HP-UX vPars and Integrity VM V6.1 Administrator Guide

HP Part Number: 5900-1577
Published: March 2012
Edition: 1.0

Legal Notices

Acknowledgments

Revision history

Manufacturing Part Number	Supported Operating Systems	Supported Versions	Document Edition Number	Publication Date
5900–1577	HP-UX	11i v3	1.0	March 2012

Contents

Figures

Tables

Examples

1 Introduction

HP-UX vPars and Integrity VM Version 6.1 is a new product bundle that brings together vPars and Integrity VM functionality in a single, common, easy-to-use management environment. This product is a follow-on to HP-UX Virtual Partitions v6.0 and HP Integrity VM V4.3. The V6.1 product supports either a VM-only environment or a vPar-only environment, but not a mixed VM and vPar environment. A VM-only environment has the same user experience as an Integrity VM V4.3 follow-on release. Similarly, a vPar-only environment has the same user experience as a follow-on release of vPars V6.0.

This release features support for:

- HP-UX Virtual Partitions V6.1
- HP-UX Integrity Virtual Machines V6.1
- HP-UX 11i v3 March 2012 release
- The direct I/O networking feature
- N_Port ID Virtualization (NPIV) for direct SAN access
- HP Integrity Virtual Server Manager V6.1 (formerly, HP Integrity Virtual Machines Manager) support
- HP-UX GUID Manager support

NOTE: There is no HP OpenVMS operating system support at this time.

The single SMH-launched GUI interface (Integrity Virtual Server Manager) consolidates and simplifies management functions, and HP-UX GUID Manager can ensure NPIV storage access remains secure.

This chapter describes the “converged” vPars V6.1 and Integrity VM V6.1 product, including the following topics:

- About vPars
- About Integrity VM
- New and changed terminology
- vPar and Integrity VM media
- Types of I/O
- Running applications in the vPars and Integrity VM environment

1.1 Intended audience

This document is intended for system and network administrators responsible for installing, configuring, and managing virtual partitions and virtual machines. Administrators are expected to have an in-depth knowledge of HP-UX operating system concepts, commands, and configuration. In addition, administrators must be familiar with the Integrity machine console and how to install the operating systems running in their virtual environments (vPars and virtual machines).

1.2 Common Manageability of vPars and VMs

The V6.1 environment consists of a VSP (virtual server platform) and individual virtual servers. The virtual servers are either VMs or vPars. VSP supports the common manageability framework for both VMs and vPars. When managing VMs, VSP provides the same management functionality supported by an Integrity VM Host in prior Integrity VM releases. When managing vPars, VSP supports the same management functionality supported by VSP in the vPars V6.0 release. In addition, the HP Integrity Virtual Server Manager provides the same level of support for both VMs and vPars.

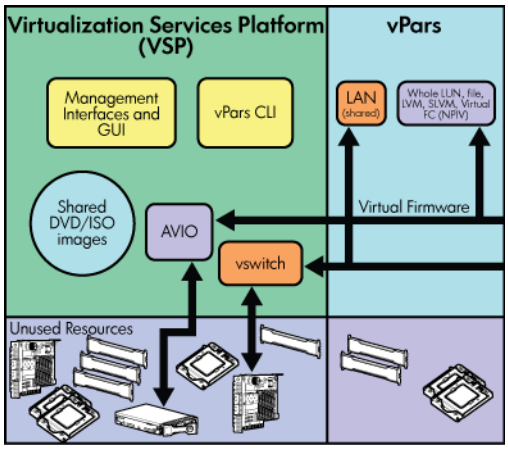
1.3 About vPars

vPars V6.1 is a follow-on release to the vPars V6.0 HP-UX Virtual Partitions product that enables you to run multiple instances of HP-UX simultaneously on a Virtualization Services Platform (VSP). Each virtual partition is assigned a subset of hardware, runs a separate instance of HP-UX, and hosts its own set of applications. As each instance of HP-UX is isolated from all other instances, vPars provides application and Operating System (OS) fault isolation. Applications running on top of HP-UX using vPars run the same as when run on HP-UX native-mode (standalone). No changes, recompilation, or re-certification is required for applications running in the virtual partitions unless otherwise noted. The patch level of each instance of HP-UX can be different. vPars V6.1 offers a common manageability framework that is accessible from the VSP. You can use the VSP to assign resources that includes CPU, memory, and I/O, and manage virtual partitions (vPars) from the graphical user interface (GUI). The VSP provides you control over creating, starting, stopping, modifying, and deleting virtual partitions.

With vPars V6.1, you can manage virtual partitions (as well as VMs) using the VSP manageability platform. This platform now includes a GUI. The same GUI allows you to create and manage vPars and VMs.

In vPars V6.1, each virtual partition requires a minimum of one dedicated processor core (CPU), one network port, one root disk, and memory sufficient for HP-UX and the hosted applications. The VSP also has similar requirements. In terms of scalability, it is possible to create 31 vPars in an eight socket 32 core BL890c i2 with each vPar having one dedicated physical processor core, or to create a single vPar with 31 cores on that same server. vPars V6.1 supports Superdome 2 and rx2800 i2 in addition to BL8x0c i2 servers supported in V6.0. Figure 1 (page 14) illustrates the vPars V6.1 framework.

Figure 1 vPars V6.1 framework



CAUTION: In V6.1, HP supports a vPar only or VM only environment, though creation of one type of virtual server when the other type already exists might be allowed by Integrity VM commands in some cases. You are strongly advised not to attempt creation of mixed vPar/VM configurations. A configuration of mixed vPars and VMs is not supported and might lead to unexpected behavior.

1.4 About Integrity VM

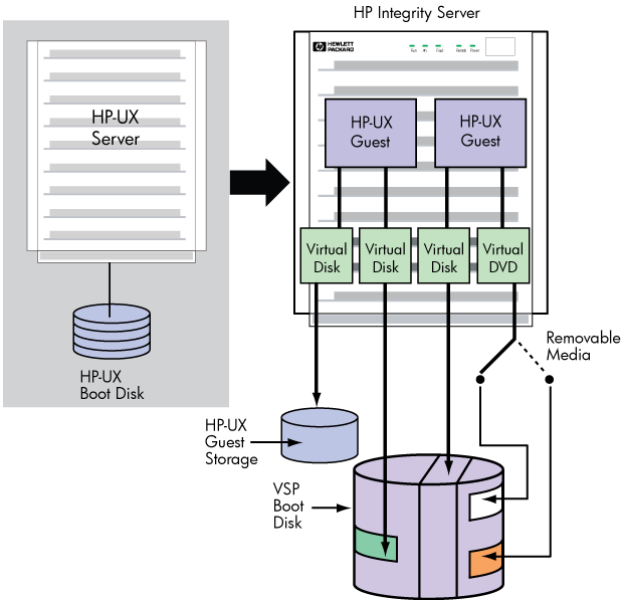
Integrity VM is a soft partitioning and virtualization technology that provides operating system isolation, with sub-CPU allocation granularity and shared I/O. Integrity VM can be installed on an Integrity server, Integrity server blade, or hardware partition (nPartition) running HP-UX. The environment consists of two types of components:

- Virtualization Services Platform (VSP)
- Virtual machines

The VSP virtualizes physical processors, memory, and I/O devices, allowing you to allocate them as virtual resources to each virtual machine.

Virtual machines are abstractions of real, physical machines. The guest operating system runs on the virtual machine just as it would run on a physical Integrity server, with no special modification. In addition to the vPars and Integrity VM V6.1 bundle, BB068AA, the VirtualBase bundle is included with the HP-UX 11i v3 March 2012 release and installed by default from the OE's. The VirtualBase bundle which contains additional functionality such as the vPars/VM AVIO network and storage drivers. The VirtualBase also provides a small guest software package that aids in local management of the guest's virtual machine.

Figure 2 Hardware Consolidation Using Integrity VM



Guests are fully loaded, operational systems, complete with operating system, applications, system management utilities, and networks, all running in the virtual machine environment that you set up for them. You boot and manage guests using the same storage media and procedures that you would if the guest operating system were running on its own dedicated physical hardware platform. Even the system administration privileges can be allocated to specific virtual machine administrators.

One way to benefit from Integrity VM is to run multiple virtual machines on the same physical machine. There is no set limit to the number of virtual machines that can be configured, but no more than 256 virtual machines can be booted simultaneously on a single VSP. Each virtual machine is isolated from the others. The VSP administrator allocates virtual resources to the guest. The guest accesses the number of CPUs that the VSP administrator allocates to it. CPU use is governed by an entitlement system that you can adjust to maximize CPU use and improve performance. A symmetric multiprocessing system can run on the virtual machine if the VSP system has CPUs for it.

Figure 2 illustrates how two HP-UX systems can be consolidated on a single Integrity server. The HP-UX boot disk is consolidated onto the same storage device as the VSP boot disk.

Because multiple virtual machines share the same physical resources, I/O devices can be allocated to multiple guests, maximizing use of the I/O devices and reducing the maintenance costs of the data center. By consolidating systems onto one platform, your data center requires less hardware and management resources.

Another use for virtual machines is to duplicate operating environments easily, maintaining isolation on each virtual machine while managing them from a single, central console. Integrity VM allows you to create and clone virtual machines with a simple command interface. You can modify existing guests and arrange networks that provide communication through the VSP's network interface or the guest local network (localnet). Because all the guests share the same physical resources, you

can be assured of identical configurations, including the hardware devices backing each guest's virtual devices. Testing upgraded software and system modifications is a simple matter of entering a few commands to create, monitor, and remove virtual machines.

Integrity VM can improve the availability and capacity of your data center. Virtual machines can be used to run isolated environments that support different applications on the same physical hardware. Application failures and system events on one virtual machine do not affect the other virtual machines. I/O devices allocated to multiple virtual machines allow more users per device, enabling the data center to support more users and applications on fewer expensive hardware platforms and devices.

△ **CAUTION:** In V6.1, HP supports a vPar only or VM only environment, though creation of one type of virtual server when the other type already exists might be allowed by Integrity VM commands in some cases. You are strongly advised not to attempt creation of mixed vPar/VM configurations. A configuration of mixed vPars and VMs is not supported and might lead to unexpected behavior.

1.5 New and changed terminology

The following new and changed terminology applies to the "converged" vPars and Integrity VM V6.1 product:

- HP Integrity Virtual Machines Manager is now named HP Integrity Virtual Server Manager
- The VM Host is now named Virtualization Services Platform (VSP).
- A guest is the guest operating system that is installed on the VM or vPar.

Virtual machines run on the same physical machine as the VSP and appear to be ordinary HP-UX processes. Each virtual machine emulates a real Integrity machine, including firmware. While virtual partitions are granted direct access to CPU and memory. The generalized term *virtual machine* is used throughout these documents, which refers to either type of machine container. A virtual machine can be referred to as a guest, VM, or vPar. A VM specifically refers to a virtual machine with virtualized (shared) CPU, memory and I/O resources, while a vPar specifically refers to a virtual partitioning of VSP CPU and memory resources and is granted direct (non virtualized) access to CPU and memory. The operating system running in a virtual machine is referred to as the guest operating system, or guest OS.

For more definitions and product descriptions, see "Glossary"

1.6 vPars and Integrity VM media

The HP-UX vPars Integrity VM V6.1 software is distributed on the HP-UX 11i v3 Operating Environment media with the Virtual Server OS (VSE-OE) and the Data Center OE (DC-OE). To install vPars and Integrity VM, select the optional software bundles for HP-UX vPars and Integrity VM (BB068AA) and Virtualization Base bundle (VirtualBase) prior to installing or updating HP-UX.

NOTE: If you are installing the HP-UX vPars and Integrity VM bundles BB068AA and VirtualBase, you must also install the HP-UX GUID Manager bundle GUIDMGR.

The HP-UX vPars and Integrity VM software for HP-UX 11i v3 is delivered in the following ways:

- As a stand-alone product on the HP-UX 11i v3 Application Software (AR) DVD
- As a product included in the HP-UX 11i v3 VSE-OE
- As a product included in the HP-UX 11i v3 DC-OE
- As a product included in the HP-UX 11i v3 Matrix OE

1.7 Types of I/O

The vPars and Integrity VM V6.1 release supports two types of I/O: accelerated virtual I/O (AVIO) and direct I/O (DIO) networking. With AVIO, I/O devices are para-virtualized. The I/O device

drivers for these devices in the guest operating system are virtualization aware, eliminating some of the virtualization overhead. However, the guest operating system still does not have direct visibility to the underlying hardware and the remaining virtualization overheads prevent the guest from achieving near native performance for certain I/O intensive workloads. With direct I/O networking, which is supported on HP Integrity Server Blade system BL8x0c i2 and HP Integrity Superdome 2 servers, a vPar and a VM can have direct control of the I/O of a device. The direct I/O networking feature minimizes the device emulation overhead, and also allows guest operating systems to control devices for which no emulation exists, enabling access to I/O hardware technology without requiring support from vPars or Integrity VM.

NOTE: Both AVIO networking and direct I/O networking support HP Virtual Connect.

Use of the *avio_lan* parameter for networking and the *avio_stor* parameter for storage provide the AVIO capability for vPars and VMs. For information about AVIO, see"Using AVIO with vPars and Integrity VM" (page 17). The `hpvmhwmgmt` command provides the *dio* parameter to designate a resource pool and allows the creation and management of a pool of direct I/O network capable devices that can be assigned to vPars or VMs. For information about direct I/O networking , see "Using direct I/O networking" (page 19).

1.7.1 Using AVIO with vPars and Integrity VM

AVIO is supported by multiple vPar/VM and guest operating systems. For AVIO support details, see the *HP–UX vPars and Integrity VM V6.1 Release Notes* and the AVIO product documentation. The vPar/VM configuration file and the `hpvmstatus` command and the `vparstatus` command display the `avio_lan` and `avio_stor` designators.

NOTE: HP strongly recommends that you use the same AVIO components from the same release on both the VSP and vPar/VMs, for example, both from the OE or both from the same Web Release (for example, WEB1103).

The following example shows the `hpvmstatus` command output of AVIO adapters for guest `avioclone`:

```
[Storage Interface Details]
Guest Device type                           :disk
Guest Adaptor type                          :avio_stor
Bus                                         :0
Device                                      :0
Function                                    :0
Target                                      :3
Lun                                         :0
Physical Storage type                       :disk
Physical device                             :/dev/rdisk/disk2

[Network Interface Details]
Physical Storage type                       :vswitch
Guest Adaptor type                          :avio_lan
Backing                                     :swlan1
Vswitch Port                                :5
Bus                                         :0
Device                                      :1
Function                                    :0
Mac Address                                 :2a-2e-5a-05-0a-ba

Physical Storage type                       :vswitch
Guest Adaptor type                          :avio_lan
Backing                                     :swlan2
Vswitch port                                :9
Bus                                         :0
Device                                      :2
```

```
Function                                        :0
Mac Address                                     :2a-2e-5a-05-0a-bc
```

Here is the output of the `vparstatus` command related to the AVIO adapters of a single vPar:

```
[IO Details]
        hba:avio_stor:0,0,0x50014C2000000002,0x50014C2800000002:npiv:/dev/fcd0
        network:avio_lan:0,1,0xb28c71e0a57b:vswitch:sitelan:portid:2
```

NOTE: The CLI accepts either `avio_lan` or `aviolan` and either `avio_stor` or `aviostor`. For example, the following `hpvmcreate` commands add both an AVIO network and an AVIO disk to the guest `aviotest`:

```
# hpvmcreate -P aviotest -O hpux -a network:aviolan::vswitch:swlan1 \
 -a disk:aviostor::disk/dev/rdisk/disk1
# hpvmcreate -P aviotest -O hpux -a network:avio_lan::vswitch:swlan1 \
 -a disk:avio_stor::disk/dev/rdisk/disk1
```

The AVIO network VSP driver allows simultaneous access to a vswitch from vPar/VMs configured with or without AVIO vPar/VM drivers for their virtual network interface cards (vNICs). Existing vPar/VMs continue to operate correctly, and their configurations remain unchanged. Virtual clients configured to use a common vswitch are allowed to share VLANs that are in use by vPar/VMs with or without AVIO configured. Ports remain distinct for each running a vPar/VM whether or not AVIO is in use. AVIO networks must use a Supported Host Physical Point Attachment (PPA) network device. For a list of supported AVIO PPAs, see the *HP-UX vPars and Integrity VM V6.1 Release Notes*.

NOTE: When using AVIO networking devices for guests that are configured as Serviceguard Packages, be sure that all Serviceguard standby `lans` are configured using PPA devices supported by AVIO. Failure to do so causes network connectivity to be lost even if the standby link is up.

AVIO increases the maximum number of devices that can be supported in a vPar/VM to 256 storage devices. The dynamic addition and deletion of AVIO storage devices is supported, and the following backing-store options are supported:

- HBA (such as /dev/fcd0) See "Configuring an NPIV HBA (vHBA)" (page 55)
- Disk (such as `/dev/rdisk/disk1`)
- Null (for DVD devices only)
- File
- Volume (lv)
- Tape
- Burner
- Changer

NOTE: Null for non-DVD devices is not supported.

With HP-UX 11i v3, the AVIO storage vPar/VM driver can receive events asynchronously from the VSP for `avio_stor` devices whenever the underlying storage, such as lun or target, changes state, for example, when a new lun or target is added or deleted or when the size of a lun changes. The asynchronous event generation occurs in addition to any notifications issued using the SCSI programming model, such as CHECK CONDITION on a subsequent I/O.

When the AVIO storage driver on the vPar/VM detects the events, it takes the appropriate actions, such as discovering the new targets. For example, if new targets are added using the `hpvmmodify -a` command, then the vPar/VM driver automatically detects the new device without the manual scan. The vPar/VM automatically detects any modification of the underlying backing storage.

Changing the underlying backing storage of a vPar/VM is best done when the it is not running to avoid damaging it. If the change is to a running vPar/VM, the administrator is responsible for knowing that the change will not adversely affect the health of the running environment. Although HP-UX vPars and Integrity VM does check to determine if the device is in use, those checks are not 100% reliable, because the vPar/VM might or might not be using the device at the time it is checked.

Backing storage can be adversely affected if the actual storage or access path is modified directly by an HP-UX server command, for example, by removing a file backing store or unmounting the file system. If the devices being changed are a result of some SAN reconfiguration, the `ioscan` command should be run on the VSP before attempting the change with the `hpvmmodify` command. If the backing storage is changed by remapping a different *`wwid`* to an existing `dsf` using: `scsimgr replace_wwid -D dsf`, the `hpvmdevmgmt -I` command needs to be run. If the backing storage is SAN presented as a different device and the change is done using: `io_redirect_dsf -d old_dsf -n new_dsf`, the vPar/VM must be modified using the `hpvmmodify` command to reference the new disk in place of the old disk.

1.7.2 Using direct I/O networking

The direct I/O networking feature supported in vPars and Integrity VM V6.1 allows administrators to assign network ports directly to a vPar/VM, giving the vPar/VM direct and exclusive access to that port on the NIC. NIC ports that are configured to be used for direct I/O are not shareable and cannot be used to back a vswitch. Before a NIC port/card can be assigned to a vPar/VM, you must first add it to the DIO pool.

NICs that support direct I/O networking on HP Integrity i2 and Superdome 2 servers provide either Function Level Assignment (FLA) or Device Level Assignment (DLA). The function in FLA refers to a single function on a multi-function NIC. A function can be a a single port on a multi-port card; some cards support multiple functions on a single port. The device in DLA refers to the entire multi-port NIC (all functions of the NIC). If a card supports FLA, each function (port) can be individually added or removed from the DIO Pool. FLA functions (ports) can be individually assigned to vPars/VMs. Each FLA function of the same card can be used by different vPars/VMs at the same time.

If a NIC supports only DLA, the entire card is added or removed from the DIO pool. You cannot assign a single port/function of a DLA card to the DIO pool. Once a DLA card is added to the DIO pool, individual functions can be assigned to vPars/VMs. To assign different functions of a DLA card to multiple vPars/VMs, each vPar/VM cannot be configured to 'reserve' resources (`resources_reserved` setting); however if multiple vPars/VMs have been assigned functions on the same DLA card (no reserved resources), only one VM can be booted at a time. For example:

- If you assign all four ports/functions of an FLA card to the DIO pool, you can assign port1 to vPar1, port2 to vPar2, and boot both vPar1 and vPart2 at the same time.
- If you assign a DLA NIC with four ports to the DIO pool, you can assign port1 to vm1 and port2 to vm2 only if `resources_reserved` is set to false. But you will only be able to boot vm1 **or** vm2; they cannot be booted at the same time.

The direct I/O networking functionality provides the following:

- 10 GB Ethernet network functions
- Support for FlexNICs created by HP Virtual Connect
- Near native network performance in vPar environments
- Improved performance over AVIO networking in VM environments
- CPU OL* operations with vPars
- DLKM operations in the vPar/VM
- Interrupt migrations in the vPar/VM and on the VSP

- Running vPars/VMs with DIO as Serviceguard nodes
- Support for HP-UX network providers

NOTE: The direct I/O networking functionality is not currently supported with the HP Auto Port Aggregation (APA) product.

For more information about using direct I/O networking, see "Using direct I/O networking" (page 158)

1.8 Running applications in vPars or Integrity VM environments

The VSP is the manageability platform for vPars or VMs. Though VSP runs the standard HPUX OE, it is a controlled environment, and no customer applications are to be installed or run on the VSP. Customer applications are to be run on an individual VM or vPar.

The following sections provide details of what can be run on the VSP as well as on the individual vPar or VM. The VSP system runs the Integrity VM or vPars software, which is responsible for allocating processor and memory resources to the running guests. The VSP system can run physical resource, performance, and software management and monitoring tools. To allow the VSP to allocate resources to the virtual environments, do not run end-user applications, such as database software, on the VSP system. Instead, run them in virtual environments.

Typical software you can run on the VSP system includes the following:

- HP-UX 11i v3 Virtual Server Operating Environment (VSE-OE)

  NOTE: vPars and Integrity VM V6.1 is included in the HP-UX VSE-OE, as well as the in the HP-UX DC-OE. You can install vPars and Integrity VM from the OE and run them on the VSP system. If you have purchased the HP-UX VSE-OE or the DC-OE, you are also entitled to run that OE on the VSP and the vPars and VM.

  For information about the software that is required on the VSP system, see Chapter 4 (page 43).

- Software installation tools (Ignite-UX and Software Distributor-UX)
- Hardware diagnostic and support tools to monitor guests (WBEM, online diagnostics, Instant Support Enterprise Edition [ISEE])
- System performance monitoring tools (GlancePlus, Measureware, OpenView Operations Agent)
- Utility pricing tools (Instant Capacity, Pay per use)
- Hardware management tools (nPartition Manager, storage and network management tools)
- Multipath storage solutions
- HP Serviceguard (which can be run on HP-UX guests as well).

Regardless of whether virtual machines or vPars are running, do not run other applications on the VSP system. Examples of applications that should not be run on the VSP are: Oracle, Workload Manager (WLM), HP SIM, and so forth. HP-UX vPars and Integrity VM installation modifies kernel parameters, making the system unsuitable for running applications.

An OS running in a virtual environment runs the way it does on a physical system. By allocating virtual resources, you provide the guest operating system and applications with access to memory, CPUs, network devices, and storage devices as if they were part of a dedicated system.

Typical software you can run in a virtual environment includes the following:

- HP-UX 11i V3 Virtual Server Operating Environment (VSE-OE)
- Software installation tools (Ignite-UX and Software Distributor-UX)
- System performance monitoring tools (GlancePlus, Measureware, OpenView Operations Agent)

Applications do not have to be changed to run on a vPar/VM OS. Operating system patches and hardware restrictions apply to vPars.

Do not run the following types of applications on a vPar/VM:

- Virtualization platform (HP-UX vPars and/or Integrity VM software)
- Utility pricing tools (run on the VSP)
- Capacity planning tools (run on the VSP)
- Applications that require direct access to physical hardware (for example, disaster-tolerant solutions)

You must purchase licenses for any software you run on a virtual machine or a vPar, including the HP-UX operating system and any HP or third-party layered software. You can purchase the licenses for HP software under the HP Virtualization Licensing program. For more information, contact your HP Support representative.

Always read the product release notes before installing any software product so that you have the latest information about changes and additions to the documentation. The following chapters describe how to install the vPars and Integrity VM software and how to create vPars and VMs to run on the VSP system.

1.9 Do not install applications on the VSP system

When you install HP-UX vPars and Integrity VM, HP-UX kernel parameters are changed to accommodate the virtual environments. This makes the system unsuitable for running any other applications. Regardless of whether guests are configured and running, the VSP system is not configured to allow applications to share system resources. You can run system management utilities and Serviceguard, as documented in the *HP Serviceguard Toolkit for Integrity Virtual Servers User Guide* at HP Serviceguard Toolkit for Integrity Virtual Servers.

- **Using backup solutions for VSP and virtual environment backups**

  Backup solutions such as HP Data Protector or Veritas NetBackup can be used on both the VSP system and the vPar/VM systems. Consult the support matrix of such products for supported versions. Install the backup (client) agents on the VSP and the vPar/VMs. HP highly recommends that the /var and /opt directories, in addition to the standard locations, be backed up regularly on the VSP system. Do not use the VSP system as a backup server. For more information, see the *HP-UX 11i v3 Installation and Update Guide*.

- **Using HP GlancePlus/iX to monitor virtual environments**

  You can use Glance on the VSP to monitor vPar/VM data, but recorded measurements can be misleading. Glance receives the CPU accounting information from the vPar/VM kernel. Because the VSP can take the vPar/VM processor away (for example, when a hardware interrupt occurs), the time spent running other vPar/VMs is reported for the state that the vPar/VM was in at the time the CPU was taken away. For more information about using Glance, see *glance*(1M).

  Glance 4.6 or later is supported running in a VSP or in a vPar/VM; however, certain measurements might not apply in a particular context or report limited results. For example, measuring CPU utilization on the VSP reports all the time spent running in vPar/VMs as "system time"; to receive "user time" or "nice time" for a given vPar/VM, you must run Glance in that vPar/VM. Similarly, memory-related faults, or system calls for a vPar/VM, are not visible from Glance running in the VSP, from Glance running in that vPar/VM. Glance also offers a number

of virtualization-related measurements. Note, Glance refers to virtual environments as logical systems.

- **Using HP Global Workload Manager (gWLM)**

  If you use gWLM within Insight Dynamics — Matrix OE to manage virtual machines, when you upgrade the VSP, make sure the gWLM agent on that host is running gWLM A.07.01 or greater. Also, the managing Insight Dynamics — VSE Central Management Station (CMS) must be running A.07.00 or greater, as described in the *HP Insight Dynamics —VSE 7.0: Integrity CMS Installation and Configuration Guide*. To upgrade the VSP, use the following procedure:

  1. Remove the gWLM agent using the following command:

     ```
     # swremove gWLM-Agent
     ```

  2. Upgrade to vPars and Integrity VM V6.1 as described in "Upgrading the VSP from previous versions of Integrity VM" (page 43).
  3. Upgrade the gWLM agent, as described in the *HP Insight Dynamics — VSE 7.0: Integrity CMS Installation and Configuration Guide*.

  If you install the current version of vPars and Integrity VM without upgrading to gWLM A.07.00 or later, and then attempt to use gWLM within Insight Dynamics — Matrix OE to manage virtual environments, you will continue to see vPars as VMs and any attempt to manage them will lead to unexpected scenarios. You will still be able to manage VMs with any gWLM version greater than A.02.50.

- **Using the HP Integrity Virtual Server Manager**

  The Virtual Server Manager product provides a graphical user interface (GUI) for vPars and Integrity VM. It is available from either of the following management interfaces:

  - HP System Management Homepage (SMH).

    For more information about using Integrity Virtual Server Manager under SMH, see the HP Integrity Virtual Server Manager v6.1 User Guide.

  - HP Insight Dynamics — Matrix OE Management Software environment in the HP Systems Insight Manager (SIM) on the Central Management Server (CMS).

    For more information about Insight Dynamics — Matrix OE, see the *HP Insight Dynamics 7.0 Getting Start Guide* on the BSC website: http://bizsupport2.austin.hp.com/bc/docs/support/SupportManual/c02048567/c02048567.pdf

    Integrity Virtual Server Manager is designed to run on HP-UX 11i v3 with vPars and Integrity VM V6.1. Earlier versions of Integrity Virtual Machines Manager (V3.0 and earlier) do not install or run on an HP-UX 11i v3 VSP and cannot be used with vPars and Integrity VM V6.1. Users upgrading to vPars and Integrity VM V6.1 must upgrade to Integrity Virtual Server Manager V6.1. In addition, users of HP Insight Dynamics Virtual Software Environment software versions earlier than V4.0 must upgrade to HP Insight Dynamics — Matrix OE v7.0 (which contains Integrity Virtual Server Manager V6.1).

- **Using HP Instant Capacity with Integrity VM**

  In an Integrity VM environment, Instant Capacity software provides meaningful functionality only on the VSP; it does not run on a virtual machine (also known as a guest). In particular, Instant Capacity commands report an error if an attempt to run on a virtual machine is made from a guest. The user cannot run a GiCAP Group Manager on a guest, nor can specify a guest in the host list for a GiCAP group member.

  In the case of a vPar v6.1, Instant Capacity commands are supported on the VSP OS. However, on the vPar v6.1 guest OS, you cannot execute Instant Capacity commands directly to activate or deactivate the cores. For an activation operation, first activate the cores on the VSP OS using the `icapmodify` command and then execute the `vparmodify` command to complete

the activation of the cores on the vPar v6.x guest OS. Similarly, for a deactivation operation, execute the `vparmodify` command on the vPar v6.1 guest OS and then execute a `icapmodify` command on the VSP OS.

iCAP commands issued from the OA activate or deactivate cores only in the VSP. `vparmodify` commands have to be issued in the VSP to move the core to and from a vPar v6.1. If there is only one core in the VSP and the rest assigned to vPars a deactivation request from the OA will fail. Cores must first be deactivated from the vPar v6.1 by issuing the `vparmodify` command on the VSP followed by an `icapmodify` command to deactivate the core from the VSP.

TiCAP is consumed once the core is active at either the VSP or vPar v6.1. If TiCAP is being used, to stop consuming TiCAP, the core must be deactivated from the vPar v6.1 as well as the VSP.

1.10 Attributes that can be changed dynamically

A dynamic change does not require a reboot of the virtual environment in question. The following attributes can be changed dynamically:

Table 1 Attributes changed dynamically

Attribute	vPars	VMs
CPU **1.** Adding or removing vPar/VM vCPU entitlement. The default is uncapped mode. In uncapped mode, this is also automatic based on overall "free" entitlement. **2.** Enabling or disabling vCPUs from within a vPar/VM. **3.** Adding or removing CPUs to and from a vPar/VM from the VSP.	**1.** N.A. **2.** No **3.** Yes	**1.** Yes **2.** Yes **3.** No
Memory • Adding or removing the memory in use by a vPar/VM. This can also be made automatic with AMR (Automatic Memory Reallocation) based on overall "free" memory.	No	Yes
Network • Adding or removing virtual switches (vswitches) on the VSP.	Yes	Yes
Storage • Adding or removing storage to or from a vPar/VM. **NOTE:** Depending on the type of storage being used, there may be additional steps required. See Section 9.2.1.5 (page 121)	Yes	Yes
Migration **1.** Migrating online. **2.** Migrating offline.	**1.** No **2.** Yes	**1.** Yes **2.** Yes

NOTE: Before you add or remove memory, networking, or storage from a vPar or a VM, be sure you know if further action is required on the vPar or VM.

1.11 Related products

Some of the HP products that you can use with vPars and Integrity VM include:

- HP-UX operating system — HP-UX vPars and Integrity VM run on HP-UX 11i v3 Integrity systems on the VSP. V6.1 requires that the VSP be installed with the HP-UX 11i v3 March 2012 release. For more information, see the *HP-UX 11i v3 Installation and Update Guide*.
- HP WBEM Services for HP-UX — Many related products, such as Virtual Server Manager and gWLM, require the VSP system be running HP WBEM Services.
- HP Matrix Operating Environment — A graphical user interface for managing HP Integrity central managed systems (CMS). Runs under HP Systems Insight Manager. For more information, see the *HP Matrix Operating Environment 7.0 Getting Started Guide*.
- HP Insight Global Workload Manager (gWLM) — As part of HP Matrix OE, this software product allows you to centrally define resource-sharing policies that you can use across multiple Integrity servers. These policies increase system utilization and facilitate controlled sharing of system resources.

  Make sure the version of gWLM is appropriate for this version of virtualization software, as described in the *HP-UX vPars and Integrity VM V6.1 Release Notes*.
- HP Integrity Virtual Server Manager — A graphical user interface for creating and managing vPars and VMs. Runs under either HP System Management Homepage (HP SMH) or HP Systems Insight Manager (HP SIM) as part of the HP Matrix OE. For more information, see the *HP Integrity Virtual Server Manager 6.1 User Guide*.
- HP Integrity VM Providers — To manage virtual environments with Virtual Server Manager, gWLM, or any Matrix OE components, install the appropriate provider software from the operating system media or the VirtualBase bundle.
- VERITAS Volume Manager— A data storage solution product that can be used to manage the physical disks on the VSP. For more information, see the *VERITAS Volume Manager Administrator's Guide*.
- HP Serviceguard — A software product that allows you to create clusters of HP-UX systems for high availability. For more information, see the *Managing Serviceguard* manual.

1.12 Using the vPars and Integrity VM documentation

The vPars and Integrity VM product bundle includes several useful sources of information, whether you are considering how to set up your vPar or virtual machine or determining how to upgrade your installation.

1.12.1 Integrity VM commands

Integrity VM commands provide a convergence point between Integrity VM and vPars. You can use Integrity VM commands to create, clone, start, and manage not only VMs, but also vPars. You can continue to use vPars commands (whose manpages are listed in Table 3 (page 26)), but Integrity VM commands provide a superset of features to accommodate both VMs and vPars.

For online information about using Integrity VM commands, see the following manpages on the VSP system:

Table 2 Integrity VM commands

Command	Description
hpvm(5)	Describes the Integrity VM environment.
hpvmclone(1M)	Describes how to create virtual machines based on existing virtual machines.
hpvmcollect(1M)	Describes how to collect virtual machine statistics.

Table 2 Integrity VM commands *(continued)*

Command	Description
hpvmconsole(1M)	Describes how to use the virtual machine console.
hpvmcreate(1M)	Describes how to create virtual machines.
hpvmdevinfo(1M)	Reports about storage for a virtual machine.
hpvmdevmgmt(1M)	Describes how to modify the way virtual devices are handled.
hpvmdevtranslate(1M)	Translates Integrity VM guest devices to agile devices.
hpvmhostgdev(1M)	Manages Integrity VSP devices available for virtual machine access.
hpvmhostrdev(1M)	Manages virtual machine access to devices used by the Integrity VSP system.
hpvmhwmgmt(1M)	Allocates resources to the specified resource pool for exclusive use by virtual machines.
hpvminfo(1M)	Describes how to get information about the VSP.
hpvmmigrate(1M)	Describes how to migrate active guests and offline virtual machines from one VSP to another.
hpvmmodify(1M)	Describes how to modify virtual machines.
hpvmmove_suspend(1M)	Moves suspend files to a different directory.
hpvmnet(1M)	Describes how to create and modify virtual networks.
hpvmnvram(1M)	Displays, creates, edits and removes EFI variables in NVRAM files from a VSP.
hpvmpubapi(3)	Describes several new public APIs.
hpvmremove(1M)	Describes how to remove a virtual machine.
hpvmresources(5)	Describes how to specify the storage and network devices used by virtual machines.
hpvmresume(1M)	Manages Integrity VSP devices available for virtual machine access.
hpvmsar(1M)	Displays performance information about one or several guests on the same host.
hpvmstart(1M)	Describes how to start virtual machines.
hpvmstatus(1M)	Describes how to get statistics about the guests.
hpvmstop(1M)	Describes how to stop a virtual machine.
hpvmsuspend(1M)	Suspends a virtual machine.

The following manpages are also provided in the HP-UX virtual environmentt:

- *hpvmcollect*(1M) — Describes how to collect virtual environment statistics.
- *hpvmdevinfo*(1M) — Reports about storage for a virtual environment.
- *hpvminfo*(1M) — Describes how to get information about the VSP.
- *hpvmmgmt*(1M) — Describes how to manage dynamic memory from the vPar/VM.
- *hpvmpubapi*(3) — Describes several new public APIs.

NOTE:

HP-UX provides the `gvsdmgr` utility, which manages AVIO HBAs. For information about the `gvsdmgr` utility, see the HP-UX `gvsdmgr` (1M) manpage.

1.12.2 vPars commands

From the VSP you can run vPars commands to create, modify, and remove virtual partitions and virtual switches. Superuser privilege is required to run the commands from the VSP. These commands cannot be run from the OA or from inside a virtual partition.

Table 2 contains a summary of the VSP commands and descriptions of their use. The following sections provide brief information about each command. For more information about the commands, see the respective manpages.

Table 3 VSP commands in vPars

Command	Description
vparboot(1M)	Boots a virtual partition.
vparcreate(1M)	Creates a new virtual partition.
vparmodify(1M)	Renames or modifies the resources of a virtual partition. It can also suspend the configuration of the virtual partition.
vparremove(1M)	Removes an existing virtual partition.
vparreset(1M)	Resets a virtual partition. Simulates, at the virtual partition level, the hard reset, soft reset (Transfer Of Control, TOC), power off, or graceful shutdown operations. When compared with the earlier versions of vPars, the `vparreset` operations more closely match that of physical hardware.
vparstatus(1M)	Displays information about one or more virtual partitions. The `vparstatus` can also display details about the available resources that can be added to a virtual partition.
vparhwmgmt(1M)	Manages the pool of CPU resources dedicated for use by the virtual partitions on the VSP.
vparnet(1M)	Creates and controls a vswitch.
vparconsole(1M)	Connects to the console of a virtual partition.

