assigned processor 8 to pset 2
# psrset -i
PSET         0
SPU_LIST     0    3
OWNID        0
GRPID        0
PERM         755
IOINTR       ALLOW
NONEMPTY     DFLTPSET
EMPTY        FAIL
LASTSPU      DFLTPSET

PSET         2
SPU_LIST     4    8
OWNID        0
GRPID        3
PERM         755
IOINTR       ALLOW
```

```
NONEMPTY    DFLTPSET
EMPTY       FAIL
LASTSPU     DFLTPSET

#
```

Example 10-2 **Destroying a Pset and Reassigning Processors**

List the local nPartition's Pset configuration using the `psrset -i` command. There are three Psets: the default Pset 0, Pset 10, and Pset 11.

```
# psrset -i
PSET        0
SPU_LIST    0    1    2    3    4
OWNID       0
GRPID       0
PERM        755
IOINTR      ALLOW
NONEMPTY    DFLTPSET
EMPTY       FAIL
LASTSPU     DFLTPSET

PSET        10
SPU_LIST    9    10   11
OWNID       0
GRPID       3
PERM        755
IOINTR      ALLOW
NONEMPTY    DFLTPSET
EMPTY       FAIL
LASTSPU     DFLTPSET

PSET        11
SPU_LIST    5    6    7    8
OWNID       0
GRPID       3
PERM        755
IOINTR      ALLOW
NONEMPTY    DFLTPSET
EMPTY       FAIL
LASTSPU     DFLTPSET

#
```

Destroy Pset 10 because it is no longer needed. Its processors (9, 10, and 11) are assigned back to the default processor set (Pset 0). Then list the new Pset configurations using the `psrset -i` command.

```
# psrset -d 10
successfully destroyed pset 10
# psrset -i
PSET         0
SPU_LIST     0    1    2    3    4    9    10   11
OWNID        0
GRPID        0
PERM         755
IOINTR       ALLOW
NONEMPTY     DFLTPSET
EMPTY        FAIL
LASTSPU      DFLTPSET

PSET         11
SPU_LIST     5    6    7    8
OWNID        0
GRPID        3
PERM         755
IOINTR       ALLOW
NONEMPTY     DFLTPSET
EMPTY        FAIL
LASTSPU      DFLTPSET

#
```

Assign processors 4 and 9 to Pset 11 using the `psrset -a...` command. Then list the new Pset configurations using `psrset -i`, which shows the processor assignments for all Psets (Pset 0 and Pset 11).

```
# psrset -a 11 4 9
successfully assigned processor 4 to pset 11
successfully assigned processor 9 to pset 11
# psrset -i
PSET         0
SPU_LIST     0    1    2    3    10   11
OWNID        0
GRPID        0
PERM         755
IOINTR       ALLOW
NONEMPTY     DFLTPSET
EMPTY        FAIL
LASTSPU      DFLTPSET

PSET         11
SPU_LIST     4    5    6    7    8    9
OWNID        0
GRPID        3
PERM         755
IOINTR       ALLOW
NONEMPTY     DFLTPSET
EMPTY        FAIL
```

```
LASTSPU     DFLTPSET

#
```

Example 10-3 Example of Running and Binding Programs in Psets

List the current Pset configuration for the local nPartition. Two Psets are configured: Pset 0 and Pset 2.

Note that processor ID 10 is not active in this nPartition (because iCOD software has deactivated it).

```
# psrset -i
PSET        0
SPU_LIST    0    2    9
OWNID       0
GRPID       0
PERM        755
IOINTR      ALLOW
NONEMPTY    DFLTPSET
EMPTY       FAIL
LASTSPU     DFLTPSET

PSET        2
SPU_LIST    1    3    4    5    6    7    8    11
OWNID       0
GRPID       3
PERM        755
IOINTR      ALLOW
NONEMPTY    DFLTPSET
EMPTY       FAIL
LASTSPU     DFLTPSET

#
```

Use the `mpsched` command to run the “potato” program and bind it to processor ID 2. Then use the `psrset -q...` command to list the Pset binding for “potato” (process ID 10368); “potato” is bound to Pset 0.

```
# mpsched -c 2 ./potato -n 7
Pid 10368: bound to processor 2 using the default process launch policy
Threads = 7
tid = 2  cpu = 2
tid = 3  cpu = 2
tid = 4  cpu = 2
tid = 5  cpu = 2
tid = 6  cpu = 2
tid = 7  cpu = 2
tid = 1  cpu = 2
```

```
# psrset -q 10368
PID  10368       PSET  0
#
```

Use the `psrset -b...` command to change the Pset binding for "potato" to Pset 2. Then use `psrset -q...` to confirm that it is bound to Pset 2, and use the `mpsched -q...` command to check its processor binding.

```
# psrset -b 2 10368
successfully bound pid 10368 to pset 2
# psrset -q 10368
PID  10368       PSET  2
# mpsched -q -p 10368
Pid 10368: bound to processor 3 using the default process launch policy
#
```

Use the `sar` command to list the current nPartition's processor usage, including Pset details. Note that processor ID 3 in (Pset 2) is heavily loaded by the "potato" program.

```
# sar -u -M -P 1

HP-UX feshd5a B.11.11 U 9000/800    10/23/01

00:17:11    pset     cpu    %usr    %sys    %wio   %idle
00:17:12       0       0       0       0       7      92
               2       1       0       0      10      90
               0       2       0       0       8      92
               2       3     100       0       0       0
               2       4       0       0       4      96
               2       5       0       0       8      92
               2       6       0       0       7      93
               2       7       0       0       9      91
               2       8       0       0       5      95
               0       9       0       1       9      90
               2      11       0       0       8      92
                  system       9       0       7      84
#
```

Use the `mpsched -u...` command to *unbind* the "potato" program (process ID 10368) from processor ID 3, to allow the program's threads to migrate to other processors in the Pset to which it is bound.

Then use `sar` to list the local nPartition's current processor usage, including all processor and Pset details.

Since the "potato" program was unbound from processor 3, its threads were able to migrate to the other processors in the Pset to which it "potato" is bound (Pset 2).

All processors in Pset 2 are being used fairly heavily, while processors in Pset 0 are 100% idle. This is due to Pset processor resource isolation: by default each program only uses processors in the Pset in which it is run. (The Pset programming interface can override this default to launch threads and processes in other Psets, given the right conditions.)

```
# mpsched -u -p 10368
Pid 10368: not bound using the default process launch policy
# sar -u -M -P 1

HP-UX feshd5a B.11.11 U 9000/800    10/23/01

00:24:26    pset     cpu    %usr    %sys    %wio   %idle
00:24:27       0       0       0       0       0     100
               2       1     101       0       0       0
               0       2       1       0       0     100
               2       3     101       0       0       0
               2       4     100       1       0       0
               2       5      96       0       0       5
               2       6     101       0       0       0
               2       7     101       0       0       0
               2       8      18       2       0      81
               0       9       0       1       0     100
               2      11      88       0       0      13
                  system      64       0       0      36
#
```

Example 10-4 **Managing Pset Permissions and Attributes**

This example modifies Pset owner, group, and access permissions; lists various Pset details; and includes other sample Pset uses by various users on the system.

Use `psrset -i` to list the current Pset configuration for the local nPartition. Three Psets are configured: Pset 0, Pset 7, and Pset 8.

```
# psrset -i
PSET        0
SPU_LIST    0    1    2    3    4    5
OWNID       0
GRPID       0
PERM        755
IOINTR      ALLOW
NONEMPTY    DFLTPSET
EMPTY       FAIL
LASTSPU     DFLTPSET
```

```
PSET        7
SPU_LIST    9    10   11
OWNID       0
GRPID       3
PERM        755
IOINTR      ALLOW
NONEMPTY    DFLTPSET
EMPTY       FAIL
LASTSPU     DFLTPSET

PSET        8
SPU_LIST    6    7    8
OWNID       0
GRPID       3
PERM        755
IOINTR      ALLOW
NONEMPTY    DFLTPSET
EMPTY       FAIL
LASTSPU     DFLTPSET

#
```

Modify the group setting for Pset 8 to be group ID 20. Also use `psrset -t...` to modify the permissions for Pset 8. Setting the permissions to 774 allows the owner (`root`) and users in group ID 20 (the group named `users`) to execute, write, and read Pset 8; all others can only read details about the Pset's configuration.

```
# psrset -t 8 GRPID=20
# psrset -t 8 PERM=774
#
```

Using the `psrset -t...` command, change the owner for Pset 7 to user ID 103 (the user named `ann`).

```
# psrset -t 7 OWNID=103
#
```

Now `ann` uses the `id` command to list the user ID and group memberships for her user account. She then lists the current Pset configuration for the local nPartition.

The `ann` user account gives her execute and read access to Pset 0, ownership of Pset 7 (including execute, write, and read access), and execute, write, and read access for Pset 8.

She (`ann`) is considered one of the "others" (access permissions 5) for Pset 0, the owner (user ID 103, with access permissions 7) for Pset 7, and a group member (group ID 20, access permission 7) for Pset 8.

```
ann $ id
uid=103(ann) gid=20(users) groups=102(prog)
ann $ psrset -i
PSET        0
SPU_LIST    0    1    2    3    4    5
OWNID       0
GRPID       0
PERM        755
IOINTR      ALLOW
NONEMPTY    DFLTPSET
EMPTY       FAIL
LASTSPU     DFLTPSET

PSET        7
SPU_LIST    9    10   11
OWNID       103
GRPID       3
PERM        755
IOINTR      ALLOW
NONEMPTY    DFLTPSET
EMPTY       FAIL
LASTSPU     DFLTPSET

PSET        8
SPU_LIST    6    7    8
OWNID       0
GRPID       20
PERM        774
IOINTR      ALLOW
NONEMPTY    DFLTPSET
EMPTY       FAIL
LASTSPU     DFLTPSET

ann $
```

Because ann is the owner for Pset 7, she has authority to modify the Pset's user, group, and access permissions attributes.

Using the `psrset -t...` command, ann sets the group for Pset 7 to 102 (the group named prog). Another `psrset -t...` command sets access permissions for Pset 7 to 770, which gives the owner (ann) and prog group members access to execute, write, and read the Pset. All others have no permissions to use or read Pset 7.

```
ann $ psrset -t 7 GRPID=102
ann $ psrset -t 7 PERM=770
ann $
```

Now ann assigns processor ID 8 to Pset 7, using the `psrset -a...` command.

Processor 8 was assigned to Pset 8, but ann can reassign it because she has write permission for Pset 8 (she is a member of group ID 20, which has execute, write, and read permissions).

Likewise, ann can assign the processor to Pset 7 because she has write permissions there (she is the owner, and has execute, write, and read permissions).

Then ann lists the new configurations for Pset 7 and Pset 8 using the `psrset -i 7 8` command.

```
ann $ psrset -a 7 8
successfully assigned processor 8 to pset 7
ann $ psrset -i 7 8
PSET        7
SPU_LIST    8    9    10   11
OWNID       103
GRPID       102
PERM        770
IOINTR      ALLOW
NONEMPTY    DFLTPSET
EMPTY       FAIL
LASTSPU     DFLTPSET

PSET        8
SPU_LIST    6    7
OWNID       0
GRPID       20
PERM        774
IOINTR      ALLOW
NONEMPTY    DFLTPSET
EMPTY       FAIL
LASTSPU     DFLTPSET

ann $
```

Now a different user in the same nPartition attempts to list and use the new Pset configurations.

This user, `joe`, lists his user ID and the IDs for the groups to which he belongs, and then lists all Pset configurations using the `psrset -i` command. Note that because `joe` does not have read permission for Pset 7, he cannot view its attribute values (he is not the owner or a member of the Pset's group, so as one of the “others” he has no permissions).

```
joe $ id
uid=102(joe) gid=20(users)
joe $ psrset -i
PSET        0
SPU_LIST    0    1    2    3    4    5
OWNID       0
GRPID       0
PERM        755
IOINTR      ALLOW
NONEMPTY    DFLTPSET
EMPTY       FAIL
LASTSPU     DFLTPSET

PSET        7
SPU_LIST    8    9    10   11
psrset: no privileges for query operation on this pset

PSET        8
SPU_LIST    6    7
OWNID       0
GRPID       20
PERM        774
IOINTR      ALLOW
NONEMPTY    DFLTPSET
EMPTY       FAIL
LASTSPU     DFLTPSET

joe $
```

When `joe` uses the `psrset -e 7...` command to attempt to execute the “potato” program in Pset 7, he cannot because he does not have execute permission in the Pset.

However, when `joe` uses the `psrset -e 8...` command to execute “potato” in Pset 8 the program is run in that Pset. He can run programs in Pset 8 because he is a member of group ID 20 and members of that group have execute, write, and read permission for the Pset.