When you use `vparcreate` command to create a vPar, resources are reserved are configured, even while the vPar is off. The vPar is also set to boot automatically whenever the VSP reboots. However, if you use the `hpvmcreate` command to create a vPar, the resource reservations are not configured, and the vPar is not set to reboot automatically. For more information, see "Reserved resources and resource over-commitment" (page 68).

1.12.3 Virtual Environment Console

The virtual environment console is a special interface for managing vPar/VMs. To start the virtual console after you create a vPar or VM, enter the either the `vparconsole` command or the `hpvmconsole` command and specify the vPar/VM name. For help using the virtual console, enter the `HE` command. For more information about the virtual console, see Section 11.6 (page 176).

1.13 Using this manual

This manual provides all the information you need to install Integrity VM, create virtual machines, install and manage guests, and use all the features of Integrity VM. Table 4 describes each chapter in this manual.

Table 4 Chapters in this Manual

Chapter	Read if...
Chapter 1: "Introduction" (page 13)	You are new to HP Integrity Virtual Machines.
"Installing and configuring VSP for HP-UX vPars and Integrity VM V6.1" (page 29)	You are installing and configuring the VSP.
Chapter 3: Installation procedures for HP-UX vPars and Integrity VM V6.1 (page 35)	You are installing the HP-UX vPars and Integrity VM V6.1 product.
"Using NPIV with vPars and Integrity VM" (page 53)	You are using NPIV to configure a vPar or VM to use vHBAs.
Chapter 6: "Creating virtual partitions" (page 59)	You are setting up a new virtual partition on a VSP system.

Table 4 Chapters in this Manual *(continued)*

Chapter	Read if...
Chapter 7: "Creating virtual machines" (page 75)	You are setting up new virtual machines on your VSP system.
Chapter 8: "Installing the HP-UX guest operating system and software" (page 101)	You are creating vPars and VMs that will run the HP-UX operating system.
Chapter 9: "Creating virtual storage devices" (page 107)	You need to make changes to the storage devices used by the VSP or virtual environments.
Chapter 10: "Creating virtual networks" (page 143)	You need to make changes to the network devices on the VSP system or to the virtual network devices used by the virtual machines.
Chapter 11: "Managing vPars/VMs" (page 165)	You need to manage existing virtual machines and resources.
Chapter 12: "Migrating virtual machines and vPars" (page 195)	You need to move vPars or VMs from one system to another.
Chapter 13: "Reporting problems with vPars and Integrity VM" (page 215)	You encounter problems while creating or using virtual environments.
Chapter 14: "Support and other resources" (page 221)	You need information about HP support.
"Appendix A: Rolling back to the previously installed version of Integrity VM" (page 225)	You plan to roll back to the previous version of Integrity VM.
Appendix B: "Sample Script for Adding Multiple Devices" (page 227)	You want to specify multiple storage devices at one time for a guest.
"Glossary" (page 233)	You do not understand the definition of a term used in the vPars and Integrity VM product documentation.

This manual and the *HP-UX vPars and Integrity VM V6.1 Release Notes* are available on the Instant Information DVD or may be viewed, downloaded, and printed from the web at http://www.hp.com/go/hpux-hpvm-docs.

2 Installing and configuring VSP for HP-UX vPars and Integrity VM V6.1

Installation of the VSP is the same whether you want to run a vPars only environment or a VM only environment. The software bundles that need to be installed on the VSP are the same in either case. This chapter describes the following:

- VSP installation requirements
- Configuring the VSP
- VSP cores
- VSP memory
- Hyperthreading setting on VSP
- VSP I/O configuration
- VSP kernel tunables

2.1 VSP installation requirements

The following software bundles must be installed on the VSP:

- HP-UX 11i v3 March 2012 (AR1203) (Base OE, VSE-OE, or DC-OE)
- BB068AA, VirtualBase (included with VSE-OE and DC-OE or as a separate install from the application media)
- HP-UX GUID Manager

NOTE: The vPars and Integrity VM V6.1 bundle BB068AA does not include the HP Integrity Virtual Server Manager software or the HP-UX GUID Manager software. You must install the HP-UX GUID Manager, but the HP Integrity Virtual Server Manager is optional. Both these products are separately installed ISU (Independent Software Unit) bundles. HP Integrity Virtual Server Manager and the HP-UX GUID Manager can be downloaded from the HP Software Depot at http://www.software.hp.com.

2.2 Configuing the VSP

After installation of the vPars and Integrity VM V6.1 product, the VSP is ready for use, you can now create and manage vPars or VMs. As mentioned earlier, the V6.1 product supports either a VM-only environment or a vPars-only environment. If you are planning to have a vPar environment, create only vPars. If you are planning to have a VM environment, create only VMs.

Before creating vPars or VMs, you should be aware of the VSP settings. Note that the core and memory setting in the VSP is configured appropriately when the product is started. There is no change needed unless otherwise documented.

2.3 VSP cores

There are two CPU pools in the VSP: one for the dedicated VSP cores, and the other for use by vPars/VMs. Use the `hpvmhwmgmt -p cpu -l` command to display the allocation of CPU cores.

2.3.1 vPar environment

As in vPars V6.0, you need at least one core reserved/dedicated to the VSP for starting vPars. In vPars6.0, the cores reserved/allocated for VSP is one by default, even when there are no vPars configured. In V6.1. VSP core is reserved/allocated automatically when the first vPar is created. Also note that in vPars V6.0, all the non-VSP cores (cores available for allocation to vPars) were

not visible to HPUX commands in the VSP. In V6.1, all the non-VSP cores are visible to the VSP. When vPars are started, these cores are deallocated from the VSP.

A single VSP core can service vPar management requests and moderate to heavy I/O loads. When the VSP core becomes saturated, the response time of vPar commands and other applications being run on the VSP might increase. If the CPU saturation makes the response time of the vPar commands intolerable, use the `vparhwmgmt` command to increase the core count in the VSP. You can modify the VSP core count to a value greater than one, if additional processing power is required to support high I/O workloads or a large configuration such as BL890c i2. You can use performance tools such as Glance and Top to determine the CPU utilization of the VSP.

The sum of the VSP and vPar core counts cannot exceed the number of cores on the system. Hence, any adjustment you make to the VSP core count will affect the cores available to the vPars. While adjusting the VSP core count, if you exceed the system core count, and if the vPars are already configured, an error occurs. In such a situation, to satisfy the core count required for the VSP, first adjust the core count of one or more vPars using the `vparmodify` command, then adjust the VSP CPU core count using the `vparhwmgmt` command.

IMPORTANT: If the system is brought down due to a faulty CPU core and the cores are deconfigured, then the vPars might not boot during the subsequent boot of the VSP. This is possible if the sum of the remaining cores is less than the sum of the cores allocated to the VSP and vPars as displayed from the `vparhwmgmt` command. You can clear this condition by removing the cores from the vPars or VSP to meet the configuration requirements.

For example, the VSP has one core allocated to it and the vPars has the remaining seven cores, and a core is deconfigured as it has a hardware issue. After you boot the VSP, none of the vPars are bootable until the number of cores assigned to the vPars is reduced by the number of deconfigured cores. In this case, you must reduce the CPU count of one of the vPars by one.

2.3.2 VM environment

As in Integrity VM V6.1 environment, there is no requirement for a dedicated VSP core. By default, there are no dedicated VSP cores. All the cores on the VSP are in the vPar/VM pool and are shared by the guests. However, you can configure dedicated cores in the VSP, if there is a need, using the `hpvmhwmgmt` command, but that is not required or recommended in a VM-only environment.

IMPORTANT: If the system is brought down due to a faulty CPU core and the cores are deconfigured, some of the VMs might not boot during the subsequent boot of the VSP. You might have to reduce the number of vCPUs of the affected VMs and/or reduce the overall CPU entitlement for all the VMs to boot the affected VMs.

2.3.3 Default cores after installing V6.1

If there are no existing VM or vPar configurations on the VSP (for example, a brand new installation), then you will see that the cores reserved/allocated for the VSP is zero by default. If there are existing vPar configurations (for example, upgrading an existing vPar V6.0 system), you will see that the cores reserved for VSP is non-zero. If there are existing VM configurations (for example, upgrading an existing VM V4.3 system), you will see that the cores reserved for the VSP is zero, matching the Integrity VM Host in a V4.3 environment.

2.4 VSP memory

As mentioned in other sections, the VSP has a controlled environment tuned for supporting the vPars and Integrity VM V6.1 product functionality. This applies to memory availability in the VSP too. HP strongly discourages running of customer applications or other workloads on the VSP, because this might affect the memory availability for the optimal functioning of vPars and Integrity VM product.

The Integrity VM product, on startup, reserves a significant portion of the free system memory available on the VSP for the vPar/VM memory pool. The memory left as available in the VSP is sufficient for the optimal functioning of the vPars and Integrity VM product functionality and features on the VSP.

Typically, about 92% of free memory available at the Integrity VM product start time (after HP-UX has booted up on the VSP) is reserved for the vPar/VM memory pool. The amount reserved also depends on the total system memory and the total number of system cores. On a small memory system (32G or less), you would find that a lower percentage of system memory is reserved for the vPar/VM memory pool;, whereas on a large memory system, you would find that a higher percentage is reserved for the vPar/VM pool. Use the `hpvmhwmgmt -p memory -l` command to display the allocation of memory. The `command.log` also has information about the free memory available when the vPar/VM memory pool was allocated.

Note that only 64MB or larger contiguous chunks of memory is reserved for the vPar/VM memory pool, and therefore, the system memory fragmentation at Integrity VM product start time affects the amount of memory that can be reserved for the vPar/VM memory pool. If the Integrity VM product is stopped and is restarted only after a long time, it is possible that there is not enough 64MB or more contiguous memory ranges in the VSP to match the memory that was reserved for vPar/VM pool previously. This can lead to an over-commitment of the memory assigned to the VMs or vPars.

NOTE: HP strongly recommends that you restart the VSP when the Integrity VM product needs to be restarted so that system memory fragmentation impacts on vPar/VM memory pool size can be minimized. There is still a chance of not being able to get the exact same amount of memory reserved earlier for the vPa/VMr pool leading to an over-commitment. This situation would be very rare, but not impossible. In that event, memory assigned to an individual VM or vPar needs to be adjusted to come out of the overcommitted state.

Memory availability for VSP use can be controlled using the `HPVM_MEMORY_OVERHEAD_PERCENT` configuration variable. If this variable is set to an appropriate value in the `hpvmconf` file, that value is used to determine the amount of memory reserved for vPar/VM the memory pool. For example, if `HPVM_MEMORY_OVERHEAD_PERCENT` is set to N, then (100-N)% of free system memory available at Integrity VM product start time (after HP-UX has booted up on the VSP) is reserved for the vPar/VM memory pool. Note that a VSP restart (or Integrity VM product restart) is required for this change to take effect. HP strongly recommends that customers do not use this `configuration` variable to change VSP memory availability unless otherwise documented or recommended by HP field personnel.

2.5 VSP memory overhead estimation

As explained earlier, VSP requires some amount of memory for the optimal functioning of the V6.1 product. Similarly each vPar or VM also has some amount of memory overhead depending on the size of the vPar or VM. Given below is a rough estimate of the memory overhead required for the VSP as well as individual vPars and VMs.

The vPar/VM memory pool reserved is roughly about 92% of the system free memory available at the time of V6.1 product startup. The remaining memory is left out as free memory available for VSP use. This is in addition to the memory taken up by HP-UX to boot on the VSP. The memory taken up by HP-UX to boot depends on the size of the system, including total memory, number of cores and the I/O devices on the system. So the overall VSP memory overhead is the sum of the memory HP-UX needs to boot up on the system plus the free memory left in the VSP for optimal functioning of the VSP. In terms of overall system memory, the VSP memory overhead typically equates roughly to 1500MB + 8.5% of the total physical memory.

Note that the calculation for how much memory is in VSP versus what is available for vPars and VMS is done at product start time. You can see the `hpvmhwmgmt –p memory –l` output to see how much memory is available for vPar/VM memory pool size.

In addition to the VSP memory overhead, individual vPars and VMs have a memory overhead depending on their size. A rough estimation of the individual guest memory overhead can be done using the following formula:

Guest memory overhead = cpu_count * (guest_mem * 0.4% + 64M)

Where:

guest_mem is the guest (vPar/VM) memory size
cpu_count is 1 for vPar and *num_vcpus* for VM. (Memory overhead in a vPar is independent of number of cpus in a vPar) .

For example, a 16G 4vCPU VM, the overhead is roughly about 512M. For the same 16G VM with 1 vCPU, it would be around 128M. For a 16G, 16 cpu vPar, as well as a 16G, 1 CPU vPar, memory overhead would be 128M.

When you create a 16G 4vCPU VM, additional 512M is used up by the VM. This memory is taken from the vPar/VM pool. Note that there may be some amount of memory taken up from the VSP memory when a vPar or VM is started. However, that is in most cases negligible compared to the vPar/VMoverhead memory taken up from the vPar/VM pool.

The `hpvmstatus -s` command output has additional information displayed to take into account the guest memory overhead required. In the following example:

```
# hpvmstatus -s
…
        Available memory for vPars and VMs = 24320 Mbytes
        Available memory for 7 (max avail.) CPU VM = 23552 Mbytes
        Available memory for 7 (max avail.) CPU vPar = 24192 Mbytes
…
```

Although available memory in the vPar/VM memory pool is 24320 MB, if you want to create a 7 vCPU VM with all the available memory, the VM can be created only with 23552 MB. The difference is the overhead memory required for a 7vCPU 23552MB VM. Note that the numbers shown in `hpvmstatus` display are approximate and the actual memory allowed to create a VM might have slight variations.

2.6 Hyperthreading setting on the VSP

By default, the VSP has the hyperthreading (firmware setting) ON in the npartition or server; whereas `lcpu_attr` is OFF in the VSP. This setting provides optimal performance and responsiveness for the VSP. HP strongly discourages changing the default hyperthreading settings in the VSP, unless it is recommended in the documentation or there is some other compelling reason to do so.

Hyperthreading is supported in individual vPars. The `setboot` command in individual vPars shows that `HT` is ON. You can turn on `lcpu_attr` in individual vPars using the `kctune` command. By default, `lcpu_attr` is OFF in the vPar (default behavior of HP-UX). Note that even when `lcpu_attr` is OFF in the VSP, each vPar can have its individual `lcpu_attr` enabled to get hyperthreading functionality in the vPar.

As in previous releases, hyperthreading is not supported for VMs. Therefore, individual VMs will not show any hyperthreading capability.

2.7 VSP I/O configuration

To map direct I/O devices between the VSP and the vPars or VMs:

- From the VSP:

  ```
  # hpvmdevinfo -P vm
  ```

- From the vPar or VM:

  ```
  # hpvmdevinfo
  ```

For AVIO, HP strongly recommends that you use the same AVIO components from the same release on both the VSP and vPar/VMs, for example, both from the OE or both from the same Web Release (for example, WEB1103).

The AVIO network VSP driver allows simultaneous access to a vswitch from vPar/VMs configured with or without AVIO vPar/VM drivers for their virtual network interface cards (vNICs). Existing vPar/VMs continue to operate correctly, and their configurations remain unchanged. Virtual clients configured to use a common vswitch are allowed to share VLANs that are in use by vPar/VMs with or without AVIO configured. Ports remain distinct for each running a vPar/VM whether or not AVIO is in use. AVIO networks must use a Supported Host Physical Point Attachment (PPA) network device. For a list of supported AVIO PPAs, see the *HP-UX vPars and Integrity VM V6.1 Release Notes* at http://www.hp.com/go/hpux-hpvm-docs.

NOTE: When using AVIO networking devices for vPar/VMs that are configured as Serviceguard Packages, be sure that all Serviceguard standby `lans` are configured using PPA devices supported by AVIO. Failure to do so causes network connectivity to be lost even if the standby link is up.

2.8 VSP kernel tunables

Upon installation, tunables are set to the following values:

- maxdsiz_64bit – 34359738368 (4294967296)
- filecache_min – 134217728 or 1% (1707212800)
- filecache_max – 134217728 or 1% (17072390144)
- lockable_mem_pct – 99% (90%)
- base_pagesize — 64 (4)
- vx_ninode — 131072 (0)
- vxfs_ifree_timelag — -1 (0)
- vxfs_bc_bufhwm — 64000 (0)

The first value is the modified value and the number in parenthesis is the default .

3 Installation procedures for HP-UX vPars and Integrity VM V6.1

This chapter describes the requirements and procedures for installing vPars and Integrity VM V6.1. The topics include:

- Installing vPars and Integrity VM V6.1
- Bundle names
- Verifying the installation
- Removing vPars and Integrity VM V6.1

3.1 Installing vPars and Integrity VM V6.1

The following sections provide the installation procedures for vPars and Integrity VM V6.1.

3.1.1 vPars V6.1 installation requirements

vPars V6.1 is installed on the VSP after installing the BB068AA and VirtualBase bundles on the VSP from a distribution media. Table 5 (page 35) lists the software products that must be installed on the VSP and on vPars V6.1.

Table 5 Software required to be installed on VSP and vPar OS

VSP	vPar OS
HP-UX 11i v3 March 2012 (AR1203) (Base OE, VSE-OE, or DC-OE)	HP-UX 11i v3 September 2011 (AR1109) or later (any OE) on HP Integrity Server Blade system BL8x0c i2 servers HP-UX 11i v3 March 2012 (AR1203) HP Integrity Superdome 2 servers
BB068AA, VirtualBase	VirtualBase
HP-UX GUID Manager	HP-UX GUID Manager (optional)
Following Serviceguard patches (applicable only if Serviceguard is installed): • PHSS_42136 • PHSS_42137	Following Serviceguard patches (applicable only if Serviceguard is installed): • PHSS_42136 • PHSS_42137

NOTE: The vPars V6.1 bundle BB068AA does not include the vPars V6.1 GUI (Integrity Virtual Server Manager) software. The Integrity Virtual Server Manager is a separately installed ISU bundle. Integrity Virtual Server Manager can be downloaded from the HP Software Depot at http://www.software.hp.com.

Table 6 VxVM and System Fault Management

VxVM	System Fault Management
If using VxVM 5.0.1, install PHKL_42649, PHCO_42677and PHCO_42648 patches.	Update System Fault Management software to: • ProviderSvcsBase C.06.00.05.02 Provider Services Base • SysFaultMgmt C.07.04.07.03 HPUX System Fault Management

NOTE: Both ProviderSvcsBase and SysFaultMgmt are included in the WBEM Management bundle — https://h20392.www2.hp.com/portal/swdepot/displayProductInfo.do?productNumber=WBEMMgmtBundle

3.1.2 Installing vPars V6.1

vPars V6.1 may be packaged in a software depot and installed in the VSP using the `swinstall` command. The minimum OE version for the VSP is HP-UX 11i v3 March 2012 (AR1203). The minimum OE version for a vPar is HP-UX 11i v3 September 2011 (AR1109) on HP Integrity Server Blade system BL8x0c i2 servers and HP-UX 11i v3 March 2012 (AR1203) on HP Integrity Superdome 2 servers. After vPars are created using either the GUI or the CLI, the deployment of HP-UX OEs can be performed by an Ignite-UX server using specific entries in the `/etc/bootptab` file or using a Direct Boot Profile (dbprofile) entry in the EFI of the vPar.

To install HP-UX Virtual Partitions V6.1 software:

1. Install HP-UX 1203 (Base OE, VSE-OE, or DC-OE) and the required patches on the VSP. See the vPars V6.1 release notes for the required patches. Go to the next step after HP-UX boots.
2. Install Virtual Partition Software products BB068AA and VirtualBase on the VSP either with the OE (it is included with VSE-OE and DC-OE) or as a separate install from the application media:

   ```
   swinstall -x autoreboot=true -s <VirtualPartitionDepot> \
   BB068AA VirtualBase
   ```

   Ensure that the HP-UX GUID Manager is already installed. If not, you can install it with BB068AA and VirtualBase.
3. Allow HP-UX to boot.
4. Create one or more virtual switches using the `vparnet` command.
5. Create virtual partitions on the VSP after HP-UX boots. Use the `vparcreate` and `vparmodify` commands to create and modify vPars on the VSP. For information about configuring NPIV, see the "Configuring an NPIV HBA (vHBA)" (page 55) section.

   You need to specify the number of CPUs, the memory size, data disks, and storage.

   NOTE: In vPars V6.1, a boot disk is created when you install HP-UX on the vPar. The boot disk is specifically defined during installation when the EFI NVRAM is updated with the hardware path of the installation disk. This process occurs automatically after the partition OS is installed. Hence, there is no need to specify a boot disk when configuring virtual partitions in vPars V6.1.

   ---
6. Install HP-UX 1109, or later, on the vPar.

   After the storage is added to the vPar, HP-UX can be installed from an Ignite server using `lanboot` or from an install media such as DVD or ISO image. Any OE can be installed on a vPar.

   #### Installing HP-UX OE in a vPar from an Ignite Server:

   Start the vPar from VSP using the `vparboot` command. Connect to the virtual console instance by using the `vparconsole` command and wait for the boot to reach the EFI shell prompt, and then use `lanboot` to install HP-UX from an Ignite Server. For more information, see "Installing HP-UX 11i on a vPar using an Ignite-UX server" (page 69).

   #### Installation of HP-UX OE in a vPar from a boot media (DVD or ISO image file):

   Add the media as a virtual DVD to the vPar using the `vparmodify` command. Boot the vPar to EFI and select virtual disk to boot HP-UX. See the vPar Commands Manual to know how to specify and add a virtual DVD to a vPars V6.1.

7. If the vPar is using HP-UX 11i v3 September 2011, install VirtualBase on the vPar from the DVD:

```
swinstall -x autoreboot=true -s <DVD dsf> VirtualBase
```

NOTE: VirtualBase is installed by default with HP-UX 11i v3 March 2012.

3.1.3 Integrity VM installation requirements

To prepare your VSP system for Integrity VM installation, your configuration must satisfy the hardware, software, and network requirements described in this section.

Before you install this product, read the *HP-UX vPars and Integrity VM V6.1 Release Notes*, which are available on the product media. The most up-to-date release notes are available on http://www.hp.com/go/hpux-hpvm-docs.

You must install Integrity VM Version 6.1 software on a system that is running HP-UX 11i v3. You can install Integrity VM on a hard partition (nPar) running HP-UX, but do not attempt to install Integrity VM on a virtual partition (vPar). Integrity VM cannot be installed on a system that has HP-UX Virtual Partitions (vPars) software installed. There is a check during the Integrity VM installation that prevents this occurrence. If you override the Integrity VM installation warnings and force this installation, you receive errors during the start of Integrity VM.

The resources on the VSP system (such as disks, network bandwidth, memory, and processing power), are shared by the VSP and all the running guests. Guests running simultaneously share the remaining memory and processing power. By default, network devices are also sharable among guests. Some resources must be made exclusive to the VSP, such as the VSP operating system boot disk.

Table 7 describes the minimum configuration requirements for installing Integrity VM on the VSP system.

Table 7 Requirements for installing Integrity VM V6.1

Resource	Description
Computer	An Integrity server
Operating system	HP-UX 11i v3 March 2012 running on an Integrity server, as well as any appropriate software patches (see the *HP-UX vPars and Integrity VM V6.1 Release Notes*). The license for Integrity VM includes the license for running the HP-UX 11i v3 Base Operating Environment (BOE) on the VSP system.
Local area network (LAN) card	Required for network connection and configuration.
Source installation media	An appropriate source for installing software (DVD or network connection).
Disk storage	Sufficient disk space for the following: • The VSP operating system (see the *HP-UX 11i v3 Installation and Update Guide*) • The VSP software (50 MB) • Disk space for each guest operating system, including swap space **NOTE:** The VSP requires swap space only as recommended by HP-UX 11i v3. Otherwise, no swap space is required for Integrity VM. • Disk space for the applications running on each guest • 4.7 MB for each running guest as the allowance for backing up configuration files For information about configuring storage devices for guests, see Chapter 9 (page 107).

Table 7 Requirements for installing Integrity VM V6.1 *(continued)*

Resource	Description
Memory	For information about memory overhead estimation, see "VSP memory overhead estimation" (page 31).
Network configuration	A configured and operational network. To allow guests network access, the VSP must have at least one functioning network interface card (NIC). For more information about configuring network devices for virtual machines, see Chapter 10 (page 143).

NOTE: The VSP overhead is valid for VSPs with base_pagesize=64.

3.1.4 Installing Integrity VM V6.1

This section describes the installation procedure to use if you have never installed Integrity VM on your system. If you are updating your HP-UX 11i v2 operating system to HP-UX 11i v3, and subsequently want to update Integrity VM V3.5 to Integrity VM 6.1, see Section 4.1 (page 43). If you are updating Integrity VM V4.0 or later to Integrity VM V6.1, see Section 4.2.

Once you have read the product release notes and verified that you have met the system requirements, install the vPars and Integrity VM software as described in this section.

NOTE: Installing the vPars and Integrity VM software requires the system to reboot. Therefore, the `swinstall` command line installation includes the `autoreboot=true` parameter.

To install the HP-UX vPars and Integrity VM software, follow these steps:

1. If you have the installation media, mount it.

   If you are installing from the network, identify the VSP and path name that correspond to the software distribution depot that contains the BB068AA and VirtualBase bundles (for example, `my.server.foo.com:/depot/path`).

2. Use the `swinstall` command to install Integrity VM and specify the path to the depot. For example, the following command installs Integrity VM:

   ```
   # swinstall -x autoreboot=true -s my.server.foo.com:/depot/path BB068AA VirtualBase
   ```

   If you are using the GUI (`swinstall -i`), perform the following steps:

   a. Enter the following commands:

      ```
      # export DISPLAY=your display variable
      # swinstall
      ```

   b. Select the vPars and Integrity VM V6.1 bundle, BB068AA, and the VirtualBase bundle from the list presented by the GUI.

   The VSP and guest configuration files are stored at `/var/opt/hpvm`. The new configuration files are not compatible with those of previous versions of Integrity VM. Therefore, if you are upgrading to the current version, the guest configuration files (except the `/ISO-Images/` directory) are saved to the `/var/opt/hpvm/backups` directory. If you revert to the older version of Integrity VM, you can use the backup configuration files to restore your VSP and guest configurations.

3. Unmount and remove any installation media. The VSP system automatically reboots, if necessary.
4. Once the Integrity VM software is installed and running, the VSP is available. Enter the following command to get information about the status of the guests:

   ```
   # hpvmstatus
   hpvmstatus: No guest information is available.
   hpvmstatus: Unable to continue.
   ```

The installation is now complete, with the following results:

- The VirtualBase software is installed in the `/opt/hpvm/guest-images` directory.
- The Integrity VM software and data files are installed in the `/var/opt/hpvm` directory.
- The Integrity VM commands are installed in the `/opt/hpvm/bin` directory.

You can now create virtual machines using the `hpvmcreate` command, as described in Chapter 7 (page 75).

Integrity VM installation modifies certain kernel parameters. For this reason, you should not install any other applications besides Integrity VM on the VSP system, regardless of whether guests are running or not. The system management applications should be installed on the VSP, like SMH and Glance. The kernel parameters that are modified when you install Integrity VM are:

- maxdsiz_64bit – 34359738368 (4294967296)
- filecache_min – 134217728 or 1% (1707212800)
- filecache_max – 134217728 or 1% (17072390144)
- lockable_mem_pct – 99% (90%)
- base_pagesize — 64 (4)
- vx_ninode — 131072 (0)
- vxfs_ifree_timelag — -1 (0)
- vxfs_bc_bufhwm — 64000 (0)

The first value is the modified value and the number in parenthesis is the default.

NOTE: With vPars and Integritry VM V6.1, the VSP has VSP has 64K as the base_pagesize tunable setting. For more details, see the white paper on this topic, *Tunable Base Page Size* available from:

http://www.hp.com/go/hpux-core-docs/

3.2 Bundle names

The HP-UX vPars and Integrity VM V6.1 release contains the following software:

- `BB068AA` — B.06.10.00 vPars and Integrity VM
- `VirtualBase` — Base virtualization software for vPar/VM and VSP.
- `HP-UX 11i V3` — HP-UX 11i v3 Virtual Server Operating Environment [VSE-OE] that is provided for the VSP system
- `PRM-Sw Krn` — Installed as part of the HP-UX VSE-OE and required by Integrity VM

3.3 Inhibitors to installing HP-UX vPars and Integrity VM

The following items block HP-UX vPars and Integrity VM Version 6.1 from installing:

- Hierarchical Files System (HFS) files in the `/etc/fstab` file. You must remove any entries before installing Integrity VM. Check for these entries with the following command:

  ```
  # grep -i hfs /etc/fstab
  ```

- HP System Insight Manager (HP SIM) Server bundle. Check for this product with the following command:

  ```
  # swlist | grep HPSIM-HP-UX
  ```

- HP-UX Virtual Partitions bundle, v5.0 or earlier. Check for this product with the following command:

  ```
  # swlist -l bundle | grep VirtualPartition
  ```

3.4 Verifying the installation of vPars and Integrity VM

To verify that Integrity VM installed successfully, enter the following command:

```
# hpvminfo
hpvminfo: Running on an HPVSP.
```

To see exactly what versions of specific bundles are installed, enter the `swlist` command and look for the following bundles:

```
BB068AA B.06.10 HP-UX vPars & Integrity VM v6

GUIDMGR A.01.00.579 HP-UX GUID Manager

VirtualBase B.06.10 Base Virtualization Software
```

(Specific version numbers on your installation may not match the examples in this manual.)

When you install Integrity VM, the file `/etc/rc.config.d/hpvmconf` is created to record the product configuration.

3.5 Removing vPars or Integrity VM

To remove the vPars and Integrity VM product, you must remove the following software bundles:

- BB068AA
- VirtualBase

```
# swremove -x autoreboot=true  BB068AA VirtualBase
```

To remove the trial version of Integrity VM, T2801AA, enter the following command:

```
# swremove -x autoreboot=true T2801AA
# rm  -rf /opt/hpvm
```

vPars and VMs are not affected by this procedure. To remove vPars/VMs, see the procedures in "Removing a virtual partition" (page 67) and Section 7.7 (page 99).

3.6 Reserving VSP devices

You can protect the storage and network resources used by the VSP against usage and corruption by virtual machines by marking the VSP devices as restricted devices. For example, you can reserve the disk storage on which the VSP operating system and swap space reside, which prevents guests from being able to access the same disk storage devices. The `hpvmdevmgmt` command allows you to establish restricted devices.

For example, to restrict the `/dev/rdisk/disk1` device, enter the following command:

```
# hpvmdevmgmt -a rdev:/dev/rdisk/disk1
```

To complete the restriction of volumes, each device included in the volume must also be restricted. For more information about using the `hpvmdevmgmt` command, see Section 11.11 (page 189).

3.7 Troubleshooting installation problems

If the installation verification succeed with warnings, report the problem using the procedures described in Chapter 13 (page 215). Some problems encountered in the process of installing Integrity VM are described in the following sections.

3.7.1 Error messages during installation

One or more of the following messages might be displayed during Integrity VM installation:

```
could not write monParams: Device is busy

hpvmnet * already exists

/sbin/init.d/hpvm start ran without running /sbin/init.d/hpvm stop
```

You can ignore these messages.

3.7.2 Warning during startup if HP SIM is installed

Integrity VM generates a warning during startup if HP SIM is installed, but Integrity VM continues to start. The HP SIM product consumes memory and CPU resources that would otherwise be available for running virtual machines. The amount of memory and CPU consumed varies depending on the specific system configuration. On systems that are already tight on memory or CPU resources, this might cause virtual machines to fail to start.

4 Upgrading the VSP from previous versions of Integrity VM

This chapter describes how to upgrade the VSP from an older version. The chapter topics include:

- Upgrading the VSP from previous version of Integrity VM V3.x to V6.1
- Updating previous versions of the VSP

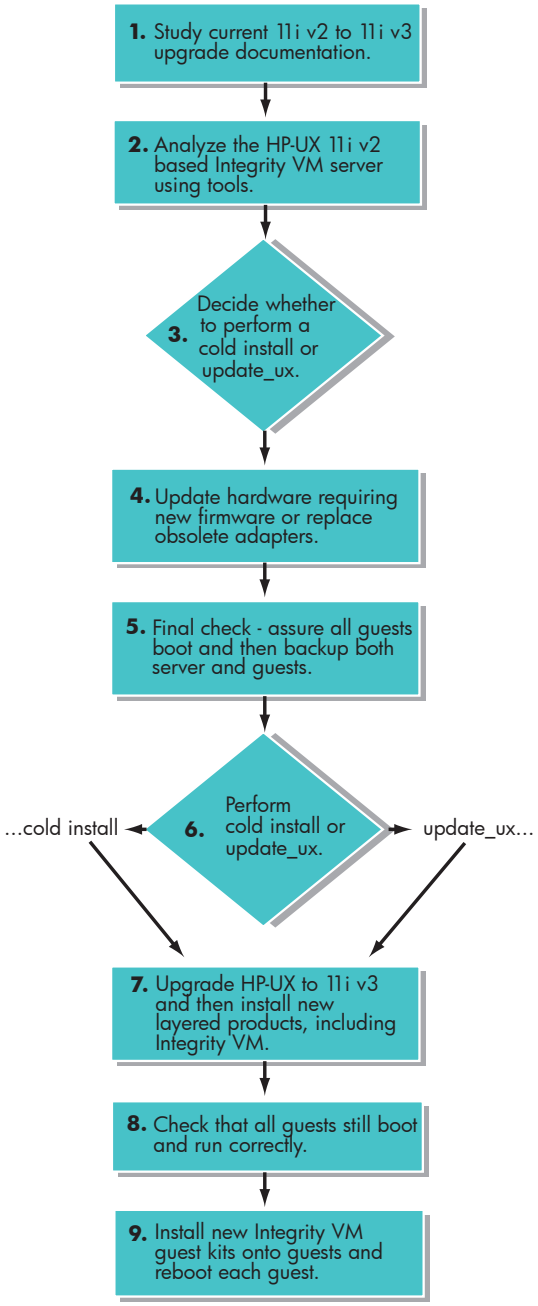
4.1 Upgrading the VSP from previous versions of Integrity VM V3.x to Integrity VM V6.1

The Integrity VM Version 6.1 VSP requires the HP-UX 11i v3 operating system. Only HP-UX 11i v2 servers running Integrity VM Version 3.0 or Version 3.5 can be upgraded to the HP-UX 11i v3 Integrity VM Version 6.1 release. This section describes the process to follow when upgrading an HP-UX 11i v2 based Integrity VM server to an HP-UX 11i v3 based Integrity VM server. If you are upgrading the VSP from Integrity VM V4.0 or later to Integrity VM V6.1, see Section 4.2.

HP-UX 11i v3 supports many features that are backward compatible with 11i v2, allowing 11i v2 applications to run without modifications. The primary goal of this section is to provide direction to the administrator performing the upgrade of the VSP to make sure that all configured virtual machines (guests) boot and run after completing the upgrade to 11i v3.

Figure 3 provides a flowchart of the upgrade procedure from 11i v2 to 11i v3.

Figure 3 Upgrade Procedure



NOTE: If you plan to perform a cold install, follow these steps:

1. Offline migrate all virtual machines to another VSP, for example, system X (which has shared access to the storage).
2. Remove all virtual machines using the `hpvmremove` command.
3. Back up the `/var/opt/hpvm` directory.
4. Perform the cold install of HP-UX only, restarting the system. (Step 6 in the flowchart)
5. Restore the `/var/opt/hpvm` directory.
6. Install the Integrity VM. (Step 7 in the flowchart)
7. Offline migrate the virtual machines from system X.
8. Start the virtual machines. (Step 9 in the flowchart)

The first thing the administrator must do is to identify subsystems on the 11i v2 Integrity VM server that are incompatible with or that are not supported on 11i v3. Some incompatibility issues can be exposed by tools, and others are found in referenced documents. The most common update problems are caused by the following:

- Unsupported hardware adapters or firmware
- Memory and system disk space requirements (HP-UX 11i v3 has increased both of these.)
- Obsolete or unsupported storage multipath solutions
- Layered products requiring an 11i v3 compatible version

4.1.1 Study the current HP-UX 11i v2 to HP-UX 11i v3 update documentation

The first stage of upgrading an Integrity VM V3.0 or V3.5 server to an Integrity VM V4.x or V6.1 server is to review the following HP–UX 11i v3 operating system update documents:

- HP-UX 11i v2 to 11i v3 Mass Storage Stack Update Guide
- Read Before Installing or Upgrading
- HP-UX 11i v3 Installation and Update Guide
- HP-UX 11i Version 3 Release Notes
- Serviceguard Specific Documentation

The following websites provide a general reference covering the features and hardware supported in HP-UX 11i v3. Read these documents and become familiar with the information before beginning the upgrade procedure.

- QuickSpecs for HP-UX 11i v3 Update 2 features and operating environments
- Upgrading to HP-UX 11i v3

As you are reading, pay particular attention to the new mass storage model, called the agile device reference model, for naming and identifying devices. The 11i v2 model is called the legacy device reference model. The new agile device model uses worldwide device identifiers (WWIDs) to identify devices. The WWID is a device attribute that is independent of the device's location in a SAN or in an adapter/controller access path. Therefore, the agile device names are persistent with respect to changes in the access path and can utilize multiple paths through a single device name.

The legacy devices require multiple device names to access the same device through multiple paths. Many Integrity VM customers use multipath solutions such as Secure Path, which allow them to use a single device name to access all paths. Some of these 11i v2 multipath solutions will continue to work, while others you must remove. The general solution for this particular problem is to replace the existing multipath device with the new agile device name, with its inherent multipath support, once the upgrade has completed.

NOTE: Dynamic Root Disk (DRD), an HP-UX system administration toolset, is available to clone an HP-UX system image to an inactive disk for software maintenance or recovery. The bundle name is `DynRootDisk` and the product name is DRD. Administrators can use DRD to reduce downtime for system maintenance by creating an inactive clone of the booted system, then applying patches and products to the clone. The modified clone can then be booted at a convenient time. DRD is available for download from a software depot. For information about HP-UX Dynamic Root Disk, see HP-UX 11i v3 Documentation.

4.1.2 Analyze HP-UX 11i v2 based Integrity VM server

Analyzing HP-UX 11i v2 based Integrity VM server is the most important stage of the Integrity VM server upgrade. During this analysis, it is important to discover any incompatible hardware and software subsystems. You can use the HP-UX 11i v2 to 11i v3 Mass Storage Check Utility (`msv2v3check`) to assist in the analysis.

The `msv2v3check` tool is free software provided on the http://software.hp.com website. Go to this website, search for msv2v3check, and download this free tool.

These analysis utilities are aimed primarily at mass storage problems and problems that are specific to existing virtual machines. In most cases, you can take actions to resolve these incompatibilities before doing the upgrade, such as loading new firmware. Other solutions might require waiting until after the upgrade, such as substituting agile devices for an 11i v2 multipath solution. Another area of particular concern is the layered products running on your 11i v2 based Integrity VM server. Analyze each layered product to determine its upgrade impact:

- No change - Layered product is compatible.
- Delete/reinstall - Layered product requires a new version to work on 11i v3.
- Delay upgrade – Layered product needs a new version that has not yet released.

For more information, see the HP-UX 11i v3 documentation on the BSC website at: HP-UX 11i v3 Manuals.

4.1.2.1 Run the HP-UX msv2v3check tool

The HP-UX `msv2v3check` command reviews all mass storage controllers and devices on your system for HP-UX 11i v3 compatibility and support. In addition, `msv2v3check` attempts to verify that your system meets other 11i v3 system requirements, particularly the minimum memory required and supported platforms. For more information, see the `getconf` (1M) and `model` (1M) HP-UX commands.

The `msv2v3check` command looks at only mass storage controllers (host bus adapters) and devices for HP-UX 11i v3 compatibility and support. This includes the following:

- Ultra0 SCSI (C8xx) host bus adapters and attached HP supported SCSI devices
- Ultra320 SCSI (MPT) host bus adapters and attached HP supported SCSI devices
- Serial Attached SCSI (SAS) host bus adapters and attached HP supported SAS devices
- Smart Array RAID (CISS) host bus adapters and attached HP supported RAID devices
- Fibre Channel (FCD/TD) host bus adapters and attached HP supported Fibre Channel devices
- HP supported SCSI disk enclosures and arrays
- HP supported Fibre Channel disk enclosures and arrays

The `msv2v3check` command creates the following log file in the `/var/adm/msv2v3check/` directory:

```
/var/adm/msv2v3check/mmddyy_hhmm is the full log file that contains all
notes, warnings, and error messages from an invocation of msv2v3check,
where mmddyy_hhmm represents the month, day, year, hours and minutes
at the time the msv2v3check utility was started.
```

Once the `msv2v3check` utility has completed, a validation result is displayed that indicates the number of errors and warnings detected on your system configuration:

- An error is a critical message that indicates that your system does not support HP-UX 11i v3 in its current configuration. Do not ignore this message.
- A warning indicates a task that might require user action, for example, upgrading the firmware on a disk device, or manually reviewing the firmware of a Fibre Channel disk array.

Review all warnings and make the necessary corrections before upgrading to HP-UX 11i v3.

For supported I/O drivers, devices, adapters; see the documentation on the BSC website:HP Manuals.

4.1.2.2 Determine HP-UX 11i v3 memory and system disk requirements

Integrity VM V6.1 memory requirements vary depending on the number and size of virtual machines supported by the Integrity VM server. When upgrading from an 11i v2 Integrity VM server, use the following steps to determine the amount of memory required for the 11i v3 Integrity VM server:

1. When your 11i v2 Integrity VM server is running at peak load, use the Integrity VM `hpvmstatus -s` command to display the available memory.
2. If the available memory is less than 1 GB, then it is highly likely that your server requires additional memory to run the same load with 11i v3 and Integrity VM V6.1. Before upgrading, add the appropriate amount of memory to ensure that there is at least 1 GB of memory available during peak loading.

NOTE: Different operating environments have different minimum memory requirements

4.1.2.3 Determine version requirements for HP-UX OE and vPars and Integrity VM

Of the HP-UX 11i v2 servers running Integrity VM, only those running Version 3.0 or Version 3.5 can be upgraded to the HP-UX 11i v3 Integrity VM V6.1 release. HP recommends that all virtual machines (guests) be upgraded with Integrity VM Version 6.1 guest kits to take advantage of performance enhancements and bug fixes. Guests that booted and ran on the 11i v2 Integrity VM server will continue to function with equivalent or improved performance after the upgrade.

Existing guest configuration information, operating system software, and application data are not affected when you upgrade Integrity VM. However, when you upgrade, also reinstall the guest kit that is provided with Integrity VM. This operation requires you to reboot the guest. For more information, see Section 8.3 (page 104).

When upgrading from a prior release, support for specific guests types or backing stores may have been removed. For support information, see the *HP-UX vPars and Integrity VM V6.1 Release Notes* at http://www.hp.com/go/hpux-hpvm-docs.

The following are the HP-UX OEs:

- HP-UX 11i v3 Base OE (BOE)

  The BOE provides and integrated HP-UX operating environment for customers who require less complex installations. The Base OE includes the entire original Foundation Operating Environment (FOE), and offers complete HP-UX functionality including security, networking, web functionality, and software management applications.

- HP-UX 11i v3 Virtual Server OE (VSE-OE)

  The VSE-OE provides an integrated HP-UX operating environment for customers who seeking higher resource utilization or who are embarking on consolidation projects and need virtualization for a flexibile UNIX environment. The VSE-OE contains all the products included in the BOE (and the original EOE) and adds a host of other products including the entire VSE suite. The VSE-OE includes HP-UX vPars and Integrity VM (BB068AA) and the VirtualBase bundle.

- HP-UX 11i v3 Data Center OE (DC-OE)

  Business-critical virtualization builtin —The Data Center OE is the offering for customers who are consolidating, or building an infrastructure for the future. Because the powerful software within the DC-OE is integrated and tested with the operating system, it is an effective choice for a highly available virtualized environment. DC-OE is a complete, fully tested, and integrated

UNIX offering. The DC-OE includes HP-UX vPars and Integrity VM (BB068AA) and the VirtualBase bundle.

- HP-UX 11i High Availability OE (HA-OE)

  For customers requiring continuous access to data and applications, the HA-OE delivers the protection of Serviceguard and related software. The HA-OE also delivers all the software in the Base OE plus what has shipped until now in the Enterprise OE.

Table 8 lists the HP-UX 11i v2 to HP-UX 11i v3 supported OE server upgrades.

Table 8 Supported Operating Environments

Original 11i v2 Operating Environments	New 11i v3 Operating Environments
Foundation OE	Base OE
Technical Computing OE	Base OE
Enterprise OE	Virtual Server OE
Mission Critical OE	Data Center OE

NOTE: Many software subsystems require upgrades on the 11i v2 Integrity VM server before updating to HP–UX 11i v3. The most obvious of these is that Integrity VM must be upgraded to V3.0 or V3.5 before beginning the HP-UX upgrade. Other layered products, such as Serviceguard, require version upgrades before updating the operating system to 11i v3. Analyze each layered product for required upgrades.

Remove HP Integrity Virtual Machines Manager Version 3.0 or earlier before upgrading to Integrity VM Version 6.1. After installing Integrity VM V6.1, install Integrity Virtual Server Manager V6.1 and HP-UX GUID Manager V1.0. If you are upgrading an Integrity VSP from 11i v2 to 11i v3 and are using Veritas file systems and volumes, update to Veritas V5.0 and become familiar with the *Veritas 5.0 Installation Guide*.

4.1.3 Decide whether to perform a cold-install or an update

The preferred method for upgrading an HP-UX 11i v2 based Integrity VSP to an 11i v3 based VSP is to use the `Update-UX` program. The `update-ux` command takes as input the new 11i v3 OE depot. The `update-ux` command strives to maintain all your current user, storage, and network configurations. There are some 11i v2 multipath solutions that are not compatible with 11i v3. This same set of multipath solutions must be dealt with whichever method you choose. In most cases, the multipath conversion is to use the agile devices on 11i v3 in place of the device names that the multipath solutions invented. The `Update-Ux` program also strives to keep volume definitions the same. This is helpful because a cold-install most likely changes all the device names requiring a mapping of devices to volumes and to guests.

One reason to choose a cold-install over an `update-ux` update is the ease by which you can immediately return to the 11i v2 environment. The `update-ux` path changes the original 11i v2 system configuration making a restore from backups the only way to return to the original 11i v2 system. The cold-install can and should be given separate disks to use allowing the original 11i v2 system disks to remain unchanged. Because the original disks can remain unchanged, there is less of a need to back up the 11i v2 based Integrity VSP.

NOTE: HP recommends a full back up of both the Integrity VSP and guests before updating.

Whether an `update-ux` or a cold-install upgrade is chosen, the administrator needs to study the documentation that covers the differences between HP-UX 11i v2 and HP-UX 11i v3. To obtain input on potential upgrade problems, the administrator should also run the HP-UX `msv2v3check` tool.

If you have installed the evaluation version of Integrity VM (software bundle `T2801AA`), remove the evaluation software before installing the Integrity VM product (see Section 3.5 (page 40)).

4.1.4 Perform required hardware and firmware upgrades

Perform all hardware and firmware upgrades that are supported on 11i v2 and that are needed for 11i v3 while still running on 11i v2. This allows the administrator to verify that all guests are fully functional with the changes before upgrading to 11i v3. Read the following information:

- Hardware Specific Information
- System Firmware Support Matrix
- Ethernet System Driver Support Matrix
- HP 9000 and HP Integrity Server Connectivity/

4.1.5 Perform either a cold-install or an update

If the cold-install upgrade path is chosen, the administrator is taking the responsibility for fully configuring the 11i v3 Integrity VSP to be functionally equivalent to the 11i v2 Integrity VSP configuration. Integrity VM V6.1 provides the `hpvmdevtranslate` utility to assist in mapping the legacy devices used by guests on the 11i v2 VSP to the new 11i v3 agile devices.

The `hpvmdevtranslate` utility produces the script `/var/opt/hpvm/common/hpvm_ev_convert`. This script needs to be reviewed and edited before running it to make the conversions. Device conversions that cannot be made are listed as comments labeled `ERROR:`. The administrator is responsible for determining the conversion of the ERROR lines. The `hpvmdevtranslate` utility translates only devices that provide unique world wide identifiers (WWIDs).

After evaluating your 11i v2 Integrity VSP and performing appropriate backups, use the following steps with the `hpvmdevtranslate` utility as part of a cold-install:

1. Choose the system disks that are to be used for the 11i v3 VSP and mark them as reserved disks.

   ```
   # hpvmdevmgmt -a rdev:device_name
   ```

2. Back up and collect all relevant configuration from the 11i v2 VSP.
3. Back up the `/var/opt/hpvm` directory, so that you can easily restore it to the 11i v3 system after the cold-install.

   NOTE: DRD can be used to clone an HP-UX system image to an inactive disk for recovery. For information about DRD, see the Dynamic Root Disk documentation on the BSC website at: Business Support Center Manuals.

4. Verify that all current guests that run on 11i v2 can boot and run successfully. Guests that cannot boot on 11i v2 cannot be expected to boot after the upgrade to 11i v3.
5. After verifying the guests, back up all relevant configuration data for each guest for a potential return to 11i v2.
6. Shut down the Integrity VM guests gracefully by logging into each one and shutting it down.
7. Shut down the Integrity VSP.
8. Using the HP-UX cold-install procedure, install the appropriate 11i v3 OE using the selected system disks. For information about performing a cold-install, see the *HP-UX 11i v3 Installation and Update Guide*.
9. Remove any blocking layered products that might block the Integrity VM installation. See Section 3.3 (page 39) for a list products.
10. Remove layered products that might cause problems or that require a new 11i v3 compatible version after the HP-UX 11i v3 upgrade.

11. Determine the order of installation of layered products, including vPars and Integrity VM V6.1 (BB068AA), so that all dependencies are met. For example, if Veritas is used to provide backing storage for guests, install it before Integrity VM.
12. Install all 11i v3 compatible layered products that are required for equivalent functionality to the 11i v2 VSP.
13. Install Integrity VM Version 6.1 to the 11i v3 VSP.

NOTE: See Section 3.3 (page 39) for a list of products that block the Integrity VM installation.

---

14. Stop Integrity VM using `/sbin/init.d/hpvm stop`.
15. Using the appropriate recovery tool, restore the 11i v2 `/var/opt/hpvm` directory over the existing 11i v3 `/var/opt/hpvm` directory on the 11i v3 VSP.
16. Start Integrity VM using `/sbin/init.d/hpvm start`.
17. Run the translator:

    ```
    # hpvmdevtranslate -a /var/opt/hpvm/common/hpvm_mgmtdb_pre1131
    ```

18. Edit the script, `/var/opt/hpvm/common/hpvm_dev_convert`, taking note of ERROR lines and commenting out the exit line that prevents the running of the script.
19. Continue with the remaining 11i v3 Integrity VSP configuration until the host is functionally equivalent to the former 11i v2 Integrity VSP.

If you choose the update path, follow these steps:

1. Create a recovery image.
2. Verify that all current guests that run on 11i v2 can boot and run successfully. Guests that cannot boot on 11i v2 cannot be expected to boot after the update to 11i v3.
3. After verifying the guests, back up all relevant configuration data for each guest for a potential return to 11i v2.
4. Install the latest `Update-UX` bundle from the OE media.
5. Update the OS/OE from the HP-UX 11i v3 OE media using the `update-ux` command. For example:

   ```
   # swinstall -s /dev/dvd Update-Ux
   update-ux -s /dev/dvd/HPUX11i-VSE-OE BB068AA
   ```

   NOTE: There is a new `update-ux` option, `-p`, which previews and update task by running the session through the analysis phase first.

   If you are updating from the VSE-OE depot, specify the following:

   ```
   # swinstall -s my.server.foo.com:/OEdepot/path Update-UX
   update-ux -s my.server.foo.com:/OEdepot/path HPUX11i-VSE-OE BB068AA
   ```

6. Remove any blocking layered products that might block the Integrity VM installation. See Section 3.3 (page 39) for a list products.
7. Remove layered products that might cause problems or that require a new 11i v3 compatible version after the HP-UX 11i v3 update.
8. Determine the order of installation of layered products, including vPars and Integrity VM V6.1 (BB068AA), so that all dependencies are met. For example, if VERITAS is used to provide backing storage for guests, install it before Integrity VM.
9. Install Integrity VM Version 6.1 to the 11i v3 VSP.
10. Update non-OE applications from the Application media using the `swinstall` command.

    For example, if you plan to install Integrity Virtual Server Manager, switch to the AR disk and specify the following:

    ```
    # swinstall -s my.server.foo.com:/Ardepot/path VMMGR
    ```

11. Create the recovery image.

4.1.6 Verifying vPars/VM after installing layered products

Follow these steps after installing layered products:

1. Start and stop each guest, one at a time, and make sure that they boot to their OS.
2. Use the guest troubleshooting section, Chapter 13 (page 215), to resolve guest booting problems.
3. Upgrade each guest with the new guest kit.
4. Make sure there are no network issues
5. If the guest OS is no longer supported, upgrade the guest OS.

NOTE: When Integrity VM is stopped either with the `/sbin/init.d/hpvm stop` command or as a result of removing or updating the version of Integrity VM on the VSP, messages of the following form might be logged in the `/var/opt/hpvm/common/command.log` file:

```
ERROR|host|root|Unable to communicate with the FSS agent
```

The messages, which are a result of interactions with the performance metrics processes `scopeux` and `perfd`, are normally transient and stop after about a minute. Approximately 60-70 messages might be generated in that time. You can clear this condition by either rebooting the VSP or by stopping and restarting the metrics collection processes.

To stop and restart the `perfd` process, use the following commands:

```
# /sbin/init.d/pctl stop
# /sbin/init.d/pctl start
```

To stop and restart the `scopeux` process, use the following commands:

```
# /sbin/init.d/ovpa stop
# /sbin/init.d/ovpa start
```

4.1.7 Upgrade troubleshooting issues

After you upgrade to 11i v3, examine the following issues:

- Mass storage issues

  The Integrity VM V6.1 release supports the use of both legacy and agile devices in guests. It is not necessary to convert guests to use strictly agile devices. If, however, problems occur with guests using multipath solutions that are based on legacy devices, change the backing device to use the equivalent agile device. For information about mass storage compatibility issues, see the documentation on the BSC website:

  HP-UX 11i v3 Manuals.

- Platform issues

  For 11i v3 platform support, see the following matrix:

  HP-UX Integrity Server Support Matrix

- Serviceguard issues

  For information about the Storage Multi-Pathing choices in HP-UX Serviceguard environments, see the Serviceguard website:

  HP Serviceguard Solutions

- Other issues

4.2 Updating previous versions of the VSP to Integrity VM V6.1

This section describes the process of updating a previous version of the VSP to Integrity VM V6.1.

Once you have read the product release notes and verified that you have met the system requirements, install the Integrity VM software as described in this section.

NOTE: Installing the vPars and Integrity VM software requires the system to reboot. Therefore, the `swinstall` command line installation includes the `autoreboot=true` parameter.

To install the HP Integrity VM software, follow these steps:

1. If you have the installation media, mount it.

   If you are installing from the network, identify the VSP and path name that correspond to the software distribution depot that contains the BB068AA bundle (for example, `my.server.foo.com:/depot/path`).

2. Use the `swinstall` command to install Integrity VM and specify the path to the depot. For example:

   ```
   # swinstall -x autoreboot=true -s my.server.foo.com:/depot/path BB068AA VirtualBase
   ```

   Install the HP-UX GUID Manager with the BB068AA and VirtualBase bundles.

   If you are using the GUI (`swinstall -i`), perform the following steps:

   a. Enter the following commands:

      ```
      # export DISPLAY=your display variable
      # swinstall
      ```

   b. Select the HP-UX vPars and Integrity VM bundle (BB068AA) from the list presented by the GUI.

   The VSP and guest configuration files are stored at `/var/opt/hpvm`. The new configuration files are not compatible with those of previous versions of Integrity VM. Therefore, if you are upgrading to the current version, the guest configuration files (except the `/ISO-Images/` and /backup directories) are saved to the `/var/opt/hpvm/backups` directory. If you revert to the older version of Integrity VM, you can use the backup configuration files to restore your VSP and guest configurations.

3. Unmount and remove any installation media. The VSP system automatically reboots, if necessary.
4. Once the Integrity VM software is installed and running, the VSP is available. Enter the following command to get information about the status of the guests:

   ```
   # hpvmstatus
   hpvmstatus: No guest information is available.
   hpvmstatus: Unable to continue.
   ```

5 Using NPIV with vPars and Integrity VM

NPIV allows creation of multiple virtual Fibre Channel ports (VFCs) over one physical port (PFC) on a Virtualization Services Platform (VSP). The VSP serves as the management platform for the vPars and virtual machines. Each of these virtual ports should be created with a unique World Wide Name (WWN) to identify it, just like the unique embedded WWN by which a physical port is identified. The NPIV feature is about creating such virtual ports over a physical port on the VSP and then allocating them as resources to the virtual environments (vPars and VMs). This means that the resource that is added to the vPar or VM is a virtual Host Bus Adapter or virtual HBA (vHBA). The VM or vPar then discovers targets and LUNs behind the vHBA using the same mechanism that is used on a standalone system to discover targets and LUNs behind a physical HBA. As in the case of a standalone system, a vPar or VM using NPIV will automatically discover new targets and LUNs behind the vHBA. With NPIV, VMs and vPars can support two kinds of device – legacy shared I/O using AVIO (AVIO LUNs), and the LUNs seen with the vHBA (NPIV HBAs). Unlike legacy shared storage, the NPIV LUNs need not be visible by the VSP and therefore, the LUNs that the vPar or VM will see behind the vHBA can be managed and provisioned the same way as on a standalone system. Note that NPIV devices can co-exist with legacy AVIO devices in the same vPar or VM.

5.1 NPIV can satisfy these requirements

The following requirements can be met using the NPIV feature:

- Storage isolation between vPar/VM and the VSP, and among vPars.
- Isolation of I/O traffic across vPars/VMs and VSP.
- Running of applications that require un-virtualized device access on the vPar.
- Monitoring the server/storage environment using charge back applications.
- Streamlining vPar and VM migrations.

5.2 Dependencies and prerequisites

The NPIV functionality requires a hardware I/O stack which explicitly supports NPIV, from the HBAs, through the interconnect modules and SAN fabric and on to the individual Fibre Channel devices. The supported HBAs include, but are not be limited to:

- Qlogic QMH2562 8Gb FB HBA for HP Blade System C-Class (451871-B21)
- HP single-port 8Gb PCIe Fibre Channel HBA (AH400A)
- HP dual-port 8Gb PCIe Fibre Channel HBA (AH401A)

Supported I/O Interconnect Modules include but may not be limited to:

- HP BLc Virtual Connect 8Gb 24-Port Fibre Channel Module for c-Class BladeSystem (466482-B21)
- HP BLc Virtual Connect 8Gb 20-Port Fibre Channel Module for c-Class BladeSystem (572018-B21)
- HP BLc 4Gb Virtual Connect Fibre Channel Module for c-Class BladeSystem (409513-B22)

For the SAN fabric, refer to your vendor's documentation to determine support. HP supports NPIV with its 4 GB and 8 GB SAN switches.

5.3 NPIV – supported limits

Table 9 (page 54) lists the supported limits associated with NPIV in vPars and Integrity VM V6.1.

Table 9 NPIV supported limits in vPars and Integrity VM V6.1

Limit description	Supported limit
NPIV HBAs per vPar/VM	8
Number of NPIV HBAs per physical HBA	16
Number of paths supported per NPIV device	8
Number of LUNs per NPIV HBA	2048
Number of NPIV devices per vPar/VM	2048

NOTE: In configurations where multiple NPIV HBAs created from a single physical HBA are used by different vPars/VMs, all the I/O from these vPars/VMs share a single physical HBA, which could lead to performance bottlenecks in high I/O scenarios.

For a more balanced performance, HP recommends that you spread NPIV HBAs for vPars/VMs across multiple physical adapters.

5.3.1 Verifying that the VSP can support NPIV

Before creating a vHBA, check the physical HBAs on the system to verify that they support NPIV. The `fcmsutil` command can do this check. Run this command on a VSP Fibre Channel HBA:

`/opt/fcms/bin/fcmsutil /dev/fcdX`

where X is the instance number of the Fibre Channel HBA and can be obtained from the `ioscan -kfnC fc` command:

```
# ioscan -kfnC fc
Class     I  H/W Path     Driver S/W State   H/W Type     Description
===================================================================
fc        0  0/2/0/0/0/0  fcd   CLAIMED     INTERFACE    HP AH401A 8Gb Dual Port PCIe Fibre Channel Adapter (FC
 Port 1)
   /dev/fcd0
fc        1  0/2/0/0/0/1  fcd   CLAIMED     INTERFACE    HP AH401A 8Gb Dual Port PCIe Fibre Channel Adapter (FC
 Port 2)
   /dev/fcd1
```

The following command shows you whether NPIV is supported on the VSP:

```
# /opt/fcms/bin/fcmsutil /dev/fcd0
Vendor ID is = 0x1077
                           Device ID is = 0x2532
             PCI Sub-system Vendor ID is = 0x103C
                    PCI Sub-system ID is = 0x3263
                               PCI Mode = PCI Express x8
                      ISP Code version = 5.4.0
                      ISP Chip version = 2
                              Topology = PTTOPT_FABRIC
                            Link Speed = 4Gb
                        Local N_Port_id is = 0x010800
                     Previous N_Port_id is = None
                N_Port Node World Wide Name = 0x5001438002344785
                N_Port Port World Wide Name = 0x5001438002344784
                Switch Port World Wide Name = 0x200800051e0351f4
                Switch Node World Wide Name = 0x100000051e0351f4
                  N_Port Symbolic Port Name = porti3_fcd0
                  N_Port Symbolic Node Name = porti3_HP-UX_B.11.31
                              Driver state = ONLINE
                            Hardware Path is = 0/2/0/0/0/0
                         Maximum Frame Size = 2048
           Driver-Firmware Dump Available = NO
           Driver-Firmware Dump Timestamp = N/A
                                     TYPE = PFC
                           NPIV Supported = YES
                           Driver Version = @(#) fcd B.11.31.1103 Aug  2 2011
```

If NPIV is supported, running the command again with the new `npiv_info` option provides information about all of the running virtual HBAs currently associated with this physical HBA:

```
# /opt/fcms/bin/fcmsutil /dev/fcd0 npiv_info
PFC Hardware Path               = 0/0/0/5/0/0/0
PFC DSF                         = /dev/fcd0
PFC Class Instance              = 0
PFC Driver state                = ONLINE
PFC Port WWN                    = 0x5001438001459910
PFC Node WWN                    = 0x5001438001459911
PFC Switch Port WWN             = 0x201400051ef06bd3
PFC Switch Node WWN             = 0x100000051ef06bd3

FlexFC Virtual Fibre Channel (VFC)
----------------------------------
Maximum Supported FlexFC VFC    = 16
Number Active FlexFC VFC        = 0

HPVM Virtual Fibre Channel (VFC)
--------------------------------
Maximum Supported HPVM VFC      = 16
Number Active HPVM VFC          = 1

The following provides the list of VFC(s) associated with this PFC:

Type                            = HPVM VFC
VFC Index                       = 17
VFC Guest ID                    = 0x4
VFC Port WWN                    = 0x50014c2000000007
VFC Node WWN                    = 0x50014c2800000023
VFC Driver state                = ONLINE
VFC DSF                         = /dev/fcd6
VFC Class Instance              = 6
```

5.3.2 WWNs – Finding and using WWNs

You can allocate and manage unique WWNs for NPIV HBAs using the HP-UX GUID Manager which is a client-server based product. Using this application ensures that you do not have to perform these tasks manually.

IMPORTANT: HP recommends using the HP-UX GUID Manager to allocate and maintain WWNs and vWWNs. For more information about HP-UX GUID Manager, see the *HP-UX GUID Manager Administrator Guide* available on the HP website at http://www.hp.com/go/hpux-hpvm-docs.

5.3.3 Configuring an NPIV HBA (vHBA)

The overall configuration process for NPIV HBA is the same as with AVIO. For configuring NPIV HBAs, a new 'storage' type called NPIV is introduced.

The following sections describe how an NPIV HBA resource is specified and how the existing vPar commands are used to configure and manage them.

5.3.3.1 Specifying an NPIV HBA resource

An NPIV resource is specified using the following format:

`devicetype:adaptertype:bus,device,vWWP,vWWN:storage:device`

where:

devicetype	The virtual device type as seen in the vPar. For NPIV this will be **`hba`**.
adaptertype	The adapter type as seen in the vPar. For NPIV, the adaptor type is **`avio_stor`**.
bus	The PCI bus number for the virtual device and can range from 0 to 7.
device	The PCI slot number for the virtual device and can range from 0 to 7.

vWWP	A valid (64 bit), unique (virtual) Port WWN that is assigned to the NPIV HBA. This is analogous to the unique Port WWN that is associated with physical HBAs.
vWWN	A valid (64 bit), unique (virtual) Node WWN that is assigned to the NPIV HBA. This is analogous to the unique Node WWN that is associated with physical HBAs.
storage	The physical storage type in the host. For NPIV, this is **`npiv`**.
device	The physical device in the host corresponding to the virtual device. For NPIV this corresponds to the device special file for the physical port on which the virtual NPIV instance is created.

Example 1 Create an NPIV HBA using the GUID server for WWNs

This resource specification creates an NPIV HBA over the physical HBA corresponding to the FCD instance `/dev/fcd0`, and lets the VSP select the bus and device numbers.

```
hba:avio_stor::npiv:/dev/fcd0
hba:avio_stor:,,,:npiv:/dev/fcd0
```

NOTE: HP recommends using the GUID Manager application to obtain the WWNs.

It is optional to use the commas. The comma is an alternate way to get WWNs from the GUID server. If you use the commas, ensure that you specify 3 commas.

Example 2 Create an NPIV HBA manually specifying WWNs

This resource specification creates an NPIV HBA over the physical HBA corresponding to the FCD instance `/dev/fcd0`, using a port WWN of 0x50060b00006499a0 and a node WWN of 0x50060b00006499a8.

```
hba:avio_stor:1,1,0x50060b00006499a0,0x50060b00006499a8:npiv:/dev/fcd0
```

In the resource string, you can skip the bus and slot numbers for an NPIV HBA; VSP picks the next available bus and slot number for the NPIV HBA. The virtual node WWN and port WWN cannot be skipped.

5.3.3.2 vPar commands in NPIV context

The following sections describe the vPar commands used in the NPIV context.

To add an NPIV resource – vparcreate and vparmodify

You can specify an NPIV HBA resource using `vparcreate` while creating the vPar or using the `vparmodify` command after the vPar is created. For information about the resource string format used to specify the NPIV HBA, see the "Specifying an NPIV HBA resource" (page 55) section.

IMPORTANT: Before creating an NPIV HBA, ensure that the physical HBAs on the system support NPIV. You can use the `fcmsutil` command to determine whether NPIV is supported. An alternate way to determine which card supports NPIV is to add the product number of the card that currently supports NPIV or point them to quickspec if it is available for vPars V6.1.

Example 3 Finding out if WWNs are supported

```
# ioscan -kfNC fc
Class     I  H/W Path     Driver S/W State   H/W Type     Description
=============================================================================
fc        0  0/2/0/0/0/0  fcd   CLAIMED     INTERFACE    HP AH401A 8Gb Dual Port PCIe Fibre Channel Adapter (FC
 Port 1)
fc        1  0/2/0/0/0/1  fcd   CLAIMED     INTERFACE    HP AH401A 8Gb Dual Port PCIe Fibre Channel Adapter (FC
 Port 2)

# fcmsutil /dev/fcd0

                          Vendor ID is = 0x1077
                          Device ID is = 0x2532
            PCI Sub-system Vendor ID is = 0x103C
                   PCI Sub-system ID is = 0x3263
                               PCI Mode = PCI Express x8
                      ISP Code version = 5.4.0
                      ISP Chip version = 2
                              Topology = PTTOPT_FABRIC
                            Link Speed = 4Gb
                  Local N_Port_id is = 0x010800
                Previous N_Port_id is = None
            N_Port Node World Wide Name = 0x5001438002344785
            N_Port Port World Wide Name = 0x5001438002344784
            Switch Port World Wide Name = 0x200800051e0351f4
            Switch Node World Wide Name = 0x100000051e0351f4
              N_Port Symbolic Port Name = pinki3_fcd0
              N_Port Symbolic Node Name = pinki3_HP-UX_B.11.31
                          Driver state = ONLINE
                    Hardware Path is = 0/2/0/0/0/0
                  Maximum Frame Size = 2048
         Driver-Firmware Dump Available = NO
         Driver-Firmware Dump Timestamp = N/A
                                  TYPE = PFC
                       NPIV Supported = YES
                       Driver Version = @(#) fcd B.11.31.1103 Aug  2 2010
```

Create a vPar named vPar1 with 4 virtual CPUs and an NPIV HBA created on `/dev/fcd0` using the GUID server to assign port and node WWNs. You can also use the `vparstatus -a` command to display NPIV capable fiber channel devices.

```
vparcreate -P vPar1 -c 4 -a hba:avio_stor::npiv:/dev/fcd0
```

Add an NPIV HBA created on `/dev/fcd1` using a virtual port WWN of 0x50060b00006499b9 and virtual node WWN of 0x50060b00006499ba to the vPar named vPar1. Obtain the port and node WWNs from your storage administrator or other source.

```
vparmodify -P vPar1 -a hba:avio_stor:,,0x50060b00006499b9,
0x50060b00006499ba:npiv:/dev/fcd1
```

NOTE: The `vparmodify` command cannot be used to change any attribute of the NPIV HBA after it is created.

To view – vparstatus

The `vparstatus` command output includes the NPIV HBA in the I/O details for vPars that have the NPIV HBA configured. All other `vparstatus` output remains unchanged.

To delete a NPIV resource – vparmodify

The `vparmodify` command is used to delete an NPIV HBA from a vPar. For the deletion to take effect, you must stop and start the vPar.

Example 4 Deleting an NPIV resource

```
vparmodify -P vPar1 -d hba:avio_stor:,,0x50060b00006499b9,
0x50060b00006499ba:npiv:/dev/fcd1
```

For the relevant vPar, you can use this syntax by copying it from the I/O details of the `vparstatus` command output and pasting it where required.

For more information about NPIV, see the white paper, *HP-UX vPars 6.0 & 6.1 and Integrity VM 4.3 & 6.1 N_Port ID Virtualization (NPIV)*, on the BSC website.

6 Creating virtual partitions

To create virtual partitions, you must run appropriate commands from the VSP or use the HP-UX Integrity Virtual Server Manager, the GUI application that can be accessed from the **Tools** page in HP SMH of the VSP.

This chapter discusses the various tasks that can be performed from the VSP using commands. For tasks you can perform using the GUI, see the *HP-UX Integrity Virtual Server Manager Help* that comes with the GUI application.

△ **CAUTION:** In V6.1, HP supports a vPar only or VM only environment, though creation of one type of virtual server when the other type already exists might be allowed by Integrity VM commands in some cases. You are strongly advised not to attempt creation of mixed vPar/VM configurations. A configuration of mixed vPars and VMs is not supported and might lead to unexpected behavior.

NOTE: In V6.1, Integrity VM commands also support configuration and management of vPars. You can choose Integrity VM commands or vPar commands based on your familiarity with the type of command.

Using NPIV HBAs generates virtual WWNs, and administrators are responsible for tracking WWNs and guaranteeing their uniqueness across the Storage Area Network (SAN). The GUIDMgr provides a mechanism to manage unique WWNs, so that the same name is not re-allocated until freed, to avoid potential data corruption when Logical Storage is presented to multiple vPars by mistake. The GUIDMgr helps accomplish this task.

GUIDMgr is integrated with vPars V6.1 to support NPIV, and it is also integrated with HP Integrity Virtual Server Manager to support managing the database.

6.1 Creating a virtual partition

When you create a virtual partition, you must specify its attributes. Later you can change these characteristics.

You can set the attributes of a virtual partition using the following commands:

- `vparcreate`, which creates new virtual partitions.
- `vparmodify`, which modifies the existing virtual partitions.

Both these commands accept the same options for specifying the attributes of a virtual partition. Table 10 (page 60) describes each characteristic and command option.

NOTE: When you use the `vparcreate` command to create a vPar, by default it reserves any resources assigned to that vPar, even when the vPar is off. For more information about reserved resources, see "Reserved resources and resource over-commitment" (page 68). Additionally, the vPar is set to AutoBoot when the VSP is restarted. You can use the `hpvmmodify -B` command to adjust the AutoBoot setting.

Table 10 Attributes of a virtual partition in vPars V6.1

vPar attribute	Description	Command option	Default value
vPar ID (name or number)	You can specify either a number or name.	`-p vpar_id`	If you do not specify either a number or a name, a vPar name in the format vParXXXX (where XXXX represents the vPar Id number), with leading zeros is automatically assigned to the newly created vPar.

vPar name	The vPar name can have up to 256 alphanumeric characters, including A-Z, a-z, 0-9, the dash (—), the underscore (_), and period (.). The vPar name must not start with a dash.	`-P vpar_name`
		OR
		`-p vpar_id`
			If you do not specify either a number or name, a vPar name in the format vParXXXX (where XXXX represents the vPar Id number), with leading zeros is automatically assigned to the newly created vPar.

CPU	You can specify the number of CPUs that a vPar can use. A running vPar cannot use more CPUs than the number of physical CPUs on the VSP system.
	You can set min and max values too. The minimum and maximum values are boundary values that are enforced if the number of CPUs in this vPar changes in the future.
		`-a cpu::num`
		`-a cpu::[num]:[min][:[max]]`
		OR
		`-a core::num`
		`-a core::[num]:[min][:[max]]`
			If you leave out this option when you create a vPar, the default is 1 CPU core. If you set any of num, min, or max to 0, the default value is assigned. For vPars V6.1, the defaults are, num=1, min=1, and max=512.

Memory	The memory is specified in megabytes. The minimum amount of memory you allocate to a vPar must be the total of the following: • The amount of memory required by the operating environment in the vPar. • The amount of memory required by the applications running on the vPar.	`-a mem::mem_size`	If you leave out this option when you create a vPar, the default memory allocated is 2 GB.