```
joe $ psrset -e 7 ./potato
psrset: no privileges to perform operation
joe $ psrset -e 8 ./potato
Threads = 2
tid = 1  cpu = 6
tid = 2  cpu = 7
...
```

11 Virtual Partitions (vPars) Management on nPartitions

This chapter describes how to create, configure, and manage HP's virtual partitions within an nPartition (hard partition) system environment. Each virtual partition can boot a single instance of the HP-UX B.11.11 operating system.

The HP **virtual partitions (vPars)** software is an *optional* feature that you can use to further subdivide a server's resources into multiple, smaller virtual machines through software partitioning.

By configuring multiple virtual partitions within an nPartition, you can boot multiple instances of HP-UX B.11.11 in a single nPartition.

For detailed tasks for configuring virtual partitions within an nPartition, see *Procedures for Managing Virtual Partitions on HP nPartition Servers* on page 475.

NOTE This chapter describes the current A.02.02 vPars software release, which supports HP rp7405/rp7410, HP rp8400, and HP Superdome servers.

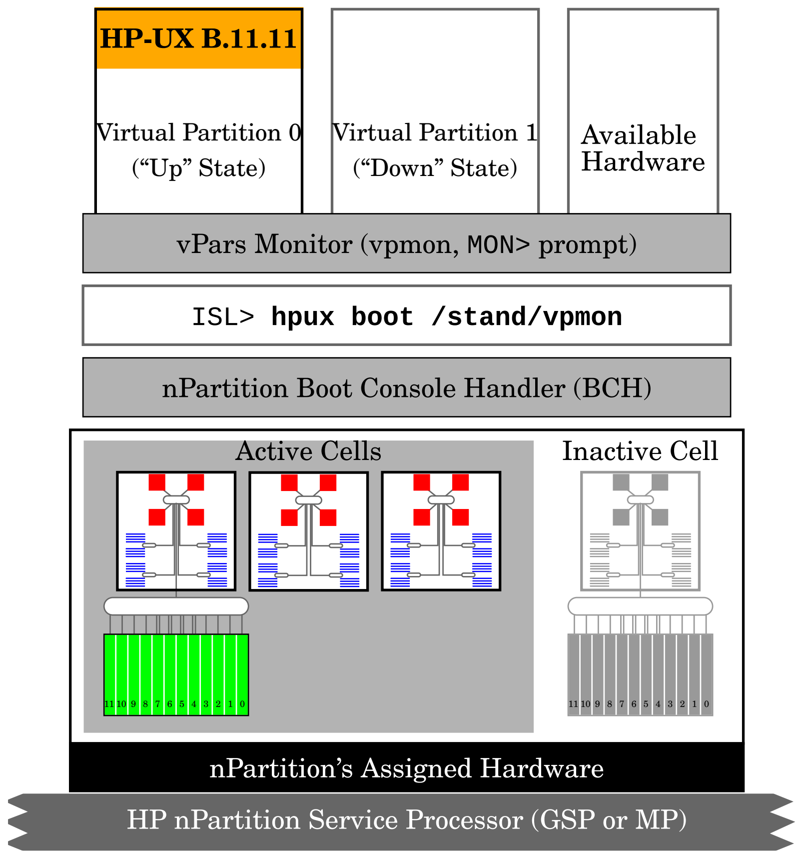
For an introduction to nPartition features, refer to the chapter *nPartition System Overviews* on page 31.

Also refer to the book *Installing and Managing HP-UX Virtual Partitions (vPars)* for more details about HP's vPars software.

Introduction to Managing Virtual Partitions on nPartitions

Figure 11-1 shows how an nPartition can boot vPars software from its BCH interface, thus enabling one or more virtual partitions to run HP-UX B.11.11 on a subset of the nPartition's active hardware.

Figure 11-1 Overview of HP Virtual Partitions (vPars) in an nPartition



Virtual Partition Hardware

On HP nPartition servers, each virtual partition is assigned a subset of its nPartition's hardware. Only the *active hardware assigned to the local nPartition* can be used by virtual partitions within the nPartition.

Hardware that is assigned to remote nPartitions and hardware that is inactive cannot be used by virtual partitions in the local nPartition.

Each virtual partition runs its own instance of HP-UX and has its own dedicated hardware resources. You can reconfigure virtual partitions and can dynamically reallocate certain processors among virtual partitions in the local nPartition, but you cannot share resources across virtual partitions or across nPartitions.

The hardware assigned to each virtual partition includes: processors (CPUs), memory, and input/output busses. Each I/O bus can have a PCI card installed and devices attached.

Each virtual partition should be assigned at least one I/O bus (LBA) that has a boot device with HP-UX B.11.11 and the vPars software product installed. An I/O bus with a network interface card also should be assigned to each virtual partition to support networking. See *vPars Requirements and Recommendations* on page 454 for more details.

vPars Software, Booting, and Consoles

Running virtual partitions in an nPartition involves installing the HP-UX virtual partitions software product, configuring one or more virtual partitions, and then booting the vPars monitor (`/stand/vpmon`) on the nPartition and loading/booting HP-UX on each of the virtual partitions.

By booting the `/stand/vpmon` virtual partitions monitor instead of the `/stand/vmunix` HP-UX kernel, an nPartition provides an additional boot loader specifically for virtual partitions.

Each nPartition can be configured to automatically boot virtual partitions, and virtual partitions can individually be configured to be booted manually or automatically. See *Procedures for Managing Virtual Partitions on HP nPartition Servers* on page 475 for details.

Only one vPars monitor is booted per nPartition. All virtual partitions in an nPartition share the same console device: the nPartition's console. See *Virtual Partition Console and Log Use on nPartition Servers* on page 465 for details.

vPars HP-UX B.11.11 Kernel

The vPars software installation builds a relocatable, vPars-enabled HP-UX B.11.11 kernel and installs patches, commands, and `vpmon` to support the vPars software environment. See *Installing and Configuring vPars on nPartition Servers* on page 472 for details.

You can load and run a vPars-enabled HP-UX B.11.11 kernel in both vPars environments and non-vPars environments. You *do not* need to reconfigure a vPars-enabled kernel for non-vPars use.

nPartition and vPars Performance

In general in HP nPartition virtual partitions environments, HP-UX B.11.11 and application *performance is nearly equivalent* to the performance given by a non-vPars nPartition that has the same hardware and software resources and configuration.

Also see the document *HP-UX Virtual Partitions Ordering and Configuration Guide* for more virtual partitions performance info.

The main performance factor for virtual partitions running in nPartitions is the underlying nPartition's hardware configuration: the cells and corresponding processors, memory, and I/O assigned to and actively used in the nPartition.

As in non-vPars nPartition environments, all memory is interleaved across all active cells in the nPartition when virtual partitions are running in an nPartition. Also on all HP nPartition servers, each processor has its own runway bus for communication to memory and I/O.

As a result, the locations (hardware paths) of processors assigned to a virtual partition *do not* affect performance. In general all processors have the same memory latency when accessing any significant amount of memory in an nPartition.

The rest of this chapter covers requirements, guidelines, procedures, and tools for using virtual partitions on HP nPartition-capable servers.

Configuring Virtual Partition Resources and Attributes

When creating or reconfiguring a virtual partition, you manage **resources and attributes** that determine the virtual partition's configuration and capabilities.

Each virtual partition has three types of resources: `cpu`, `io`, and `mem`, which specify processor(s), I/O, and memory allocated exclusively for the virtual partition.

The virtual partition resource configuration determines which hardware is dedicated for the virtual partition's use, by indicating hardware paths, quantities, and limits.

Each virtual partition also has three types of attributes: general attributes, hardware attributes, and boot attributes.

NOTE

To modify most virtual partition *hardware* resources or attributes, you must ensure that the virtual partition being modified is in a "Down" state.

Also note that some virtual partition attributes are required and some are optional.

See the *vparmodify* (1M) and *vparresources* (5) manpages for details.

The following list includes details and command-line options for setting virtual partition attributes. Also see the *vparcreate* (1M) and *vparmodify* (1M) command manpages.

- **Virtual Partition General Attributes**

  The **general virtual partition attributes** include the *name* of the virtual partition and the *`static`* attribute.

  The *name* attribute (`-p` and `-P`) defines the virtual partition's name, which you use when referencing or managing the virtual partition using commands.

The *static* attribute (`-S`) defines whether the virtual partition can be reconfigured. See *Dynamic and Static Virtual Partitions* on page 461 for details.

- **Virtual Partition Hardware Resource Attributes**

  Virtual partition hardware resource attributes include specifications for the processors, I/O, and memory that are dedicated for use by the virtual partition.

  You can add (`-a`), delete (`-d`), and modify (`-m`) virtual partition hardware resources and attributes.

  Also see the *vparresources* (5) manpage for details.

  Descriptions of processor (`cpu`), I/O (`io`), and memory (`mem`) virtual partition hardware resource attributes are in the following list.

  — Processors (`cpu`) resources — You can specify the following attributes for processors:

  The *path* of one or more processors that are bound to the virtual partition. For example, to set the processor at hardware path 0/10 to be bound to the virtual partition:

  ```
  # vparmodify -p name -m cpu:0/10
  ```

  A *minimum* and *maximum* number of processors allowed in the virtual partition. For example, to set the minimum number of processors to 2 and the maximum to 4 processors:

  ```
  # vparmodify -p name -m cpu:::2:4
  ```

  The *total* number of processors in the virtual partition. For example, to set the total number of processors to 6:

  ```
  # vparmodify -p name -m cpu::6
  ```

  — Input/Output (`io`) — You can optionally specify `boot`, `altboot`, and other attributes for each I/O device path assigned to a virtual partition.

  The `boot` attribute specifies the primary (PRI) boot device path for the virtual partition, which is stored in the vPars database (vpdb) and is separate from the nPartition boot device path settings.

The `altboot` attribute specifies the alternate (ALT) boot device path for the virtual partition, which also is separate from nPartition boot settings.

For example, to set the specified virtual partition's PRI boot device path to 0/0/6/0/0.5 (the corresponding nPartition's PRI path is *not* changed, however):

`#` **`vparmodify -p`** ***`name`*** **`-m io:0/0/6/0/0.5:boot`**

— Memory (`mem`) — You can specify the total (`-m mem::`*`size`*) memory *`size`* in MBytes for a virtual partition, and can increase (`-a`) or decrease (`-d`) the amount of memory,

As needed, the specified *`size`* is rounded up to a 64 MByte boundary.

For example, to configure a virtual partition to have 2 GBytes (2048 MBytes) of memory allocated:

`#` **`vparmodify -p`** ***`name`*** **`-m mem::2048`**

NOTE

HP recommends that you *only specify the total amount of memory* to be allocated for each virtual partition. On all supported HP virtual partition systems there is no benefit to specifying the base and range for memory.

Each nPartition's memory is interleaved across all active cells in the nPartition and thus all useful ranges of virtual partition memory will span all cells.

- **Virtual Partition Boot Attributes**

  Virtual partition boot attributes include the *`autoboot`* setting, the *`kernel path`* attribute, the *`boot options`* attribute, and *`io resources`* attributes.

  The *`autoboot`* attribute (`-B`) determines whether a virtual partition is booted (`-B auto`) or not booted (`-B manual`) when the virtual partition is reset. This attribute also affects virtual partition boot behavior when the vPars monitor is loaded with the `vpmon -a` option or when the `vparload -auto` command is issued from the `MON>` prompt.

The *kernel path* attribute (-b) specifies the path of the vPars-enabled HP-UX B.11.11 kernel that is to be booted when the virtual partition is loaded. By default the /stand/vmunix kernel on the boot device is used.

The *boot options* attribute (-o) specifies the options that are applied when the virtual partition's HP-UX B.11.11 kernel is booted. These boot options are equivalent to the secondary system loader options described in the *hpux* (1M) manpage.

You can use the *io resources* attributes (-a io..., -m io...) to designate primary (PRI) and alternate (ALT) boot device paths for a virtual partition, as explained in *Virtual Partition Hardware Resource Attributes* on page 446.

Tools for Managing Virtual Partitions on nPartition Servers

The main tools for virtual partitions administration are the HP-UX vPars commands and the Virtual Partition Manager (`vparmgr`) utility.

This section briefly lists these and other tools and commands you can use for managing virtual partitions on HP nPartition servers.

- **HP-UX Virtual Partitions Commands**

  The HP-UX vPars commands create, modify, and provide status and configuration info about the virtual partitions in the *currently active* vPars database (`/stand/vpdb`), or any other accessible vPars database that you specify.

  The vPars commands list status or modify configuration details for virtual partitions in the local nPartition. They cannot modify or list info about virtual partitions running in remote nPartitions.

  Using all vPars commands requires `root` permission.

  In most cases the vPars commands are used after you have booted one or more virtual partitions in an nPartition. However, you also can use some vPars commands when you have booted HP-UX in a non-vPars nPartition environment, such as when initially configuring virtual partitions.

  See the *vpartition* (5) manpage for a list and descriptions of the vPars commands, including `vparstatus`, `vparcreate`, `vparmodify`, `vparboot`, `vparreset`, and others.

- **Virtual Partition Manager (`vparmgr`) Utility**

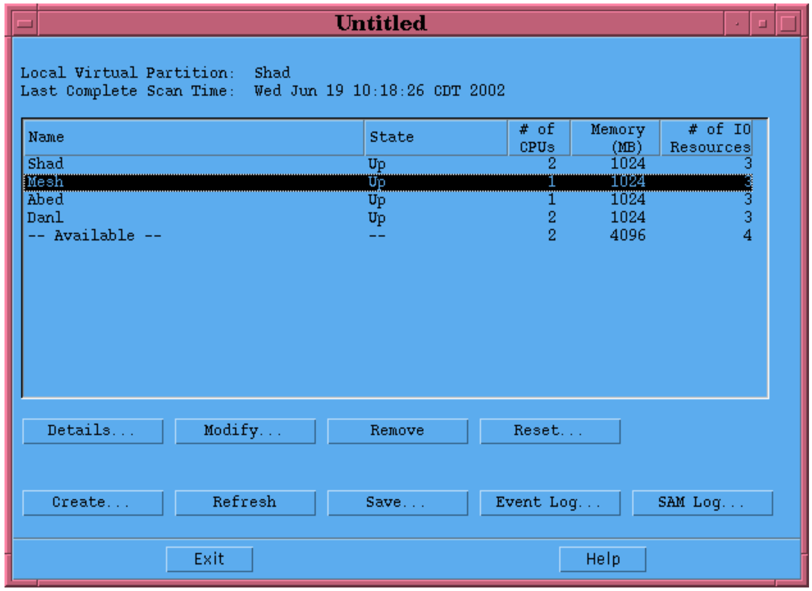
  The Virtual Partition Manager utility provides a graphical interface to the HP-UX vPars commands. Using Virtual Partition Manager, you can perform virtual partitions administration tasks from HP-UX running on a virtual partition. You cannot use Virtual Partition Manager when HP-UX is booted in non-vPars mode.

NOTE The Virtual Partition Manager utility *is not installed* as part of the virtual partitions software product installation. Instead, you must install Virtual Partition Manager separately, as described in the the book *Installing and Managing HP-UX Virtual Partitions (vPars)*.

Using the Virtual Partition Manager utility requires `root` permission.

Because Virtual Partition Manager is an X window graphical utility, you must set and export the virtual partition system's DISPLAY environment variable before launching `vparmgr`. The DISPLAY variable specifies where (which X server) the system displays X windows. You also must use the `xhost` command on the X server (your local system) to grant access for the virtual partition system to display windows on the X server. See the *X* (1) and *xhost* (1) manpages for details.

The following window is the Virtual Partition Manager utility's status window, which is the first window displayed after any alert messages. This window lists the status of all virtual partitions defined in the current vPars database as well as general details about available resources.



Virtual Partition Manager has online help that you can view at any time by clicking the **Help** button, which displays info in a separate Web browser. You also can view Virtual Partition Manager help from a Web browser by issuing the following command:

```
/opt/netscape/netscape file:/opt/webadmin/vparmgr/help/C/overview.html
```

See the online help for complete details on using the Virtual Partition Manager.

All Virtual Partition Manager tasks also can be performed using the HP-UX vPars commands, which are described in the *vpartition* (5) manpage.

- HP-UX nPartition Commands: `parstatus` and Others

  The `parstatus` command can list nPartition status info as well as details about hardware assigned to the local nPartition and other hardware throughout the entire nPartition server complex.

  See the *parstatus* (1) manpage for details.

  The other HP-UX nPartition commands, such as `parmodify`, also are supported when running HP-UX in a virtual partition on an nPartition. Likewise the Partition Manager tool is supported.

- HP-UX `setboot` Command

  The HP-UX `setboot` command affects the current virtual partition's boot settings (stored in its `/stand/vpdb`) when you use it in a virtual partition environment.

  When used in a *non-vPars* nPartition environment, the `setboot` command affects the local nPartition's boot settings. The nPartition boot device paths are stored in the nPartition's Partition Configuration Data portion of the server's Complex Profile.

- Tools for Boot Device AUTO File Management

  HP-UX commands to set (`mkboot`) and list (`lifcp`) a device's AUTO file: **`mkboot -a`** ***`STRING`*** **`/dev/dsk/...`** and **`lifcp /dev/dsk/...:AUTO -`**

  ISL commands to list (`hpux show...`) and set (`hpux set...`) an AUTO file: **`hpux show autofile`** and **`hpux set autofile`** ***`STRING`***

  vPars monitor (`MON>`) command to list an AUTO file: **`getauto`**

  See the *mkboot* (1M), *lifcp* (1), and *hpux* (1M) manpages.

- HP nPartition Virtual Front Panel (VFP)

  The nPartition VFP indicates the boot status for all cells and virtual partitions in the nPartition. As long as at least one virtual partition is running HP-UX the VFP will display an HP-UX "heartbeat".

  For details, refer to the chapter *Using Console and Service Processor Interfaces* on page 125.

- HP nPartition Console and Virtual Partition Consoles

  Each nPartition console provides access to BCH for the nPartition, allows you to boot HP-UX or the vPars monitor on an nPartition, and permits access to all virtual partition console interfaces in the nPartition.

  For details, see *Virtual Partition Console and Log Use on nPartition Servers* on page 465.

- HP vPars Monitor (`vpmon`) Commands

  At the vPars monitor (`MON>`) prompt enter **`?`** or the **`help`** command to list all available vPars monitor commands.

The vPars monitor commands include: `reboot` (reboot the nPartition), `vparload` (load/boot one or more virtual partitions), `scan` (scan and list all active hardware in the local nPartition), `log` (list recent history from the vPars monitor's event log), and other commands.

The vPars monitor's `MON>` prompt is available when the nPartition's monarch processor is not assigned to a virtual partition that has been loaded/booted.

- HP nPartition Service Processor (GSP or MP) Commands

  Service processor commands that reboot or reset an nPartition affect all virtual partitions within the nPartition.

  For details, refer to the chapter *Using Console and Service Processor Interfaces* on page 125.

- HP nPartition Server Chassis Log Viewer (SL)

  HP virtual partitions-related details that are accessible as chassis logs include the HP-UX "heartbeat" emitted when HP-UX is running on each virtual partition. Otherwise, all vPars-specific event logs are stored in the vPars event log.

  For details, see the section *Virtual Partition Console and Log Use on nPartition Servers* on page 465.

vPars Requirements and Recommendations

HP offers the following requirements and recommendations for configuring virtual partitions in HP nPartition environments.

Additional recommendations for avoiding obstacles to loading/booting virtual partitions are in *Fault-Tolerant Virtual Partition Configurations for nPartitions* on page 457.

NOTE See the *HP-UX Virtual Partitions Ordering and Configuration Guide* for the latest requirements.

Configuration Requirements and Recommendations for Virtual Partitions

- ❏ The following software releases, or later, **must** be installed for complete vPars support on HP nPartition-capable servers:
  - — Any HP-UX B.11.11 release (December 2000 or later).
  - — The A.02.02 virtual partitions software product.
  - — The Partition Manager B.11.11.01.05 product, which must be installed *before* the A.02.02 vPars software is installed.
  - — The Superdome SMS Software V1.2 release (including firmware).
- ❏ Each nPartition in which virtual partitions are configured **must** have *no more than* eight cells assigned to it, and all the nPartition's cells **must** reside in the same cabinet.
- ❏ Each nPartition **must** have *no more than* eight virtual partitions configured.

  If you require more than eight virtual partitions in the same HP nPartition complex, configure the virtual partitions in multiple nPartitions.
- ❏ Hardware to be used by virtual partitions within an nPartition **must** be *assigned* to the local nPartition and must be *active* hardware.

Because each nPartition only provides access to the hardware that is assigned to and active within the local nPartition, any virtual partitions in the nPartition are limited to using this same set of currently available nPartition hardware.

Adding or removing hardware from an nPartition changes the local set of hardware that is available to virtual partitions in the nPartition. Likewise, making nPartition hardware inactive makes it unavailable to virtual partitions.

- ❏ At least one processor **must** be bound to each virtual partition.

  Only bound processors can handle I/O interrupts. Other processors in the virtual partition can be either bound or unbound.

- ❏ A multiple of 64 MBytes of memory **must** be assigned to each virtual partition.

  When you specify the memory size of each virtual partition, the commands involved automatically round the memory assignment upward as required to a 64-MByte boundary.

  Memory in HP nPartitions is interleaved across all active cells in the local nPartition. As a result the memory used by each virtual partition may physically reside on all active cells in the nPartition where the virtual partitions exist.

- ❏ Each virtual partition **must** have at *least one* I/O bus (LBA) assigned to it.

  On HP nPartition servers, each LBA corresponds to a PCI card slot in an I/O chassis attached to an active cell in the local nPartition.

  For I/O slot details, see the section *Planning Virtual Partition Configurations for HP nPartition Servers* on page 467.

- ❏ Each virtual partition **must** have at least one bootable disk accessible through a PCI card in one of the I/O busses assigned to the virtual partition.

  The bootable disk must have both HP-UX B.11.11 and the HP virtual partitions software package installed.

- ❏ The HP processor pay per use (PPU, or iCOD Utility) product is *not yet supported for virtual partitions* and **must not** be installed or configured for nPartition systems running vPars.

The HP processor Instant Capacity on Demand (iCOD Purchase) release B.05.00 software may be installed and used with vPars software on HP nPartition-capable servers.

A future release of iCOD Utility (pay per use) also will support processor capacity on demand for nPartition servers running vPars.

- ❏ Each virtual partition **should** have at least one LAN card or port available through one of the I/O busses assigned to the virtual partition.

  The LAN port is required if HP-UX networking is to be supported.

  HP recommends that, for best performance, you do not configure HP-UX lan0 to use the nPartition's Core I/O LAN (on HP Superdome servers, the LAN at hardware path `*cell*/0/0/1/0`).

  The HP Superdome Core I/O card is a PCI-1x card that possibly provides lower performance than a comparable PCI-2x or PCI-4x card.

- ❏ If you require that a virtual partition not be reconfigured then you **should** set the virtual partition to be "static". For details, see *Dynamic and Static Virtual Partitions* on page 461.

The next section gives detailed guidelines for creating fault-tolerant virtual partition configurations on nPartitions.

Fault-Tolerant Virtual Partition Configurations for nPartitions

You **should** consider the following recommendation if you intend to establish virtual partition configurations that are tolerant of failures, such as hardware self-test failures, or nPartition configuration changes, such as removing a cell from an nPartition.

NOTE The guidelines in this section only apply to virtual partitions that are configured in multiple-cell nPartitions.

These guidelines are not for single-cell nPartitions.

Configuration Guidelines for Fault-Tolerant Virtual Partitions

To configure virtual partitions in an nPartition so that the virtual partitions still can load/boot HP-UX when not all expected hardware is active or available, use the info in the following list.

- Fault-Tolerant Virtual Partition Processor (`cpu`) Configuration

  To plan and create fault-tolerant virtual partition processor configurations you need to know how many processors are available for use in an nPartition.

  The number of processors available for virtual partition use is the number of *configured* processors on *active* cells in the nPartition.

  Guidelines for **virtual partition processor fault tolerance**:

  - ❑ The number of virtual partition *`bound`* processors and the *`min`* number of virtual partition processors should anticipate that one or more cells or processors might remain inactive or be deconfigured.

    The total of all *`min`* processor attributes for all virtual partitions within an nPartition should no more than:

    (*active_cells* - 1) * (*processors_per_cell*)

    where *active_cells* is the number of currently active cells, and *processors_per_cell* is the average number of configured processors on each cell.

The *bound* attribute value cannot exceed the `min` attribute, and so the total of all *bound* processor attributes for all virtual partitions within an nPartition also should not exceed the following amount:

[(*active_cells* - 1) * (*processors_per_cell*)]

When a bound processor is unavailable, the vPars monitor assigns and binds a different processor in its place. Thus, even when some of a virtual partition's requested bound processors are unavailable, the virtual partition still has the same number of bound processors requested.

In a multiple-cell nPartition, this guideline for *bound* and `min` processors can accommodate a cell being made inactive or deleted without preventing any virtual partitions from loading/booting HP-UX B.11.11.

- ❏ The *max* number of processors for each virtual partition within an nPartition can exceed the number of processors assigned to an nPartition.

  However, for example, note that it is not practical to set *max* to greater than 64 because HP-UX B.11.11 running on HP Superdome hardware does not support configurations beyond 64 processors.

- ❏ The `total` number of processors for each virtual partition can be any value, but the `total` processor attribute never needs to be greater than 64.

  To help permit the desired virtual partition processor configurations to be loaded and booted, you should ensure that the sum of all virtual partition `total` processor attributes does not exceed the number of assigned and available processors in the nPartition.

  If not enough processors are available for a virtual partition to be assigned the configured `total`, then each virtual partition is assigned as many processors as possible after all virtual partitions are assigned their *bound* and `min` processors.

- Fault-Tolerant Virtual Partition Input/Output (`io`) Configuration

  You may need to use alternate I/O devices for virtual partitions when nPartition I/O is unavailable, such as when a cell or I/O chassis is inactive due to a failure or reconfiguration.

  Guidelines for **virtual partition input/output fault tolerance**:

  ❏ If possible, configure disks in *separate I/O chassis* for each virtual partition's `boot` and `altboot` attributes (the PRI and ALT device paths for the virtual partition).

  Having each virtual partition's `boot` and `altboot` devices on separate I/O chassis permits the virtual partitions to load and boot HP-UX even in the event of a cell failure or I/O chassis failure.

  ❏ If possible, plan for an alternate LAN to be available in a PCI slot on a separate I/O chassis from the primary LAN.

  As with the `boot` and `altboot` devices, this configuration permits a LAN device to still be available if a cell or I/O chassis fails or is inactive.

- Fault-Tolerant Virtual Partition Memory (`mem`) Configuration

  All configured/allocated memory DIMMs on all active cells in an nPartition are available for use by virtual partitions.

  If less nPartition memory is available than is configured for virtual partitions within the nPartition, one or more of the virtual partitions might be unable to load/boot.

  Guidelines for **virtual partition memory fault tolerance**:

  ❏ Specify the *`size`* of memory for each virtual partition so that the total amount of memory allocated for *all* virtual partitions is *less than or equal* to the total memory on all active cells in the nPartition.

  By configuring *less* memory to all virtual partitions than is available on the active cells in the nPartition, you can potentially avoid virtual partition load/boot problems in the event of a memory or cell failure.

  For example, when a DIMM fails on an nPartition's cell its entire rank (4 DIMMs) is deallocated. Likewise, all memory in an inactive cell is unavailable for use.

- ❏ Do not specify a *base* and *range* for memory.

  There is no benefit to specifying virtual partition memory in terms of base and range, and doing so can potentially prevent a virtual partition from booting if the base and range are unavailable.

  For example: if one or more cells does not boot, then less memory is available and some previously available bases and ranges will not exist. This can prevent virtual partitions from loading/booting if they are configured to use the unavailable memory range.

Dynamic and Static Virtual Partitions

Each virtual partition has a static/dynamic attribute that determines whether resource changes can be made to the virtual partition.

A **static virtual partition** cannot have any modifications made to its resource profile. This means that the virtual partition's processor, memory, and I/O characteristics and assignments cannot be changed, even if the virtual partition is not running (in a "Down" state).

A **dynamic virtual partition** can have its resource profile changed through the use of the `vparmodify` command.

To toggle between the `static` and `dynamic` virtual partition attribute settings, use the `vparmodify` command's `-S` option:

`vparmodify -p` ***`vpname`*** **`-S static`**

`vparmodify -p` ***`vpname`*** **`-S dynamic`**

You also can toggle this attribute between `dynamic` and `static` in a single command. For example, the following command sets the virtual partition named "Shad" to be dynamic, then modifies its total number of CPUs, then sets the virtual partition to be static.

```
# vparmodify -p Shad -S dynamic -m cpu::3 -S static
```

See the *vparmodify* (1M) manpage for details.

Note that some resource changes require that the virtual partition not only be dynamic but also be in a "Down" state.

For example, changing I/O attributes or adding and removing processors may be possible while a dynamic virtual partition is running, but changing memory or I/O assignments requires a virtual partition to be both *dynamic* and *down*.

To check virtual partition static/dynamic attribute settings, use `vparstatus`.