Table 10 Attributes of a virtual partition in vPars V6.1 *(continued)*

vPar attribute	Description	Command option	Default value
I/O (virtual devices)	You can allocate virtual network switches and virtual storage devices to the virtual partition. The VSP presents devices to the virtual partition as virtual devices. The vPar network consists of vNICs and vswitches. For virtual partitions to communicate either with other virtual partitions or outside the VSP system, each virtual partition's virtual network must be associated with a vswitch. If you start a virtual machine without a vswitch, the virtual machine has no network communication channel. Virtual storage devices are backed by physical devices on the VSP system. You can specify one of the following devices – disk, dvd, tape, changer, burner, or hba.	`-a rsrc`	If you leave out this option when you create a vPar, the vPar will not have access to network and storage devices.
Virtual iLO Remote Console	You can access the Virtual iLO Remote Console of the virtual partition using telnet or ssh. This attribute is the IP address that is used to connect to the Virtual iLO Remote Console of the vPar. The address must be specified in IPv4 dot-decimal notation. If the `-K` option is specified, then the `-L` option too must be specified. By default, the root user may access the console of the vPar using the `vparconsole` command or through the Virtual iLO Remote Console, if configured. There is no need to configure a console account if the root user for this purpose does not violate security policy. However, access to the console, through the `vparconsole` command or the remote console, can be granted to groups or individual users, with either administrative or operator virtual iLO permissions.	`-K console_ip`	If you leave out this option when you create a vPar, the remote console is not started, that is, the virtual console can be accessed only using the `vparconsole` command.
IPv4 subnet mask for accessing the Virtual iLO Remote Console	To access the Virtual iLO Remote Console of the vPar if you have specified its IP address using the `-K` option, then you must specify the IPv4 subnet mask for accessing the Virtual iLO Remote Console for the vPar.	`-L netmask`	Not applicable.

Table 10 Attributes of a virtual partition in vPars V6.1 *(continued)*

vPar attribute	Description	Command option	Default value
Group with administrator or operator privileges	If you omit this option, group accounts will not have admin or oper privileges.	`-g group:{admin|oper}`	If you do not specify the group authorization, then only the root user has access to the virtual console.
User with administrator or operator privileges	If you omit this option, no user accounts, other than root, have admin or oper privileges.	`-u user:{admin|oper}`	If you do not specify the user authorization, then only the root user has access to the virtual console.

Example 5 Create a default virtual partition

Run the `vparcreate` command to create a basic virtual partition with the default values of 1 CPU, 2 GB memory, and no I/O. Later, use the `vparmodify` command to add I/O and modify other attributes.

```
# vparcreate
[Creating vPar0001.]
# vparmodify -a network:avio_lan::vswitch:sitelan \
-a hba:avio_stor::npiv:/dev/fcd0
```

Example 6 Create the virtual partition called Oslo

Create the vPar named Oslo in the local system, specifying 1 GB of memory, 2 CPUs, and virtual I/O resources.

```
vparcreate -p Oslo -a mem::1024 -a cpu::2 \
-a disk:avio_stor::disk:/dev/rdisk/disk3 \
-a network:avio_lan::vswitch:sitelan
```

Example 7 Create a virtual partition with vHBA

Create a virtual partition named vpar001 with a virtual disk backed by a whole disk, a virtual network interface backed by virtual switch sitelan, and a virtualized HBA using NPIV port assuming a GUID manager is available to assign World Wide Port Name and World Wide Node Name.

```
# vparcreate -p vpar001 -a disk:avio_stor::disk:/dev/rdisk/disk70 \
-a network:avio_lan::vswitch:sitelan -a hba:avio_stor::npiv:/dev/fcd0
# vparstatus -p vpar001 -v
[Virtual Partition Details]
          Number:    1
          Name:      daytona1
          RunState:  UP
          State:     Active

          [CPU Details]
          Min/Max:                 1/127
          System assigned [Count]: 1

          [Memory Details]
          Total Memory(MB):        2048

          [OL* Details]
          Operation : CPU change
          CPU Count: 1
          Status: PASS

          [Authorized Administrators]
          Oper Groups:
          Admin Groups:
          Oper Users:
          Admin Users:

          [Remote Console]
          Remote Console Ip Address:      10.92.101.70
          Remote Console Net Mask:        255.255.252.0

          [IO Details]
                  disk:avio_stor:0,0,0:disk:/dev/rdisk/disk70
                  network:avio_lan:0,1,0x9605006e53ec:vswitch:sitelan:portid:9
                  hba:avio_stor:0,2,0x50014C200000009E,0x50014C280000002D:npiv:/dev/fcd0
```

In this example output, the last I/O resource is an NPIV device at hardware address bus 0 device 2. It uses World Wide Port Name (0x50014C200000009E) and World Wide Node Name (0x50014C280000002D). The World Wide Port Name can be used by the storage administrator to present LUNs to the virtual partition.

For additional information about configuring NPIV, see the *vparresources3*(5) manpage and the relevant NPIV sections in this guide.

6.1.1 Specifying CPU/Core min and max limits

The syntax to specify min and max CPUs assigned to a virtual partition is:

`-[a|m] cpu::[num]:[min][:[max]]`

where:

`-a` add (used with `vparcreate` or `vparmodify`)

`-m` modify (used with `vparmodify`)

`min` the minimum number of CPUs for the virtual partition and the minimum number of CPUs that must remain assigned to the partition

`max` the maximum number of CPUs that can be assigned to the virtual partition

NOTE: The virtual partition can be either UP or DOWN when setting the min or max value. Hence, a reboot is not necessary when you modify the min and max value. When the partition is UP, the CPU count can only be adjusted if the HP-UX OS on the vPar is running. CPU counts cannot be adjusted while the OS on the vPar is in EFI state.

Example 8 Setting the minimum number of CPUs to 2

```
machinename1# vparmodify -p machinename2 -m cpu:::2
```

Example 9 Setting the minimum number of CPUs to 2 and the maximum to 4

```
machinename1# vparmodify -p machinename2 -m cpu:::2:4
```

6.1.2 Adding and deleting CPUs/Cores by total

The basic syntax for adding and deleting CPUs is:

`-[a|d|m] cpu::`*`num`*

where:

`-a|d|m` specifies adding, deleting, or modifying the *`total`* count of CPUs.

`num` specifies the number of CPUs.

NOTE: The virtual partition can be either UP or DOWN when using the cpu::num syntax.

When the virtual partition is active, CPUs that were added using the cpu::num syntax can be deleted only by using `cpu::`*`num`* syntax.

The total increases or decreases by *`num`* when the `-a` or `-d` option is used and is set to *`num`* when the `-m` option is used.

vPars does not support assignment of resources based on hardware path or socket locality.

6.2 Booting a virtual partition

You can boot and manage vPars using the same storage media and procedures that you would if the vPar operating system were running on its own dedicated physical hardware platform. You can allocate administration privileges to specific virtual partition administrators.

The vPar must be in the DOWN run state to be able to be booted. To boot the vPar, you must run the `vparboot` command or provide the `-c "pc –on"` parameters to `vparconsole`.

vPars provides individual consoles for each vPar. You can access the console from the VSP using the `vparconsole` command. Start the console before you run the `vparboot` command in case there is need to interact with EFI. You can also provide the `-f -i -c "pc –on"` parameters to

the `vparconsole` command to start, and enter the console in interactive mode right after the start.

Example 10 Boot the virtual partition called Oslo

```
vparboot -p Oslo
```

OR

```
vparconsole -P Oslo -fic 'pc -on'
```

6.3 Modifying a virtual partition

You can modify all the attributes that you specify while creating a vPar. You can rename the vPar, modify the resources, and change group and user level authorization. Some attributes can be modified dynamically, that is, a reboot is not required, while others require a reboot.

Just as the `vparcreate` command, the `vparmodify` command must be run from the VSP.

The same options used for creating a virtual partition are applicable for modifying the virtual partition.

Modifying CPU resources – dynamically

In vPars V6.1, you can modify only the CPU cores dynamically. You can change the CPU core count of a vPar while it is running. You need not reboot the vPar after you modify the CPU core count.

Modifying memory and I/O resources – statically

To modify the memory size and the I/O devices of a vPar, the vPar must be in the DOWN run state. After you modify the memory and I/O, reboot the vPar. As an exception, an individual storage device can be added to the virtual partition, as long as a free target exists on an existing virtual partition device. If all 128 target addresses on a device are full, and a new device must be added, the virtual partition must be in the DOWN state to add that device.

Modifying vPar name and number

The vPar must be in the DOWN run state to modify the name. You can modify the name of a virtual partition using `vparmodify -P` to add a name that does not exist in the current virtual partition database. The vPar number cannot be modified. The only way you can get a different number is to destroy the current vPar configuration and recreate it. When it is recreated you can specify the vPar number with the `-p` option.

Example 11 Modify a virtual partition with vHBA

Add a virtualized hba, using NPIV ports (manually assigning port and node WWNs)

```
# vparmodify -p Oslo \
-a hba:avio_stor:,,0x100000110a030000,0x100000110a030001:npiv:/dev/fcd0
```

NOTE: If you want to create a vPar that has access to the same storage as a different vPar or the VSP, or re-create a vPar configuration, giving it the same storage access, you can manually assign the port and node worldwide names, as in the example.

6.4 Viewing information specific to a virtual partition

You can view information about a vPar by specifying either the name or the number of the vPar. You must use the `vparstatus` command from the VSP to view vPar information.

By default, the `vparstatus` command displays summary information. To view detailed information you must use the `-v` option. You can also view the vPar information in machine-readable format using the `-M` option.

The information that the `vparstatus` command (and `hpvmstatus` command) displays includes the following:

- Number and name of the vPar.
- State of the vPar – active or inactive.
- Run-state of the active or inactive vPar.
- Summary of CPU, I/O, and memory resource allocations.
  In both summary and detailed machine-readable format, the following information for the specified vPar is displayed:
  - Total memory size in MB.
  - The number of CPUs assigned to the virtual partition.
  - The virtual I/O devices assigned to the virtual partition in the resource statement format.

To view summary information about all the vPars, run the following command:

```
# vparstatus
```

To view the detailed attributes and resources of a specific virtual partition named vpar1, run the following command:

```
# vparstatus -p vpar1 -v
```

To view the detailed attributes and resources of a specific virtual partition named vpar1 in machine-readable format, run the following command:

```
# vparstatus -p vpar1 -M
```

To view the revisions of partition management tools, run the following command:

```
# vparstatus -r
```

6.5 Resetting a virtual partition

When a virtual partition is unresponsive, instead of shutting down the vPar, you can reset or restart the vPar.

△ **CAUTION:** When the `vparreset` command is used accidentally, serious consequences can occur. Hence, the `-f` (force) option is required with the command.

You can perform any of the different reset operations described here.

Hard reset

The hard reset is equivalent to specifying `RS` command in the management processor. You should only do a hard reset if you cannot get the OS to issue its own reboot or shutdown process. The virtual partition will be restarted after the hard reset.

To hard reset a vPar named Oslo, run the following command:

```
vparreset -f -p Oslo -h
```

Power off

The power off option `-d` is useful to break out of a reboot loop, that is, when you do not want the vPar to be rebooted. In such a case, you must manually restart the vPar using the `vparboot` command.

To power off a vPar named Oslo, run the following command:

```
vparreset -f -p Oslo -d
```

IMPORTANT: In the case of both hard reset and power-off, the operating system of the vPar is abruptly shut down and the crash dump of the OS is not saved. HP recommends shutting down the vPar from the vPar using the `shutdown` command.

Soft reset (transfer of control - TOC)

When you do not specify any option with the `vparreset` command, a soft reset is performed by default. In a soft reset, the crash dump of the OS running on the vPar is saved. This enables the HP engineers to debug the problem that caused the unresponsiveness. The `-t` option is used for a soft reset. The virtual partition is restarted after the soft reset is issued.

To soft reset a vPar named Oslo, run the following command:

```
vparreset -f -p Oslo
```

OR

```
vparreset -f -p Oslo -t
```

Shutdown

When a vPar must be completely shut down and not be rebooted, the `-g` option can be used. You can issue a graceful shut down to the OS by using the `vparreset` command. The HP-UX OS responds to a graceful shutdown request when you perform the `reboot -h` operation.

To shut down a vPar named Oslo, run the following command:

```
vparreset -f -p Oslo -g
```

NOTE: With portable image support in HP-UX 11i v3 March 2012, this constraint does not exist any more. See the `kctune gio_portable_image` in the HP-UX Portable Image documentation on the BSC website.

6.6 Removing a virtual partition

When you want to permanently delete a virtual partition, you can use the `vparremove` command. The vPar must be in the DOWN run-state before you can delete the vPar. To bring a vPar to the DOWN run-state, you can either power down (`vparreset` command with `-d` option) the vPar or shutdown the vPar (`vparreset` command with `-g` option).

NOTE: The preferred method for stopping a vPar is to log in to it, stop all the applications, and then issue the `/etc/shutdown -h` command.

CAUTION: When the `vparremove` command is used accidently, serious consequences can occur.

Hence, the `-f` (force) option is required with the command.

To remove a vPar named Oslo, run the following command:

```
vparremove -p Oslo -f
```

6.6.1 Removing and recreating a virtual partition

If you remove a vPar configuration, and recreate it using the `vparcreate` command, the newly created vPar might have not have the same hardware paths for network and storage devices. This could change the LAN instance number. In such a case, you must update the `netconf` file with the new instance number. When the LAN instance number is incorrect, the network is inaccessible and startup scripts will hang until they timeout.

NOTE: To modify `netconf`, you might have to mount all the file systems using the `mountall` command to be able to access the editor that is required to modify the `netconf` file. To avoid the long boot time, boot to single user mode and modify the `netconf` file with the new LAN instance number.

NOTE: With portable image support in HP-UX 11i v3 March 2012, this constraint does not exist any more. See the usage of `kctune gio_portable_image` in the Portable Image documentation on the BSC website,

6.7 Reserved resources and resource over-commitment

HP-UX vPars and Integrity VM V6.1 allows the reservation of resources for virtual machines and virtual partitions. Reservations imply that a resource will be available when it is needed, with the intention of assuring that a virtual machine or virtual partition can boot at any time. The reserved resources setting is managed for each individual virtual machine and virtual partition and is set using the `resources_reserved` attribute (managed with the `-x` option of the `hpvmcreate` and the `hpvmmodify` command). The default behavior of the `vparcreate` command is to set `resources_reserved` to true when a virtual partition is created. However the `hpvmcreate` command does not reserve resources by default when creating virtual machines or virtual partitions. The `resources_reserved` attribute can be managed using the `hpvmmodify` command.

Resources that are reserved include memory, CPU and I/O devices. If a resource is assigned to a virtual machine or virtual partition that has the `resources_reserved=true`, that same resource cannot be assigned to a different virtual machine or virtual partition that also has `resources_reserved=true`. It is also not possible to assign a resource to a virtual partition or virtual machine that has `resources_reserved=true`, if that resource is not currently available. For example, if all the CPUs have been assigned to other reserving virtual machines or virtual partitions, then it is not possible to assign CPUs to any additional reserving virtual machines or virtual partitions. It is possible to assign resources to non-reserving virtual machines and virtual partitions, however, it is not possible to boot them (because the resources assigned to that virtual machine or virtual partition are reserved by other virtual machines or virtual partitions).

6.7.1 Resource over-commitment

In some circumstances, it is possible for resources to be over-committed, meaning that more resources are assigned to reserving virtual machines and virtual partitions than are currently available. This can occur if CPU, memory or other I/O devices are lost, such as removed during a VSP shutdown or from a hardware failure. If all the resources of the VSP are assigned to virtual partitions or virtual machines, and then later some of those resources are lost, the VSP will be in an over-committed state. When this occurs, it is not possible to boot all of the reserving virtual machines or virtual partitions. Virtual machines and virtual partitions with reserving resources will be allowed to boot on a first-come first-served basis. And virtual machines and virtual partitions without resource reservations will not be allowed to boot.

If CPU or memory resources are overcommitted, this state is logged in the `/var/opt/hpvm/common/command.log` file. Additionally the overcommitted state is noted when displaying system resources using the `hpvmstatus -s` command. For example:

```
# hpvmstatus -s
[HPVM Server System Resources]

        *** VSP resources are over-committed ***
        vPar/VM types supported by this VSP = Shared
        Processor speed = 1596 Mhz
        Total physical memory = 16278 Mbytes
        Total number of operable system cores = 8
        CPU cores allocated for VSP = 1
        CPU cores allocated for vPars and VMs = 7
        CPU cores currently in use or reserved for later use = 0
```

```
Available VSP memory = 884 Mbytes
Available swap space = 7032 Mbytes
Total memory allocated for vPars and VMs = 12544 Mbytes
Memory in use by vPars and VMs = 13568 Mbytes
Available memory for vPars or VMs is overdrawn by 1024 Mbytes
Available memory for 0 (max avail.) CPU VM = N/A Mbytes
Available memory for 0 (max avail.) CPU vPar = N/A Mbytes
Maximum vcpus for an HP-UX virtual machine = 7
Maximum vcpus for an OpenVMS virtual machine = 7
Maximum available vcpus for a VM is overdrawn by 1
Available CPU cores for virtual partitions are overdrawn by 1
```

When CPU or memory resources are overdrawn, either these resources must be brought back online, or reservations on existing virtual machines or virtual partitions must be reduced or removed. Resource reservations can be removed using the `hpvmmodify -P name -x resources_reserved=false` command. Memory reservations can be reduced by using the `hpvmmodify -r` option to reduce virtual machine or virtual partition memory size. CPU reservations can be reduced by using either the `-c` or `-e` options to reduce the CPU count or CPU entitlement. Note `-e` does not apply to a virtual partition, because it always has 100% entitlement to a CPU. Use the information provided by `hpvmstatus -s` to determine the amount of overcommitted resource, and then use `hpvmmodify` to reduce that resource's commitments appropriately.

When virtual machines are configured to use reserved resources, and CPUs are overcommitted, it might require a combination of reducing both CPU entitlement, and CPU counts to bring the system back into a committed state. When determining the appropriate amount of entitlement or CPU count to reduce, Integrity VM packs CPU entitlement as efficiently as possible. For example, if there are two virtual machines that require 50% entitlement of three CPUs, and there are seven CPUs available in total on the VSP, that would mean that four CPUs would be available at 100%, if needed (however, those two virtual machines might spread across 6 CPUs if those CPUs are not otherwise in use.)

While it is not possible to increase resources on a reserving virtual machine or virtual partition when the VSP is in an over-committed state, it is possible to reduce resource commitments, even if that reduction still leaves the VSP in an over-committed state.

It is still possible to change resource assignments of virtual machines and virtual partitions when the VSP is in an overcommitted state, if those virtual machines or virtual partitions do not reserve resources. However it is not possible to set the `resources_reserved=true`.

6.8 Installing the HP-UX Operating Environment on a vPar

For HP Integrity Server Blade BL8x0c i2 servers, you can install HP-UX 11i v3 September 2011 (AR1109) or later on the vPar. For HP Integrity Superdome 2 servers, you must install the HP-UX 11i v3 March 2012 (AR1203) or later release. The server must be an HP Integrity Server Blade system BL8x0c i2 server or newer machine.

There are multiple ways to install HP-UX 11i on a virtual partition. HP-UX 11i installation on a vPar using an Ignite-UX server is described here.

6.8.1 Installing HP-UX 11i on a vPar using an Ignite-UX server

This approach describes the use of the network to directly install HP-UX 11i from an Ignite-UX server. For more information about Ignite-UX based installation, see the *Ignite-UX Administration Guide* for HP-UX 11i available on the website at: http://www.hp.com/go/ignite-ux-docs.

NOTE: Before using the Ignite-UX server approach, ensure the following:

- The vPar is created and assigned a network interface.
- At least one disk has been added to the vPar with sufficient space to install HP-UX 11i on it.
- The Ignite-UX server is set up and accessible from the LAN interface assigned to the vPar.

To install HP-UX 11i on a vPar using an Ignite-UX server:

1. Boot the vPar to EFI.
2. Create a dbprofile from EFI shell by running the following command:

   ```
   Shell> dbprofile –dn mybootprofile –sip <IP_address_of_ignite-server>
   -cip <IP_address_of_vPar> -gip <IP_address_of_gateway> -m
   <network_mask> -b "/opt/ignite/boot/nbp.efi"
   ```

   NOTE: **`IP_address_of_gateway`** is the IP address of gateway from the LAN domain of the vPar to the LAN domain of the Ignite server. **`network_mask`** is the netmask (in dotted notation) of the LAN to which vPar is connected.

3. Boot from the install kernel using the following command:

   ```
   lanboot select –dn mybootprofile
   ```

4. Follow the steps as prompted by the install kernel to install HP-UX 11i.

6.9 Setting up a vPar to use NPIV HBAs

This section describes the steps to configure a vPar with NPIV HBAs, install a vPar image on an NPIV disk, and bring the vPar up.

6.9.1 Configuring storage for a vPar with NPIV HBAs

You can assign storage for a vPar with an NPIV HBA either before or after the vPar starts up. In both cases, the vPar will boot if it already has a non-NPIV boot device configured. If it does not have a boot device yet, then the vPar boot will halt at EFI.

To configure storage for a vPar with NPIV HBA:

1. Start the vPar.

   Once the vPar is started, the virtual port instance corresponding to the NPIV HBA assigned to the vPar would have logged into the FC fabric to which the physical HBA is connected.

2. Obtain the port WWN assigned to the NPIV HBA using the `vparstatus` command on VSP.
3. Note the port WWN number, and work with your storage administrator to get the required storage provisioned.

   The storage administrator must use the storage management utility corresponding to the storage device from which the administrator plans to provision storage, and then create LUNs of the required capacity.

4. Present the LUNs to the port WWN corresponding to the NPIV HBA.

NOTE: The steps to provision storage are the same as for a native host, except that in this case, the HBA and the WWN involved are virtual in nature.

You can present the same LUN to a vPar using both NPIV and AVIO.

Example 12 Using `vparstatus` to determine the WWN of an NPIV HBA

```
vparstatus -P Vpar1 -d
[Virtual Partition Devices]

[Storage Interface Details]
disk:avio_stor:0,0,0:avio_stor:/dev/rdisk/disk31
disk:avio_stor:0,0,1:lv:/dev/vg_on_host/rlvol3
hba:avio_stor:0,4,0x50060b00006499b9,0x50060b00006499ba:npiv:/dev/fcd1
hba:avio_stor:1,3,0x50060b00006499a0,0x50060b00006499a8:npiv:/dev/fcd0

[Network Interface Details]
network:lan:0,1,0xF2AF8F8647BF:vswitch:vswitch1:portid:1
network:lan:0,5,0x569FC1F96205:vswitch:vswitch1:portid:3

[Misc Interface Details]
serial:com1::tty:console
```

In this example, the port WWNs are:

- 0x50060b00006499b9
- 0x50060b00006499a0

These WWNs must be used for LUN masking or fabric zoning.

6.9.2 Identifying NPIV HBAs and devices in a vPar

To identify an NPIV HBA from the set of HBAs for the vPar, use the `ioscan` command.

Example 13 Identifying NPIV HBAs and devices in a vPar

```
# ioscan -kfNd gvsd
Class     I  H/W Path  Driver S/W State   H/W Type     Description

ext_bus   0  0/0/0/0   gvsd   CLAIMED     INTERFACE    VPAR AVIO Stor Adapter
ext_bus   1  0/0/4/0   gvsd   CLAIMED     INTERFACE    VPAR NPIV Stor Adapter
ext_bus   4  0/1/3/0   gvsd   CLAIMED     INTERFACE    VPAR NPIV Stor Adapter
```

Note that the `ioscan` output listing the NPIV devices in the vPar is exactly the same as a similar listing of SAN LUNs in a native host.

Example 14 Identifying NPIV HBAs and devices in a vPar by specifying hardware path

```
# ioscan -kfNH 0/0/4/0
Class     I  H/W Path  Driver S/W State   H/W Type     Description
================================================================
ext_bus   1  0/0/4/0   gvsd    CLAIMED     INTERFACE    VPAR NPIV Stor Adapter
tgtpath   3  0/0/4/0.0x207000c0ffda0287                    estp    CLAIMED     TGT_PATH     Virtual Storage
HBA target served by gvsd driver, target port id 0x105ef
lunpath   5  0/0/4/0.0x207000c0ffda0287.0x0                eslpt   CLAIMED     LUN_PATH     LUN path for ctl1
lunpath   8  0/0/4/0.0x207000c0ffda0287.0x4001000000000000 eslpt   CLAIMED     LUN_PATH     LUN path for disk7
lunpath   9  0/0/4/0.0x207000c0ffda0287.0x401d000000000000 eslpt   CLAIMED     LUN_PATH     LUN path for disk8
tgtpath   4  0/0/4/0.0x247000c0ffda0287                    estp    CLAIMED     TGT_PATH     Virtual Storage
HBA target served by gvsd driver, target port id 0x104ef
lunpath   6  0/0/4/0.0x247000c0ffda0287.0x0                eslpt   CLAIMED     LUN_PATH     LUN path for ctl2
lunpath  11  0/0/4/0.0x247000c0ffda0287.0x4001000000000000 eslpt   CLAIMED     LUN_PATH     LUN path for disk7
lunpath  12  0/0/4/0.0x247000c0ffda0287.0x401d000000000000 eslpt   CLAIMED     LUN_PATH     LUN path for disk8
```

6.9.3 Installing the vPar image on NPIV disks

After the LUNs are presented to the NPIV HBA, you can install the vPar image on an NPIV device.

6.10 Deactivating a vPar configuration

You can deactivate a vPar to remove or deallocate resources from it, while maintaining its configuration settings. This is a way of managing shadow configurations, and allows the shadow

configuration on a per vPar basis. The `-x active_config=false` option must be used with either the `vparcreate` or the `vparmodify` command.

You can deactivate a vPar configuration only if the vPar is in the inactive state, that is, the run-state must be DOWN.

To deactivate a single vPar configuration, the `vparmodify` command must be used with the `-x active_config=false` option. Once this is done, the vPar instance no longer consumes or reserves the resources allocated to it, and those resources may be distributed to other partitions or the VSP, or those resources may be used to a create different vPar instance.

To reactivate the vPar configuration use `vparmodify` with the `-x active_config=true`. However, note that a vPar configuration cannot be reactivated unless the resources it requires are available and not reserved by other vPar instances. A vPar can still be managed while its configuration is deactivated; however, it cannot be booted.

Example 15 Deactivating a vPar named Gold

```
vparmodify -p Gold -x active_config=false
```

6.11 vPar V6.1 local MCA support

On a system running vPars V6.0, any Machine Check Abort (MCA) encountered in an individual vPar (or the VSP) will result in a system crash that brings down all of the vPars. With vPars V6.1, a certain class of recoverable, local MCAs caused by a CPU in an individual vPar are isolated to that vPar and do not impact other running vPars.

The vPar OS first tries to automatically recover from such MCAs without bringing down the vPar (APR – automatic process recovery supported by HP-UX). If that is not possible, the individual vPar goes through a crash dump and is rebooted to recover from the error. If the vPar is rebooted, a tombstones file is generated in the individual vPar.

The type of MCAs recovered from typically includes user process register file errors, kernel process register file errors, TLB errors etc affecting a single vPar. In all other cases of local MCAs affecting individual vPars or any type of local MCA affecting VSP cores and any global MCAs, a server or nPartition crash happens impacting VSP and all vPars. In most cases, a VSP core dump is also generated. In all cases, MCA logfiles are generated in the standard locations depending on the platform.

You should be aware of the following behavior:

- If a CPU core experiences an excessive number of MCAs from which the vPar recovered either through APR or through rebooting the vPar, system firmware and or diagnostics might deconfigure or deactivate the CPU. In this case, when the vPar reboots, it will not contain a deactivated or deconfigured CPU core and the MCA error records belonging to the affected CPU core might not be available in the `/var/tombstones` directory.
- If another MCA (of any type) happens on any other CPU core when recovery of an earlier MCA has not yet completed, this might cause the server or partition to be reset.
- If you stop or reset a vPar before it completely boots up after processing a local MCA, this action might lead to the server or partition being reset depending on the platform. On Superdome 2, this might also result in the nPartition status being displayed as MCA, even though the vPar has actually recovered from the MCA.
- When a local MCA affecting an individual vPar cannot be contained or isolated, it will trigger a server or nPartiton reset. In most cases, this will manifest as an INIT received by the VSP resulting in a VSP crash dump and reboot. Hence, if there is an unexpected crash of the VSP indicating that it was TOC'd, check the system firmware logs to determine if there was an MCA that caused this. The VSP `crashdump` itself might not have any information about the MCA.

6.12 Recovering a nonresponsive vPar

In the rare event that the OS of a vPar becomes nonresponsive, there is no prompt from a network connection nor through the virtual console (vparconsole). In such a situation, you may need to manually reset the partition. To recover a vPar that is nonresponsive, you can use the `vparreset` command.

To recover a nonresponsive vPar, use the `vparreset -g` command. This results in a graceful shutdown of the OS. The vPar can then be restarted. However, if the OS is nonresponsive, this operation might not be successful. When `vparreset -g` does not succeed, consider the following additional operations:

- Use the `vparreset -t` command if you want to initiate a transfer of control operation. This option restarts the vPar. Also, a dump of the OS is collected if there is sufficient space, which provides any troubleshooting data to determine the cause of the hang. The dump information can later be used to troubleshoot the unresponsiveness of the vPar.
- Use the `vparreset -h` command to hard reset the vPar. This option restarts the vPar without collecting an OS dump.
- Use the `vparreset -d` command to power down the vPar. This option does not restart the vPar.

7 Creating virtual machines

After you install the vPars and Integrity VM product, you can create virtual machines and virtual resources for the virtual machines to use. This chapter includes the following topics:

- Specifying VM characteristics
- Using the `hpvmcreate` command
- Starting VMs
- Changing VMs configurations
- Cloning VMs
- Stopping VMs
- Removing VMs
- Troubleshooting virtual machine creation problems

7.1 Specifying virtual machine characteristics

When you create a new virtual machine, you specify its characteristics. Later, you can change the virtual machine characteristics.

You can set the characteristics of a virtual machine using the following commands:

- `hpvmcreate`, which creates new virtual machines.
- `hpvmclone`, which creates new virtual machines based on existing virtual machine.
- `hpvmmigrate`, which moves virtual machines from one system to another.
- `hpvmmodify`, which modifies existing virtual machiness.

All of these commands accept the same options for specifying VM characteristics. Table 11 describes each characteristic and command option.

Table 11 Characteristics of an VM

Virtual Machine Characteristic	Default Setting	Command Option	Where Described
Virtual machine name	You must specify a name when you create or modify the virtual machine. You cannot modify this characteristic.	`-P vm-name`	Section 7.1.1 (page 76)
Operating system type	If you do not specify the operating system type, it is set to UNKNOWN.	`-O os_type [:version]`	Section 7.1.3 (page 77)
Virtual CPUs (vCPUs)	If you omit this option when you create the virtual machine, the default is one vCPU.	`-c number_vcpus`	Section 7.1.5 (page 77)
Virtual machine type	If not specified, by default a shared VM is created.	`-x vm_type=type`	"Virtual machine type" (page 76)
CPU entitlement	If you omit this option when you create the virtual machine, the default is 10%.	`-epercent[:max_percent]` `-E cycles[:max_cycles]`	Section 7.1.6 (page 78)
Memory	If you omit this option when you create the virtual machine, the default is 2 GB.	`-r amount`	Section 7.1.7 (page 78)

Table 11 Characteristics of an VM *(continued)*

Virtual Machine Characteristic	Default Setting	Command Option	Where Described
Virtual devices	If you omit this option when you create the virtual machine, it has access to no network and storage devices.	`-a rsrc`	Section 7.1.10 (page 80)
Virtual machine label	If you omit this option, the virtual machine has no label.	`-l vm_label`	Section 7.1.11 (page 83)
Startup behavior	If you omit the option, it is set to `auto`, and the virtual machine starts when Integrity VM is started.	`-B start_attribute`	Section 7.1.12 (page 83)
Dynamic memory	If you omit the option, dynamic memory is not enabled for the guest.	`-x keyword=parameter`	Section 7.1.13 (page 84)
Group with administrator or operator privileges	If you omit this option, no group accounts have `admin` or `oper` privileges.	`-g [+]group[:admin | oper]`	Section 11.5 (page 174)
Resource reservations	If not specified, resources will not be reserved when the virtual machine is off.	`–x resources_reserved=[true|false]`	"Reserved resources" (page 77)
User with administrator or operator privileges	If you omit this option, no user accounts have `admin` or `oper` privileges.	`-u [+]user[:admin | oper]`	Section 11.5 (page 174)

7.1.1 Virtual machine name

Use the `-P vm-name` option to specify the name of the new virtual machine. This option is required for the `hpvmcreate` command. In the following example, the new virtual machine is named `host1`. On the VSP, enter the following command:

```
# hpvmcreate -P host1
```

The virtual machine name can be up to 256 alphanumeric characters, including A-Z, a-z, 0-9, the dash (—), the underscore (_), and period (.). The virtual machine name must not start with a dash.

7.1.2 Virtual machine type

Use the `-x vm_type=type` to specify the virtual machine type. A virtual machine that shares CPU resources with other virtual machine types can be specified with the *`shared`* type. While a virtual machine that has exclusive access to CPU resources is specified with the *`vpar`* . By default, a *`shared`* type is created.

Note that when creating a vPar, resource reservations and AutoBoot are not set by default, as is the default when using the `vparcreate` command. The following two commands are functionally equivalent:

```
vparcreate -P vparName
```

```
hpvmcreate -P vparName -B auto -x vm_type=vpar -x resources_reserved=true
```

Before changing the *`vm_type`* of the existing virtual machine, the virtual machine must be shut down, and the Matrix OE Portable Image product must be installed and enabled.

Note that when a vPar is created, its CPU entitlement (`-e`) is automatically set to (forced) 100%. Converting a shared VM to a vPar automatically adjusts its entitlement to 100%, while converting a vPar to a shared VM does not change the entitlement (meaning it remains 100% unless modified using the `-e` option).

7.1.3 Guest operating system type

Use the `-O os_type` option to specify the type of operating system that will run on the virtual machine. This option is not required.

For `os_type`, specify `hpux`. For specific information about installing HP-UX guests, see Chapter 8 (page 101).

If you do not supply the operating system type, it defaults to `UNKNOWN`. When you install the operating system and boot the guest, this guest configuration parameter is automatically set to the appropriate operating system type.

When a running guest transitions from running in the machine console to running in the operating system, the operating system type is detected. If the operating system type is different from the information in the guest's configuration file, it is automatically updated to reflect the current operating system.

7.1.4 Reserved resources

Use the `-x resources_reserve=true` to specify whether CPU, memory and device resources should be reserved while the virtual machine is in the off state. Resource reservations attempt to guarantee that resources will be available so that the virtual machine can be started at any time. For more information about reserved resources, see "Reserved resources and resource over-commitment" (page 79)

7.1.5 Virtual CPUs

Use the `-c number_vcpus` option to the command to specify the number of virtual CPUs (vCPUs) that the virtual machine can use. If you do not specify the number of vCPUs, the default is 1. For example, to set the new virtual machine `host1` to have two vCPUs, enter the following command:

```
# hpvmcreate -P host1 -c 2
```

Every virtual machine has at least one vCPU. A running virtual machine cannot use more vCPUs than the number of physical CPU-cores on the VSP system. (For the purpose of this discussion, the term "physical CPU" refers to a processing entity on which a software thread can be scheduled.)

Do not set the number of vCPUs higher than the physical number of CPU-cores, as this can cause undesirable behavior.

NOTE: Even if hyperthreading and `lcpu_attr` is turned ON in the VSP, the number of vCPUs in a VM cannot be more than the number of physical CPU-cores on the system.

HP strongly discourages changing the hyperthreading/lcpu_attr settings on the VSP.

The following command specifies the number of virtual CPUs to set:

```
# hpvmcreate -c number_vcpus[:minimum[:maximum]]
```

The minimum and maximum values are boundary values, which are enforced if the number of virtual CPUs this virtual machines changes in the future. The default value is one (1) virtual CPU for the virtual machine. The number of virtual CPUs should not be set higher than physical number of CPUs on the VSP, as this can cause undesirable behavior.

The default minimum and maximum boundary values are a minimum of one (1) virtual CPU, and a maximum of sixteen (16) virtual CPUs.

NOTE: HP Integrity VM does not support real-time applications running in the guest. Scheduling and precise timing properties that can be relied upon on physical hardware are not guaranteed to be preserved in a virtual machine. In particular, changing the `hires_timeout_enable(5)` HP-UX tunable may not have the desired effect.

You can change the number of enabled CPUs in HP-UX guests, using the `hpvmmgmt -c num` command. This command sets the number of enabled virtual CPUs to the number indicated by

num, and disables the others. Disabled virtual CPUs no longer show up in the guest in commands such as top or GlancePlus, and no longer consume resources on the VSP. However, disabled virtual CPUs still appear on the VSP, for example in the hpvmsar command.

7.1.6 Entitlement

Use the -e or -E option to specify the virtual machine's entitlement.

Virtual machine entitlement is the minimum amount of processing power guaranteed to the virtual machine from each virtual CPU. When you create a virtual machine, you can use the -e option to specify the entitlement as a percentage, from 5% to 100%. If you do not specify the entitlement, the virtual machine receives 10% entitlement by default.

Alternatively, you can use the -E option to specify the entitlement as the number of CPU clock cycles per second to be guaranteed to each vCPU on the virtual machine.

For example, to specify an entitlement of 20% for the new virtual machine host1, enter the following command:

```
# hpvmcreate -P host1 -e 20
```

When the virtual machine starts, the VSP ensures that sufficient processing power is available for every running virtual machine to receive its entitlement. For virtual machines with multiple virtual CPUs, the entitlement is guaranteed on each vCPU in the virtual machine's configuration. For example, if a virtual machine has four vCPUs, and the entitlement is set at 12%, the VSP ensures that the equivalent of at least 48% of one physical CPU is available to that virtual machine.

To allow multiple virtual machines to run at the same time, make sure that the entitlement of each virtual machine does not prevent the others from obtaining sufficient processor resources. The sum of all entitlements across all active virtual machines cannot total more than 100% for any physical processor. If available processor resources are insufficient, the virtual machine is not allowed to boot; error messages are displayed to indicate the specific problem.

If a virtual machine is busy and sufficient processing resources are available on the VSP system, the virtual machine can receive more than its entitlement. When there is contention for processing resources (on a VSP system with busy virtual machines), each virtual machine is limited to its entitlement.

For help managing CPU power across multiple virtual machines, install the HP Global Workload Manager (gWLM) on the VSP system. For more information, see *HP Integrity Essentials Global Workload Manager Administrator's Guide*.

7.1.7 Guest memory allocation

Use the -r *amount* option to specify the amount of virtual memory to be allocated to the guest. If you do not specify the memory allocation, the default is 2 GB. For example, to allocate three gigabytes to the virtual machine host1, enter the following command:

```
# hpvmcreate -P host1 -r 3G
```

The amount of memory to allocate is the total of the following:

- The amount of memory required by the guest operating system.
- The amount of memory required by the applications running on the guest.

The amount of memory should be at least the total of these two amounts. If there is not enough memory in the current configuration, Integrity VM issues a warning but allows you to create the virtual machine. This allows you to create virtual machines for future configurations. When the virtual machine is started, the VSP checks memory resources, including those allocated to running guests, and makes sure that there is sufficient memory to run the virtual machine. In addition to the amount of memory you specify for the virtual machine, the VSP requires a certain amount of overhead for booting the guest operating system. The amount of memory allocated to all the running guests cannot exceed the amount of physical memory minus the amount used by the VSP for its

operating system and its administrative functions. For more information about the memory requirements of the VSP, see Section 3.1.3 (page 37).

Guest memory allocation can be viewed and allocated dynamically (that is, without stopping the guest) by using dynamic memory parameters, as described in Section 11.9 (page 180).

7.1.8 Automatic cell balancing

When creating a guest, Integrity VM determines the best fitting locality domain for the new guest when the VSP is predominantly Cell Local Memory(CLM). The `hpvmstatus -C` command provides a list of guests with their memory type.

7.1.9 Reserved resources and resource over-commitment

HP-UX vPars and Integrity VM V6.1 allows the reservation of resources for virtual machines and virtual partitions. Reservations imply that a resource will be available when it is needed, with the intention of assuring that a virtual machine or virtual partition can boot at any time. The reserved resources setting is managed for each individual virtual machine and virtual partition and is set using the `resources_reserved` attribute (managed with the `-x` option of the `hpvmcreate` and the `hpvmmodify` command). The default behavior of the `vparcreate` command is to set `resources_reserved` to true when a virtual partition is created. However the `hpvmcreate` command does not reserve resources by default when creating virtual machines or virtual partitions. The `resources_reserved` attribute can be managed using the `hpvmmodify` command.

Resources that are reserved include memory, CPU and I/O devices. If a resource is assigned to a virtual machine or virtual partition that has the `resources_reserved=true`, that same resource cannot be assigned to a different virtual machine or virtual partition that also has `resources_reserved=true`. It is also not possible to assign a resource to a virtual partition or virtual machine that has `resources_reserve=true`, if that resource is not currently available. For example if all the CPUs have been assigned to other reserving virtual machines or virtual partitions, then it is not possible to assign CPUs to any additional reserving virtual machines or virtual partitions. It is possible to assign resources to non-reserving virtual machines and virtual partitions, however, it is not possible to boot them (because the resources assigned to that virtual machine or virtual partition are reserved by other virtual machines or virtual partitions).

7.1.9.1 Resource over-commitment

In some circumstances, it is possible for resources to be over-committed, meaning that more resources are assigned to reserving virtual machines and virtual partitions than are currently available. This can occur if CPU, memory or other I/O devices are lost, such as removed during a VSP shutdown or from a hardware failure. If all the resources of the VSP are assigned to virtual partitions or virtual machines, and then later some of those resources are lost, the VSP will be in an over-committed state. When this occurs, it is not possible to boot all of the reserving virtual machines or virtual partitions. Virtual machines and virtual partitions with reserving resources will be allowed to boot on a first-come first-served basis. And virtual machines and virtual partitions without resource reservations will not be allowed to boot.

If CPU or memory resources are overcommitted, this state is logged in the `/var/opt/hpvm/common/command.log` file. Additionally the overcommitted state is noted when displaying system resources using the `hpvmstatus -s` command. For example:

```
# hpvmstatus -s
[HPVM Server System Resources]

        *** VSP resources are over-committed ***
        vPar/VM types supported by this VSP = Shared
        Processor speed = 1596 Mhz
        Total physical memory = 16278 Mbytes
        Total number of operable system cores = 8
        CPU cores allocated for VSP = 1
        CPU cores allocated for vPars and VMs = 7
```

```
CPU cores currently in use or reserved for later use = 0
Available VSP memory = 884 Mbytes
Available swap space = 7032 Mbytes
Total memory allocated for vPars and VMs = 12544 Mbytes
Memory in use by vPars and VMs = 13568 Mbytes
Available memory for vPars or VMs is overdrawn by 1024 Mbytes
Available memory for 0 (max avail.) CPU VM = N/A Mbytes
Available memory for 0 (max avail.) CPU vPar = N/A Mbytes
Maximum vcpus for an HP-UX virtual machine = 7
Maximum vcpus for an OpenVMS virtual machine = 7
Maximum available vcpus for a VM is overdrawn by 1
Available CPU cores for virtual partitions are overdrawn by 1
```

When CPU or memory resources are overdrawn, either these resources must be brought back online, or reservations on existing virtual machines or virtual partitions must be reduced or removed. Resource reservations can be removed using the `hpvmmodify -P name -x resources_reserved=false` command. Memory reservations can be reduced by using the `hpvmmodify -r` option to reduce virtual machine or virtual partition memory size. CPU reservations can be reduced by using either the `-c` or `-e` options to reduce the CPU count or CPU entitlement. Note `-e` does not apply to a virtual partition, because it always has 100% entitlement to a CPU. Use the information provided by `hpvmstatus -s` to determine the amount of overcommitted resource, and then use `hpvmmodify` to reduce that resource's commitments appropriately.

When virtual machines are configured to use reserved resources, and CPUs are overcommitted, it might require a combination of reducing both CPU entitlement, and CPU counts to bring the system back into a committed state. When determining the appropriate amount of entitlement or CPU count to reduce, Integrity VM packs CPU entitlement as efficiently as possible. For example, if there are two virtual machines that require 50% entitlement of three CPUs, and there are seven CPUs available in total on the VSP, that would mean that four CPUs would be available at 100%, if needed (however, those two virtual machines might spread across 6 CPUs if those CPUs are not otherwise in use.)

While it is not possible to increase resources on a reserving virtual machine or virtual partition when the VSP is in an over-committed state, it is possible to reduce resource commitments, even if that reduction still leaves the VSP in an over-committed state.

It is still possible to change resource assignments of virtual machines and virtual partitions when the VSP is in an overcommitted state, if those virtual machines or virtual partitions do not reserve resources. However it is not possible to set the `resources_reserved=true`.

7.1.10 Virtual devices

Use the `-a` option to allocate virtual network switches and virtual storage devices to the virtual machine. The VSP presents devices to the virtual machine as “virtual devices.” Attached I/O devices, such as tape, DVD burner, and autochanger, are not presented as virtual devices; they are presented as physical I/O devices. You specify both the physical device to allocate to the virtual machine and the virtual device name that the virtual machine will use to access the device. The following sections provide brief instructions for creating virtual network devices and virtual storage devices.

7.1.10.1 Creating virtual network devices

The guest virtual network consists of:

- Virtual network interface cards (vNICs)
- Virtual switches (vswitches)

For virtual machines to communicate either with other virtual machines or outside the VSP system, each virtual machine's virtual network must be associated with a virtual switch (vswitch). If you start a virtual machine without a vswitch, the virtual machine has no network communication channel.

Traffic from an AVIO guest LAN network device is directed to the pNIC directly by a separate host module rather than by the vswitch. You can create vswitches before or after creating guests that access the vswitches. If you create the virtual machine before creating the vswitch, the virtual machine is created and warning messages display the specific problem. This allows you to create virtual machines for future configurations.

To create a vswitch, enter the `hpvmnet -c` command. Include the `-S` option to specify the name of the virtual switch. For example:

```
# hpvmnet -c -S vswitch-name -n nic-id
```

where:

- *`vswitch-name`* is the name you assign to the vswitch. You must specify the name of the vswitch.
- *`nic-id`* is the pNIC ID on the VSP. If you omit the *`nic-id`*, the vswitch is created for the `localnet`.

To start the vswitch, enter the `hpvmnet -b` command. For example:

```
# hpvmnet -b -S vswitch-name
```

For more information about using the `hpvmnet` command, see Section 10.2.1 (page 144).

To create the virtual machine and allocate the vswitch to it, use the `-a` option to the `hpvmcreate` command. For example:

```
# hpvmcreate -P vm-name -a network:adapter-type:[hardware-address]:vswitch:vswitch-name
```

where *`hardware-address`* (optional) is the vNIC PCI bus number, device, and MAC address. If you omit the hardware address, it is generated for you. HP recommends that you allow this information to be automatically generated. In this case, omit the *`hardware-address`* value from the command line, but retain the colon character separator. For example:

```
# hpvmcreate -P vm-name -a network:adapter-type:vswitch:vswitch-name
```

The *`adapter-type`* is `avio_lan`.

On the guest, use standard operating commands and utilities to associate the vNIC with an IP address, or use DHCP just as you would for a physically independent machine.

By default, vswitches are sharable; you can allocate the same vswitch to multiple virtual machines. The `hpvmnet` command displays the status of the vswitches, including the mode. The vswitches are always in SHARED mode.

Virtual LANs allow virtual machines to communicate with other virtual machines using the same VLAN, either on the same VSP or on different VSP systems. You associate the VLAN port number with a vswitch, then allocate that vswitch to virtual machines that communicate on that VLAN. For more information about HP-UX VLANs, see the manual *Using HP-UX VLANs*.

NOTE: If the guest is configured with a number of VLAN devices, but it does not have sufficient memory, some of the devices might be missing after the guest is booted. To resolve this issue, increase the size of the guest memory with the `hpvmmodify -r` command.

For more information about creating and managing VLANs on virtual switches, see Section 10.4 (page 152).

7.1.10.2 Creating virtual storage devices

When you create a virtual machine, you specify the virtual storage devices that the virtual machine uses. Virtual storage devices are backed by physical devices on the VSP system (backing stores). The VSP system must have storage for the VSP and for all of the virtual machines.

Use the `-a` option to create and allocate the virtual device to the virtual machine. For example:

```
# hpvmcreate -a VM-guest-storage-specification:VM-Host-storage-specification
```

where:

- *VM-guest-storage-specification* defines where and what storage is seen in the virtual machine. This is formatted as:

  device:*adapter-type*:*hardware-address*:

  You can specify one of the following devices:

  - disk
  - dvd
  - tape
  - changer
  - burner
  - hba

- *adapter-type* is avio_stor on an HP-UX guest.
- *hardware-address* or pcibus, pcislot,aviotgt (optional) specifies the virtual device PCI bus number, PCI slot number, and AVIO target number. If you do not specify this information, it is generated automatically. HP recommends that you allow the hardware address to be generated automatically. To omit the hardware address, use the following format (including two colons):

  device:adapter-type::*VM-Host-storage-specification*

- *VM-Host-storage-specification* defines where and how the virtual machine storage is supplied on the VSP. Specify it using the following format:

  storage:*location*

  Where *storage* is one of the following:

  - disk
  - lv
  - file
  - null
  - attach

  And *location* is a VSP system file.

For complete information about constructing storage specifications for virtual machines, see Section 9.2.2.1 (page 122).

The type of VSP backing store can affect the performance of the virtual machine. Use the ioscan command to obtain information about the current device configuration on the VSP system, and try to distribute the workload of the virtual machines across the physical backing stores.

When you share a physical backing storage device among virtual machines. potential conflicts are not always obvious. For example, if you use a file in a file system on /dev/disk/disk1 as a backing store, the raw device (/dev/rdisk/disk1) cannot also be used as a backing store. For more information about specifying virtual devices, see Chapter 9 (page 107).

Integrity VM checks the current physical configuration when you create a virtual machine using the hpvmcreate command. If the virtual machine uses backing stores that are not available, the virtual machine is created, and warning messages provide details. If you use the hpvmstart command to start a virtual machine that requires physical resources that are not available on the VSP system, the virtual machine is not allowed to start, and error messages provide detailed information about the problem.

After you create a virtual machine, you can use the hpvmmodify command to add, remove, or modify storage devices for the virtual machine. To add a device to an existing virtual machine, include the -a option, the same way you would on an hpvmcreate command. For example,

the following command modifies the virtual machine named `host1`, adding a virtual DVD device backed by the physical disk device `/c1t1d2`. The virtual hardware address is omitted and will be generated automatically.

```
# hpvmmodify -P host1 -a dvd:avio_stor::disk:/dev/rdisk/disk2
```

You can modify storage devices while the virtual machine is running. It is not necessary to restart the virtual machine; however, it may be necessary to re-scan for devices on the virtual machine.

Some devices should be restricted to use by the VSP and to each guest (for example, boot devices and swap devices). Specify restricted devices using the `hpvmdevmgmt` command. For more information about sharing and restricting devices, see Section 11.11.2.4 (page 192).

Any alternate boot devices should be set with the same care that you would use on a physical system. If the primary boot device fails for any reason, a virtual machine set to `autoboot` attempts to boot from devices in the specified boot order until either an option succeeds or it reaches the EFI Shell. Make sure that any specified boot options, and the boot order, are appropriate for the guest. For more information about the `autoboot` setting, see Table 14.

7.1.11 Creating virtual machine labels

The `-l` option specifies the label of the virtual machine. The virtual machine label is a descriptive label unique to this virtual machine. The label can be useful in identifying a specific virtual machine in the `hpvmstatus -V` display. The label can contain up to 256 alphanumeric characters, including A-Z, a-z, 0-9, the dash (—), the underscore (_), and the period (.). If white space is desired, the label must be quoted ("").

7.1.12 Specifying the virtual machine boot attribute

The `-B` option specifies the startup behavior of the virtual machine. The `start_attr` attribute can have the following (case-insensitive) values:

- `auto`: Automatically start the virtual machine when Integrity VM is initialized on the host.
- `manual`: Manually start the virtual machine.

If the `start_attr` attribute is set to `auto`, the virtual machine is started when Integrity VM is initialized. This is the default. This occurs when the VSP system is booted, and when the Integrity VM software is stopped and restarted on a running VSP. For example, when you upgrade Integrity VM to a new version on a running system, the software is started automatically. The VSP attempts to start all virtual machines for which the attribute is set to `auto`. If insufficient resources exist, some virtual machines may fail to start.

If the attribute is set to `manual`, the virtual machine will not be started automatically when Integrity VM is initialized on the VSP. The virtual machine can then be started manually with the `hpvmstart` command or through its virtual console.

This option does not set the virtual machine's console to enable booting when the virtual machine is started. This function must be set with the virtual machine's console.

NOTE: If the `start_attr` attribute is set to Auto, the virtual machine also starts after you install or upgrade Integrity VM.

In addition to automatically starting guests when Integrity VM starts, this feature also determines a startup order to best utilize VSP processor and memory resources. On cellular systems with cell local memory (CLM) configured, the goal is to start the guests so that CLM is utilized first. For each guest with the `start_attr` attribute set to `auto`, the startup order is based on a memory weight and a processor weight added together.

A rough estimate of the memory weight calculation is:

100 * guest memory size / available host memory + 2 (if the guest resources can fit into a cell's available CLM and processors)

A rough estimate of the processor weight calculation is:

(minimum guest cpu entitlement * number of virtual processors) / (100 * number of host processors)

Guests are expected to start in order of highest weight to lowest. You can adjust the order by setting the `sched_preference` attribute (Section 3.2.6). If a guest fails to start for any reason, the sequence continues with the next guest. For memory placement on a non cell-based system or cell-based system with all interleaved (ILM) memory configured, the boot order has little affect.

In general, on these configurations, the largest guests boot first. On cell-based systems with CLM configured, expected memory placement depends on the calculated weights, the `sched_preference` setting and the VSP memory configuration:

- If `sched_preference` is not set or set to “cell” and the guest resources fit into one cell, CLM is used.
- If there is not enough CLM and there is enough ILM, ILM is used.
- If `sched_preference` is set to “ilm” and there is enough ILM, ILM is used.
- If there is not enough ILM, the memory is allocated from all cells (striped).
- If there is insufficient ILM but the guest resources fit into one cell, CLM is used. Otherwise the memory is striped.

7.1.13 Specifying dynamic memory parameters

Specifies whether the new virtual machine (shared VM type only) uses dynamic memory and the values associated with it by including the following keywords:

- `dynamic_memory_control={0|1}`
- `ram_dyn_type={none|any|driver}`
- `ram_dyn_min=`*`amount`*
- `ram_dyn_max=`*`amount`*
- `ram_dyn_target_start=`*`amount`*
- `ram_dyn_entitlement=`*`amount`*
- `amr_enable={0|1}`
- `amr_chunk_size=`*`amount`*

For more information about using dynamic memory for guests, see Section 11.9 (page 180).

7.1.14 Configuration limits

Table 12 lists the configuration limits for Integrity VM Version 6.1.

Table 12 Configuration Limits

Description	Support
# vCPUs/VM — Maximum (Integrity VM V6.1 Max vCPU = 16)	min (#pCPUs, Max vCPU)
# vCPUs/pCPU — Maximum	20
# VMs per VSP — Maximum	256
# pCPUs in VSP	HP-UX limit
Memory per VM — Minimum (HP-UX 11i v3)	2 GB or the minimum required for HP-UX 11i v3 to boot
Memory per VM — Maximum (HP-UX)	128 GB
# virtual SCSI devices / VM or vPar— Maximum	256 AVIO
# virtual NICs / VM or vPar— Maximum	62

Table 12 Configuration Limits *(continued)*

Description	Support
# virtual switches — Maximum	50
# virtual NICs / vswitch	511
# file backing store devices / VM or vPar — Maximum	30
# virtual AVIO storage devices /VM	Maximum 256
# file backing store devices /VM	Maximum 30
Maximum size of backing store for AVIO (disk, lvol, file)	> 2TB
Maximum # PCI functions per vParVM for DIO	16

7.1.15 Sizing guidelines

The sizing guidelines for Integrity Virtual Machines Version 4.0 and later are different from that of previous releases due to several factors, including the change of VSP operating system to HP-UX 11i v3. As a result, the formulas used to calculate virtual machine capacity are outlined in the white paper *Hardware Consolidation with Integrity Virtual Machines*. The sizing information and related calculations are updated in revisions to this white paper dated September 2008 or later. The latest version of this white paper is available from:

http://www.hp.com/go/virtualization-manuals

7.1.16 Default guest settings for HP-UX

Table 13 lists the default guest settings for HP-UX and `Unknown` guests. An `Unknown` guest is a virtual machine that has not booted with any operating system. When an `Unknown` guest type boots, the appropriate operating system type is applied to the guest configuration.

The following guest OS specific settings are applied if you specify the operating system type with the `-O` option to the `hpvmcreate` command.

Table 13 Guest Default Settings

	HP-UX Guest Default Settings	Unknown Guest Operating System Default Settings
Maximum CPUs	16	16
Default CPUs	1	1
Default memory	2 GB	2 GB
Minimum memory	512 MB[1]	32 MB
Maximum memory	128 GB	128 GB
Default reserved memory	64 MB	64 MB
Minimum reserved memory	32 MB	32 MB
Maximum reserved memory	128 GB	128 GB

[1] The minimum memory requirement for HP-UX 11i v2 is 512 MB. The minimum memory requirement for HP-UX 11i v3 is 1 GB (see "System Requirements" section in the *HP-UX 11i v3 Installation and Update Guide*); however, the *HP-UX 11i*

v3 Installation and Update Guide warns that cold installations with 1 GB or less memory might fail or take a long time to complete. Therefore, 2 GB is recommended for cold installations of HP-UX 11i v3.

NOTE: The amount of memory you should allocate to the guest must be sufficient to allow the guest operating system to boot. This amount might differ from the defaults documented here. For specific memory requirements, see the product documentation for the operating system and applications on the guest.

7.2 Using the `hpvmcreate` command

To create a virtual machine, enter the `hpvmcreate` command. Enter the `-P` option to specify the virtual machine name (up to 256 alphanumeric characters). All other options are optional and may be added to the virtual machine configuration later using the `hpvmmodify` command.

Table 14 describes the options you can use with the `hpvmcreate` command.

Table 14 Options to the `hpvmcreate` Command

Option	Description
`-P vm-name`	Virtual machine name. You must specify a name when you create or modify the virtual machine. You cannot modify this characteristic.
`-O os_type[:version]`	Specifies the type and version of the operating system. If you do not specify the operating system type, it is set to UNKNOWN. The version is specific to the operating system type and can consist of up to 256 alphanumeric characters, including A-Z, a-z, 0–9, the dash (—), the underscore (_), and the period (.).
`-c number_vcpus`	Virtual CPUs (vCPUs) allocated. If you omit this option when you create the virtual machine, the default is one vCPU.
`-e percent[:max_percent]` `-E cycles[:max_cycles]`	CPU entitlement allocated. If you omit this option when you create the virtual machine, the default is 10%.
`-r amount`	Memory allocated. If you omit this option when you create the virtual machine, the default is 2 GB.
`-a rsrc`	Virtual devices created. If you omit this option when you create the virtual machine, it has access to no network and storage devices.
`-l vm_label`	The label for the virtual machine (an optional text string associated with the virtual machine).
`-B start_attribute`	The startup behavior of the virtual machine (`auto` or `manual`).

Table 14 Options to the `hpvmcreate` Command *(continued)*

Option	Description
`-x keyword=parameter`	Specifies values for dynamic memory setting associated with the guest, including: • `dynamic_memory_control` • `ram_dyn_type` • `ram_dyn_min` • `ram_dyn_max` • `ram_dyn_target_start` • `ram_dyn_entitlement=amount` • `amr_enable={0|1}` • `amr_chunk_size=amount` • `sched_preference` • `graceful_stop_timeout` For more information about dynamic memory, see Section 11.9 (page 180). Also specifies values for Online VM Migration: • migrate_copy_phase_timeout={number of seconds} • migrate_frozen_phase_timeout={number of seconds} • migrate_init_phase_timeout={number of seconds} • migrate_io_quiesce_phase_timeout={number of seconds} • online_migration={enabled | disabled} • tunables={name=value[,name=value,...]} For information about Online VM Migration, see Chapter 12 (page 195) Specifies arbitrary virtual machine or vPar attributes that control their behavior: • `vm_type={vpar|shared}` • `resources_reserved={0|1}` • `active_config={0|1}`
`-F`	Suppresses all resource conflict checks and associated warning messages (force mode). This option is primarily intended for use by scripts and other noninteractive applications. Note that you will receive no notification of potential resource problems for a virtual machine created with the `F` option. **NOTE:** The `-F` option is deprecated in Integrity VM commands; this option should be used only at the direction of HP Support.
`-s`	Verifies the virtual machine configuration and returns warnings or errors, but does not create the virtual machine. This option is used to invoke the `hpvmcreate` command's resource checking for a virtual machine configuration without actually creating the virtual machine. If the `-s` option is not specified, the virtual machine is created even if resource warnings occur.
`-g group[:admin | oper]`	Group with administrator or operator privileges over the virtual machine. Enter the group name for *`group`*, and enter either `admin` or `oper`.
`-u user[:admin | oper]`	User with administrator or operator privileges over the virtual machine. Enter the user name for *`user`*, and enter either `admin` or `oper`.

Table 14 Options to the `hpvmcreate` Command *(continued)*

Option	Description
`-i package-name`	Specifies whether the virtual machine is managed by Serviceguard or gWLM (or both). For the argument, specify one or more of the following parameters: • `SG` indicates that the VSP is a Serviceguard cluster node. • `SG-pkgname` indicates that the VSP is a Serviceguard package. • `GWLM` indicates that the VSP is managed by gWLM. • `NONE` indicates there are no external managers. For a node that is managed by both Serviceguard and gWLM, parameters are separated with a comma. For example: `SG_host1,gWLM`. **CAUTION:** This option is used by Integrity VM software; do not use this option without express instruction by HP.
`-j {0 | 1}`	Specified whether the virtual machine is a distributed guest (that is, managed by Serviceguard and can be failed over to another cluster member).
`-K console_IP_Addr`	Specifies the IP address used to connect to the guest's virtual iLO Remote Console. The address must be specified in IPv4 dot notation. The `–L` option must also be specified.
`-L console_IP_Addr_Netmask`	Specifies the IPv4 subnet mask used with the option when setting up the IP interface to be used for accessing the virtual iLO Remote Console for this guest. The address is entered in dot notation form.

You can specify an NPIV HBA with this syntax: hba:avio_stor::npiv:/dev/fcd0. For additional information about virtual I/O resource statements, see the `hpvmresources(5)` manpage.

7.2.1 Example of virtual machine creation

To create a virtual machine named `ux1`, enter the following command:

```
# hpvmcreate -P ux1
```

This command creates a virtual machine named `ux1` with no network access and no allocated storage devices. To view the characteristics of the virtual machine, enter the `hpvmstatus` command. For example:

```
# hpvmstatus
[Virtual Machines]
Virtual Machine Name VM #  Type OS Type State     #VCPUs #Devs #Nets Memory
==================== ===== ==== ======= ========= ====== ===== ===== =======
vPar0002                 2 VP   HPUX    Off            3     0     0 2048 MB
guest1                   3 SH   UNKNOWN Off            4     0     0   10 GB
ux1                      1 SH   HPUX    Off            4     2     0    3 GB
```

The `ux1` virtual machine has been assigned virtual machine number 1, has been created with an UNKNOWN operating system type, one vCPU, no storage devices, no network devices, and 3 GB of memory. For more information about running virtual machines under Serviceguard, see *HP Serviceguard Toolkit for Integrity Virtual Servers User Guide* at HP Serviceguard Toolkit for Integrity Virtual Servers.

7.3 Starting virtual machines

To start the virtual machine, enter the `hpvmstart` command. You can specify either the virtual machine name or the virtual machine number (listed in the `hpvmstatus` display under `VM #`.)

The `hpvmstart` command syntax is:

```
# hpvmstart {-P vm-name | -p vm_number} [-F | -s | -Q]
```

Table 15 describes the options to the `hpvmstart` command.

Table 15 Options to the `hpvmstart` Command

Option	Description
`-P vm-name`	Specifies the name of the virtual machine. Specify either the `-P` option or the `-p` option.
`-p vm_number`	Specifies the number of the virtual machine. To determine the virtual machine number, enter the `hpvmstatus` command.
`-F`	Suppresses all resource conflict checks and associated warning messages (force mode). Use force mode for troubleshooting purposes only. **NOTE:** The `-F` option is deprecated in Integrity VM commands; this option should be used only at the direction of HP Support.
`-s`	Sanity-checks the virtual machine configuration and returns warnings or errors, but doesn't start the virtual machine.
`-Q`	Quietly performs the command. The default is to prompt for confirmation of the command before performing it.

For example, to start the new virtual machine `host1`, enter the following command:

```
# hpvmstart -P host1
(C) Copyright 2000 - 2008 Hewlett-Packard Development Company, L.P.
Opening minor device and creating guest machine container
Creation of VM, minor device 2
Allocating guest memory: 2048MB
  allocating low RAM (0-80000000, 2048MB)
/opt/hpvm/lbin/hpvmapp (/var/opt/hpvm/uuids/8ba249f2-3399-11db-aacc-00306ef392e0
/vmm_config.current): Allocated 2147483648 bytes at 0x6000000100000000
    locking memory: 0-80000000
  allocating firmware RAM (ffaa0000-ffab5000, 84KB)
/opt/hpvm/lbin/hpvmapp (/var/opt/hpvm/uuids/8ba249f2-3399-11db-aacc-00306ef392e0
/vmm_config.current): Allocated 860 bytes at 0x6000000180000000
    locked SAL RAM: 00000000ffaa0000 (4KB)
    locked ESI RAM: 00000000ffaa1000 (4KB)
    locked PAL RAM: 00000000ffaa4000 (4KB)
    locked Min Save State: 00000000ffaa5000 (1KB)
RAM alignment: 40000000
Memory base low : 6000000100000000
Memory base FW  : 6000000180000000
Loading boot image
Image initial IP=102000 GP=62C000
Initialize guest memory mapping tables
Starting event polling thread
Starting thread initialization
Daemonizing....
hpvmstart: Successful start initiation of guest 'host1'
```

The `hpvmstatus` command displays the allocation of memory and devices. After you start the virtual machine, the `hpvmstatus` command displays the virtual machine status as `On (EFI)`, because the virtual machine is powered on but the guest operating system is not running. Because the operating system has not been installed, the guest OS type is listed as `UNKNOWN`.

```
# hpvmstatus
[Virtual Machines]
Virtual Machine Name VM #  OS Type State     #VCPUs #Devs #Nets Memory
==================== ===== ======= ========= ====== ===== ===== =======
config1                  1 HPUX    Off            1     5     1  512 MB
config2                  2 HPUX    Off            1     7     1    1 GB
guest1                   5 HPUX    On (OS)        1     5     1    1 GB
host1                   13 UNKNOWN On (EFI)       1     0     0    2 GB
```

For more information about using the `hpvmstatus` command, see Chapter 11 (page 165).

NOTE: When configuring or starting Integrity VM guests, the following warning message might be displayed if storage associated with the guest appears to be performing very poorly.

```
hpvmcreate: WARNING (host): Device /dev/rdisk/c6t9d0 took 32 seconds to open.
```

7.4 Changing virtual machine configurations

You can create a virtual machine with characteristics that the VSP cannot supply at the time of creation. This allows you to create virtual machines to run after system configuration changes. For example, the following command creates the virtual machine `host1` with 3 vCPUs and 4 GB of allocated memory:

```
# hpvmcreate -P host1 -c 3 -r 4G
HPVM guest host1 configuration problems:
    Warning 1: Guest's vcpus exceeds server's physical cpus.
    Warning 2: Insufficient cpu resource for guest.
These problems may prevent HPVM guest host1 from starting.
hpvmcreate: The creation process is continuing.
```

Because the VSP is not currently configured to support the new virtual machine, warning messages indicate the specific characteristics that are inadequate.

When you start a virtual machine, the VSP determines whether the current system configuration can support the virtual machine's characteristics. The ability of the system to run the virtual machine can be affected by the other virtual machines that are currently running, because they share the physical processors and memory. Any allocated vswitches must be started, and storage devices must be made available to the virtual machine. If the virtual machine cannot be started, the following type of message is generated:

```
# hpvmstart -P host1
HPVM guest host1 configuration problems:
Warning 1: Insufficient free memory for guest.
Warning 2: Insufficient cpu resource for guest.
   These problems may prevent HPVM guest host1 from booting.
hpvmstart: Unable to continue.
```

You can either change the system configuration, or modify the virtual machine. To modify the characteristics of a virtual machine, use the `hpvmmodify` command. When you use the `hpvmmodify` command to modify a guest, the entire guest configuration is reevaluated. Any problems that might prevent the guest from starting are reported. For example, if a guest has a reference to a host device that no longer exists, and you enter an `hpvmmodify` command that modifies the guest but does not fix the bad reference, a warning message is generated. Table 16 describes the options you can use on the `hpvmmodify` command.

For example, to modify the characteristics of the problematic virtual machine `host1` to remove vCPUs and memory, enter the following command:

```
# hpvmmodify -P host1 -c 1 -r 2G
```

This command changes the following characteristics of the virtual machine named `host1`:

- The `-c 1` option specifies one vCPU.
- The `-r 2G` option specifies two GB of memory.

NOTE: Note, you can specify an NPIV HBA with this syntax: **`hba:avio_stor::npiv:/dev/fcd0`**. For additional information about shared I/O resource statements, see `hpvmresources(5)`.

Table 16 Options to the `hpvmmodify` Command

Option	Description
`-P` *vm-name*	Specifies the name of the virtual machine. You must specify either the `-P` option or the `-p` option..
`-p` *vm_number*	Specifies the number of the virtual machine. To determine the virtual machine number, enter the `hpvmstatus` command.
`-F`	Suppresses all resource conflict checks and associated warning messages (force mode). Use force mode for troubleshooting purposes only. **NOTE:** The `-F` option is deprecated in Integrity VM commands; this option should be used only at the direction of HP Support.
`-s`	Sanity-checks the virtual machine configuration and returns warnings or errors, but does not start the virtual machine.
`-N` *new-vm-name*	Specifies a new name for the virtual machine. The name can consist of up to 256 alphanumeric characters including A-Z, a-z, 0-9, the dash (-), the underscore character (_), and the period (.). The virtual machine name cannot start with a dash (—).
`-l` *vm_label*	Modifies the descriptive label for this virtual machine. The label can contain up to 256 alphanumeric characters, including A-Z, a-z, 0-9, the dash (—), the underscore (_), and the period (.). To include spaces, the label must be quoted (" ").
`-B` *start_attr*	Modifies the startup behavior of the virtual machine. For *start_attr*, enter one of the following: `auto`: Automatically starts the virtual machine when Integrity VM is initialized on the VSP. `manual`: The virtual machine is not started automatically. Use the `hpvmstart` command to start the virtual machine manually.
`-O` *os_type*[:*version*]	Modifies the type and version of the operating system running on the virtual machine. For the *os_type*, specify one of the following (case-insensitive) values: `hpux`
`-c` *number_vcpus*	Modifies the number of virtual CPUs this virtual machine detects at boot time. If unspecified, the number defaults to one. The maximum number of vCPUs that you can allocate to a virtual machine is the number of physical processors on the VSP system.

`-e` *percent[:max_percent]* | `-E` *cycles[:max_cycles]*	Modifies the virtual machine's CPU entitlement in CPU cycles. To specify the percentage of CPU power, enter the following option:
	`-e` *percent[:max_percent]*
	To specify the clock cycles, enter one of the following options:
	`-E` *cycles[:max_cycles]*`M (for megahertz)`
	`-E` *cycles[:max_cycles]*`G (for gigahertz)`

`-g` *group*[:*admin* | *oper*]	Specifies a group authorization. The specified administrative level (`admin` or `oper`) is applied to the specified user group.
`-K` *console_IP_Addr*	Specifies the IP address used to connect to the guest's virtual iLO Remote Console. The address must be specified in IPv4 dot notation or 0. If 0 is entered, then the guest will no longer have virtual iLO Remote Console access using IP.
`-L` *console_IP_Addr_Netmask*	Specifies the IPv4 subnet mask used with the option when setting up the IP interface to be used for accessing the virtual iLO Remote Console for this guest. The address is entered in dot notation form.
`-u` *user*[:*admin* | *oper*]	Specifies a user authorization. The specified administrative level (`admin` or `oper`) is applied to the specified user.

Table 16 Options to the `hpvmmodify` Command *(continued)*

Option	Description
`-a` *rsrc*	Adds a virtual storage or network device to the virtual machine. For more information, see *hpvmresources*(5).
`-m` *rsrc*	Modifies an existing I/O resource for a virtual machine. The resource is specified as described below. You must specify the hardware address of the device to modify. The physical device portion of the `rsrc` specifies a new physical device that replaces the one in use.
`-d` *rsrc*	Deletes a virtual resource.
`-r` *amount*	Modifies the amount of memory available to this virtual machine. Specify the amount as either *amount*`M` (for megabtyes) or *amount*`G` (for gigabytes).
`-i` *package-name*	Specifies whether the virtual machine is managed by Serviceguard or gWLM (or both). For the argument, specify one or more of the following parameters: • `SG` indicates that the VSP is a Serviceguard cluster node. • `SG-`*pkgname* indicates that the VSP is a Serviceguard package. • `GWLM` indicates that the VSP is managed by gWLM. • `NONE` indicates there are no external managers. For a node that is managed by both Serviceguard and gWLM, parameters are separated with a comma. For example: `SG_host1,gWLM`. Do not specify this option. This option is used internally by Integrity VM.

Table 16 Options to the `hpvmmodify` Command *(continued)*

Option	Description
`-j [0|1]`	Specifies whether the virtual machine is a distributed guest (that is, managed by Serviceguard) and can be failed over to another cluster member running Integrity VM. Do not specify this option. This option is used internally by Integrity VM.
`-x keyword=parameter`	Specifies values for dynamic memory setting associated with the guest, including: • `dynamic_memory_control` • `ram_dyn_type` • `ram_dyn_min` • `ram_dyn_max` • `ram_dyn_target_start` • `ram_dyn_entitlement=amount` • `amr_enable={0|1}` • `amr_chunk_size=amount` • `runnable_status` • `not_runnable_reason` • `graceful_stop_timeout` • `sched_preference` • `suspend={enable | disable}` • `suspend_file=delete` Specifies settings for Online VM Migration: • `online_migration` • `migrate_init_phase_timeout` • `migrate_copy_phase_timeout` • `migrate_io_quiesce_phase_timeout` • `migrate_frozen_phase_timeout` For more information about dynamic memory, see Section 11.9 (page 180). Specifies virtual machine or vPar attributes that control their behavior: • `vm_type={vpar|shared}` • `resources_reserved={0|1}` • `active_config={0|1}`

The `hpvmmodify` command generated no warnings, so the VSP system is ready to start the virtual machine.

After you make the necessary modifications, use the `hpvmstart` command to start the virtual machine. For example:

```
# hpvmstart -P host1
(C) Copyright 2000 - 2008 Hewlett-Packard Development Company, L.P.
Initializing System Event Log
Initializing Forward Progress Log
Opening minor device and creating guest machine container
Creation of VM, minor device 2
Allocating guest memory: 2048MB
  allocating low RAM (0-40000000, 2048MB)
/opt/hpvm/lbin/hpvmapp (/var/opt/hpvm/uuids/8ba249f2-3399-11db-aacc-00306ef392e0
/vmm_config.next): Allocated 1073741824 bytes at 0x6000000100000000
    locking memory: 0-40000000
  allocating firmware RAM (ffaa0000-ffab5000, 84KB)
/opt/hpvm/lbin/hpvmapp (/var/opt/hpvm/uuids/8ba249f2-3399-11db-aacc-00306ef392e0
/vmm_config.next): Allocated 860 bytes at 0x6000000140000000
    locked SAL RAM: 00000000ffaa0000 (4KB)
    locked ESI RAM: 00000000ffaa1000 (4KB)
```

```
    locked PAL RAM: 00000000ffaa4000 (4KB)
    locked Min Save State: 00000000ffaa5000 (1KB)
RAM alignment: 40000000
Memory base low : 6000000100000000
Memory base FW  : 6000000140000000
Loading boot image
Image initial IP=102000 GP=62C000
Initialize guest memory mapping tables
Starting event polling thread
Starting thread initialization
Daemonizing....
hpvmstart: Successful start initiation of guest 'host1'
```

The virtual machine `host1` is started. Now the guest operating system must be installed.