```
# vparstatus
[Virtual Partition]
                                                                          Boot
Virtual Partition Name         State Attributes Kernel Path               Opts
============================== ===== ========== ========================= =====
Shad                           Down  Dyn,Manl   /stand/vmunix
Mesh                           Up    Stat,Manl  /stand/vmunix             boot
```

```
[Virtual Partition Resource Summary]
                                                CPU    Num        Memory (MB)
                                      CPU     Bound/   IO   # Ranges/
Virtual Partition Name              Min/Max  Unbound  devs  Total MB    Total MB
==============================  ================  ====  ====================
Shad                                  2/  8    2   0     8    0/  0         2048
Mesh                                  2/ 12    2   6     3    0/  0         2048
# vparmodify -p Mesh -m cpu::4
vparmodify: Error: Virtual partition Mesh is static, cannot modify resources.
#
```

Virtual Partition Configuration Data on nPartitions

This section covers configuration data issues related to using virtual partitions in nPartitions.

Virtual partition configuration data by default is stored in the `/stand/vpdb` file, although you can specify that another file be used as the vPars database.

When you have multiple virtual partitions booted (in an "Up" state) on an nPartition, the vPars databases for all booted virtual partitions are kept coherent; any changes to virtual partition configurations are saved in each of the booted virtual partition's databases.

Virtual partition configuration data *is not* stored in the nPartition complex profile data. As a result, virtual partition changes do not affect nPartition configurations.

The following list describes some issues related to managing both nPartition and virtual partitions configuration data in an nPartition.

- Boot Paths for Virtual Partitions and nPartitions

  Each nPartition's boot path variables (PRI, HAA, ALT) are stored in the nPartition's profile data.

  The virtual partition boot device paths (PRI, ALT) are stored in the vPars configuration database.

  The `parmodify` and `parstatus` commands always can report and modify nPartition boot device path details.

  When one or more virtual partitions is booted in an nPartition, the `setboot` command affects the *current virtual partition's* boot paths and settings.

  When HP-UX is booted in a *non-vPars nPartition*, the `setboot` command and others affect the local nPartition.

- Virtual Partition Configuration Data Coherency

  Only the vPars configuration database (`/stand/vpdb`) residing on each virtual partition's currently booted device's filesystems is updated to reflect any changes.

Any booted virtual partition that has multiple boot devices (such as `boot` and `altboot`) can have one current and one outdated copy of virtual partition data.

For example: when a virtual partition boots from its ALT device path and has a config change, and then the virtual partition's nPartition reboots and loads `/stand/vpmon` from the same virtual partition's PRI device path, then as a result the latest virtual partition config changes are not used (but they still reside on the virtual partition's ALT device). Instead, in this example, the `/stand/vpdb` vPars database on the virtual partition's PRI device is used for configuring the nPartition's virtual partitions.

Virtual Partition Console and Log Use on nPartition Servers

On HP nPartition-capable servers, each nPartition has its own console and its own console log that stores a record of recent console activity.

When an nPartition has one or more virtual partitions booted, the nPartition console serves as the console for all virtual partitions loaded/booted in the nPartition.

NOTE To switch among virtual partition console interfaces, type **Control-a** when accessing the corresponding nPartition console.

In the following example, the user of the nPartition console repeatedly types **Control-a** to cycle through the available virtual partition consoles.

```
feshd3a / Shad [HP Release B.11.11] (see /etc/issue)
Console Login:
Control-a
[Mesh]
Control-a
[Abed]
Control-a
[Danl]
Control-a
[Shad]
Control-a
[Mesh]
```

The above example starts with console access to the virtual partition named "Shad", then switches to the "Mesh" virtual partition console, "Abed" console, "Danl" console, and then back to the "Shad" and finally the "Mesh" virtual partition console.

The service processor (GSP or MP) console log stores nPartition console output, including BCH output and HP-UX `/dev/console` for nPartitions.On nPartition server running virtual partitions, all virtual partitions in the nPartition emit their `/dev/console` output to the nPartition console. Thus, when HP-UX B.11.11 is running in multiple virtual partitions in an nPartition, the nPartition console will display `/dev/console` output for more than one instance of HP-UX.

The nPartition console log also records vPars monitor (`vpmon`, `MON>` prompt) output for its nPartition, because the vPars monitor interface is accessed and displayed through the virtual partition's nPartition console.The `vpmon` event logs file—which is viewable from the `vparstatus -e` command or the vPars monitor's `log` command—only records virtual partition events. It does not record any nPartition chassis codes.The server chassis logs—which are viewable from the service processor (GSP or MP) Show Chassis Log menu—record nPartition and server complex hardware events. nPartition chassis logs do not record virtual partition configuration or virtual partition-specific load/boot events.However, as in non-vPars nPartition environments, the chassis logs do record HP-UX "heartbeat" events and related timeout counter details.

The vPars monitor prompt (`MON>`) is shared by all virtual partitions in the same nPartition and gives access to commands for loading/booting virtual partitions, displaying virtual partition and system info, reviewing event log history, and performing other tasks.

If multiple nPartitions in a server are running virtual partitions, each nPartition has its own vPars monitor, just as each nPartition runs its own instance of Boot Console Handler (BCH).

Planning Virtual Partition Configurations for HP nPartition Servers

You can use this section to help plan the virtual partition configurations you will establish in nPartitions.

This section covers the following topics:

Also see the book *Installing and Managing HP-UX Virtual Partitions (vPars)* for other virtual partition planning info.

Virtual Partition Hardware Paths

You may need to reference the following hardware path info for HP nPartition-capable servers when planning virtual partition configurations.

Figure 11-2 HP nPartition Hardware Paths for Virtual Partition Configurations

Processor (CPU) Hardware Paths

cell/10
cell/11
cell/12
cell/13

For all nPartition hardware paths, *cell* is the global cell number (0–15).

HP Superdome — PCI Card Slot Hardware Paths

Slot	11	10	9	8	7	6	5	4	3	2	1	0
Path (LBA)	*cell*/0/8	*cell*/0/9	*cell*/0/10	*cell*/0/11	*cell*/0/12	*cell*/0/14	*cell*/0/6	*cell*/0/4	*cell*/0/3	*cell*/0/2	*cell*/0/1	*cell*/0/0

HP rp7405/rp7410 and HP rp8400 — PCI Card Slot Hardware Paths

Slot	1	2	3	4	5	6	7	8
Path (LBA)	*cell*/0/8	*cell*/0/10	*cell*/0/12	*cell*/0/14	*cell*/0/6	*cell*/0/4	*cell*/0/2	*cell*/0/1

Listing Available nPartition Hardware Resources

This section presents how you can list all available hardware resources in an nPartition server.

The following are common methods of listing available hardware:

- **`vparstatus -A`** lists the available processor, I/O, and memory resources in the local nPartition. This command lists the hardware that has not yet been assigned to any virtual partition.

  Note that `vparstatus -A` command lists processors and I/O using a form of the hardware path notation where a period (`.`) separates each hardware path field rather than a slash (`/`).

- **`ioscan`** lists the assigned and active hardware in the local nPartition or the current virtual partition.

  When HP-UX is booted in a *non-vPars* environment, `ioscan` lists all all active hardware in the local nPartition.

  When HP-UX is booted in a virtual partition, `ioscan` lists only the I/O devices assigned to the current virtual partition and lists processors on active cells in the local nPartition (but it *does not* list bound processors in other virtual partitions).

  Note that `ioscan` can list some processors that are not assigned to the current virtual partition, including unbound processors assigned to other virtual partitions.

- **`mpsched -s`** lists only the assigned and active processors in the current virtual partition or nPartition.

  Note, however, that `mpsched -s` lists the HP-UX CPU IDs for processors, not their hardware paths.

- **`parstatus -C`** lists the configurations of all cells in an HP nPartition-capable server complex, including the number of processors, amount of memory, I/O chassis connections, current usage status, and nPartition assignment.

Also refer to the chapter *Listing and Managing Server Hardware* on page 305 for details.

Virtual Partition Configuration Planning

Table 11-1 is for planning virtual partition attributes and resource assignments.

For each virtual partition, you **must** specify: a virtual partition name, at least one bound CPU, at least one I/O slot, a PRI boot path, and enough memory to boot HP-UX. The *default virtual partition attributes* establish a dynamic configuration that manually boots the `/stand/vmunix` kernel with no boot options.

The “Host Info” column in Table 11-1 includes configuration details for HP-UX networking (the hostname, IP, gateway, and so on).

Also see *Configuring Virtual Partition Resources and Attributes* on page 445.

Table 11-1 Configuration Planning Table for a Virtual Partition

<table>
<tr><th>Virtual Partition Name and Host Info</th><th>Attributes (* = Default)</th><th>Processor Resources</th><th>Input/Output Resources</th><th>Memory Resources</th></tr>
<tr><td>Virtual Partition Name:
____________</td><td>Configuration:
Static
or
Dynamic*</td><td>Min: _____</td><td>Boot Path (PRI):
__/ 0 /__/__/__.__
Note: also must assign the LBA.</td><td rowspan="5">Size (MBytes):
____
Note:
1024 MBytes = 1 GByte</td></tr>
<tr><td>Hostname:
____________</td><td>Boot:
Manual*
or
Automatic</td><td>Max: _____</td><td>Altboot (ALT):
__/ 0 /__/__/__.__
Note: also must assign the LBA.</td></tr>
<tr><td>IP Address:
___.___.___.___</td><td>Kernel Path:
/stand/_______
For example:
<code>/stand/vmunix</code>*</td><td>Total: _____</td><td rowspan="3">Assigned Slots (LBAs):
___/ 0 /___
___/ 0 /___
___/ 0 /___
___/ 0 /___
___/ 0 /___
___/ 0 /___
___/ 0 /___
___/ 0 /___</td></tr>
<tr><td>Gateway:
___.___.___.___
Subnet:
___.___.___.___</td><td rowspan="2">Boot Options:
_________</td><td rowspan="2">Bound CPUs:
___/___
___/___
___/___
___/___
___/___
___/___
___/___
___/___</td></tr>
<tr><td>DNS IP:
___.___.___.___
Domain:
____________</td></tr>
</table>

Installing and Configuring vPars on nPartition Servers

This section covers information on installing the vPars software product onto disks that will be used for booting HP-UX on a virtual partition that is loaded in an nPartition.

After a boot disk has both HP-UX and the vPars software product installed, the disk can be booted to be used in either virtual partition or non-vPars environments.

The vPars software install process can occur when you have booted HP-UX from each disk so that HP-UX is running in a non-vPars environment.

You also can install *both* the HP-UX and vPars software simultaneously on a virtual partition when its nPartition is running in vPars mode. To do this issue the `vparboot -p` *`vpname`* `-I...` command from a different virtual partition in the same nPartition to load/boot the virtual partition from an Ignite-UX server. For details see the *vparboot* (1M) manpage or the vPars install documentation listed below.

NOTE The HP vPars software product must be installed on *every boot device* that will be used by virtual partitions. Each virtual partition must have a boot disk where HP-UX B.11.11 and the vPars software product are installed.

NOTE Before installing the HP vPars software product on an nPartition, you must install the Partition Manager B.11.11.01.05 product (or a later Partition Manager release).

For the latest Partition Manager software, see the **`http://software.hp.com`** Web site.

vPars Software Installation for an HP nPartition

This procedure gives a high-level overview of a process for manually installing HP's virtual partitions software product on an nPartition's disks.

The book *Read Before Installing HP-UX Virtual Partitions* has important information you should read before performing this procedure. Also refer to the book *Installing and Managing HP-UX Virtual Partitions (vPars)*.

Step 1. Boot HP-UX on the nPartition using the boot device that will be the primary (PRI) boot device for one of the nPartition's virtual partitions.

Step 2. Install the HP virtual partitions software product on the booted device.

As part of the vPars software installation, the nPartition is rebooted and a new `/stand/vmunix` HP-UX kernel is built.

Step 3. Create the first virtual partition on the device onto which you have installed the vPars software product.

NOTE Only perform this step for the boot disk from which `/stand/vpmon` will be loaded. Do not perform this step for other boot disks.

Other boot disks are automatically updated with copies of the vPars database as needed.

To complete this step, first boot HP-UX from the device, and then use the `vparcreate` and `vparmodify` commands to create the first virtual partition for the nPartition.

Creating the first virtual partition establishes a vPars database (`/stand/vpdb`) for the nPartition.

You can assign each virtual partition resources that are part of the *local* nPartition. Only hardware that is *assigned* to the local nPartition and is *active* can be used by the virtual partitions within the nPartition.

By default the `vparcreate`, `vparmodify`, and `vparstatus` commands use the `/stand/vpdb` file. Although the vPars monitor is not running as you perform this step, these commands will read and write to `/stand/vpdb` if you do not specify a different vPars database file using the `-D` option.

If you want to create multiple virtual partitions within an nPartition you can do so as part of this step, by issuing a `vparcreate` command for each new virtual partition within the local nPartition.

You also can create any additional virtual partitions later, after you have booted the vPars monitor and loaded/booted HP-UX B.11.11 on the first virtual partition.

Step 4. Issue the **`shutdown -r`** command to reboot HP-UX on the nPartition and return to the nPartition's BCH interface.

As needed, interrupt the autoboot process to interact with the nPartition at the BCH interface.

Step 5. Install the vPars software product on each remaining boot device that is to be used to boot HP-UX on one of the nPartition's virtual partitions.

For each virtual partition boot device, you can boot HP-UX from the device in non-vPars mode and then install the vPars software product on the device.

An alternate install method is to load/boot HP-UX on the first virtual partition, and then simultaneously install HP-UX and vPars software on other virtual partitions by using the the `vparboot -p` *`vpname`* `-I...` command. See the *vparboot* (1M) manpage.

You must install both HP-UX and the vPars software for every virtual partition. For example, if you plan to have three virtual partitions in an nPartition then you need at least three boot devices with HP-UX and vPars software installed.

If you intend to have multiple boot disks for a virtual partition—for example, a PRI device and an ALT device—then you need to install HP-UX and the vPars software product on *both* the PRI and ALT devices for the virtual partition.

Procedures for Managing Virtual Partitions on HP nPartition Servers

This section has procedures for performing typical virtual partitions configuration and management tasks on HP nPartition-capable servers.

The following virtual partitions procedures are covered here:

NOTE The procedures described here use the HP-UX virtual partitions commands and *not* the graphical Virtual Partition Manager utility.

You can perform all tasks in this section with either the commands or Virtual Partition Manager. For info on the Virtual Partition Manager graphical utility see its online help or see the section *Tools for Managing Virtual Partitions on nPartition Servers* on page 449.

See the section *Installing and Configuring vPars on nPartition Servers* on page 472 for details on installing vPars software on nPartition systems.

Also see the section *Managing nPartitions from a Virtual Partition* on page 506 for issues you should be aware of when you perform nPartition config tasks from a vPars environment.

The book *Installing and Managing HP-UX Virtual Partitions (vPars)* also has detailed virtual partitions management information.

Creating a New Virtual Partition

This section gives details on how to create a new virtual partition.

NOTE Before creating a virtual partition you should already have planned how the local nPartition's resources will be assigned to the virtual partitions running in the local nPartition.

See the section *Planning Virtual Partition Configurations for HP nPartition Servers* on page 467 for info on planning virtual partitions configurations for use in nPartitions.

vPars Management: Creating a New Virtual Partition in an nPartition

This procedure creates a new virtual partition from HP-UX running in a virtual partition on an nPartition.

The book *Installing and Managing HP-UX Virtual Partitions (vPars)* also has detailed virtual partitions information.

Step 1. Login to HP-UX running on one of the virtual partitions within an nPartition.

Step 2. Complete all virtual partition resource planning and confirm that the resources are available for the new virtual partition you intend to create.

See the section *Planning Virtual Partition Configurations for HP nPartition Servers* on page 467 for planning details.

Issue the **`vparstatus -A`** command to list the processors, memory, and I/O busses that are not yet assigned to any virtual partition.

The `vparstatus -A` command should list all hardware you plan to assign to the new virtual partition. If any hardware you planned to assign is not available then you must either must revise your plans or unassign or otherwise make the hardware available.