NOTE: You might receive the following note-level message in the `/var/opt/hpvm/common/command.log` file under certain circumstances:

```
mm/dd/yy hh:mm:ss|NOTE|host|root|Unable to open file '/dev/rdisk/diskxxx' - Device busy.
```

This note might be logged if:

- A guest is configured with an attached `avio_stor` burner:

  `resource: -a burner:avio_stor::[b,d,t]:attach:pass-through-device-path`

- The guest is then booted to EFI.
- Then the `hpvmmodify` command is run to add a device or remove a device other than the burner.

You may safely ignore this message.

For information about creating HP-UX guests, see Chapter 8.

7.5 Cloning virtual machines

Once you have created a guest, you can quickly and easily create additional guests by using the `hpvmclone` command. Like the `hpvmcreate`, `hpvmmigrate`, and `hpvmmodify` commands, the `hpvmclone` command accepts the command options listed in Table 11 (page 75) for specifying virtual devices, network interfaces, and other virtual machine characteristics. This allows you to create new guests with similar characteristics but different virtual resources.

Table 17 describes the options you can use with the `hpvmclone` command.

Table 17 Options to the `hpvmclone` Command

Option	Description
`-P` *vm-name*	Specifies the name of the existing virtual machine to be cloned. You must specify either the `-P` option or the `-p` option.
`-p` *vm-number*	Specifies the number of the existing virtual machine to be cloned. You must specify either the `-P` option or the `-p` option.
`-K` *console_IP_Addr*	Specifies the IP address used to connect to the guest's virtual iLO Remote Console. The address must be specified in IPv4 dot notation or 0. If 0 is entered, then the guest will no longer have virtual iLO Remote Console access using IP.
`-L`*console_IP_Addr_Netmask*	Specifies the IPv4 subnet mask used with the option when setting up the IP interface to be used for accessing the virtual iLO Remote Console for this guest. The address is entered in dot notation form.
`-N` *clone-vm-name*	Specifies the name of the new virtual machine (the clone). The *clone-vm-name* can be up to 256 alphanumeric characters. The same virtual machine name cannot already exist on the same VSP system.

Table 17 Options to the `hpvmclone` Command *(continued)*

Option	Description

`-e percent[:max_percent] | -E cycles[:max_cycles]`	Specifies the virtual machine's CPU entitlement in CPU cycles. To specify the percentage of CPU power, enter the following option:
	`-e percent[:max_percent]`
	To specify the clock cycles, enter one of the following options:
	`-E cycles[:max_cycles]M (for megahertz)`
	`-E cycles[:max_cycles]G (for gigahertz)`

`-l vm_label`	Specifies a descriptive label for this virtual machine. The label can contain up to 256 alphanumeric characters, including A-Z, a-z, 0-9, the dash (—), the underscore (_), and the period (.). To include spaces, the label must be quoted (" ").
`-B start_attr`	Specifies the startup behavior of the virtual machine. For *`start_attr`*, enter one of the following keywords: `auto`: Automatically starts the virtual machine when the VSP is started (autoboot). `manual`: The virtual machine is not started automatically. Use the `hpvmstart` command to start the virtual machine manually.
`-O os_type[:version]`	Specifies the type and version of the operating system running on the virtual machine. For the *`os_type`* parameter, you can specify one of the following (case-insensitive) values: `hpux`
`-a rsrc`	Creates a virtual device for the new virtual machine (clone). Specify the virtual and physical device information for *`rsrc`* . For information about forming a virtual storage device specification, see Chapter 9. For information about forming a virtual network device specification, see Chapter 10.
`-d rsrc`	Deletes a virtual device that is defined on the existing virtual machine in the clone virtual machine configuration. Specify the virtual and physical device information for *`rsrc`* . For information about forming a virtual storage device specification, see Chapter 9. For information about forming a virtual network device specification, see Chapter 10.
`-m rsrc`	Modifies a virtual device that is defined on the existing virtual machine in the clone virtual machine configuration. Specify the virtual and physical device information for *`rsrc`* . For information about forming a virtual storage device specification, see Chapter 9. For information about forming a virtual network device specification, see Chapter 10.
`-F`	Suppresses all resource-conflict checks and associated warning messages (force mode). Use force mode for troubleshooting purposes only. **NOTE:** The `-F` option is deprecated in Integrity VM commands; this option should be used only at the direction of HP Support.
`-c number_vcpus`	Specifies the number of vCPUs this virtual machine detects at boot time. If unspecified, the number defaults to one. The maximum number of vCPUs that you can allocate to a virtual machine is the number of physical processors on the VSP system.

Table 17 Options to the `hpvmclone` Command *(continued)*

Option	Description
`-r amount`	Specifies the amount of memory available to this virtual machine. Specify the amount as either *amount*`M` (for megabtyes) or *amount*`G` (for gigabytes).
`-S amount`	Specifies that the cloned guest must share the same virtual LAN (VLAN) ports as the source guest. By default, the `hpvmclone` command allocates VLAN ports that are different from those allocated to the guest that is the source of the clone operation. For more information about using VLANS on virtual machines, see Section 10.4 (page 152).
`-g group[:{admin|oper}]`	Specifies a group authorization. The specified administrative level (`admin` or `oper`) is applied to the specified user group.
`-u user[:{admin|oper}]`	Specifies a user authorization. The specified administrative level (`admin` or `oper`) is applied to the specified user group.
`-x keyword=parameter`	Specifies values for dynamic memory setting associated with the guest, including: • `dynamic_memory_control` • `ram_dyn_type` • `ram_dyn_min` • `ram_dyn_max` • `ram_dyn_target_start` • `ram_dyn_entitlement=`*amount* • `amr_enable={0|1}` • `amr_chunk_size=`*amount* • `graceful_stop_timeout` • `mac_address` • `sched_preference` • `serial_number` • `tunables` • `suspend={enable | disable}` • `suspend_file=delete` For Online VM Migration: • `online_migration` • `migrate_frozen_phase_timeout` • `migrate_copy_phase_timeout` • `migrate_io_quiesce_timeout` • `migrate_init_phase_timeout` For more information about dynamic memory, see Section 11.9 (page 180). Specifies virtual machine or vPar attributes that control their behavior: • `vm_type={vpar|shared}` • `resources_reserved={0|1}` • `active_config={0|1}` To specify the serial number of the new virtual machine, enter `serial_number={new | same}`

For example, to clone the virtual machine named `host3`, to create a new virtual machine named `clone1`, enter the following commands. First display the current guest status on the VSP:

```
# hpvmstatus
[Virtual Machines]
Virtual Machine Name VM #  OS Type State     #VCPUs #Devs #Nets Memory  Runsysid
==================== ===== ======= ========= ====== ===== ===== ======= ========
```

```
host1                2 HPUX    On (OS)        1     1     1    2 GB        0
host2                3 UNKNOWN Off            1     1     1    1 GB        0
host3                4 HPUX    Off            1     1     1    2 GB        0
```

You can create a clone of `host3` by entering the following command. The new virtual machine is named `clone1`:

```
# hpvmclone -P host3 -N clone1
```

To see the results of the command, enter the `hpvmstatus` command again:

```
# hpvmstatus
[Virtual Machines]
Virtual Machine Name VM #  OS Type State     #VCPUs #Devs #Nets Memory  Runsysid
==================== ===== ======= ========= ====== ===== ===== ======= ========
host1                2 HPUX    On (OS)        1     1     1    2 GB        0
host2                3 UNKNOWN Off            1     1     1    1 GB        0
host3                4 HPUX    Off            1     1     1    2 GB        0
clone1               5 HPUX    Off            1     1     1    2 GB        0
```

The `hpvmclone` command creates a copy of an existing virtual machine and its configuration information. This command copies the configuration files of the existing guest. It does not copy the actual data and software associated with the guest. Use the `-b` option to specify a storage device to be physically duplicated in the cloning process. The `clone_vm_name` must not already exist on the same VSP.

The new virtual machine's configuration information can be modified from the original configuration file by using command options. If no options are specified, all original parameters are retained. This will cause resource conflicts if both the original and clone virtual machines are booted together.

Resources are checked to determine whether the virtual machine could boot by itself on the server. Any problems are reported as WARNINGS. These warnings will not prevent the new virtual machine from being created. These conditions will, however, prevent the guest from starting.

Backing storage devices (for example, directories and files) cannot be shared, and therefore they cannot be used by two running guests at the same time. In this case, you must either enter a different backing store, or run only one of the guests at a time. For more information, see "Creating virtual storage devices" (page 107).

Use the `-b` option to specify a storage device to be physically duplicated in the cloning process. This feature allows the user to specify any number of storage devices and supports all of the possible physical device types (`disk`, `lv`, and `file`).

Because there is no guarantee that other virtual machines would be running at the same time the new virtual machine would be running, use the following command to check the device for dependents:

```
# hpvmdevmgmt -l entry_name
```

For more information about the `hpvmdevmgmt` command and the guest device management database, see Chapter 9.

7.6 Stopping virtual machines

To stop a running virtual machine, use the `hpvmstop` command. You must confirm this command. Table 18 describes the options to the `hpvmstop` command:

Table 18 Options to the `hpvmstop` Command

Option	Description
`-P` *vm-name*	Specifies the name of the virtual machine.
`-p` *vm_number*	Specifies the number of the virtual machine. To display the virtual machine number, enter the `hpvmstatus` command.
`-a`	Specifies all the virtual machines that are running. You must also specify the `-F` option.
`-h`	Performs a hard stop on the virtual machine, similar to a power failure. This is the default.

Table 18 Options to the `hpvmstop` Command *(continued)*

Option	Description
`-g`	Performs a graceful shutdown on the virtual machine.
`-F`	Forces the command to act without requiring confirmation. **NOTE:** The `-F` option is deprecated in Integrity VM commands; this option should be used only at the direction of HP Support.
`-Q`	Performs the operation without requiring you to confirm the command.
`-q`	Makes certain scripted operations less verbose (quiet mode).

For example, the following command stops the virtual machine named `host1`. The `hpvmstatus` command shows that the virtual machine is `Off`.

```
# hpvmstop -P host1
hpvmstop: Stop the virtual machine 'host1'? [n/y]: y

# hpvmstatus

[Virtual Machines]
Virtual Machine Name VM #  OS Type State     #VCPUs #Devs #Nets Memory
==================== ===== ======= ========= ====== ===== ===== =======
config1                  1 HPUX    Off             1     5     1  512 MB
config2                  2 HPUX    Off             1     7     1    1 GB
guest1                   5 HPUX    On (OS)         1     5     1    1 GB
host1                   12 UNKNOWN Off             1     0     0    2 GB
```

The default action of this command (if you press **Enter**) is to not perform the command operation. To continue the operation, you must enter **`y`**.

To enter the command without requiring a confirmation (for example, in a script), enter the following command:

```
# hpvmstop -P host1 -Q
#
```

To quickly shut down all three virtual machines that are running on the VSP, enter the following command:

```
# hpvmstop -a -F
Stopping virtual machine host1
Stopping virtual machine host2
Stopping virtual machine host3
```

NOTE: When stopping a guest that is running a heavy I/O load, the `hpvmstop` command can exhaust its timeout allotted for the stop and exit. When this happens, the SIGKILL has been sent to the running hpvmapp process and will be received by that process when pending I/Os complete. The SIGKILL then terminates the guest.

This is expected behavior for an I/O intensive process receiving a SIGKILL. This behavior is not specific to Integrity VM, but is how the signal-delivery mechanism works in the HP-UX operating system.

You can also use the `hpvmconsole` command to force the virtual machine to shut down. However, after you install the guest operating system, you should use the standard operating system commands and procedures on the guest to shut it down.

NOTE: To stop a guest, HP recommends that you perform an operating system shutdown from a privileged account on the guest using their native operating system commands. If the guest is not responding, use the `hpvmstop -g` command on the VSP. Do not stop a guest by killing the `hpvmapp` process.

7.7 Removing virtual machines

To remove a virtual machine from the VSP, use the `hpvmremove` command. By default, you are required to confirm this action. Table 19 describes the options to the `hpvmremove` command.

Table 19 Options to the `hpvmremove` Command

Option	Description
`-P vm-name`	Specifies the name of the virtual machine. You must include either the `–P` or `–p` option.
`-p vm_number`	Specifies the number of the virtual machine. To display the virtual machine number, enter the `hpvmstatus` command.
`-F`	Forces the command to act regardless of errors. **NOTE:** The `-F` option is deprecated in Integrity VM commands; this option should be used only at the direction of HP Support.
`-Q`	Performs the command without requiring user input to confirm.

For example, the following command removes the virtual machine named `host1`. The subsequent `hpvmstatus` command shows that `host1` is gone:

```
# hpvmremove -P host1
hpvmremove: Remove the virtual machine 'host1'? [n/y]: y
# hpvmstatus
[Virtual Machines]
Virtual Machine Name VM #  OS Type State     #VCPUs #Devs #Nets Memory
==================== ===== ======= ========= ====== ===== ===== =======
config1                  1 HPUX    Off             1     5     1  512 MB
config2                  2 HPUX    Off             1     7     1    1 GB
guest1                   5 HPUX    On (OS)         1     5     1    1 GB
```

The default action of this command (if you press **Enter**) is to not perform the command action. To perform the action, you must enter **`y`**.

This command removes `host1` and all its configuration files, and restores any resources allocated to that guest to the VSP's pool of available resources. (Any guest operating system and application data on the VSP storage devices are not affected.)

To remove the guest without requiring user confirmation (for example, in a script), enter the following command:

```
# hpvmremove -P host1 -Q
```

7.8 Troubleshooting virtual machine creation problems

If you encounter problems with creating virtual machines, report them through your support channel. For information about collecting information to report the problem, see Chapter 13.

The following section describes a problem that might be encountered during virtual machine creation.

7.8.1 Configuration error on starting the virtual machine

When you start the virtual machine, the following message is displayed:

```
Configuration error: Device does not show up in guest
```

If you encounter this type of problem:

1. Verify that the path name to the file-backing store is correct and that the physical storage device is mounted.
2. Verify that the size of the physical storage device is divisible by 512 bytes (for a disk device) or 2048 (for a DVD device).
3. Modify the virtual machine using the `hpvmmodify` command.

8 Installing the HP-UX guest operating system and software

To create HP-UX guests, install the HP-UX operating system on the vpar or virtual machine. To install the HP-UX guest operating system, follow the procedures in the following sections:

- Installing the HP-UX guest operating system
- Installing the HP-UX vPar/VM VirtualBase software
- Troubleshooting HP-UX guest creation

NOTE: The following topics apply to both vPars and Integrity VM.

8.1 Installing the HP-UX guest operating system

You can install the HP-UX 11i v3 operating system as a guest OS, See the *HP-UX vPars and Integrity VM V6.1 Release Notes*, *Chapter 11 vPars and Integrity VM Support Policy* for a list of supported versions of the HP-UX operating system.

To install the HP-UX operating system on the virtual machine, follow this procedure:

1. Start the virtual machine from the VSP administrator account using the `hpvmstart` command. For example, to start the virtual machine called `host1`, enter the following command. The `hpvmstatus` command shows that the virtual machine is started.

```
# hpvmstart -P host1
(C) Copyright 2000 - 2008 Hewlett-Packard Development Company, L.P.
Initializing System Event Log
Initializing Forward Progress Log
Opening minor device and creating guest machine container
Creation of VM, minor device 2
Allocating guest memory: 2048MB
  allocating low RAM (0-40000000, 2048MB)
/opt/hpvm/lbin/hpvmapp (/var/opt/hpvm/uuids/8ba249f2-3399-11db-aacc-00306ef392e0
/vmm_config.next): Allocated 1073741824 bytes at 0x6000000100000000
    locking memory: 0-40000000
  allocating firmware RAM (ffaa0000-ffab5000, 84KB)
/opt/hpvm/lbin/hpvmapp (/var/opt/hpvm/uuids/8ba249f2-3399-11db-aacc-00306ef392e0
/vmm_config.next): Allocated 860 bytes at 0x6000000140000000
    locked SAL RAM: 00000000ffaa0000 (4KB)
    locked ESI RAM: 00000000ffaa1000 (4KB)
    locked PAL RAM: 00000000ffaa4000 (4KB)
    locked Min Save State: 00000000ffaa5000 (1KB)
RAM alignment: 40000000
Memory base low : 6000000100000000
Memory base FW  : 6000000140000000
Loading boot image
Image initial IP=102000 GP=62C000
Initialize guest memory mapping tables
Starting event polling thread
Starting thread initialization
Daemonizing....
hpvmstart: Successful start initiation of guest 'host1'

# hpvmstatus

[Virtual Machines]
Virtual Machine Name VM #  OS Type State     #VCPUs #Devs #Nets Memory  Runsysid
==================== ===== ======= ========= ====== ===== ===== ======= ========
config1                  1 HPUX    Off            1     5     1  512 MB         0
config2                  2 HPUX    Off            1     7     1    1 GB         0
guest1                   5 HPUX    On (OS)        1     5     1    1 GB         0
host1                   12 UNKNOWN On (EFI)       1     0     0    2 GB         0
```

2. To boot the guest from the virtual console, enter the following command:

```
# hpvmconsole -P host1
 vMP MAIN MENU
```

```
        CO: Console
        CM: Command Menu
        CL: Console Log
        SL: Show Event Logs
        VM: Virtual Machine Menu
        HE: Main Help Menu
         X: Exit Connection

[host1] vMP>
```

The `hpvmconsole` command opens the virtual machine console. From the virtual console, you can control the virtual machine just as if it were a physical Integrity server.

3. In response to the virtual machine prompt, enter the `co` command:

```
[host1] vMP> co

EFI Boot Manager ver 1.10 [14.62] [Build: Wed Jun  4 11:37:36 2008]

Please select a boot option

    EFI Shell [Built-in]
    Boot option maintenance menu


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

4. Select `Boot option maintenance menu`.

```
EFI Boot Maintenance Manager ver 1.10 [14.62]
Main Menu. Select an Operation

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

        Manage BootNext setting
        Set Auto Boot TimeOut

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

        Cold Reset
        Exit
```

5. Select `Add a Boot Option`.

```
EFI Boot Maintenance Manager ver 1.10 [14.62]

Add a Boot Option.  Select a Volume

    Removable Media Boot [Acpi(PNP0604,0)]
    Load File [Acpi(PNP0A03,0)/Pci(1|0)/Mac(763AE48F393F)]
    Load File [EFI Shell [Built-in]]
    Legacy Boot
    Exit
```

To install from virtual DVD, select `Removable Media Boot`.

To install from the Ignite-UX server, select the entry with your MAC address. For example:

```
Device Path Acpi(PNP0A03,0)/Pci(1|0)/Mac(763AE48F393F)

Enter New Description:  lan0boot
New BootOption Data. ASCII/Unicode strings only, with max of 240 characters
Enter BootOption Data Type [A-Ascii U-Unicode N-No BootOption] :  N
```

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

6. Exit the EFI Boot Maintenance Management screen to return to the EFI Boot Manager screen. Boot from the new boot entry, indicated by the virtual machine's MAC address:.

```
EFI Boot Maintenance Manager ver 1.10 [14.62]

Add a Boot Option.  Select a Volume

    Removable Media Boot [Acpi(PNP0604,0)]
    Load File [Acpi(PNP0A03,0)/Pci(1|0)/Mac(763AE48F393F)]
    Load File [EFI Shell [Built-in]]
    Legacy Boot
    Exit
```

The installation process continues just as if the virtual machine were an Ignite-UX client.

When the basic installation process is complete, the software is copied from the distribution media to the guest's disk. Then the operating system reboots. If this reboot fails, restart it, as follows:

1. Enter the EFI shell by enter the `co` command at the virtual machine console prompt:

```
[host1] vMP> CO


        (Use Ctrl-B to return to vMP main menu.)


- - - - - - - - - - Prior Console Output - - - - - - - - - -

Shell>
```

2. Enter `fs0:`

```
Shell> fs0:
```

3. Enter `hpux`:

```
fs0\> hpux
```

The guest boots from `fs0`.

If you used a DVD to install the guest operating system, remove the virtual DVD, as follows:

1. Determine the bus, device, and target ID by entering the following command:

```
# hpvmstatus -P host1
```

2. Delete the virtual DVD by entering the following command (substituting the correct PCI bus, slot, and target number for 0,0,0):

```
# hpvmmodify -P host1 -d dvd:avio_stor::0,0,0
```

3. If necessary, restart the guest to remove the DVD from the guest configuration.

8.2 Do not create golden images of the VSP for guest installation

Do not use the VSP to create golden images to be used for guest OS installations using Ignite-UX.

An Integrity system can be used to create a golden image suitable for OS installation on a virtual machine, provided it has all of the VSP software completely removed. To do so, remove both BB068AA bundle and the VirtualBase bundle:

```
# swremove -x autoreboot=true BB068AA VirtualBase
```

Before using the system to create a golden image, verify that neither of these bundles are installed. That is, errors should result when querying the system with `swlist`:

```
# swlist BB068AA VirtualBase
# Initializing...
```

```
# Contacting target "foo"...
ERROR:    Software "BB068AA" was not found on host "foo:/".
ERROR:    Software "VirtualBase" was not found on host "foo:/".
```

For more information about using Ignite-UX golden images, see the *Ignite-UX Administration Guide*.

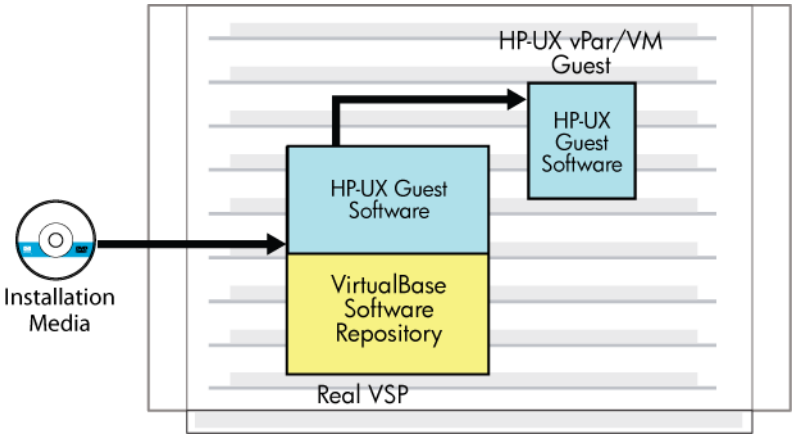
8.3 Installing VirtualBase on a vPar/VM

After you install the guest OS, you might need to install or update VirtualBase on the vPar/VM . Without VirtualBase, you cannot use dynamic memory with VMs and with VirtualBase, the vPar/VM can be managed by Integrity Virtual Server Manager. Additionally, you can use Matrix OE V7.0 to manage VMs.

NOTE: VirtualBase is installed by default with the HP-UX 11i v3 March 2012 release.

Figure 4 illustrates the process. A copy of VirtualBase is installed onto the VSP system when vPars and Integrity VM is installed or upgraded.

Figure 4 Installing VirtualBase on a vPar/VM



A copy of VirtualBase is stored on the VSP system in the `/opt/hpvm/guest-images` directory. A subdirectory contains an SD tape depot with VirtualBase for the HP-UX operating system, as shown in the following example:

```
# cd /opt/hpvm/guest-images
# ls
common    hpux
```

Before installing the VirtualBase bundle, preview the install task allowing the installation analysis. This provides the opportunity to identify and address any warnings that might result from this preview before proceeding with the installation. For example, the analysis phase includes checks for installation of the appropriate AVIO drivers on the guest. To preview the installation, use the `-p` option of `swinstall` as shown in the following example:

```
swinstall -p -x autoreboot=true -s path to hpvm_guest_depot.11iv#.sd VirtualBase
```

Installing the vPars/VM VirtualBase software kit causes the vPar and VM to reboot.

NOTE: If you receive the following messages during boot or reboot of a vPars or VM, you can safely ignore them:

```
montecito_proc_features: PROC_GET_FEATURES returned 0xfffffffffffffff8
CPU00 failed to synchronize its interval timer with CPU03 within 46 Ticks
CPU00 failed to synchronize its interval timer with CPU04 within 46 Ticks
CPU00 failed to synchronize its interval timer with CPU06 within 46 Ticks
CPU00 failed to synchronize its interval timer with CPU07 within 46 Ticks
CPU00 failed to synchronize its interval timer with CPU02 within 45 Ticks
CPU00 failed to synchronize its interval timer with CPU04 within 41 Ticks
CPU00 failed to synchronize its interval timer with CPU06 within 41 Ticks
CPU00 failed to synchronize its interval timer with CPU07 within 41 Ticks
CPU00 failed to synchronize its interval timer with CPU03 within 43 Ticks
Installing Socket Protocol families AF_INET and AF_INET6
```

Each subdirectory in `/opt/hpvm/guest-images` contains a `README.txt` file that describes how to install the software for that type of vPar/VM. For information about any additional software updates that you should also install on your vPar/VM, see the *HP-UX vPars and Integrity VM V6.1 Release Notes* at http://www.hp.com/go/hpux-hpvm-docs.

8.4 Troubleshooting HP-UX guest creation

The following section describes a problem that might occur during HP-UX guest installation.

8.4.1 The guest hangs in the EFI shell

The guest hangs in the EFI when you are starting the guest and you get the following message:

```
Shell> \efi\hpux\hpux

'\efi\hpux\hpux' not found
Exit status code: Invalid Parameter
```

The EFI boot parameters were probably not set up correctly during guest operating system installation. Choose the correct EFI partition from which to boot. For example:

```
Shell> fs3:
fs3:\> hpux
```

Installation continues from the specified partition.

9 Creating virtual storage devices

This chapter describes what vPars and Integrity VM V6.1 storage is, how to configure it, and how to use it. The topics included in this chapter are:

- Introduction to vPar/VM storage
- Configuring vPar/VM storage
- Using vPar/VM storage

NOTE: The information in this chapter also applies to vPars, as well as VMs.

9.1 Introduction to vParVM storage

The way you configure and manage vPar/VM storage affects the way virtual machines perform. To get the most benefit from using virtual machines and virtual partitions, learn how vPars and Integrity VM V6.1 makes storage devices available to virtual machines and virtual partitions. The following sections describe:

- Storage goals
- Storage architectures
- Storage implementations

9.1.1 Storage goals

To successfully configure and manage virtual storage, it is helpful to understand the basic goals of the vPars and Integrity VM V6.1 storage subsystem, including:

- Storage utilization
- Storage availability
- Storage performance
- Storage security
- Storage configurability

9.1.1.1 Storage utilization

The main purpose of vPars and Integrity VM V6.1 is to increase system resource utilization on Integrity servers. The vPar/VM storage subsystem meets this goal by permitting multiple vPars/VMs to share a variety of physical storage adapters and devices that are available on an Integrity server. Furthermore, the vPars/VM storage subsystem allows for a single storage LUN on the VSP to be carved up into smaller entities that can be used as separate individual disks or DVDs on the virtual platform.

9.1.1.2 Storage availability

Like HP Integrity servers, it is expected that virtual machines and virtual partitions will have several different storage device types available for use. The vPar/VM storage subsystem provides for disks, DVDs, tapes and media changers to be used by a guest OS. Additionally, the way that virtualization abstracts the physical hardware provides a common supportable interface with which a guest OS can interact. Because a guest OS accesses only vPars and Integrity VM virtual hardware, the guest OS can use physical hardware that it does not support on an Integrity server.

9.1.1.3 Storage performance

Each release of the vPar/VM storage subsystem strives to improve performance. Performance is improved in each release by lowering costs of virtualization, exploiting new features in the VSP,

and tuning operating systems for the virtual platform. At the same time, vPars and Integrity VM V6.1 provides more virtualization choices to VSP administrators, so that they can find the best balance between virtualization and performance to meet their needs.

9.1.1.4 Storage security

To avoid problems while supporting multiple vPars/VMs on one physical machine, vPars and Integrity VM V6.1 isolates each virtual machine and virtual partition. Using Integrity VM commands, the VSP administrator determines the physical storage resources that each virtual machine and virtual partition can access. This storage isolation is maintained by the vPar/VM storage subsystem through DMA boundary checks on each vPar/VM I/O operation, thereby ensuring that one virtual machine or virtual partition does not access the memory of another.

9.1.1.5 Storage configurability

VSP administrators expect the vPars/V<s to be as easily configurable as HP Integrity servers. The vPar/VM storage subsystem allows for easy changes of the storage devices through vPars and Integrity VM commands. Using these commands, the VSP administrator dynamically adds, deletes, and modifies storage devices on virtual machines and virtual partitions. Guest administrators can change some storage, limited in scope by the VSP administrator, using the virtual console.

9.1.2 Storage architectures

To provide the flexibility required to meet a variety of data center needs, the vPar/VM storage subsystem consists of two storage architectures, shared I/O and attached I/O.

9.1.2.1 Shared I/O

The shared I/O architecture is a means by which a vPar/VM accesses an entirely virtualized storage subsystem provided by vPars and Integrity VM V6.1. The vPar/VM storage subsystem emulates real hardware to the vPar/VM while interacting with the VSP to complete the vPar/VM I/O operation to the VSP storage entity. This abstraction provides the ability of a VSP administrator to share physical VSP storage hardware across multiple virtual machines and to allocate that storage at sub-LUN levels.

The sharing of individual storage LUNs is accomplished by dividing a VSP LUN into smaller parts, like logical volumes, or files. Each of these sub-LUN VSP entities can then be used as media for separate virtual storage devices. The vPars/VMs access the virtual storage devices as real storage devices, with no knowledge that the virtual storage media is actually a sub-LUN VSP entity.

The way the virtual storage media is accessed by the vPar/VM storage subsystem allows vPars/VMs to share physical VSP storage adapters. All virtual storage media is accessed through user-defined interfaces on the VSP. The VSP maintains complete control of the physical hardware and handles the vPar/VM I/O operations just as it would be handled for any other user application. Thus, just as hardware is shared among normal applications running on the VSP, vPar/VM I/O is shared across the physical storage as well.

This architecture also provides for whole LUNs to be virtualized. While this does not increase storage utilization, it does provide higher storage availability. Because the LUN is virtualized, the guest OS does not have to support the physical VSP LUN. It only has to be able to support the virtualized version of it. Thus by using shared I/O, a vPar/VM can run with any physical hardware that is supported by the VSP.

Finally, all vPar/VM I/O requests in shared I/O are processed by virtual adapters. A virtual adapter is either an emulation of a real adapter that a native guest OS driver accesses as real hardware, or a special driver loaded into the guest OS. In either case, the virtual adapter uses internal vPar/VM storage subsystem calls to handle communication of vPar/VM I/O to the virtual devices. This connection between the virtual adapter and the virtual devices need not resemble anything in an HP Integrity server system. It is emulated so that the vPar/VM does not know the difference.

9.1.2.2 Attached I/O

Attached I/O allows a vPar/VM to access to a VSP LUN directly. In this architecture, the vPar/VM storage subsystem attaches a LUN on the VSP to a virtualized storage adapter. A LUN can be a disk, DVD, tape, media changer, or other peripheral device types. Because attached I/O does not require device virtualization, the performance of attached I/O might be better than shared I/O.

The main difference between shared I/O and attached I/O is the degree to which a physical storage subsystem is virtualized. In shared I/O, an entire storage subsystem is virtualized. Therefore, all physical adapters on the VSP and all the storage connected to those adapters may be shared among vPars/VMs. In attached I/O, only the storage adapter is virtualized. Therefore, only the VSP physical storage adapters may be shared. At least one LUN, the attached LUN, cannot be shared. It is owned and solely controlled by the vPar/VM to which it is attached.

To provide the vPar/VM with complete control over attached devices, the vPar/VM storage subsystem interprets I/O requests from the guest device drivers into I/O requests that can be completed by the VSP storage subsystem on the vPar/VM's behalf. In the process, the VSP storage subsystem sends all the actual data and responses back the vPar/VM device drivers. With all this data, the vPar/VM device driver is in complete control over the device. As such, the guest OS must have built-in support for the attached VSP LUN to use it.

Attached I/O uses a virtual adapter to communicate with the guest OS and the attached LUN. The virtual adapter either can be an emulation of a real adapter or it can be controlled by a special driver loaded into the guest OS. Either solution produces a virtual adapter that communicates with both virtual devices and attached physical devices.

9.1.3 Attached device support in AVIO

AVIO storage supports attached devices (tapes, changers and burners) on HP-UX 11i v2 and HP-UX 11i v3 guests. Attached devices configured using AVIO (`avio_stor` adapter)::

- Allow sharing of tapes, changers, and burners among multiple guests and host
- Support of USB 2.0 DVD burners
- Improved performance

AVIO (*avio_stor* adapter type) supports USB 2.0 DVD burners.

To identify USB CD/DVD devices, use the `ioscan -fun` command.

NOTE: Because vPars/VM might do four to six calls to `open()` on a DVD when accessing it, and `hpvmcreate` or `hpvmmodify` command might take more than a minute to complete when there is no media in the drive. Example commands that could appear to hang are:

```
# hpvmcreate -P guest -a dvd:avio_stor::disk:/dev/rdisk/disk5
# hpvmcreate -P guest -a dvd:avio_stor::null:/dev/rdisk/disk5
# hpvmmodify -P guest -a dvd:avio_stor::disk:/dev/rdisk/disk5
# hpvmmodify -P guest -a dvd:avio_stor::null:/dev/rdisk/disk5
```

9.1.3.1 Resource syntax

AVIO storage requires the hardware path of the lunpath class (displayed only in ioscan with the `-N` option) to be specified in place of device special files in a resource specifier. Here is the syntax of the resource specifier:

`tape|changer|burner:avio_stor:bus,device,target:attach_path:`*`new style lunpath hardware path of the attached device`*

The following example shows the resource specifier with the `avio_stor` adapter:

```
tape:avio_stor:0,4,0:attach_path:0/7/1/1.0x500104f00048b29e.0x0
```

To find the lunpath hardware path of a device, see Section 6.1.3.2. Once the lunpath hardware path is obtained, use the `hpvmmodify` command to add the tape to a guest. For example, use the following command to assign the second lunpath to guest1:

```
# hpvmmodify -P guest1 -a tape:avio_stor::attach_path:0/7/1/1.0x500104f00048b29e.0x0
```

The following examples add, delete, and modify attached devices:

Add

```
# hpvmmodify -P guest1 -a tape:avio_stor:0,5,0:attach_path:0/1/1/0.0x50060b0000332254.0x0
# hpvmmodify -P guest1 -a changer:avio_stor:0,5,1:attach_path:0/1/1/0.0x50060b0000332253.0x0
# hpvmmodify -P guest1 -a burner:avio_stor:0,5,2:attach_path:0/1/1/0.0x50060b0000332252.0x0
```

Delete

```
# hpvmmodify -P guest1 -d tape:avio_stor:0,5,0:attach_path:0/1/1/0.0x50060b0000332254.0x0
# hpvmmodify -P guest1 -d changer:avio_stor:0,5,1:attach_path:0/1/1/0.0x50060b0000332253.0x0
# hpvmmodify -P guest1 -d burner:avio_stor:0,5,2:attach_path:0/1/1/0.0x50060b0000332252.0x0
```

Modify

```
# hpvmmodify -P guest1 -m tape:avio_stor:0,5,0:attach_path:0/1/1/0.0x50060b0000332254.0x0
# hpvmmodify -P guest1 -m changer:avio_stor:0,5,1:attach_path:0/1/1/0.0x50060b0000332253.0x0
# hpvmmodify -P guest1 -m burner:avio_stor:0,5,2:attach_path:0/1/1/0.0x50060b0000332252.0x0
```

NOTE: When a guest application uses an attached device, the other guest's (or VSP) access to the attached device path is denied.

9.1.3.2 Finding the lunpath hardware path

To obtain the lunpath hardware path for an attached device, use the `ioscan` command with the `-m lun` option. For example, in this case of a tape having two paths. the `ioscan` output looks like this:

```
# ioscan -m lun /dev/rtape/tape1_BEST
Class     I  Lun H/W Path  Driver  S/W State   H/W Type     Health  Description
===================================================================
tape      1  64000/0xfa00/0x0   estape  CLAIMED     DEVICE       online  STK     T9940B
             0/1/1/1.0x500104f00048b29d.0x0
             0/7/1/1.0x500104f00048b29e.0x0
                      /dev/rtape/tape1_BEST    /dev/rtape/tape1_BESTn
                      /dev/rtape/tape1_BESTb   /dev/rtape/tape1_BESTnb
```

You can use the `ioscan` command to find the device special file corresponding to a lunpath hardware path. For example, in the previous case, to find the device special file for lunpath hardware path 0/7/1/1.0x500104f00048b29e.0x0, invoke the following `ioscan` command line:

```
# ioscan -kfnNH 0/7/1/1.0x500104f00048b29e.0x0
Class     I  H/W Path  Driver S/W State   H/W Type     Description
===================================================================
lunpath  21  0/7/1/1.0x500104f00048b29e.0x0  eslpt   CLAIMED     LUN_PATH     LUN path for tape1
```

The DSF for tape1 is /dev/rtape/tape1_BEST*.

9.1.3.3 Sharing an attached device

Attached devices can be shared among multiple vPars/VMs in a VSP using a single physical HBA port (initiator) or multiple physical HBA ports (initiators) in the VSP. This section describes how to share attached devices. To share a tape device, do the following:

1. Identify the tape device(s):