```
# vparstatus -A
[Unbound CPUs (path)]:  0.13
                        1.11
                        1.12
                        1.13
                        2.10
```

```
                        2.11
                        2.12
                        2.13
                        6.10
                        6.11
                        6.12
                        6.13
[Available CPUs]:  12

[Available I/O devices (path)]:  0.0.1
                                 0.0.3
                                 0.0.8
                                 0.0.9
                                 0.0.10
                                 0.0.11
                                 0.0.12
                                 0.0.14
                                 2.0
                                 2.0.0
                                 2.0.1
                                 2.0.2
                                 2.0.3
                                 2.0.4
                                 2.0.6
                                 2.0.8
                                 2.0.9
                                 2.0.10
                                 2.0.11
                                 2.0.12
                                 2.0.14

[Unbound memory (Base  /Range)]:  0x0/64
                (bytes) (MB)      0x8000000/6080
[Available memory (MB)]:  6144
#
```

Step 3. Issue the **`vparcreate -p...`** command to create the new virtual partition and as needed use **`vparmodify -p...`** to further configure the new virtual partition.

When using the `vparcreate` command you *must specify* the name for the new virtual partition (`-p` *vpname*).

You also should specify the resources that are to be assigned for exclusive use by the virtual partition, including processor (`-a cpu...`) memory (`-a mem...`) and input/output (`-a io...`) resources.

You also can include other virtual partition attributes and settings (such as the autoboot setting) as part of the `vparcreate` command that establishes the new virtual partition.

```
# vparcreate -p Mesh -a mem::2048 -a io:2/0/14 -a io:2/0/0 -B manual -S dynamic
-a io:2/0/14/0/0.6:BOOT -a cpu:2/10
#
```

To further modify the virtual partition, issue the `vparmodify` command after `vparcreate` has created the new virtual partition.

For other details on creating and configuring virtual partitions see the section *Configuring Virtual Partition Resources and Attributes* on page 445.

Step 4. Issue the **`vparstatus`** command to list the configuration and boot status for the newly created virtual partition.

For detailed virtual partition information, use the **`vparstatus -v -p...`** command.

If you need to change any of the new virtual partition's configuration details, use the **`vparmodify`** command.

```
# vparstatus
[Virtual Partition]
                                                                          Boot
Virtual Partition Name         State Attributes Kernel Path               Opts
============================== ===== ========== ========================= =====
Shad                           Up    Dyn,Manl   /stand/vmunix
Mesh                           Down  Dyn,Manl   /stand/vmunix

[Virtual Partition Resource Summary]
                                            CPU    Num        Memory (MB)
                                  CPU     Bound/   IO   # Ranges/
Virtual Partition Name           Min/Max  Unbound  devs  Total MB    Total MB
==============================   ================  ====  ====================
Shad                               2/  8    2   2     7    0/  0         2048
Mesh                               1/ 16    1   0     3    0/  0         2048
# vparstatus -v -p Mesh
[Virtual Partition Details]
Name:          Mesh
State:         Down
Attributes:    Dynamic,Manual
Kernel Path:   /stand/vmunix
Boot Opts:

[CPU Details]
Min/Max:  1/16
```

```
Bound by User [Path]:  2.10
Bound by Monitor [Path]:
Unbound [Path]:

[IO Details]
   2.0.14
   2.0.0
   2.0.14.0.0.6  BOOT

[Memory Details]
Specified [Base  /Range]:
          (bytes) (MB)
Total Memory (MB):  2048
#
```

Deleting a Virtual Partition

This section describes the procedure for deleting a virtual partition and related issues you may encounter.

NOTE The virtual partition you delete *must* be in a "Down" state.

You can delete virtual partitions that are defined in the currently active vPars database (`vpdb`) used by the *local* nPartition.

You also can delete virtual partitions from an alternate vPars database that you specify using the `vparremove` command's `-D` *`database`* option.

You *cannot* modify or delete vPars configuration info from inaccessible vPars databases, such as those on currently unused boot disks. (Currently unused boot disks include: disks assigned to a virtual partition that is not in an "Up" state; or disks not currently booted by a virtual partition such as the ALT boot device for a virtual partition that has booted its PRI device.)

---

vPars Management: Deleting a Virtual Partition from an nPartition

This procedure deletes a virtual partition's configuration info from the currently active vPars database.

Also see the book *Installing and Managing HP-UX Virtual Partitions (vPars)* for virtual partition management info.

Step 1. Login to HP-UX running on one of the virtual partitions in an nPartition.

Step 2. Issue the **`vparstatus`** command to list the current boot status and high-level configuration information for all virtual partitions defined in the currently active vPars database (`vpdb`) used by the local nPartition

```
# vparstatus
[Virtual Partition]
                                                                            Boot
Virtual Partition Name         State Attributes Kernel Path                 Opts
============================== ===== ========== ========================== =====
Shad                           Up    Dyn,Auto   /stand/vmunix
Mesh                           Down  Dyn,Manl   /stand/vmunix               boot

[Virtual Partition Resource Summary]
                                           CPU    Num        Memory (MB)
```

```
                                  CPU     Bound/   IO   # Ranges/
Virtual Partition Name          Min/Max  Unbound  devs  Total MB    Total MB
==============================  ================  ====  ====================
Shad                              2/  8    2   0     7    0/  0         2048
Mesh                              2/ 12    2   0     3    0/  0            0
#
```

Step 3. Issue the **vparremove -p *vpname*** command to delete the specified virtual partition (*vpname*) and then issue the **vparstatus** command to list the new configuration status.

You can delete only virtual partitions that are in a "Down" state, as reported by the `vparstatus` command (see the example in the previous step).

See the *vparremove* (1M) manpage for details.

```
# vparremove -p Mesh
Remove virtual partition Mesh? [n]  y
#
# vparstatus
[Virtual Partition]
                                                                         Boot
Virtual Partition Name         State Attributes Kernel Path              Opts
============================== ===== ========== ========================= =====
Shad                           Up    Dyn,Auto   /stand/vmunix

[Virtual Partition Resource Summary]
                                           CPU    Num       Memory (MB)
                                  CPU     Bound/   IO   # Ranges/
Virtual Partition Name          Min/Max  Unbound  devs  Total MB    Total MB
==============================  ================  ====  ====================
Shad                              2/  8    2   0     7    0/  0         2048
#
```

Modifying Virtual Partition Configuration Attributes

This section describes how to add or remove resources from a virtual partition, and how to change a virtual partition's attribute settings.

Many virtual partition hardware resource changes require that the virtual partition being modified *is not running* (not in an "Up" state).

vPars Management: Modifying Virtual Partition Attributes in nPartition Environments

This procedure describes how to modify a virtual partition's attributes and resource configuration.

Step 1. Login to HP-UX running on one of the virtual partitions in an nPartition, or login to HP-UX running in non-vPars mode on an nPartition.

You can modify virtual partition attributes from HP-UX running in a virtual partition.

You also can modify vPars database configurations when you have booted HP-UX in non-vPars mode. By default the `vparcreate`, `vparmodify`, and `vparstatus` commands use the `/stand/vpdb` file.

To modify a vPars database other than the `/stand/vpdb` file, use the `-D` option to specify its location.

Step 2. Issue the **`vparstatus`** command to list the current status of the virtual partition you plan to update.

For detailed info on a virtual partition use the `vparstatus -v -p...` command.

```
# vparstatus
[Virtual Partition]
                                                                          Boot
Virtual Partition Name          State Attributes Kernel Path              Opts
============================== ===== ========== ========================= =====
Shad                            Up    Dyn,Manl   /stand/vmunix
Mesh                            Up    Dyn,Manl   /stand/vmunix            boot

[Virtual Partition Resource Summary]
                                               CPU    Num       Memory (MB)
                                      CPU     Bound/   IO   # Ranges/
Virtual Partition Name          Min/Max  Unbound  devs  Total MB    Total MB
==============================  ================  ====  ====================
```

```
Shad                                  2/  8    2   6     7    0/  0          2048
Mesh                                  2/ 12    2   6     3    0/  0          2048
#
```

Step 3. Issue the **vparmodify -p...** command to modify the specified virtual partition.

See the section *Configuring Virtual Partition Resources and Attributes* on page 445 for descriptions of the virtual partitions configuration options.

Also see the *vparmodify* (1M) manpage for details.

For example, the following commands set the virtual partition named "Mesh" to be configured for autoboot; configure "Shad" to have 4 processors; and configure "Mesh" to have 12 processors:

```
# vparmodify -p Mesh -B auto
# vparmodify -p Shad -m cpu::4
# vparmodify -p Mesh -m cpu::12
```

Step 4. Issue the `vparstatus` command to list the new status for the virtual partition you modified in the previous step.

You can make further changes to the virtual partition by issuing additional `vparmodify` commands.

```
# vparstatus
[Virtual Partition]
                                                                          Boot
Virtual Partition Name         State Attributes Kernel Path               Opts
============================== ===== ========== ========================= =====
Shad                           Up    Dyn,Manl   /stand/vmunix
Mesh                           Up    Dyn,Auto   /stand/vmunix             boot

[Virtual Partition Resource Summary]
                                         CPU    Num        Memory (MB)
                                  CPU    Bound/  IO   # Ranges/
Virtual Partition Name          Min/Max  Unbound devs Total MB    Total MB
==============================  ================ ==== ====================
Shad                              2/  8    2   2     7    0/  0          2048
Mesh                              2/ 12    2  10     3    0/  0          2048
#
```

Listing the Status for Virtual Partitions

This section covers methods for listing the status of virtual partitions in an nPartition.

You can list virtual partition status details from HP-UX running on a virtual partition. You also can list vPars database details for nPartitions that are not booted in vPars mode.

In addition to the procedure listed below, you can list limited virtual partition status info from the vPars monitor (the `MON>` prompt) by using the `vparinfo` monitor command.

vPars Management: Listing Virtual Partition Status from an nPartition

This procedure lists the boot status and configuration details for virtual partitions that are defined in an nPartition.

Step 1. Login to HP-UX running on any of the virtual partitions in the nPartition.

Step 2. Issue the **`vparstatus`** command for details about virtual partitions.

You can list the status for all virtual partitions, a specific virtual partition, or the vPars monitor's event log.

- To list a summary that includes the status for all currently defined virtual partitions, issue the **`vparstatus`** command with no arguments or options.
- To list detailed information about a specific virtual partition, issue the **`vparstatus -v -p`** ***vpname*** command.
- To display history from the vPars monitor's event log issue the **`vparstatus -e`** command.

  The vPars monitor event log details are available only when the nPartition is booted in vPars mode.

```
# vparstatus
[Virtual Partition]
                                                                           Boot
Virtual Partition Name         State Attributes Kernel Path                Opts
============================== ===== ========== ========================== =====
Shad                           Up    Dyn,Auto   /stand/vmunix
Mesh                           Down  Dyn,Manl   /stand/vmunix              boot
```

```
[Virtual Partition Resource Summary]
                                              CPU    Num        Memory (MB)
                                    CPU     Bound/   IO    # Ranges/
Virtual Partition Name            Min/Max  Unbound  devs  Total MB     Total MB
==============================  ================  ====  ====================
Shad                                2/  8    2   0     8    0/  0         2048
Mesh                                2/ 12    2   2     3    0/  0         2048
#
# vparstatus -e
INFO:CPU0:MON:[17:56:51 5/20/2002 GMT] VPAR Monitor version 0.2 started
INFO:CPU0:MON:Version string: @(#) $Revision: vpmon:    vw: --    selectors: CUP
11.11_BL2001_1101 'cup_vpar_pib3' 'cup_shep_sd_vpars'  Sun May  5 20:22:18 PDT
2002 $
INFO:CPU0:MON:cell num 6 does not contain i/o chassis
INFO:CPU0:MON:cell num 1 does not contain i/o chassis
INFO:CPU0:MON:Partition Shad monarch set to 0/10
INFO:CPU0:MON:Partition Mesh monarch set to 2/10
INFO:CPU0:MON:[17:57:45 5/20/2002 GMT] Shad is active
INFO:CPU0:MON:PDC_STABLE return size = 3f0
INFO:CPU0:MON:[17:58:15 5/20/2002 GMT] Shad is up

#
```

Booting HP-UX on Virtual Partitions

This section provides a procedure for loading and booting HP-UX on a virtual partition that is running in an nPartition.

As part of the virtual partition boot process, you will boot the `/stand/vpmon` vPars monitor from the nPartition BCH interface instead of booting the `/stand/vmunix` HP-UX kernel.

From the vPars monitor (the `MON>` prompt) running on an nPartition, you can load one or more virtual partitions. Each virtual partition then can boot a single instance of the HP-UX kernel.

Before performing this procedure, review the following list for an overview of situations you may encounter when loading and booting virtual partitions on an nPartition server.

- If one or more virtual partitions already has loaded/booted HP-UX on the nPartition, you can load/boot additional virtual partitions from HP-UX running on one of the existing virtual partitions.

  In this situation, you can issue the `vparboot` command to load other virtual partitions; see the *vparboot* (1M) manpage for details.

  However, you can only load virtual partitions that are defined in the currently active vPars configuration database, which typically is the `/stand/vpdb` file on the same boot device where `/stand/vpmon` was booted.

- If HP-UX is booted in non-vPars mode on an nPartition, you must shut down HP-UX on the nPartition and from the nPartition's BCH interface boot the `/stand/vpmon` vPars monitor before loading any virtual partitions.

These above situations also are addressed in the following procedure.

vPars Management: Booting HP-UX B.11.11 on Virtual Partitions in an nPartition

This procedure describes how to boot HP-UX on one or more virtual partitions in a single nPartition.

Also refer to the chapters *An Overview of nPartition Boot and Reset* on page 161 and *Booting and Resetting nPartitions* on page 197 for details on booting nPartitions.

Step 1. Login to the service processor (GSP or MP) for the server where the virtual partitions will be booted.

Step 2. Access the console for the nPartition in which the virtual partitions will boot HP-UX.

From the service processor main menu enter **CO** to access the nPartition console menu, and select the nPartition in which you will boot the virtual partitions.

```
GSP> CO

    Partitions available:

     #   Name
    ---  ----
     0)  feshd4a
     1)  feshd4b
     Q)  Quit

    Please select partition number: 0
```

Step 3. Access HP-UX or the BCH interface for the nPartition, and if needed reboot the instance of HP-UX running on the nPartition.

When accessing the nPartition console, if you can interact with a BCH command prompt such as the following:

```
Main Menu: Enter command or menu >
```

then you can proceed to the next step and *skip the rest of this step*.

If you cannot interact with a BCH menu or HP-UX login prompt or command line, then the nPartition might be booting or might be hanged.

You can use the server's Virtual Front Panel (VFP) to check the nPartition's current boot state. Refer to the chapter *Using Console and Service Processor Interfaces* on page 125 for details.

Otherwise, if HP-UX is running in the nPartition, first check to see whether HP-UX booted in vPars-mode or non-vPars mode.

Enter the **`vparstatus -w`** command, and determine the current nPartition boot state:

- If `vparstatus` reports `Error: Virtual partition monitor not running` then the nPartition is *not running vPars*.

  Enter `shutdown -r` to reboot the nPartition and as needed interrupt the autoboot process to access the nPartition's BCH interface.

After entering the `shutdown` command and accessing the BCH interface you can *proceed with the next step*.

- If `vparstatus` reports `The current virtual partition is...` then the nPartition is *running one or more virtual partitions*.

NOTE This note applies only when an nPartition is running one or more virtual partitions.

Because at least one virtual partition already has loaded/booted HP-UX on the nPartition, you should check whether the virtual partition you wish to boot already is loaded, or whether the virtual partition can be loaded without rebooting HP-UX.

From HP-UX running on the virtual partition, enter the `vparstatus` command.

If the virtual partition you wish to load/boot *is not listed* in the `vparstatus` output, you may need to reboot the nPartition and its virtual partitions and you can *proceed with the rest of this step and procedure*.

If the virtual partition you wish to load/boot is listed, then check its boot state. If the virtual partition is "Up" then it *already has loaded/booted* HP-UX. If the virtual partition is "Down" then you can load/boot it using the `vparboot` command (see the *vparboot* (1M) manpage). In either of these two cases you can *skip the rest of this procedure*.

If you are certain that you need to reboot the nPartition and its virtual partitions, proceed with the rest of this step.

To access BCH, shut down HP-UX on all virtual partitions and reset the virtual partition.

See the procedure *Rebooting or Shutting Down Virtual Partitions* on page 494 for complete details on shutting down HP-UX on all virtual partitions and returning to the BCH interface. You must shut down and halt (`shutdown -h`) HP-UX in each virtual partition, and then at the vPars monitor's `MON>` prompt enter the `reboot` command to reset the nPartition. When the nPartition resets all active cells in the nPartition are reset; after the cells reset you should interrupt the nPartition's autoboot process if needed and then access the BCH interface.

Once you have access to the BCH interface proceed with the next step.

Step 4. From the BCH interface enter the **`BOOT`** ***`device`*** command, where *`device`* is the disk where the desired vPars configuration database (the `/stand/vpdb` file) resides.

When using the `BOOT` command you can specify a boot path variable (for example, `BOOT PRI`) or a hardware path for the boot device (for example, `BOOT 0/0/1/0/1.3`).

In addition to having the vPars database (`vpdb`), the device must have both HP-UX B.11.11 and the vPars software product installed.

Step 5. Instruct BCH to stop the boot process at the ISL prompt by entering **`y`** at the "Do you wish to stop" prompt.

```
Do you wish to stop at the ISL prompt prior to booting? (y/n)
>> y
```

If the boot device's AUTO is set to boot `/stand/vpmon` then you can instead enter `n` (for "do not stop at ISL") and have the nPartition proceed to boot the vPars monitor automatically.

However, you must direct the ISL/SSL interfaces to load `/stand/vpmon` if the boot device's AUTO file *does not contain* the string `hpux boot /stand/vpmon`.

You can check the AUTO file contents from the ISL prompt by entering the `hpux show autofile` command. By default the AUTO file is set to `hpux`, which loads the `/stand/vmunix` kernel.

See *Configuring Virtual Partition Autoboot* on page 498 for details on configuring a boot device's AUTO file.

Step 6. As necessary, from the ISL interface enter the **`hpux boot /stand/vpmon`** command to boot the vPars monitor (`vpmon`) on the local nPartition.

```
ISL> hpux boot /stand/vpmon

Boot
: disk(0/0/6/0/0.5.0.0.0.0.0;0)/stand/vpmon
565248 + 156368 + 16872200 start 0x23000
```

```
Welcome to VPMON (type '?' for a list of commands)

MON>
```

If you stopped at the ISL interface, you must perform this step.

You do not need to perform this step if the boot device AUTO file is set to `hpux boot /stand/vpmon` and you did not stop at the ISL prompt in the previous step.

Step 7. At the vPars monitor's `MON>` prompt, enter the **`vparinfo`** command to list details about the virtual partitions currently defined in the vPars database (`vpdb`).

Especially note the list of virtual partitions that the `vparinfo` command displays as the "Names of the partitions in the database".

```
MON> vparinfo

Resources not assigned to any partition
-------------------------------------
0           0xfffffffffc000000      1       0  TYPE=14  SV_MODEL=170
0/0         0xfffffff808000000      1       0  TYPE= 7  SV_MODEL= 12
0/0/0       0xfffffff804000000      1       0  TYPE=13  SV_MODEL= 10
0/0/1       0xfffffff804002000      1       0  TYPE=13  SV_MODEL= 10

....

Names of the partitions in the database:
----------------------------------------

Shad
Mesh

Available Free Memory: 0 MB

Available MEM RANGE: 0x0000000000000000-0x00000000ffffffff (4194304 Kb)
MON>
```

To see detailed information about a particular virtual partition you can use the `vparinfo` *vpname* command, where *vpname* is the virtual partition's name. This detailed information includes the resources assigned to the virtual partition and other details such as the virtual partition boot path(s) and the virtual partition's autoboot setting.

Step 8. From the vPars monitor's `MON>` prompt, use the **`vparload`** command load/boot HP-UX on a virtual partition.

You can specify any of the following `vparload` commands at the vPars monitor `MON>` prompt:

- To load/boot HP-UX on *all virtual partitions* that are defined in the current vPars configuration database, enter: **`vparload -all`**
- To load/boot HP-UX on *a single virtual partition*, enter: **`vparload -p`** ***`vpname`***

  where *`vpname`* is the name of the virtual partition, as reported by the `vparinfo` command in the previous step.
- To load/boot HP-UX only on *the autoboot-enabled virtual partitions*, enter: **`vparload -auto`**

  This command loads only the virtual partitions that have the autoboot attribute configured (the boot attribute is set to `auto`).

The following example shows the virtual partition named "Shad" being loaded from the vPars monitor prompt and booting HP-UX.

```
MON> vparload -p Shad
[MON] Booting Shad...
[MON] Console client set to Shad
[MON] Console server set to Shad

[Shad]

[MON] Shad loaded
gate64: sysvec_vaddr = 0xc0002000 for 2 pages
NOTICE: nfs3_link(): File system was registered at index 3.
NOTICE: autofs_link(): File system was registered at index 6.
NOTICE: cachefs_link(): File system was registered at index 7.

    Host is virtual System Console slave
Logical volume 64, 0x3 configured as ROOT
Logical volume 64, 0x2 configured as SWAP

....

     HP-UX Start-up in progress
     __________________________

     Configure system crash dumps ........................................ OK
     Mount file systems .................................................. OK
     Virtual Partitions Initialization ................................... OK
     Update kernel and loadable modules .................................. N/A
     Initialize loadable modules ......................................... N/A
```

```
Setting hostname .................................................... OK
```

....

Step 9. Login to HP-UX running in the virtual partition you loaded/booted in the previous steps of this procedure.

Because the *nPartition console is shared by all virtual partitions* within the nPartition, you should login using a method other than the system console login for most virtual partition access purposes.

Use the `telnet` command or another remote login method to access HP-UX running on the virtual partition.

After you login to a virtual partition running on an nPartition, you can list the current virtual partition using `vparstatus -w`, and can list the local nPartition's partition number using the `parstatus -w` command.

```
# vparstatus -w
The current virtual partition is Shad.
# parstatus -w
The local partition number is 0.
#
```

Rebooting or Shutting Down Virtual Partitions

This section describes how to reboot HP-UX in a virtual partition and how to shut down all virtual partitions running in an nPartition.

After shutting down all virtual partitions in an nPartition you can reboot the vPars monitor to reset the nPartition and, after resetting, make the nPartition's BCH interface available.

NOTE The nPartition *reboot for reconfig* and *ready for reconfig state* involve methods of resetting an nPartition that require additional considerations not covered in this section.

See the section *Managing nPartitions from a Virtual Partition* on page 506 for details and procedures to perform a *reboot for reconfig* or to shut down an nPartition to a *ready for reconfig state* when one or more virtual partitions is running in the nPartition.

When you reboot (`shutdown -r`) HP-UX running in a virtual partition, HP-UX will automatically attempt to reboot on the virtual partition *if* the virtual partition is configured for autoboot *and* none of the other virtual partitions in the same nPartition have initiated a reboot for reconfig or a shutdown to the ready for reconfig state.

You can interrupt the virtual partition autoboot process when accessing the virtual partition's console interface through its nPartition's console.

When you halt (`shutdown -h`) HP-UX running in a virtual partition the virtual partition shuts down to a "Down" state and HP-UX does not reboot. After you have halted a virtual partition you can load/boot HP-UX on the virtual partition using the `vparboot` command from HP-UX running on any of the other virtual partitions in the same nPartition. Also see the procedure *vPars Management: Booting HP-UX B.11.11 on Virtual Partitions in an nPartition* on page 487 for other methods and details.

To shut down *all virtual partitions* in an nPartition, login to each virtual partition through its console and halt HP-UX (`shutdown -h`). Then from the vPars monitor's `MON>` prompt you can enter `reboot` to exit the vPars monitor and reset the nPartition's active cells.

vPars Management: Rebooting or Shutting Down Virtual Partitions in an nPartition

The following procedure is for performing a normal reboot (`shutdown -r`) or a shutdown-and-halt (`shutdown -h`) in a virtual partition that is running in an nPartition.

This procedure also describes how to halt all virtual partitions in an nPartition and return to the nPartition's BCH interface.

Step 1. Login to HP-UX running on the virtual partition that you want to shutdown or reboot.

You can login to the virtual partition remotely by using the `telnet` command or another remote login command, or can login through the virtual partition's console.

If you plan to *shut down all virtual partitions* in the nPartition, you should gain console login access to the virtual partitions.

You also will need virtual partition console access for this procedure if you must interrupt the virtual partition's HP-UX autoboot process.

To access the virtual partition's console, first login to the service processor (GSP or MP) for server where the virtual partition's nPartition resides and then access the nPartition's console. As needed, in the nPartition console type **Control-a** to switch among the virtual partition consoles.

Step 2. Enter the **`vparstatus -w`** command to confirm that you are logged into the virtual partition that you want to shut down.

You also can check the virtual partition's autoboot setting by using the `setboot` or the `vparstatus` command.

```
# vparstatus -w
The current virtual partition is Shad.
# setboot
Primary bootpath : 0/0/6/0/0.5.0
Alternate bootpath : 0/0/6/0/0.6.0

Autoboot is ON (enabled)
Autosearch is ON (enabled)

Note: The interpretation of Autoboot and Autosearch has changed
for
```

```
systems that support hardware partitions. Please refer to the
manpage.
#
```

Step 3. From HP-UX running in the virtual partition you want to shut down, enter the **shutdown** command with the appropriate command-line options.
If shutting down *all virtual partitions* in the nPartition, use **shutdown -h**.

To shut down and halt HP-UX on the virtual partition, enter the **shutdown -h** command along with any additional command-line options you need.

To shut down HP-UX on the virtual partition and allow the virtual partition to autoboot HP-UX if it is configured to do so, enter the **shutdown -r** command along with any additional options you need.

See the *shutdown* (1M) manpage for complete details about all options.

Step 4. If you are shutting down *all virtual partitions* in the nPartition, type **Control-a** to switch to the next virtual partition's console login prompt, login to the virtual partition, and then repeat *Step 2* and *Step 3* to shut down HP-UX in the virtual partition.

Typing **Control-a** switches among the virtual partition consoles that are available through an nPartition's console, when one or more virtual partitions are loaded/booted in the nPartition.

NOTE When you type **Control-a** repeatedly in the nPartition console and remain at the vPars monitor (MON> or [MON]), then no virtual partitions are loaded or booted in the nPartition.

Step 5. If you are shutting down *all virtual partitions* in the nPartition, then after you have halted HP-UX running in each virtual partition in the nPartition you can reset the nPartition by entering the **reboot** command from the MON> prompt.

The vPars monitor reboot command resets all active cells in the nPartition.

After the cells have reset and completed self-tests, the cells participate in partition rendezvous, form an nPartition, and finally the nPartition's BCH interface is made available through the nPartition console.

If the nPartition is configured to autoboot, you can interrupt the autoboot process by typing any key at the appropriate time when accessing the nPartition's console.

If autoboot is configured for an nPartition, you will see a message similar to the following in the nPartition's console during the nPartition reset process.

```
Attempting to boot using the primary path.
-------------------------------------------------------------

 To discontinue, press any key within 10 seconds.
```

Configuring Virtual Partition Autoboot

This section describes how you can configure an nPartition to automatically boot the virtual partitions Monitor and to also boot all virtual partitions that have autoboot configured.

For details on automatically booting HP-UX in *non-vPars mode* on an nPartition, refer to the chapter *Booting and Resetting nPartitions* on page 197.

As the following procedure describes, setting up the virtual partitions autoboot process involves first configuring the nPartition's boot device paths and path flags to boot the device where the current vPars database resides, then configuring that boot device's AUTO file to specify that the vPars monitor be loaded with the `-a` option, and finally configuring the virtual partitions that you want load automatically to have their boot attribute set to auto.

vPars Management: Configuring Virtual Partition Boot Settings

This procedure configures an nPartition to autoboot the vPars monitor (`MON>` prompt) and also automatically load/boot the virtual partitions that have autoboot configured.

Step 1. From the nPartition's BCH interface, configure the nPartition to automatically boot the device where the current vPars database and vPars monitor resides.

First configure one of the nPartition's boot path variables (PRI, HAA, or ALT) to reference the device where the current vPars database (`/stand/vpdb`) resides.

Then configure the nPartition's path flags to boot the chosen device path.

Refer to the chapter *Booting and Resetting nPartitions* on page 197 for details on nPartition autoboot, including device path and path flag configuration.

Step 2. Boot HP-UX in *non-vPars mode* from the device you configured in the previous step.

From the BCH interface, issue the BOOT command and specify the boot path variable you set in the previous step. For example: `BOOT HAA` to boot the HAA device path.

Step 3. Login to HP-UX and configure the chosen boot device's AUTO file.

After HP-UX has booted in non-vPars mode on the nPartition, login as `root`, use the `lvdisplay` command to list device file for the boot device, and then use the `mkboot` command to configure the boot device's AUTO file. You also can use the `lifcp` command to display the contents of the AUTO file.

For example, the following `mkboot` command sets the AUTO file for the `/dev/dsk/c1t5d0` device, and the `lifcp` command displays the contents of the device's AUTO file.

```
# mkboot -a "hpux boot /stand/vpmon -a" /dev/dsk/c1t5d0
# lifcp /dev/dsk/c1t5d0:AUTO -
hpux boot /stand/vpmon -a
#
```

The `vpmon -a` option specifies to automatically load/boot all virtual partitions that have autoboot configured when the vPars monitor is loaded.

Also see the example *Autoboot Configuration Example for Virtual Partitions (vPars)* on page 500 for more example.

Step 4. From HP-UX check all virtual partition boot attributes and reconfigure any boot attributes to establish the virtual partitions autoboot configuration you desire.

Use the `vparstatus` command to list details about all virtual partitions, including boot attributes. Note that when you issue this command when HP-UX is booted in non-vPars mode, the command presents configuration info based on the `/stand/vpdb` vPars database.

Then as needed use the `vparmodify` command to reconfigure any boot attributes. For example, the following command sets the virtual partition named "Mesh" to automatically load/boot HP-UX when possible.

```
# vparmodify -p Mesh -B auto
```

Each virtual partition that you want to boot automatically must have its boot attribute set to `auto`.

Step 5. Reboot the nPartition, and as desired observe its boot progress from the nPartition's Virtual Front Panel or its console.

The result of this nPartition reboot is to automatically load/boot the virtual partitions that you have configured for autoboot.

Issue the **shutdown -r** command to shut down HP-UX and reboot the nPartition.

When the nPartition reboots to its BCH interface, it will proceed to boot the device path you specified using nPartition boot paths and path flags. The nPartition then will execute the device's AUTO file contents that you specified, to load the /stand/vpmon vPars monitor. Finally, because the vPars monitor is invoked with the -a option, it will automatically load/boot all virtual partitions that have autoboot configured.

Example 11-1 Autoboot Configuration Example for Virtual Partitions (vPars)

In this example, the user first confirms that the vPars database (/stand/vpdb) and vPars monitor (/stand/vpmon) are in the /stand directory and thus can be referenced and booted.

The bdf command displays the logical volume associated with the /stand directory, and the the lvdisplay command then displays the device file associated with the logical volume.

```
# ls /stand/vp*
/stand/vpdb   /stand/vpmon
# bdf /stand
Filesystem          kbytes    used   avail %used Mounted on
/dev/vg00/lvol1     512499   71581  389668   16% /stand
# lvdisplay -vk /dev/vg00/lvol1 | grep dev
LV Name                      /dev/vg00/lvol1
VG Name                      /dev/vg00
   /dev/dsk/c1t5d0    128       128
#
```

The first lifcp command that follows displays the original contents of the boot device's AUTO file. Originally, this device is configured with the AUTO file default, hpux, which invokes the hpux loader with no options and thus the /stand/vmunix kernel would be booted.

The mkboot command that follows sets the AUTO file contents so that the hpux loader will boot the /stand/vpmon vPars monitor with the -a option. Issuing the lifcp command again shows the new contents of the device's AUTO file.

```
# lifcp /dev/dsk/c1t5d0:AUTO -
hpux
# mkboot -a "hpux boot /stand/vpmon -a" /dev/dsk/c1t5d0
# lifcp /dev/dsk/c1t5d0:AUTO -
hpux boot /stand/vpmon -a
#
```

Next the user issues the `vparstatus` command to list the current settings for the virtual partitions defined in the `/stand/vpdb` file. Because the `vparstatus` command is issued when the local nPartition is booted in non-vPars mode, the command lists info based on the `vpdb` file rather than the vPars monitor (which is not running).

```
# vparstatus
vparstatus: Warning: Virtual partition monitor not running, Requested resources
shown.
[Virtual Partition]
                                                                        Boot
Virtual Partition Name          State Attributes Kernel Path            Opts
=============================== ===== ========== ========================= =====
Shad                            N/A   Dyn,Manl   /stand/vmunix
Mesh                            N/A   Dyn,Manl   /stand/vmunix

[Virtual Partition Resource Summary]
                                              CPU    Num        Memory (MB)
                                  CPU       Bound/   IO    # Ranges/
Virtual Partition Name          Min/Max   Unbound  devs  Total MB    Total MB
==============================  ========  =======  ====  ====================
Shad                              2/  3     2   0     6    0/  0         2048
Mesh                              1/  2     1   1     6    0/  0         1024
#
```

In this example both virtual partitions, named “Shad” and “Mesh”, originally are configured to be booted manually, as shown in the previous `vparstatus` command output: the boot attribute for each is listed as “Manl” (`manual`).

Next the `vparmodify` command reconfigures the boot attribute for the virtual partition named “Mesh” to `auto`.

After changing the boot attribute, issuing the `vparstatus` command shows updated info about “Mesh” and lists its boot attribute as “Auto” (`auto`).

```
# vparmodify -p Mesh -B auto
# vparstatus -p Mesh
vparstatus: Warning: Virtual partition monitor not running, Requested resources
shown.
[Virtual Partition]
                                                                        Boot
Virtual Partition Name          State Attributes Kernel Path            Opts
=============================== ===== ========== ========================= =====
Mesh                            N/A   Dyn,Auto   /stand/vmunix

[Virtual Partition Resource Summary]
                                              CPU    Num        Memory (MB)
                                  CPU       Bound/   IO    # Ranges/
```

```
Virtual Partition Name          Min/Max  Unbound  devs  Total MB    Total MB
==============================  ================  ====  ====================
Mesh                              1/  2    1   1     6    0/  0         1024
#
```

This example nPartition now is configured so that when the nPartition reboots it will automatically boot from a device that will automatically load a vPars monitor, which then will automatically load/boot the virtual partition named “Mesh”.

In this example’s next step the user reboots the nPartition by issuing the `shutdown -r` command.

```
# shutdown -r

SHUTDOWN PROGRAM
06/26/02 17:57:23 CDT
Waiting a grace period of 60 seconds for users to logout.
Do not turn off the power or press reset during this time.


Broadcast Message from root (console) Wed Jun 26 17:58:23...
SYSTEM BEING BROUGHT DOWN NOW ! ! !

Do you want to continue? (You must respond with 'y' or 'n'.):   y

/sbin/auto_parms: DHCP access is disabled (see /etc/auto_parms.log)


     System shutdown in progress
     ___________________________

     Stopping OpC agent processes (opcagt). ................... OK
     Stop CDE login server .................................... OK
```

Because the nPartition is booted in non-vPars mode, the `shutdown -r` command shuts down HP-UX and resets the nPartition’s active cells.

After the cells boot and the nPartition reaches its BCH interface, the autoboot process begins.

The following example output shows that the nPartition automatically boots the primary (PRI) boot device path, whose AUTO file is configured to load the vPars monitor and automatically load/boot the virtual partitions whose boot attribute is `auto`.

The end result of this example nPartition shutdown-and-reboot is that the nPartition has loaded/booted the virtual partition named “Mesh”.

```
Firmware Version  35.3

Duplex Console IO Dependent Code (IODC) revision 1
   --------------------------------------------------------------------------
     (c) Copyright 1995-2002, Hewlett-Packard Company, All rights reserved
   --------------------------------------------------------------------------

....

     Primary Boot Path:  0/0/1/0/1.5
          Boot Actions:  Boot from this path.
                         If unsuccessful, go to BCH.

HA Alternate Boot Path:  0/0/1/0/1.6
          Boot Actions:  Go to BCH.


   Alternate Boot Path:  0/0/1/0/1.4
          Boot Actions:  Go to BCH.

          Console Path:  0/0/0/0/0.0


Attempting to boot using the primary path.
-------------------------------------------------------------


 To discontinue, press any key within 10 seconds.

 10 seconds expired.
 Proceeding...

Initializing boot Device.

Boot IO Dependent Code (IODC) Revision 0

Boot Path Initialized.

HARD Booted.

ISL Revision A.00.43  Apr 12, 2000

ISL booting  hpux boot /stand/vpmon -a

Boot
: disk(0/0/1/0/1.5.0.0.0.0.0;0)/stand/vpmon
585728 + 164600 + 16896360 start 0x23000
[MON] Booting Mesh...
MON> [MON] Console client set to Mesh

[MON] Mesh loaded
```

```
....

     HP-UX Start-up in progress
     __________________________

     Configure system crash dumps ....................................... OK
     VxVM device node check ............................................. OK

....

     Start CDE login server ............................................. OK

The system is ready.

2/0/1/0/0.5  feshd4b (Mesh)  [HP Release B.11.11]
Console Login:
```

In this example, the nPartition has completed the reboot and autoboot process and has automatically loaded/booted the virtual partition named "Mesh", which has its boot attribute set to `auto`.

As the following output shows, when the user logs in to HP-UX running on the virtual partition, the `vparstatus` and `parstatus` commands report that the *current virtual partition* is "Mesh", the *local nPartition* is partition number 0, and the virtual partition named "Shad" is in a "Down" state. Shad was not automatically loaded/booted because its boot attribute is set to `manual` (listed as "Manl" in the output below).

```
Console Login: root
Password:

# vparstatus -w
The current virtual partition is Mesh.
# parstatus -w
The local partition number is 0.
# vparstatus
[Virtual Partition]
                                                                         Boot
Virtual Partition Name         State Attributes Kernel Path               Opts
============================== ===== ========== ========================= =====
Shad                           Down  Dyn,Manl   /stand/vmunix
Mesh                           Up    Dyn,Auto   /stand/vmunix

[Virtual Partition Resource Summary]
                                              CPU    Num        Memory (MB)
                                     CPU     Bound/   IO   # Ranges/
Virtual Partition Name         Min/Max  Unbound  devs  Total MB    Total MB
==============================  ================  ====  ====================
```

```
Shad                                     2/  3    2   0      6    0/  0          2048
Mesh                                     1/  2    1   1      6    0/  0          1024
#
```

Managing nPartitions from a Virtual Partition

This section covers the nPartition management issues that are unique to virtual partition environments.

NOTE

In virtual partition environments, you can apply the same nPartition configuration tools and principles that you use in non-vPars environments—but you *must take additional steps* to perform a reboot for reconfig of an nPartition that has more than one virtual partition loaded/booted.

The procedures in this section address the minor differences in performing nPartition reconfiguration and reboot for reconfig processes when using virtual partitions in an nPartition.

The following procedures are covered in this section:

- *vPars Management: Determining if an nPartition is Running vPars* on page 507
- *vPars Management: Performing a Reboot for Reconfig or Shutdown to Ready for Reconfig from a Virtual Partition* on page 510
- *vPars Management of nPartitions: Adding and Removing nPartition Cells from a Virtual Partition* on page 513
- *vPars Management of nPartitions: Reconfiguring nPartition Attributes from a Virtual Partition* on page 518

Also refer to the chapter *Managing nPartitions* on page 243 for complete nPartition management procedures.

Virtual partition configuration procedures are covered in the section *Procedures for Managing Virtual Partitions on HP nPartition Servers* on page 475.

Determining if an nPartition is Running vPars

This section gives you several methods for determining if an nPartition has loaded/booted HP-UX in or more virtual partitions. For example, typing **Control-a** at the nPartition console, or using the **`vparstatus`** command.

vPars Management: Determining if an nPartition is Running vPars

Step 1. Login to the service processor (GSP or MP) for the server where the nPartition resides, and access the nPartition's console. From the main menu, enter **`CO`** for the console menu and select the nPartition's console.

Step 2. At the nPartition's console, determine the current boot state.

The nPartition's current console prompt, if any, provides the first clue about the boot state:

- If you can interact with a BCH command prompt such as the following:

  ```
  Main Menu: Enter command or menu >
  ```

  Then the nPartition has not booted HP-UX or any virtual partitions.

- If you can interact with a vPars monitor prompt (`MON>`) then the nPartition has *at least* booted the `/stand/vpmon` vPars monitor.

  It is possible that one or more virtual partitions also are loaded/booted.

- If you can interact with the HP-UX console login prompt or the HP-UX command line then you still need to determine if HP-UX has booted in a vPars or in non-vPars mode.

If the nPartition console does not have any interactive prompt or command line, then you can use the Virtual Front Panel to help determine if the nPartition is in the process of booting/resetting or if HP-UX has hanged.

Step 3. When the nPartition's console gives access to the vPars monitor or HP-UX, use additional techniques to determine how many (if any) virtual partitions are loaded/booted.

You can type **Control-a** to switch among the virtual partition consoles. Each time you type **Control-a** the name of the current virtual partition or monitor (`[MON]`) is displayed in the console window.

If HP-UX is running, login as `root` and issue the **`vparstatus -w`** command to list the current virtual partition's name. The **`vparstatus`** command with no options lists all virtual partitions.

Example 11-2 Checking if Virtual Partitions are Running on an nPartition

The following examples show different nPartition boot states on systems that have the HP vPars software installed.

In the following example, the vPars monitor has booted on the nPartition but virtual partitions have not yet been loaded or booted. Typing **Control-a** repeatedly did not switch to any virtual partitions—only the monitor (`[MON]`) is running.

```
MON>
Control-a
[MON]
Control-a
[MON]
Return
MON>
```

In the following example, HP-UX is running on the nPartition. Although the vPars software is installed, the nPartition is running in non-vPars mode. The vPars monitor is not running, indicating that `/stand/vmunix` was booted from BCH, not the `/stand/vpmon` vPars monitor.

```
# vparstatus
vparstatus: Error: Virtual partition monitor not running.
#
```

Finally, in the following example, HP-UX is running on a virtual partition in the nPartition. The current virtual partition is "Shad" and it is the only loaded/booted virtual partition in the nPartition: the other virtual partition named "Mesh" is in a "Down" state.

```
# vparstatus -w
The current virtual partition is Shad.
# vparstatus
[Virtual Partition]
                                                                       Boot
Virtual Partition Name         State Attributes Kernel Path              Opts
============================== ===== ========== ========================= =====
Shad                           Up    Dyn,Manl   /stand/vmunix
Mesh                           Down  Dyn,Manl   /stand/vmunix             boot
```

```
[Virtual Partition Resource Summary]
                                            CPU    Num        Memory (MB)
                                  CPU     Bound/   IO    # Ranges/
Virtual Partition Name          Min/Max  Unbound  devs  Total MB    Total MB
==============================  ================  ====  ====================
Shad                              2/  8    2   0     8    0/  0         2048
Mesh                              2/ 12    2   2     3    0/  0         2048
#
```

Reboot for Reconfig or Shutdown to *Ready for Reconfig* from a Virtual Partition

This section describes how to perform a *reboot for reconfig* and how to shut down an nPartition to the *ready for reconfig* state on nPartitions that are running HP-UX one or more virtual partitions.

A **reboot for reconfig** resets all cell in the nPartition, performs any nPartition reconfigurations, and reboots the nPartition to its BCH interface and allows the nPartition to autoboot, if it is configured to autoboot.

A shut down to the **ready for reconfig state** resets all cell in the nPartition, performs any nPartition reconfigurations, and then holds all cells at a boot-is-blocked state, which makes the nPartition *inactive*.

Refer to the chapter *Booting and Resetting nPartitions* on page 197 for complete details.

vPars Management: Performing a *Reboot for Reconfig* or Shutdown to *Ready for Reconfig* from a Virtual Partition

This procedure covers how to perform a *reboot for reconfig* of an nPartition that is running virtual partitions.

This procedure also describes how to reset an nPartition to a *ready for reconfig* state, for an nPartition that is running vPars.

Step 1. Login to HP-UX running on any of the nPartition's virtual partitions.

Step 2. Issue the **`parstatus -w`** command to list the local nPartition's partition number, and confirm that you are logged into the virtual partition on the nPartition you want to reboot.

Step 3. Issue the **`vparstatus`** command to list details about all virtual partitions currently defined in the local nPartition.

Check the command's output to determine whether a reboot for reconfig or a shutdown to ready for reconfig already has been initiated from one of the virtual partitions in the nPartition.

The following note in the `vparstatus` output is presented when either type of reconfig shutdown has been initiated.

```
Note:  A profile change is pending.  The hard partition must be
rebooted to complete it.
```

If this note is presented, then in all other virtual partitions at an “Up” or “Load” or “Boot” state issue the **`shutdown -r`** command and *skip the next step*.

Step 4. Issue the **`shutdown`** command with the options appropriate for the type of reboot you want to perform.

You can perform a reboot for reconfig or reset the nPartition to the ready for reconfig (inactive) state.

Use either of the following lists for details.

To perform the *reboot for reconfig* of the local nPartition:

- First issue the **`shutdown -R`** command in the *current* virtual partition.
- Then in all other virtual partitions at an “Up” or “Load” or “Boot” state, issue the **`shutdown -r`** command.

  Any virtual partitions in a “Load” or “Boot” state must be shut down *after* they finish loading/booting HP-UX.

  If the nPartition has only one virtual partition—or if all other virtual partitions are in a “Down” or “Shut” state—you do not need to shut down other virtual partitions.

To reset the nPartition to a *ready for reconfig (inactive)* state:

- First issue the **`shutdown -R -H`** command in the *current* virtual partition.
- Then in all other virtual partitions at an “Up” or “Load” or “Boot” state, issue the **`shutdown -r`** command.

  Any virtual partitions in a “Load” or “Boot” state must be shut down *after* they finish loading/booting HP-UX.

  If the nPartition has only one virtual partition—or if all other virtual partitions are in a “Down” or “Shut” state—you do not need to shut down other virtual partitions.

Step 5. Monitor the nPartitions boot state by using its Virtual Front Panel. You can access the VFP from the service processor (GSP or MP) main menu.

Refer to the chapter *An Overview of nPartition Boot and Reset* on page 161 for boot status details.

Adding or Removing nPartition Cells from a Virtual Partition

This section describes how you can add cells and remove cells from an nPartition that is running HP-UX in one or more virtual partitions.

See *vPars Management of nPartitions: Reconfiguring nPartition Attributes from a Virtual Partition* on page 518 for details on reconfiguring other nPartition attributes, such as boot paths or the nPartition name, from a virtual partition.

Complete nPartition configuration procedures are given in the chapter *Managing nPartitions* on page 243.

vPars Management of nPartitions:
Adding and Removing nPartition Cells from a Virtual Partition

This procedure covers how to add and remove cells from an nPartition that is running the HP virtual partitions software product.

In this procedure, you make changes to the *local* nPartition from HP-UX running in a virtual partition in the nPartition.

For details on modifying remote nPartitions, or nPartitions not currently running vPars, refer to the chapter *Managing nPartitions* on page 243.

NOTE

After *removing* one or more cells from an nPartition, any virtual partitions defined within the nPartition may need to be reconfigured if they explicitly used processor or I/O resources on the removed cell(s).

The vPars software product automatically adjusts virtual partition configurations as needed to account for any expected hardware that is unavailable; as a result, *the vPars database may automatically be changed* after a cell is made inactive or is removed from its nPartition.

After *adding* one or more cells to an nPartition, to use the resources on the new cell(s) you also must modify the nPartition's virtual partition configurations. For example, for a virtual partition to bind the new cell's processors or use any attached I/O slots you must add the resource to the virtual partition by using the `vparmodify` command or the Virtual Partition Manager utility.

Likewise, after an nPartition's inactive cell is made active you also must modify the nPartition's virtual partition configurations to explicitly use the cell's processor or I/O resources.

The following procedure initiates an nPartition cell assignment change from HP-UX running on one virtual partition (using `parmodify` or Partition Manager) and then *if required* performs a reboot for reconfig of the nPartition.

Step 1. Login to HP-UX running on one of the virtual partitions in the nPartition you want to reconfigure.

You must login as `root` to perform this procedure.

Step 2. List the local nPartition's partition number to confirm that you are logged into the nPartition you want to modify.

Issue the **`parstatus -w`** command or use an equivalent Partition Manager procedure to determine the local partition number.

Step 3. Issue the **`vparstatus -w`** command to list the current virtual partition's name, and then issue the **`vparstatus`** command (with no options) to list all virtual partitions defined in the local nPartition.

```
# parstatus -w
The local partition number is 0.
# vparstatus -w
The current virtual partition is Shad.
#
# vparstatus
[Virtual Partition]
                                                                          Boot
Virtual Partition Name         State Attributes Kernel Path               Opts
============================== ===== ========== ========================= =====
Shad                           Up    Dyn,Manl   /stand/vmunix
Mesh                           Down  Dyn,Manl   /stand/vmunix             -iS

[Virtual Partition Resource Summary]
                                             CPU    Num        Memory (MB)
                                   CPU     Bound/   IO    # Ranges/
Virtual Partition Name           Min/Max  Unbound  devs   Total MB    Total MB
==============================  ================  ====   ====================
Shad                               2/  8    2   2     7    0/  0          2048
Mesh                               2/  8    2   2     3    0/  0          2048
#
```

After you add or remove a cell from the local nPartition, *if required* you will initiate the nPartition's reboot for reconfig from the *current virtual partition only* and will initiate a normal reboot (`shutdown -r`) from all other "Up" (loaded/booted) virtual partitions in the nPartition.

Step 4. Request that the cell(s) be added or removed from the local nPartition.

Use the `parmodify` command or Partition Manager to initiate the cell addition or removal. For detailed information on adding or removing the cell see the following list:

- You should adhere to HP's nPartition requirements and guidelines when adding or removing cells from an nPartition; refer to the chapter *Planning nPartition Configurations* on page 109.
- For details on adding or removing a cell from an nPartition, refer to the chapter *Managing nPartitions* on page 243.

After you have used the `parmodify` command or Partition Manager to modify the local nPartition's configuration, proceed with the next step.

Step 5. Determine if the local nPartition must be shut down to perform a *reboot for reconfig*.

The cell addition and removal procedures referenced in the previous step describe the situations where a reboot for reconfig of the local nPartition is required.

You **must** perform a reboot for reconfig if you removed an *active* cell or you specified the `-B` option to the `parmodify` command. You also **should** perform a reboot for reconfig if you added a cell with a "y" use-on-next-boot value.

You **do not** need to perform a reboot for reconfig if you removed an inactive cell (and did not specify the `-B` option) or if you added a cell with a "n" use-on-next-boot value (and did not specify the `-B` option).

Step 6. *Only if required* to complete the nPartition configuration change, perform a reboot for reconfig of the local nPartition.

First issue the **`vparstatus`** command to list the state (such as "Up" or "Down") of all virtual partitions in the local nPartition.

```
# vparstatus
[Virtual Partition]
                                                                      Boot
Virtual Partition Name        State Attributes Kernel Path            Opts
```

```
============================== ===== ========== ========================= =====
Shad                           Up    Dyn,Manl   /stand/vmunix
Mesh                           Load  Dyn,Manl   /stand/vmunix             boot

[Virtual Partition Resource Summary]
                                          CPU    Num        Memory (MB)
                                 CPU     Bound/   IO   # Ranges/
Virtual Partition Name         Min/Max  Unbound  devs  Total MB    Total MB
==============================  ================  ====  ====================
Shad                             2/  8    2   0      7    0/  0         2048
Mesh                             2/ 12    2   0      3    0/  0         2048
#
```

To perform the reboot for reconfig of the local nPartition:

- First issue the **`shutdown -R`** command in the *current* virtual partition.
- Then in all other virtual partitions that are at an “Up” or “Load” or “Boot” state, issue the **`shutdown -r`** command.

  Any virtual partitions in a “Load” or “Boot” state must be shut down *after* they finish loading/booting HP-UX.

  If the nPartition has only one virtual partition—or if all other virtual partitions are in a “Down” or “Shut” state—you do not need to shut down other virtual partitions.

For details see *vPars Management: Performing a Reboot for Reconfig or Shutdown to Ready for Reconfig from a Virtual Partition* on page 510.

After you issue the above `shutdown` commands, HP-UX on the virtual partitions is shut down and the vPars monitor is automatically rebooted.

Because a reboot for reconfig is a *reset of the nPartition hardware*, any virtual partitions that are configured for autoboot do not do so at this time. Instead, the nPartition boot process takes place—including the configured nPartition autoboot behavior (defined by the nPartition’s Path Flag settings).

As the reboot for reconfig occurs, *all cells* assigned to the nPartition will reset, any cell assignment changes for the nPartition will occur, and the cells will proceed to perform their self tests.

After the nPartition's cells complete self tests the partition rendezvous can occur and then the nPartition's BCH interface is initiated. If the nPartition is configured to autoboot, then that will occur; otherwise, the BCH interface is made available through the nPartition console interface.

To load/boot all virtual partitions, you can use the normal virtual partition boot methods. See the section *Booting HP-UX on Virtual Partitions* on page 487 for details.

Reconfiguring nPartition Attributes from a Virtual Partition

This section describes how you can modify nPartition attributes from HP-UX running in a virtual partition in the same server as the nPartition.

For details on changing nPartition cell assignments from a virtual partition, see *vPars Management of nPartitions: Adding and Removing nPartition Cells from a Virtual Partition* on page 513.

NOTE When one or more virtual partitions is loaded/booted on an nPartition, the `setboot` command *affects the current virtual partition's boot settings* and *does not* affect the local nPartition's boot settings.

In this situation, instead use the `parmodify` command to configure nPartition boot device paths from HP-UX.

nPartition configuration data is stored as part of the server Complex Profile and is separate from virtual partition configuration data, which typically is stored in the `/stand/vpmon` file on disk.

For complete nPartition configuration procedures, refer to the chapter *Managing nPartitions* on page 243. Also refer to the chapter *Booting and Resetting nPartitions* on page 197 for boot-related configuration tasks.

vPars Management of nPartitions: Reconfiguring nPartition Attributes from a Virtual Partition

This section covers the configuration of nPartition attributes other than cell assignments, when performing the nPartition configuration from HP-UX running in a virtual partition.

When performing this procedure you *do not* need to perform any reboots.

Step 1. Login to HP-UX running on any virtual partition within the nPartition whose attributes you wish to reconfigure.

Some changes, such as cell attribute changes, require that you initiate the reconfiguration from the local nPartition to which the cell is assigned. When the nPartition is running multiple virtual partitions, you can login to any of the virtual partitions on the local nPartition.

Step 2. Use the **`parstatus`** command or Partition Manager to list the nPartition's current configuration status.

Step 3. Use the **`parmodify`** command or Partition Manager to modify the nPartition's configuration. (Do not use the `setboot` command.)

For example, `parmodify -p0 -P NewName` changes nPartition number 0 to be named "NewName".

For other details, refer to the chapter *Managing nPartitions* on page 243.

Index