```
# ioscan -funNC tape
Class     I  H/W Path  Driver S/W State   H/W Type     Description
  ===================================================================
  tape      5  64000/0xfa00/0x1  estape   CLAIMED     DEVICE       HP      Ultrium 3-SCSI
                        /dev/rtape/tape5_BEST    /dev/rtape/tape5_BESTn
                        /dev/rtape/tape5_BESTb   /dev/rtape/tape5_BESTnb
  tape      6  64000/0xfa00/0x3  estape   CLAIMED     DEVICE       STK     T9840B
                        /dev/rtape/tape6_BEST    /dev/rtape/tape6_BESTn
                        /dev/rtape/tape6_BESTb   /dev/rtape/tape6_BESTnb
```

2. This system has two tape drives. Identify the lunpaths:

```
# ioscan -m lun /dev/rtape/tape5_BEST
Class     I  Lun H/W Path  Driver  S/W State   H/W Type     Health  Description
```

```
================================================================
tape      5  64000/0xfa00/0x1   estape  CLAIMED     DEVICE       online  HP      Ultrium 3-SCSI
             0/5/0/0/0/0.0x500110a0008b9de2.0x0
                      /dev/rtape/tape5_BEST    /dev/rtape/tape5_BESTn
                      /dev/rtape/tape5_BESTb   /dev/rtape/tape5_BESTnb

# ioscan -m lun /dev/rtape/tape6_BEST
Class     I  Lun H/W Path  Driver  S/W State   H/W Type     Health  Description
================================================================
tape      6  64000/0xfa00/0x3   estape  CLAIMED     DEVICE       online  STK     T9840B
             0/4/1/0.0x500104f0004732d9.0x0
             0/4/1/1.0x500104f0004732d9.0x0
             0/4/1/0.0x500104f0004732da.0x0
             0/4/1/1.0x500104f0004732da.0x0
                      /dev/rtape/tape6_BEST    /dev/rtape/tape6_BESTn
                      /dev/rtape/tape6_BESTb   /dev/rtape/tape6_BESTnb
```

Device `tape5` is connected to the VSP using a single HBA port (initiator). It has one lunpath through initiator (0/5/0/0/0/0). Device `tape6` is connected to the VSP using two HBA ports (initiators). It has four lunpaths through two initiators (0/4/1/0 and 0/4/1/1).

3. Here is an example of sharing a tape device using a single initiator (single lunpath):

```
# hpvmmodify -P guest1 -a tape:avio_stor::attach_path:0/5/0/0/0/0.0x500110a0008b9de2.0x0
# hpvmmodify -P guest2 -a tape:avio_stor::attach_path:0/5/0/0/0/0.0x500110a0008b9de2.0x0
# hpvmdevmgmt -l gdev:0/5/0/0/0/0.0x500110a0008b9de2.0x0

0/5/0/0/0/0.0x500110a0008b9de2.0x0,lunpath1:CONFIG=gdev,EXIST=YES,SHARE=NO,DEVTYPE=ATTACHPATHLUN,AGILE_DSF=
/dev/rtape/tape5_BESTn:guest1,guest2:0x01.0x00.0x03.0x500110a0008b9de1_lunpath1

# hpvmdevmgmt -m gdev:0/5/0/0/0/0.0x500110a0008b9de2.0x0:attr:SHARE=YES
# hpvmdevmgmt -l gdev:0/5/0/0/0/0.0x500110a0008b9de2.0x0

0/5/0/0/0/0.0x500110a0008b9de2.0x0,lunpath1:CONFIG=gdev,EXIST=YES,SHARE=YES,DEVTYPE=ATTACHPATHLUN,AGILE_DSF=
/dev/rtape/tape5_BESTn:guest1,guest2:0x01.0x00.0x03.0x500110a0008b9de1_lunpath1
```

The `hpvmdevmgmt -m` command can also take the following form:

```
# hpvmdevmgmt -m gdev:lunpath1:attr:SHARE=YES
```

Where "lunpath1" is the vPars and Integrity VM- generated alias for the hardware path. The vPar/VM-generated alias of the form "lunpath#" can be used as shorthand in device management commands, but it cannot be used in `hpvmcreate` or `hpvmmodify` commands.

4. Here is an example of sharing a tape device using different initiators (different lunpaths):

a. Add different paths to each vPar/VM:

```
# hpvmmodify -P guest1 -a tape:avio_stor::attach_path:0/4/1/0.0x500104f0004732d9.0x0
# hpvmmodify -P guest2 -a tape:avio_stor::attach_path:0/4/1/1.0x500104f0004732d9.0x0
```

Note that the two lunpath hardware paths in the previous example are through two different initiators (0/4/1/0/ and 0/4/1/1/).

b. List the attributes of each path (Note the value of the `AGILE_DSF` attribute is the same for both lunpaths.):

```
# hpvmdevmgmt -l gdev:0/4/1/0.0x500104f0004732d9.0x0

0/4/1/0.0x500104f0004732d9.0x0,lunpath3:CONFIG=gdev,EXIST=YES,SHARE=NO,DEVTYPE=ATTACHPATHLUN,AGILE_DSF=
/dev/rtape/tape6_BESTn:vme01,guest1:0x01.0x00.0x03.0x500104f0004732d8_lunpath3

# hpvmdevmgmt -l gdev:0/4/1/1.0x500104f0004732d9.0x0

0/4/1/1.0x500104f0004732d9.0x0,lunpath4:CONFIG=gdev,EXIST=YES,SHARE=NO,DEVTYPE=ATTACHPATHLUN,AGILE_DSF=
/dev/rtape/tape6_BESTn:guest2:0x01.0x00.0x03.0x500104f0004732d8_lunpath4
```

c. List the attributes of the parent tape DSF:

```
# hpvmdevmgmt -l gdev:/dev/rtape/tape6_BESTn
/dev/rtape/tape6_BESTn:CONFIG=gdev,EXIST=YES,SHARE=NO,DEVTYPE=ATTACH,SHARE_LUNPATHS=NO:
lunpath3,lunpath6,lunpath5,lunpath4:0x01.0x00.0x03.0x500104f0004732d8
```

d. Modify the `SHARE_LUNPATHS` attribute:

```
# hpvmdevmgmt -m gdev:/dev/rtape/tape6_BESTn:attr:SHARE_LUNPATHS=YES
```

NOTE: The `SHARE_LUNPATHS` and `SHARE` attributes take effect only after an `hpvmstop` command.

e. Relist the attribute of the parent tape DSF:

```
# hpvmdevmgmt -l gdev:/dev/rtape/tape6_BESTn
```

```
/dev/rtape/tape6_BESTn:CONFIG=gdev,EXIST=YES,SHARE=NO,DEVTYPE=ATTACH,SHARE_LUNPATHS=YES:
lunpath3,lunpath6,lunpath5,lunpath4:0x01.0x00.0x03.0x500104f0004732d8
```

9.1.3.4 Mapping AVIO storage devices on HP-UX guests

This section explains how to map an AVIO storage device on an HP-UX guest to an `hpvmstatus` display on the Integrity VSP either at the EFI console or at the HP-UX operating system.

The following example shows the output of `hpvmstatus` from the Integrity VSP:

```
# hpvmstatus -P aviotest
[Storage Interface Details]
Guest                                    Physical
Device  Adaptor    Bus Dev Ftn Tgt Lun Storage   Device
======= ========== === === === === === ========= ========================
disk    avio_stor    0   2   0  22   0 disk      /dev/rdisk/c82t15d3
```

The following statistics are displayed in this example:

- PciBus = 0
- PciDev = 2
- PciFtn = 0
- Addr (Target Id) = 22 (0x16)
- Lun = 0

Note that Addr (Target Id) is decimal in the `hpvmstatus` display, and PciFtn and Lun are always zero (0).

The Integrity VM guest EFI device path encodes PciBus, PciDev, and Addr (Target Id) from the `hpvmstatus` display:

```
                          PciDev
                             |
                             | PCIFtn
                  PciBus     | |        Addr(Target Id)
                     |       | |          |
                     V       V V          V
blk16 : Acpi(PNP0A03,0)/Pci(2|0)/Scsi(Pun16,Lun0)
```

PciFtn (PCI function) and Lun# are always zero (0). Addr (Target Id) becomes EFI Pun# and is displayed as a hexidecimal number.

The two methods for mapping an Integrity VM HP-UX 11i v2 guest hardware path or HP-UX 11i v2 Device Special File (DSF) to an Integrity VSP `hpvmstatus` display:

1. `-e` option of the `ioscan` utility

   `ioscan -fne` displays the HP-UX hardware path/DSF and the EFI device path for the device. The HP-UX hardware path encodes the following from the `hpvmstatus` display:

   - PciBus
   - PciDev
   - Addr (Target Id)

   Addr (Target Id) is encoded as an HP-UX tgt ID and an HP-UX lun ID in the HP-UX hardware path.

   HP-UX tgt ID and HP-UX lun ID are calculated from Addr (Target Id) in the `hpvmstatus` display using the following equations:

   ```
   HP-UX tgt ID = Addr(Target Id) % 16
   HP-UX lun ID = Addr(Target Id) / 16
   ```

   Note the following example:

   ```
   # ioscan -fne
                  PciDev
                     | PCIFtn
   ```

```
                    | |(Addr(Target Id) % 16) <-> HP-UX tgt ID
               PciBus | | |(Addr(Target Id) / 16) <-> HP-UX lun ID
                    | | | | |
                    V V V V V
disk       49  0/0/2/0.6.1   sdisk     CLAIMED     DEVICE       HP      Virtual Disk
                             /dev/dsk/c0t6d1   /dev/rdisk/c0t6d1
          Acpi(PNP0A03,0)/Pci(2|0)/Scsi(Pun16,Lun0)
                          ^      ^ ^          ^
                          |      | |          |
                       PciBus    | PCIFtn   Addr(Target Id)
                                 |
                             PciDev
```

In this example, exp1 / exp2 represents the quotient from exp1 divided by exp2 (integer division), and exp1 % exp2 finds modulo of exp1 divided by exp2 (that is, finds the remainder of an integer division).

2. `get_info` option of the `gvsdmgr` utility

If you are using the HP-UX DSF, the following `gvsdmgr` option can be used to get the VSD LUN ID, which is the same as the Addr (Target Id) in the `hpvmstatus` display. The `gvsdmgr` utility displays VSD LUN Id as a hexidecimal number. The first nibble of VSD LUN Id becomes HP-UX lun ID, and the second nibble becomes HP-UX tgt ID.

The following example shows the `get_info` option with the `gvdsmgr` utility:

```
# gvsdmgr get_info -D /dev/gvsd0 -q lun=/dev/rdisk/c0t6d1
Tue Oct  2 13:35:32 2007

Lun DSF                                                 : /dev/rdisk/c0t6d1
VSD LUN Id                                              : 0x16
Lun  Hardware path                                      : 0/0/2/0.6.1
LUN State                                               : UNOPENED
```

The following is a method for mapping an Integrity VM HP-UX 11i v3 guest hardware path or HP-UX 11i v3 DSF to an Integrity VSP `hpvmstatus` display using the `ioscan` utility:

```
# ioscan -m dsf /dev/rdisk/c0t6d1
Persistent DSF Legacy DSF(s)
========================================
/dev/rdisk/disk22 /dev/rdisk/c0t6d1

# ioscan -m lun /dev/rdisk/disk22
Class I Lun H/W Path Driver S/W State H/W Type Health Description
======================================================================
disk 22 64000/0xfa00/0x1 esdisk CLAIMED DEVICE online HP Virtual Disk
0/0/2/0.0x16.1x0
/dev/disk/disk22 /dev/rdisk/disk22
/dev/disk/disk22_p1 /dev/rdisk/disk22_p1
/dev/disk/disk22_p2 /dev/rdisk/disk22_p2
/dev/disk/disk22_p3 /dev/rdisk/disk22_p3
```

An HP-UX 11iv3 Lun Path hardware path displayed by the `ioscan` utility can be mapped to an `hpvmstatus` utility output as follows:

```
          PciDev
               | PCIFtn
               | | Addr(Target Id)
         PciBus | |  |    Lun
              | | |  |    |
              V V V  V    V
            0/0/2/0.0x16.1x0
```

9.1.3.5 Patch dependency

Table 7–2 lists the patch dependencies for the AVIO attached devices features.

Table 20 Patch Dependencies for AVIO Attached Devices

Patch Number	HP-UX Version	VSP	Guest	Notes
PHKL_38604	11i v3	Yes	Yes	Hard[1] dependency for guest, and soft[2] dependency for VSP.
PHKL_38605	11i v3	Yes	No	Soft dependency on VSP.
PHKL_38750	11i v3	Yes	Yes	Recommended patch.

[1] Enforced during `swinstall`.

[2] Required only if attached devices are configured. No enforcement using `swinstall`.

9.1.3.6 Error messages

This section lists possible VSP and guest error messages and their description.

- VSP error messages
  - Access error on a shared attached device

    The VSP's attempt on a shared tape is denied when it is in use by any guests. Applications receive a busy error in such cases. For example. here is the behavior of `diskinfo` on a tape which is being used by a guest:

    ```
    # diskinfo /dev/rtape/tape1_BEST
    diskinfo: can't open /dev/rtape/tape1_BEST: Device busy
    ```

- Guest error messages
  - 11i v3 guest — access error on a shared attached device

    A guest access attempt on a shared tape is denied when it is in use by the VSP or other guests. Applications receive a busy error in such cases. For example, here is the behavior on `diskinfo` on a tape that is being used by another guest.

    ```
    # diskinfo /dev/rtape/tape1_BEST
    diskinfo: can't open /dev/rtape/tape1_BEST: Device busy
    ```

  - 11i v2 guest — access error on a shared attached device

    A guest access attempt on a shared tape is denied when it is in use by the VSP or other guests. Applications receive a no-device error in such cases. For example, here is the behavior on `diskinfo` on a tape that is being used by another guest.

    ```
    # diskinfo /dev/rmt/c7t0d0BEST
    diskinfo: can't open /dev/rmt/c7t0d0BEST: No such device or address
    ```

9.1.4 vPar/VM storage implementations

This section describes the implementations of the vPar/VM storage architectures.

9.1.4.1 vPar/VM storage adapters

The AVIO storage adapter is a high performance adapter and needs guest OS drivers. AVIO supports up to 256 storage devices per guest and also leverages the VSP 11i v3 storage stack features to provide better storage manageability in the guest. VxVM is also supported as an AVIO backing store.

9.1.4.2 Sample script for adding multiple devices at once

To add 256 AVIO storage devices to a vPar/VM, HP recommends that you use the `hpvmcreate` and `hpvmmodify` commands to add multiple devices at a time using multiple `-a` options. Adding

multiple devices at a time takes less time than adding them one at a time, with one device per call to `hpvmcreate` and then one device per call in subsequent calls to `hpvmmodify`.

You can add any number of devices at a time up to the supported limit. However, you might find that adding multiple devices at a time per call to `hpvmmodify` not only takes less time than adding all of them at once, but also using one particular number of devices at a time provides better `hpvmmodify` performance than others. For example, if you are adding a total of 256 disks, adding 64 at a time might provide better performance than adding 8 at a time and better performance than adding 128 at a time. The best number to use might vary depending on many factors including how many total devices you are adding.

For a sample script for adding multiple devices, see Appendix B (page 227).

9.1.4.3 vPar/VM storage devices

vPars and Integrity VM V6.1 supports a variety of virtual and attachable devices. Disk and DVD-ROM devices support several virtual media types (see Section 9.1.4.3.1 (page 115)). Physical tapes, media changers, and CD/DVD burners are attachable; they can be used to perform data backups directly from a vPAr/VM. (See Section 9.1.4.3.2 (page 115)).

9.1.4.3.1 Virtual devices

vPar and Integrity VM V6.1 supports the following virtual disk types:

Virtual Disk Type	Backing Storage Device	For more information, see...
Virtual Disk	VSP disk	"Virtual Disks" (page 124)
Virtual LvDisk	VSP LVM or VxVM logical volume.	Section 9.2.2.3.2 (page 124)
Virtual FileDisk	VSP VxFS file	Section 9.2.2.3.3 (page 126)

The following virtual DVD-ROM types are supported:

Virtual DVD Type	Backing Storage Device	Described in ...
Virtual DVD	Disk in a VSP physical DVD drive	Section 9.2.2.3.4 (page 127)
Virtual FileDVD	ISO file on a VSP VxFS file system	Section 9.2.2.3.6 (page 129)
Virtual NullDVD (empty)	VSP physical DVD drive or VxFS directory	Section 9.2.2.3.3 (page 126)

9.1.4.3.2 Attached devices

vPars and Integrity VM V6.1 supports a suite of attached devices to complete data backups from a vPar/VM. vPars and Integrity VM attaches these devices using a special vPar/VM pass-through driver. With this pass-through driver, vPar/VM I/O requests are interpreted by vPars and Integrity VM and sent through the virtual storage subsystem to the physical device. The virtual storage subsystem sends device responses to the vPar/VM pass-through driver, which sends the responses to the virtual machine. Because the vPar/VM can see all the data and responses, support for the attached physical device must be provided by the guest OS. An attached device can be attached to only one vPar/VM at a time.

Attached devices include:

- CD/DVD burners
- Media changers
- Tape devices

The maximum transfer size can be 1 MB for any guest operating system.

9.2 Configuring vPar/VM storage

This section describes how to plan and set up vPar/VM storage, including the following topics:

- Storage considerations
- Setting up virtual storage

9.2.1 Storage considerations

When you configure storage for a vPar/VM, consider the following:

- Storage supportability
- Storage performance
- Storage multipath solutions
- Storage management
- Storage changes
- Virtual storage setup time

The following sections explain each of these considerations.

9.2.1.1 Storage supportability

Before you configure vPar/VM storage, make sure the VSP storage can be supported by the vPar/VM.

- All VSP storage available for use by a vPar/VM must meet support requirements for the Integrity server and OS version that comprise the VSP. If the physical storage is not supported by the VSP, it is not supported for use by a vPar/VM.
- All VSP storage available for use by a vPar/VM must be connected with a supported adapter and driver type. See the *HP-UX vPars and Integrity VM V6.1 Release Notes* for the list of supported types.

  If the physical storage is not connected with one of the supported adapter and driver types, it cannot be used by a vPar/VM. Use the `ioscan` command to display the VSP storage that is connected to adapters and drivers.

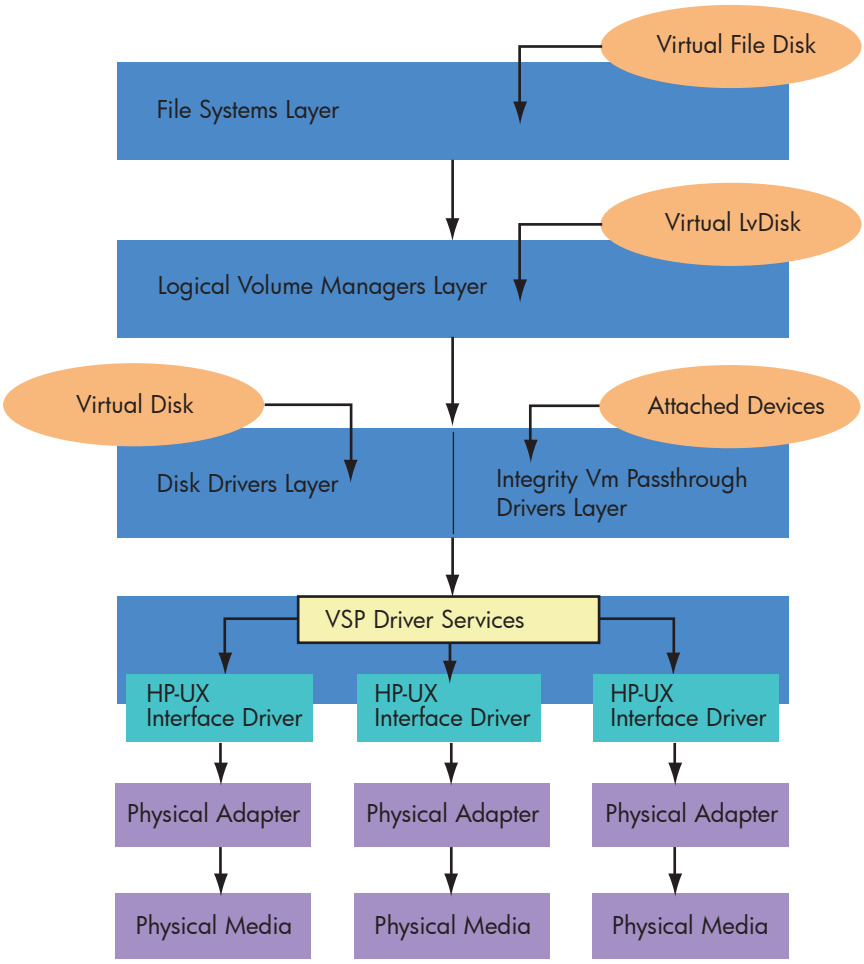
- Any VSP attachable devices available for use by a vPar/VM must be supported by the guest OS to which it is attached. If the physical device is not supported by the guest OS, the device cannot be attached to the vPar/VM.

9.2.1.2 Performance of virtual devices

To meet the performance requirements of applications running in vPars/VMs, consider the potential performance of each type of vPar/VM storage device.

Different types of virtual media have different effects on the performance of the virtual device because they communicate differently with the VSP to complete vPar/VMI/O operations. To understand the effect of the virtual device type on potential performance, consider the vPar/VM storage I/O stack illustrated in Figure 5.

Figure 5 Storage I/O Stack



For a virtual I/O operation to be completed, it has to travel round trip between the virtual storage adapter and the VSP physical storage device. The longer the path is, the longer it takes for virtual I/O to be completed. As shown in Figure 5, a virtual I/O operation must traverse each software layer in order, from where it originates to the physical media. For example, a virtual I/O operation for a Virtual FileDisk must traverse any logical volume managers the file system is on and the disk drivers that control the whole disk. Therefore, in general, the higher the virtual media is in the VSP I/O stack, the slower it operates.

The simplified I/O stack in Figure 5 does not completely illustrate all the choices that can affect the performance:

- Performance of different software layers differs.
- The interfaces to each software layer are different, allowing Integrity VM different ways to send I/O through the layers. For example, whole disks can achieve higher throughput rates than logical volumes and file systems.
- The I/O layer might have features to help performance increase beyond a lower layer. For example, a file system's buffer cache may help a Virtual FileDisk perform better on some I/O workloads than the other virtual device types, which have no such caching.

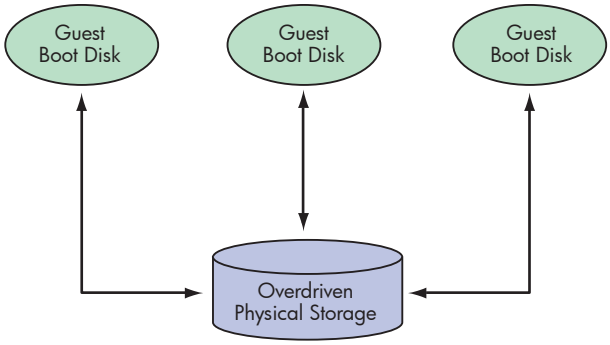
For further information on tuning performance at each software layer on the VSP, see the vPars and Integrity VM white papers on the Business Support Center website at http://www.hp.com/go/virtualization-manuals.

When you configure virtual devices, consider how the virtual media maps to the physical storage. All virtual media connects to a piece of physical media somewhere in the data center. You can

help ensure the best performance by understanding the impact of the physical storage and the way I/O accesses it.

It is important to know exactly where the virtual media is located on physical storage devices. With vPars and Integrity VM V6.1, a single physical disk might be sliced into logical volumes or files. Slicing up physical disks increases utilization, but it can affect the performance of the physical device. The guest OS treats the virtual disk as a whole disk, not as a part of a physical one. Over-slicing physical storage can overload a physical device's ability to handle virtual I/O that is meant for whole disks. Figure 6 shows a common mistake of overdriving physical storage with multiple guest OS boot disks, which are often I/O intensive.

Figure 6 Overdriving Physical Storage Hurts Performance



Provide workloads that the physical devices can handle for all the virtual devices layered on top of them. Use performance tools on the VSP, like *sar(1M)*, to see how the physical storage is keeping up with the virtual device demands.

The way the virtual media I/O gets to the physical storage backing it is also an important consideration. As shown in Figure 5, all virtual I/O goes through a general VSP I/O services layer that routes the virtual I/O to the correct VSP interface driver. The interface driver then controls the physical I/O adapter to issue virtual I/O to the physical storage device. By load balancing across these physical adapters, virtual I/O bottlenecks can be eliminated at the physical hardware layers, thereby increasing performance. Load balancing can be done by using a multipathing solution on the VSP. For help with selecting a multipath solution for a virtual media type, see Section 9.2.1.3 (page 118).

The performance of attached devices is largely determined by the type of physical device attached to the virtual machine. Tapes, media changers, and CD/DVD burners are inherently slow devices, not significantly impacted by the software overhead of Integrity VM.

9.2.1.3 Storage multipath solutions

vPars and Integrity VM virtual devices support the built-in multipathing of the HP-UX 11i v3 VSP, which is enabled by default to provide improved performance, load-balancing, and higher availability for vPars/VMs. Currently, there are no multipath solutions supported for the attachable device types of tapes, media changers, and CD/DVD burners.

There are no multiple paths inside a virtual machine to virtual devices. Multipathing is supported only on the VSP for the following reasons:

- The VSP is the only place where all virtual I/O can be properly load balanced for the best overall performance. A single virtual machine cannot account for all the other vPar/VM I/O with which it is competing on the VSP (see Figure 5).
- Running a multipath solution in a vPar/VM does not provide any high availability for a virtual device. Virtual connections between virtual adapters and their devices are never lost until an `hpvmmodify` command is used to disconnect them. The only connection ever lost is the ability of a virtual device to access its own virtual media through the VSP. Errors in communication

to the virtual media are properly emulated as media errors sent to the guest OS, not as path failures.

- The VSP does not return specific errors to Integrity VM for hardware path failures. vPars and Integrity VM does not detect such events and does not pass them to the vPar/VM.

For supported multipathing configurations, see the *HP-UX vPars and Integrity VM V6.1 Release Notes* at http://www.hp.com/go/hpux-hpvm-docs.

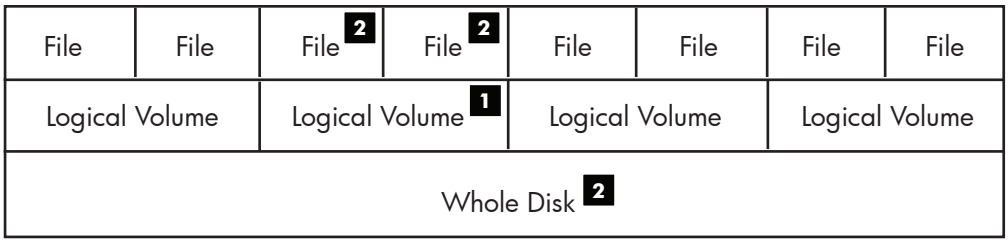
9.2.1.4 VM storage management

Before you decide how to divide VSP storage, consider the impact on the management of the storage subsystem.

A VSP administrator manages VM storage to make sure virtual media is allocated safely. This begins with understanding the VSP I/O stack and knowing from where the virtual media is being allocated.

Figure 7 shows an example of a VSP I/O stack as it applies to a single LUN.

Figure 7 Sub-LUN Storage Allocation Example



The virtual machine is allocated a logical volume from the LUN for a Virtual LvDisk.

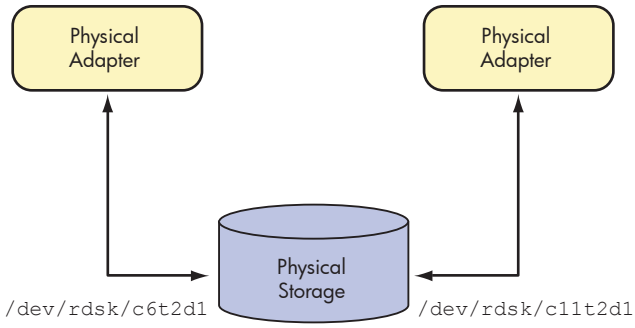
- The logical volume that has been allocated is labeled **1**.
- The parts of the disk that cannot be allocated are labeled **2**.

Those parts that are no longer available include the files that were on the logical volume and the whole disk that makes up part of the volume group. If any of these parts are allocated for other virtual devices, data corruption can occur on the Virtual LvDisk.

Those parts that are still available for reallocation include other logical volumes that are on the disk, and files that are on those other logical volumes on the disk. These pieces can be allocated without data corruption problems because they do not overlap with the Virtual LvDisk.

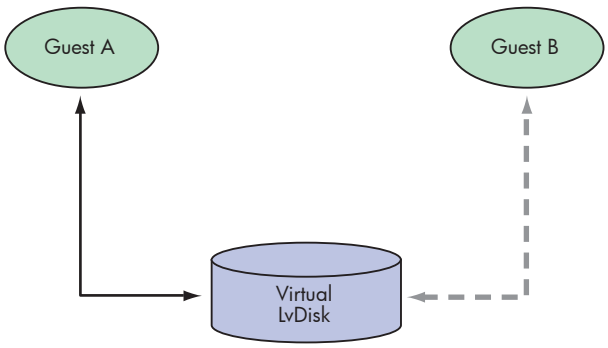
Beyond avoiding sub-LUN collisions, whole LUN collisions also need to be avoided. The same storage resource, virtual or attached, cannot be specified more than once to the same virtual machine. HP-UX 11i v3 supports both legacy per-path device files (for example, `/dev/rdisk/c6t2d0`) and agile non-path specific device files (for example, `/dev/rdisk/disk`). As shown in Figure 8, there may be more than one legacy device file that points to the same physical storage device, while there is only one agile device file per a given physical storage device. Use of agile device files is recommended to avoid whole LUN collisions.

Figure 8 Bad Multipath Virtual Media Allocation



Also, the same storage resource, virtual or attached, cannot be simultaneously shared between virtual machines, unless otherwise specifically exempted. Figure 9 shows a Virtual LvDisk being shared across virtual machines, which is not supported.

Figure 9 Bad Virtual Device Allocation



As these examples illustrate, it is important to know where storage is allocated from to avoid data corruption with vPars/VMs or even the VSP. Management utilities such as the HP System Management Homepage (HP SMH) utility allow you to track disk devices, volume groups, logical volumes, and file systems. You can use these utilities to annotate devices so that VSP administrators can see exactly which virtual machines are using each VSP storage device.

To show each disk only once, management utilities consolidate multipath devices into one disk. When you are dividing up the disk, you should use all the parts of a single disk on a single virtual machine. Allocating different parts of the same disk to different virtual machines makes it difficult to manage and to isolate problems.

When an LVM volume group is deactivated, the storage (physical volumes) used by that storage is designated as unused by HP-UX system administration tools such as System Management Homepage (SMH). This is also true for Integrity VM storage management. As a result, these physical volumes are not automatically protected from use by virtual machines as virtual disks.

You can resolve this problem in one of two ways:

- If the volume group is to remain deactivated, the VSP administrator can manually add the physical volume as a restricted device with the `hpvmdevmgmt` command.
- Or, after activating the volume group, execute the `hpvmhostrdev` command, so that the VSP storage management database is updated accordingly.

An HP-UX system administrator can deactivate a volume group using the `vgchange` command. It can also be deactivated, if it is a shared LVM (SLVM) volume group, whenever the associated Serviceguard cluster is reconfigured, or the VSP system is rebooted. Take care to check that all SLVM volume groups are activated after a VSP reboot or Serviceguard cluster reconfiguration.

9.2.1.5 VM storage changes

Depending on how you set up storage for a vPar/VM, the resulting configuration can be more or less difficult to change.

The ability to change virtual media depends on the type of virtual media used. Whole disks are not normally adjustable in terms of size, but some high-end storage enclosures might permit the adjustment of a LUN without losing that LUN's data. Logical volumes are adjustable without losing any data. Finally, files can be changed easily with VSP file system commands.

No changes to any virtual media can take place on the VSP until the virtual device that uses the media is removed from the active VM. Attempts to change virtual devices that have I/O active on them is denied by the `hpvmmodify` command. Once an active vPar/VM is allocated virtual media for a virtual device, that vPar/VM owns that media and can access it any time. VSP administrators need to coordinate with VM guest administrators about active virtual machine changes, if the two roles are served by different individuals.

This coordination might also be necessary for attached I/O devices. Once a VSP device is attached to the vPar/VM, it is controlled and owned by that vPar/VM. Modifications to the attached device, like changing a tape, can be done physically without detaching the device from the vPar/VM. However, such changes might need to be coordinated with the VSP administrator, especially if the guest administrator has no physical access to the device attached to the vPar/VM.

All types of virtual storage devices can be added and removed dynamically from vPars/VMs. That is, virtual disks, virtual DVDs, tapes, media changers, and CD/DVD burners are all hot-swappable. However, the virtual storage adapters are currently not hot-swappable. Therefore, if all the virtual storage adapters are full, you must reboot the vPar/VM when you add additional devices.

9.2.1.6 Virtual storage setup time

Some virtual devices take longer to set up than others. Whole disks are very easy to set up because they require nothing more than a character device file. This is usually created automatically when the VSP system is booted.

Logical volume creation is relatively simple. Logical volumes are used widely on HP-UX systems. The Veritas Enterprise Administrator can be used to create logical volumes. With experience, you can use logical volume commands more quickly.

Creating files for virtual devices is not hard, but takes time. Files are usually placed on top of logical volumes, so you might have to create a logical volume first.

To create empty files for virtual disks, use the `hpvmdevmgmt` command (see Section 11.11 (page 189)).

To create ISO files from physical CD/DVD media for use in virtual DVDs, use the *mkisofs* or the *dd* utility.

For attached devices, the effort and time to set them up is spent in the creation of the HP-UX pass-through device files that point to the devices being attached. Once understood, making HP-UX pass-through device files is a fast, simple process. If device drivers for the devices are installed on the VSP, use the `hpvmdevmgmt` command to quickly create the device files. Otherwise, see *scsi_ctl* for information about creating passthrough device files using *mknod*.

9.2.2 Setting up virtual storage

When you add or modify a virtual device, you must enter a resource statement (`rsrc`). The resource statement can specify either virtual network devices (as described in Chapter 10), or virtual storage devices.

This section describes how to enter resource statements for use with the `hpvmcreate` command (described in Chapter 7) and the `hpvmmodify` command (described in Chapter 11). The resource statement specifies the virtual storage device that will be seen by the vPar/VM and how it maps to the physical storage device on the VSP.

The following is an outline of a complete resource statement for specifying a virtual storage device:

VM-guest-storage-specification:*VM-Host-storage-specification*

where:

- *VM-guest-storage-specification* defines where and what storage is seen in the vPar/VM (see Section 9.2.2.1 (page 122))
- *VM-Host-storage-specification* defines where and how the vPar/VM storage is supplied on the VSP (see Section 9.2.2.2 (page 122))

For examples of how to construct resource statements, see Section 9.2.2.3 (page 123).

9.2.2.1 Storage specification

All virtual storage is addressed from virtual PCI buses. The vPar/VM virtual platform contains 8 PCI buses. Each PCI bus has 8 slots into which virtual PCI adapters can be placed. An AVIO storage adapter supports up to 128 devices per adapter (and VSP) and provides high performance and guest storage manageability.

A VSP administrator specifies this virtual adapter using the following:

device:avio_stor:*pcibus*,*pcislot*,*aviotgt*

where:

- *device* is one of the following: `disk`, `dvd`, `tape`, `changer`, `burner`, or `hba`
- *pcibus* is an integer from 0-6.

  The virtual AVIO is supported only on PCI buses 0-7.

- *pcislot* is an integer from 0-7.

  A PCI function number is not specified. It is implicitly zero because the virtual storage adapter supports only a single channel.

- *aviotgt* is an integer from 0–127 for AVIO. All supported storage device types can share the same virtual AVIO adapter by specifying the same PCI bus and slot numbers. A virtual AVIO adapter can be added only to a virtual machine if it has a device connected to it.

  All targets connected to a vPar/VM are single LUN devices. That is, virtual disks and DVDs are emulated as single LUNs and all attached devices are specified by per LUN VSP system files. The physical LUN number of an attached device has no impact. All virtual and attached AVIO LUN numbers are implicitly zero and therefore not specified.

  All supported storage device types can share the same virtual adapter. Up to 15 storage devices can be added to the same virtual adapter by specifying the same PCI bus and slot numbers.

A virtual adapter can be added only to a vPar/VM if it has a device connected to it.

Not all device types are virtualized. Disk and DVD devices are virtual device types, whose virtual media comes from the VSP. Tapes, changers, and burners are physical VSP devices. For these attached devices, the physical IDs do not determine their place on the virtual bus.

9.2.2.2 VSP storage specification

Each vPar/VM storage device is backed by some VSP storage entity. A VSP entity is defined on the VSP with a system file, which is used by vPars and Integrity VM and the VSP operating system in processing I/O to and from that storage entity.

A VSP administrator specifies these storage entities using the following specification:

storage:*location*

where:

- *storage* is one of the following: `disk`, `lv`, `file`, `null`, `attach`, or `npiv`.

  The selection of storage type defines what VSP system files apply. For example, `lv` implies the use of logical volume character device files.

  For virtual devices, the selection of VSP storage determines what type of virtual media the virtual device uses. For example, the selection of `lv` for a virtual disk, makes it a Virtual LvDisk to the VM. It does not support the attach storage type.

  A VSP storage entity can only be used for one VM device type at a time. For example, a VSP CD/DVD drive cannot be used for a Virtual DVD and an attached burner at the same time.

- *location* is a VSP system file.

  The file permissions on the VSP system file are not honored by vPars and Integrity VM. vPar/VM device types that support write operations can still do so using a VSP system file marked read only. Backing stores provided as virtual disks can be written to regardless of the file permission settings on the backing store. A backing store provided as a virtual DVD is always read-only. Attached devices do not consider file permissions when backing up data.

  More than one VSP system file might point to the same VSP storage entity. For example, if multiple paths to storage are present on the VSP, more than one disk system file can point to the same disk. Different VSP system files change how I/O is routed to the VM storage resource, but the system files point to the same storage entity. Therefore, different system files cannot constitute different vPar/VM storage resources. A given vPar/VM storage resource can only be specified once to a given vPar/VM. Therefore, only one VSP system file per VSP storage entity can be provided to a vPar/VM (see Section 9.2.1.4 (page 119)).

Not all virtual device types support all VSP storage types (see Section 9.1.4 (page 114)). Complete VM storage resource statements are discussed in the next section.

9.2.2.3 Storage resource statements

This section provides information about formulating complete valid resource statements for vPar/VM storage devices.

To specify a vPar/VM storage device for a vPar/VM, use a complete valid resource statement with the `hpvmcreate` or `hpvmmodify` command. The resource statement is a combination of the vPar/VM guest resource specification (described in Section 9.2.2.1 (page 122)) and the VSP Storage Specification (described in Section 9.2.2.2 (page 122)). This section provides examples of complete resource statements for each of the following types of virtual storage devices:

- Virtual disks
- Virtual LvDisks
- Virtual FileDisks
- Virtual DVDs
- Virtual FileDVDs
- Virtual NullDVDs
- Attachable Devices

A vPar/VM can have up to 128 AVIO devices total (number of virtual and attached devices).

The minimum size of a virtual storage resource is 512 bytes for virtual disk and 2048 bytes for a virtual DVD.

Do not specify the same storage resource, virtual or attached, for the same vPar/VM more than once (see Section 9.2.1.4 (page 119)). Unless otherwise noted, storage resources, virtual or attached, cannot be simultaneously shared by vPars/VMs.

All multipath products for storage resources must run on the VSP; multipath solutions are not supported in a vPar/VM. All multipath solutions used on the VSP must be in valid supported configurations before being used for vPar/VM storage resources (see Section 9.2.1.3 (page 118)).

The resource statements in the following subsections do not contain vPar/VM hardware addressing. The PCI bus, PCI slot, and AVIO target numbers are optional.

9.2.2.3.1 Virtual Disks

A Virtual Disk is an emulated AVIO disk whose virtual media comes from a VSP disk LUN. The VSP disk LUN is specified using a character device file. The character device file is owned by the HP-UX `esdisk` or `sdisk` driver.

Virtual Disk resources cannot be shared simultaneously across active vPars/VMs (except in certain cluster configurations, as indicated in this manual). Only one active vPar/VM at time can be given a particular Virtual Disk resource. Virtual Disk resources can be changed dynamically among active vPars/VMs.

To prevent virtual media conflicts that can result in data corruption, a proper accounting of how the VSP whole disks are allocated for use by Virtual Disks needs to be done, as described in Section 9.2.1.4 (page 119).

The Virtual Disk resource statement takes the form:

`disk:avio_stor::disk:`*`/dev/rdisk/diskX`*

where `/dev/rdisk/diskX` is an HP-UX `esdisk` character device file.

These device files can be located for a VSP LUN using the `ioscan` command. These system files are installed and removed using the `insf` and `rmsf` commands, respectively. Device files are created automatically by the VSP for any storage it identifies during boot. New devices connected or created after boot time, require the use of `ioscan` and `insf` to create the new `sdisk` device files. To remove old device files for storage that is no longer present, use the `rmsf` command. For example:

```
# ioscan

# ioscan -NfunC disk

disk        64000/0xfa00/0x10   esdisk    CLAIMED     DEVICE
HP      HSV210
/dev/disk/disk   /dev/rdisk/disk
```

9.2.2.3.2 Virtual LvDisks

A Virtual LvDisk is an emulated AVIO disk whose virtual media is provided by a raw VSP logical volume. To specify a VSP logical volume, use a character device file. The character device file is owned by either LVM or VxVM.

Virtual LvDisks cannot be shared simultaneously across active vPars/VMs. Only one active vPar/VM at time can be given a particular Virtual LvDisk resource. Virtual LvDisk resources can be changed dynamically between active vPars/VMs (see Section 9.3 (page 133)).

Logical volumes can be created using the `sam` utility or the Veritas Enterprise Administrator. Alternatively, logical volumes can be created using the commands available with the volume manager. All logical volumes are created on whole disks. The sizes of the logical volumes come from the space available from their respective volume group types; that logical volume size can be increased without loss of data in the volume. The character devices for the logical volumes are created by their respective volume managers at the time the logical volume is created. Also to avoid file system corruptions for the VSP and guest , use only raw logical volumes that contain no VSP file systems and are not currently mounted on the VSP.

To prevent data corruptions, keep an account of logical volumes for Virtual LvDisks. To help with the accounting, use all logical volumes within a given volume group for a single virtual machine. When logical volumes are configured this way, you only have to keep track of the volume groups

to prevent media conflicts. For information about tracking virtual media allocation, see Section 9.2.1.4 (page 119).

If you are using LVM, the Virtual LvDisk resource statement takes the following form:

```
disk:avio_stor::lv:/dev/vg_name/rlvol_name
```

Where */dev/vg_name/rlvol_name* is an LVM character device file for *rlvol_name* on *vg_name*. To display the LVM character device file name, enter the following command:

```
# vgdisplay -v
VG Name                     /dev/lvrackA
VG Write Access             read/write
VG Status                   available
Max LV                      255
Cur LV                      4
Open LV                     4
Max PV
Cur PV                      1
Act PV                      1
Max PE per PV               8683
VGDA                        2
PE Size (Mbytes)            4
Total PE                    8681
Alloc PE                    8192
Free PE                     489
Total PVG                   0
Total Spare PVs             0
Total Spare PVs in use      0

   --- Logical volumes ---
   LV Name                      /dev/lvrackA/disk1
   LV Status                    available/syncd
   LV Size (Mbytes)             8192
   Current LE                   2048
   Allocated PE                 2048
   Used PV                      1

   LV Name                      /dev/lvrackA/disk2
   LV Status                    available/syncd
   LV Size (Mbytes)             8192
   Current LE                   2048
   Allocated PE                 2048
   Used PV                      1

   LV Name                      /dev/lvrackA/disk3
   LV Status                    available/syncd
   LV Size (Mbytes)             8192
   Current LE                   2048
   Allocated PE                 2048
   Used PV                      1

   LV Name                      /dev/lvrackA/disk4
   LV Status                    available/syncd
   LV Size (Mbytes)             8192
   Current LE                   2048
   Allocated PE                 2048
   Used PV                      1

   --- Physical volumes ---
   PV Name                      /dev/disk/disk237
   PV Status                    available
   Total PE                     8681
   Free PE                      489
   Autoswitch                   On
```

In this example, the Virtual LvDisk resource statement is `disk:avio_stor::lv:/dev/lvrackA/rdisk2`.

To use VxVM, the Virtual LvDisk resource statement takes the following form:

`disk:avio_stor::lv:`*`/dev/vx/rdisk/dg_name/v_name`*

where */dev/vx/rdisk/dg_name/v_name* is a VxVM character device file for volume *v_name* on disk group *dg_name*. To display the VxVM character device file name, enter the following command:

```
# vxprint
Disk group: rootdg

TY NAME         ASSOC        KSTATE   LENGTH   PLOFFS   STATE    TUTIL0
PUTIL0
dg rootdg       rootdg       -        -        -        -        -       -

dm disk01       c3t0d0       -        35562538 -        -        -       -

Disk group: VxvmTest1

TY NAME         ASSOC        KSTATE   LENGTH   PLOFFS   STATE    TUTIL0
PUTIL0
dg VxvmTest1    VxvmTest1    -        -        -        -        -       -

dm disk01       c5t8d0       -        780564 -         -        -       -

v  vxvm_1       fsgen        ENABLED  2048000  -        ACTIVE   -       -
pl vxvm_1-01    vxvm_1       ENABLED  2048000  -        ACTIVE   -       -
sd disk01-01    vxvm_1-01    ENABLED  2048000  0        -        -       -

v  vxvm_2       fsgen        ENABLED  2048000  -        ACTIVE   -       -
pl vxvm_2-01    vxvm_2       ENABLED  2048000  -        ACTIVE   -       -
sd disk01-02    vxvm_2-01    ENABLED  2048000  0        -        -       -

v  vxvm_3       fsgen        ENABLED  2048000  -        ACTIVE   -       -
pl vxvm_3-01    vxvm_3       ENABLED  2048000  -        ACTIVE   -       -
sd disk01-03    vxvm_3-01    ENABLED  2048000  0        -        -       -

v  vxvm_4       fsgen        ENABLED  2048000  -        ACTIVE   -       -
pl vxvm_4-01    vxvm_4       ENABLED  2048000  -        ACTIVE   -       -
sd disk01-04    vxvm_4-01    ENABLED  2048000  0        -        -       -
```

To use VxVM, the Virtual LvDisk resource statement is `disk:avio_stor::lv:/dev/vx/rdisk/VxvmTest1/vxvm_2`. For information about adapters that support VxVM, see the *HP-UX vPars and Integrity VM V6.1 Release Notes* at http://www.hp.com/go/hpux-hpvm-docs.

For information about multipath solutions for Virtual LvDisks, see Section 9.2.1.3 (page 118).

9.2.2.3.3 Virtual FileDisks

A Virtual FileDisk is an emulated AVIO disk whose virtual media comes from a VSP file. The VSP file is specified using the absolute pathname to the file. The file can be on a VxFS file system locally mounted on the VSP. NFS file systems are not supported for Virtual FileDisks.

Virtual FileDisks cannot be shared simultaneously across active virtual machiness. Only one active vPar/VM can be given a particular Virtual FileDisk resource at a time. Virtual FileDisk resources can be changed dynamically between active vPars/VMs (see Section 9.3 (page 133)).

The file systems used for Virtual FileDisks need to be managed to prevent data corruptions. To help with accounting, it is recommended that all files under a given directory be used with a single vPar/VM. Additionally, it might help to allocate file directories from complete logical volumes or whole disks to make the accounting even easier. For more information, see Section 9.2.1.4 (page 119).

The Virtual FileDisk resource statement takes the following form:

`disk:avio_stor::file:/pathname/file`

where the `/pathname/file` specifies the VSP file used as virtual media.

A VxFS file system can be created on top of a whole disk or logical volume. For files over 2 GB, VxFS requires the file system be marked with a `largefiles` option. The `mkfs` command can be used to create the VxFS file systems directly. Once the file systems are created, `mount` can be used to mount them onto the VSP file system. Alternatively, if using logical volumes to create the file system on, the volume manager GUIs like `sam` can be used to create the file systems and their mount points, when the logical volumes are created. In any case, once the file system is mounted, you can create empty files for Virtual FileDisk using the `hpvmdevmgmt` command.

```
# mkfs -F vxfs -o largefiles /dev/disk/disk237
# mount /dev/disk/disk237 /fdev/frackA/
# hpvmdevmgmt -S 4G /fdev/frackA/disk1
```

In this example, the Virtual FileDisk resource statement is `disk:avio_stor::file:/fdev/frackA/disk1`.

Multipath options for a Virtual FileDisk device are discussed in Section 9.2.1.3 (page 118).

NOTE: Each vPar/VM can support a maximum of 30 Virtual FileDisks.

9.2.2.3.4 Virtual DVDs

A Virtual DVD is an emulated AVIO DVD-ROM with virtual media that comes from a disc inside of a CD/DVD drive on the VSP. The VSP CD/DVD drive is specified using an HP-UX sdisk character device file.

While the Virtual DVD is read-only, the slowness of the physical VSP CD/DVD drives prohibits them from being shared across active vPars/VMs. Thus only one active vPar/VM at time should be given a particular Virtual DVD resource. Virtual DVD resources can be changed dynamically between active vPars/VMs (see Section 9.3 (page 133)).

Because the Virtual DVDs are read only, they do not require management to prevent conflicts writing to the device. However, to prevent sensitive information from being accessed by the wrong vPar/VM, make sure you know which vPar/VM currently owns the device before you load a CD/DVD. This information can be found on the VSP with the `hpvmstatus` commands.

The agile Virtual DVD resource statement takes the following form:

`dvd:avio_stor::disk:/dev/rdisk/disk#`

where `/dev/rdisk/disk#` is an HP-UX `esdisk` character device file for a VSP CD/DVD drive. The legacy Virtual DVD resource statement takes the form `dvd:avio_stor::disk:/dev/rdisk/cXtYdZ`, where `/dev/rdisk/cXtYdZ` is an HP-UX `sdisk` character device file for a VSP CD/DVD drive.

Typically, the HP-UX `esdisk` and `sdisk` character files will already be created before booting the VSP. If they are not, they can be created and managed using the `ioscan`, `insf`, and `rmsf` utilities. For example:

```
# ioscan -NfunC disk

disk        7   64000/0xfa00/0x6    esdisk    CLAIMED      DEVICE
TEAC      DW-224E
                /dev/disk/disk7   /dev/rdisk/disk7

# diskinfo /dev/rdisk/disk7
SCSI describe of /dev/rdisk/disk7:
             vendor: TEAC
         product id: DW-224E
               type: CD-ROM
               size: 4300800 Kbytes
   bytes per sector: 2048
```

In this example, the Virtual DVD resource statement is `dvd:avio_stor::disk:/dev/rdisk/disk7`.

For a vPar/VM to recognize a Virtual DVD, physical media must be present inside the VSP CD/DVD drive. If media is not added at vPar/VM start time, it can be inserted into the VSP CD/DVD drive after the vPar/VM is already up. A rescan by the guest OS picks up the new media and adds the Virtual DVD to the vPar/VM.

If for some reason the VSP Administrator requires control of the VSP CD/DVD drive claimed by a vPar/VM but has no media for the VSP CD/DVD drive, then a Virtual NullDVD should be specified (see Section 9.2.2.3.6 (page 129)). Physical media can then be inserted into the VSP CD/DVD drive and become virtual media for a Virtual DVD using the `hpvmmodify` or the virtual console's insert command (see Section 9.3.1.3 (page 135)).

After the Virtual DVD is in the vPar/VM, the VSP CD/DVD drive is locked. The VSP CD/DVD drive is automatically unlocked when the vPar/VM is shut down. The VSP CD/DVD can also be changed while the vPar/VM is up using the virtual console's `eject` command. Once ejected, the Virtual DVD turns into a Virtual NullDVD and the VSP CD/DVD drive unlocks. After you place physical media in the VSP's CD/DVD drive, use the virtual console's `insert` command to turn a Virtual NullDVD back to a Virtual DVD, relocking the VSP CD/DVD drive.

Most physical VSP CD/DVD devices on HP Integrity servers have only one path to them. As such, no multipath software is available on the VSP for them.

9.2.2.3.5 Virtual FileDVDs

A Virtual FileDVD is an emulated SCSI DVD with virtual media that comes from a VSP ISO file. The VSP ISO file is specified using the absolute pathname to the ISO file. The file can be on a VxFS file systems locally mounted on the VSP. NFS file systems are not supported for Virtual FileDVDs.

The Virtual FileDVD resource statement takes the following form:

`dvd:avio_stor::file:`*`/pathname/file.ISO`*

where the *`/pathname/file.ISO`* specifies the VSP ISO file to use as virtual media.

A VSP ISO file can be created using the `mkisofs` utility or by using the `dd` command to copy CD/DVD media to a file. The VxFS file system should be enabled to support `largefiles`, because ISO files tend to be over 2 GB in size. All the ISO files that are useful to a guest OS should be placed in the same directory to take advantage of dynamic changes using the virtual console (see Section 9.3.2.3 (page 138)). The ISO files should be marked with proper permissions; they must not be world writable. For example:

```
# ls -l /var/opt/hpvm/ISO-images/hpux

total 26409104
-rw-r--r-- 1 root sys 3774611456 Jul 11 :59 0505-FOE-OE.iso
-rw-r--r-- 1 root sys 4285267968 Jul 11 17:05 0512-FOE.iso
-rw-r--r-- 1 root sys 3149987840 Jul 11 18:42 0603-FOE-D1.iso
-rw-r--r-- 1 root sys 29978624 Jul 11 18:51 0603-FOE-D2.iso
```

In this example, the Virtual FileDVD Resource Statement is:
`dvd:avio_stor::file:/var/opt/hpvm/ISOimages/hpux/0603-FOE-D1.iso`.

Virtual FileDVDs, like all files, can take advantage of the multipath options with which the file system is created. See Section 9.2.1.3 (page 118) for details.

Virtual FileDVDs are read-only and are shareable across active virtual machines. Use the `hpvmdevmgmt` command to mark them sharable.

To prevent media conflicts, you must manage Virtual FileDVDs carefully (see Section 9.2.1.4 (page 119)). You can see where the file system directory where the ISO file resides using the guest's virtual console. To simplify accounting, allocate file directories from complete logical volumes or whole disks.

A Virtual FileDVD reverts to its original resource statement when the guest shuts down or reboots. Therefore, after you install a guest from multiple CDs or DVDs, you must reload the Virtual FileDVD when the guest reboots to complete the installation. Stop the automatic EFI reboot and insert the CD/DVD using the appropriate `IN` and `EJ` commands. When the media is loaded, proceed with the installation.

NOTE: The `hpvmmodify` command might fail to change a Virtual FileDVD if the device has already been modified by the virtual console. The `hpvmstatus` command displays the current status of the Virtual FileDVD, which might not be in its original resource state. To see the original resource statement, which is required by the `hpvmmodify` command to change a Virtual FileDVD, use the `hpvmstatus -D` command.

9.2.2.3.6 Virtual NullDVDs

A Virtual NullDVD is an emulated SCSI DVD-ROM with no virtual media currently present. The next media selection may come from a VSP CD/DVD drive or VSP ISO file, depending on how the Virtual NullDVD is configured. Once the next media is selected, the Virtual NullDVD turns into either a Virtual DVD (see Section 9.2.2.3.4 (page 127)) or a Virtual FileDVD (see Section 9.2.2.3.5 (page 128)) device. As such, a Virtual NullDVD is a transitory state of an empty virtual DVD type.

The choice of how to configure a Virtual NullDVD depends on the access that the VSP administrator gives to the guest administrator. Virtual DVD changes can be initiated from the virtual console (see Section 9.3.1.3 (page 135)). All virtual DVD changes by the guest administrator are constrained by the actions of the VSP administrator.

If the VSP administrator gives access to the guest administrator to load and unload physical media on the VSP CD/DVD drive, the Virtual NullDVD can be set up with the following form of the resource specification:

`dvd:avio_stor::null:`*`/dev/rdisk/disk#`*

where *`/dev/rdisk/disk#`* is an HP-UX `esdisk` character device file that points to the VSP CD/DVD drive. The legacy sdisk device file, `/dev/rdisk/cXtYdZ`, may also be used.

This is the same as setting up a Virtual DVD (see Section 9.2.2.3.4 (page 127)), except that the VSP CD/DVD might not contain media. The media is expected to come from the guest administrator, who should have access to the VSP to make such physical media changes. For example:

```
# ioscan -NfunC disk

disk        7   64000/0xfa00/0x6    esdisk    CLAIMED     DEVICE
TEAC      DW-224E
             /dev/disk/disk7   /dev/rdisk/disk7
# diskinfo /dev/rdisk/disk7

SCSI describe of /dev/rdisk/disk7:
             vendor: TEAC
         product id: DW-224E
               type: CD-ROM
               size: 0 Kbytes
   bytes per sector: 0
```

In this example, the Virtual NullDVD resource statement is `dvd:avio_stor::null:/dev/rdisk/disk7`.

If the VSP administrator does not want to give access to the VSP CD/DVD drive to the guest administrator, you can set up a Virtual NullDVD to a file system directory containing the ISO files that the guest administrator wants to access. This resource statement would take the following form:

`dvd:avio_stor::null:`*`/pathname`*

where *`/pathname`* is the file system directory where the ISO files are located.

This is the same as setting up a Virtual FileDVD (see Section 9.2.2.3.5 (page 128)), except that the file is not specified. By specifying a file directory, the guest administrator can choose which ISO

files to use from the virtual console. The file directory must be a locally mounted VxFS file system. NFS file systems are not supported. If the ISO files are world writable, they are not available from the virtual console. For the following ISO files:

```
# ls -l /var/opt/hpvm/ISO-images/hpux

total 26409104
-rw-r--r-- 1 root sys 3774611456 Jul 11 :59 0505-FOE.iso
-rw-r--r-- 1 root sys 4285267968 Jul 11 17:05 0512-FOE.iso
-rw-r--r-- 1 root sys 3149987840 Jul 11 18:42 0603-FOE-D1.iso
-rw-r--r-- 1 root sys 29978624 Jul 11 18:51 0603-FOE-D2.iso
```

The Virtual NullDVD resource statement is `dvd:avio_stor::file:/var/opt/hpvm/ISO-images/hpux/.`

You can configure the Virtual NullDVD to be sharable or have multipath options. If the Virtual NullDVD device is configured to use the VSP CD/DVD device, it is not sharable and no multipath options are available. If the Virtual NullDVD is configured to use a file system directory, it is sharable and you can use multipath options (see Section 9.2.1.3 (page 118)). To mark the directory sharable across virtual machines, use the `hpvmdevmgmt` command. For example:

```
# hpvmdevmgmt -m gdev:/var/opt/hpvm/ISO-images/hpux/:attr:SHARE=YES
```

For more information about using the `hpvmdevmgmt` command, see Section 11.11 (page 189).

Virtual NullDVDs require no additional management beyond that required for the Virtual DVD (see Section 9.2.2.3.4 (page 127)) or Virtual FileDVD (see Section 9.2.2.3.5 (page 128)) types they become.

9.2.2.3.7 Attachable devices

vPars and Integrity VM allows you to attach physical VSP backup device types to vPars/VMs. VSP backup device types are tapes, media changers, and CD/DVD burners. These devices are specified on the VSP using HP-UX agile `esctl` device files. Use of the agile `esctl` device files are recommended, because they are per physical device not per path.

The guest OS running on the vPar/VM has full control over an attached physical device. Therefore, the guest OS must support the device being attached. For a list of supported guest OS drivers, see the device's product documentation.

The resource statements for attached devices take the following forms depending upon device type:

- For magnetic tape:
  `tape:avio_stor::attach:`*`/dev/pt/pt_tape#`*
- For media changers:
  `changer:avio_stor::attach:`*`/dev/pt/pt_autoch#`*
- For CD/DVD burners:
  `burner:avio_stor::attach:`*`/dev/pt/pt_disk#`*
  where `/dev/pt/pt_*` files are HP-UX `esctl` device files.

Attachable devices are specified as *`avio_stor`*.

To create an HP-UX `esctl` device file, follow these steps:

1. Run `ioscan` to pick up any new devices that were connected:
   ```
   # ioscan
   ```
2. Locate the device designated for attachment.
   a. Install any device special files for these new devices:
   ```
   # insf -e
   ```
   b. Verify whether the new devices were claimed by VSP:

```
# ioscan -Nfun
```

The following is an example of a claimed tape device:

```
tape       1   64000/0xfa00/0x19   estape   CLAIMED   DEVICE   HP Ultrium 1-SCSI
/dev/rtape/tape1_BEST            /dev/rtape/tape1_BESTn
     /dev/rtape/BESTb           /dev/rtape/tape1_BESTnb
```

If the device is not seen in `ioscan -fun`, proceed to step 2c. Otherwise, go to step 3.

c. If the device is not claimed, make sure the device is seen:

```
# ioscan -fk
```

The following is an example of an unclaimed media changer device:

```
Class       I  H/W Path       Driver     S/W State   H/W Type     Description
============================================================================

ext_bus     6  0/2/1/0        c8xx       CLAIMED     INTERFACE    SCSI C1010
Ultra0 Wide LVD A6828-60101
target     35  0/2/1/0.0      tgt        CLAIMED DEVICE
unknown    -1  0/2/1/0.0.0               UNCLAIMED   UNKNOWN      HP ThinStor
AutoLdr
```

If the device is not seen, there is a hardware problem or AVIO ID conflict. Consult the documentation for the particular device to resolve this issue before proceeding.

If the device is seen but not claimed, this is a result of missing drivers in the VSP. Integrity VM does not require the drivers to be loaded on the VSP for the devices to be attached. The HP-UX tape (`stape`) and changer (`schgr`) drivers are not loaded by default unless those devices are connected at install time. To load the drivers, use the `kcmodule` command to statically load the drivers. To complete the installation, the VSP must be rebooted. Any guests that are running must be shut down before loading these drivers.

The following is an example of installing the tape driver:

```
# kcmodule stape=static
```

The following is an example of installing the media changer driver:

```
# kcmodule schgr=static
```

If you are not loading the VSP drivers, proceed to step 4.

If you are loading the VSP drivers, the devices should show up in `ioscan` with device files after the VSP reboot. In which case, proceed to step 3.

3. Install `esctl` device files using the `hpvmdevmgmt` command. For example:

```
# hpvmdevmgmt -I
```

4. Locate the passthrough device file that corresponds to the device slated for attachment.

The following are examples of a tape device:

```
Agile = /dev/rtape/tape1_BEST
 ESCTL = /dev/pt/pt_tape1
```

The following are examples of media changer device:

```
Agile = /dev/rchgr/autoch1
 ESCTL = /dev/pt/pt_autoch1
```

The following are examples of CD/DVD burner device:

```
Agile = /dev/rdisk/disk7
 ESCTL = /dev/pt/pt_disk7
```

Attached devices cannot be shared simultaneously across active vPars/VMs. Only one active vPar/VM can be given a particular attached device at a time. However, like virtual devices, attached devices can be attached and detached dynamically across active vPars/VMs (see Section 9.3 (page 133)). Also, as the device is being attached to a vPar/VM, it cannot be opened by the VSP at the time of or during attachment.

Because tapes, media changers, and CD/DVD burners are not virtualized, media changes with them must be done physically. Therefore, all media changes with attached devices must be done by individuals with access to that physical storage. Changes to attached devices might require the device to be unlocked from an active guest OS. Attached devices remain in the last lock state the guest OS put it in when the device is detached or the virtual machine is shut down. Empty devices are attached and are not locked.

No multipath solutions are available for attached devices on the VSP. No multipath products are supported in the vPar/VM.

Manage attached devices to prevent the wrong vPars/VMs from viewing sensitive information. You can display which vPars/VMs are currently using attached devices using the `hpvmstatus` command.

9.2.2.4 NFS-Mounted backing stores for root, swap, and dump

vPars and Integrity VM V6.1 supports NFS-mounted backing stores for root, swap and dump. These backing-store files can now be located in NFS-mounted file systems. The following configuration requirements apply. These configuration requirements might be removed or changed in future vPars and Integrity VM releases.

- NFS-mounted backing stores can be used only for the root (that is, boot) file system, swap and dump. NFS-mounted cannot be used as file-backed virtual DVD drives.
- The maximum number of NFS-mounted backing stores per guest is four.
- NFS-mounted backing stores are supported only for HP-UX 11i v3 guests.
- NFS-mounted backing stores must be configured with AVIO.
- The following NFS mount options must be used by the VSP when mounting an NFS file system housing a guest's backing-store files:
  - NFS Version 3
  - TCP
  - Hard
  - IPv4 address or server host names mapping to IPv4 address
- The Integrity VSP (NFS client) and the NFS server systems must reside in the same IP subnet.
- Online VM Migration is supported for VMs using NFS-mounted backing stores. For OVMM to work successfully, both Integrity VSPs must mount the NFS file system housing the guest's backing-store files using the identical syntax and mount options. Both the source and target VSPs must have the NFS file system mounted at the time of the migration.

The following limitations apply to this release of the NFS-mounted backing stores feature in vPars and Integrity VM V6.1:

- Integrity VM guests configured with NFS-mounted backing stores cannot be integrated with Serviceguard as either a package (VM-as-an-SG-Package) or node (VM-as-an-SG-Node).
- The use of symbolic links on the NFS server to redirect the location of a guest's backing-store files is not allowed. However, symbolic links are still allowed inside the guest booted with an NFS backing store.
- NFS file systems housing a guest's backing stores must be mounted using IPv4. Mounting NFS backing stores using IPv6 is not allowed at this time.
- Management of Integrity VM guests configured with NFS-mounted backing stores is not supported with the following management applications:

- Logical Server Manager (LSM) 6.2
- HP Infrastructure Orchestration (HPIO) 6.2
- HP Insight Software 6.2

When creating NFS-mounted backing-store files, HP recommends that you create these files locally on the NFS server, if possible. You can use either the `hpvmdevmgmt` command, if available on the NFS server, or the `dd` command. For example, to create an 80 GB file on an HP-UX NFS server as a guest backing store in the shared directory called `/export`, use either on of the following commands:

```
/opt/hpvm/bin/hpvmdevmgmt -S 80G /export/vm1.boot
/usr/bin/dd if=/dev/zero of=/export/vm1.boot bs=1024K count=80000
```

If the local access to the NFS server is not available, you can use these same commands on the VSP inside the NFS-mounted file system.

NOTE: Creating a guest's backing-store files on an NFS client system (that is, VSP), can take significantly longer to complete than creating the backing-store files locally on the NFS server directly. Therefore, create a guest's backing-stores files directly on the NFS server, if possible.

9.3 Using vPars and Integrity VM storage

The following sections describe the roles of individuals accessing virtual storage, the commands they use, and some examples of using vPars and Integrity VM storage.

9.3.1 Storage roles

This section describes the roles that individuals play in working with vPars/VM storage. Each role has different responsibilities in using vPars/VM storage. The roles might be played by one or more individuals depending on security requirements and skill sets. The three roles are:

- VSP administrator
- Guest administrator
- Guest user

9.3.1.1 VSP administrator

The VSP administrator role is an individual responsible for the proper configuration and maintenance of the VSP for running vPars/VMs. As such, this person needs complete access to the VSP to install hardware and software. This person also needs to understand how to do HP-UX system maintenance, how to configure hardware properly, and how to set up and use various software applications and tools.

The VSP administrator uses the following commands to manage vPar/VM storage devices:

Management Function	Integrity VM Command
Add, delete, manage, and modify vPar/VM storage devices.	`hpvmmodify` (see Section 7.4 (page 90))
Display information about the storage devices for a vPar/VM.	`hpvmstatus` (see Section 11.3 (page 170))

Once a resource is added or attached to a vPar/VM and the vPar/VM is powered on, the storage resource is owned by the guest administrator. That is, the guest OS may access that storage resource at any time. A deletion, detachment or modification fails if any guest I/O is active on the resource. Dynamic storage changes on an active virtual machine must be approved by the guest administrator.

9.3.1.2 Creating vPar/VM administrator and operator accounts

In prior versions of Integrity VM, `admin` console access is available, and one such account per guest is allowed. The administrator account name must match the guest name. The new version of vPars and Integrity VM provides proper access controls and individual accountability for these accounts.

A captive virtual console account is a special-purpose user account created on the VSP for each guest administrator. These types of user accounts use `/opt/hpvm/bin/hpvmconsole` for a shell, and the desired guest's per-guest directory for a home directory. For virtual console access, the account also requires a password, and access to its associated guest. You create this account with the `hpvmcreate`, `hpvmclone`, or `hpvmmodify` command. You can establish group membership of the account using the `-g` option to those commands, or user membership, using the `-u` option to those commands.

NOTE: Do not use the `hpvmsys` group for user accounts. This group is used for security isolation between components of vPars and Integrity VM.

The HP-UX `useradd` command might not work as expected. To create user accounts for virtual console access, use the `useradd` command before you create the virtual machine. Alternatively, specify the user account directory completely in the `/etc/passwd` file, ensuring the entry is unique.

In the following example, the `useradd` command is used to create three user accounts on the VSP system (`testme1`, `testme2`, and `testme3`):

```
# useradd -r no -g users -s /opt/hpvm/bin/hpvmconsole \
 -c "Console access to guest 'testme'" \
 -d /var/opt/hpvm/guests/testme \
 testme1
# useradd -r no -g users -s /opt/hpvm/bin/hpvmconsole \
 -c "Console access to guest 'testme'" \> -d /var/opt/hpvm/guests/testme \
 testme2
# useradd -r no -g users -s /opt/hpvm/bin/hpvmconsole \
 -c "Console access to guest 'testme'" \
 -d /var/opt/hpvm/guests/testme \
 testme3
```

The following command creates the virtual machine named `testme`:

```
# hpvmcreate -P testme -u testme1:admin -u testme2 -u testme3:oper
```

At this point, users `testme2` and `testme3` both have `oper` level access to the virtual console, and user `testme1` has `admin` level access. In order to make these accounts usable, set passwords for them, as follows:

```
# passwd testme1
...
# passwd testme2
...
# passwd testme3
...
```

Because of the way the `useradd` command works, an attempt to create an additional account might result in an error. For example, the following command attempts and fails to add the `testme4` user account:

```
# useradd -r no -g users -s /opt/hpvm/bin/hpvmconsole \
> -c "Console access to guest 'testme'" \
> -d /var/opt/hpvm/guests/testme \
> testme4
'/var/opt/hpvm/guests/testme' is not a valid directory
```

To enter the command correctly, include the entire directory path. For example:

```
# useradd -r no -g users -s /opt/hpvm/bin/hpvmconsole \
> -c "Console access to guest 'testme'" \
> -d /var/opt/hpvm/guests/testme/. \
```

```
> testme4
# hpvmmodify -P testme -u testme4
# passwd testme4
```

Note the addition of the slash and period (`/.`) to the end of the argument to the `-d` option, which ensures there is no confusion with HP-UX shared home directories.

9.3.1.3 Guest administrator

The vPar/VM Guest Administrator is responsible for the proper maintenance of a guest OS. As such, this person needs access to the virtual console by the VSP administrator to control the vPar/VM. The guest administrator must understand how to maintain the guest OS, install patches and applications, and set up security for the guest users of the guest OS. Additionally, vPar/VM storage requires you to:

- Install any specific guest OS patches required by vPars and Integrity VM for proper OS operation on the virtual platform.
- Review and understand any vPar/VM storage release notes that are specific to the guest OS.
- Work with the VSP administrator to complete virtual storage changes, including managing attached VSP devices.

The guest administrator uses the virtual console to modify virtual storage. The virtual console is used to change discs of a virtual DVD device type. All modifications are bounded by what the VSP administrator configures for the virtual machine.

The virtual console commands are available from the vMP Main Menu, using the `hpvmconsole` command or by pressing **Ctrl/B** if you are already connected . The virtual console commands eject (`ej`) and insert (`in`) allow you to control the DVD device. Both commands provide submenus for displaying devices that are removable. Selecting options through the submenus completes the ejection/insertion process.

If the guest `hpvmconsole pc –cycle` command doesn't complete and restart the guest, enter **Ctrl/B** to interrupt the command and then press **Enter** to return to the virtual console. Exit the virtual console by entering the `X` command. At the VSP command prompt, enter the following command to start the guest:

```
# hpvmstart –P guestname
```

NOTE: If a guest hangs, attach to the guest's virtual console using the `hpvmconsole` command, then use **Ctrl/B** to enter the virtual console. Enter the `tc` command to reset the guest. The guest captures a memory dump of the machine state, which can be used later for offline diagnosis. Do not kill the guest from the VSP or use the virtual console to power down a hung guest. Doing so can corrupt the guest file system.

Management Function	Integrity VM Command
Eject a virtual DVD.	`vMP> ej`
Insert a virtual DVD	`vMP> in`

NOTE: When a DVD without a disk in the drive is added to a guest, specify the backing store type of `null`, for example:

```
# hpvmmodify -P guest -a dvd:avio_stor::null:/dev/rdisk/disk#
```

Run `ioscan` on the booted guest if the guest if running HP-UX.

If an empty DVD drive is given the backing store type `disk`, the following example shows the result:

```
# hpvmmodify -P testguest -a  dvd:avio_stor::disk:/dev/rdisk/disk31
hpvmmodify: WARNING (testguest): DVD or burner: '/dev/rdisk/disk31' currently has no disk. This device may not
 show up or be usable by the guest when booted.
```

If a guest boots when configured with a DVD using the `disk` backing store type when there is no disk in the drive, the guest kit utility command `hpvmdevinfo` (available for HP-UX guests) might return the following type of results:

```
# hpvmdevinfo
hpvmdevinfo: Error converting (0,0,1): Error 0
Device Type     Bus,Device,Target       Backing Store Type      Host Device Name        Virtual Machine Device
 Name
===========    =================      ==================     ================       ============================
disk            [0,0,0]                 disk                    /dev/rdisk/c2t0d0        /dev/rdisk/c0t0d0
dvd             [0,0,1]                 disk                    /dev/rdisk/disk31       ??
```

The following results indicate a problem of an empty DVD drive:

- The "Error converting (0,0,1): Error 0" message
- The "??" string in the field for the virtual machine's device name

Output appears for the `dvd` , because it is stored as part of the guest configuration on the VSP. However, because there is no disk in the drive, the drive itself is not virtualized as a device within the guest. Also note that the DVD drive does not show up in `ioscan` output in the guest.

9.3.1.4 Guest user

The guest user runs applications on a guest OS. Access is provided and limited by the guest administrator. There are no Integrity VM storage requirements for application users of the guest OS.

There are no Integrity VM storage commands for application users in the guest OS. The guest users use Integrity VM storage on the guest OS the same way as they normally use storage on an HP Integrity server. Any required Integrity VM storage changes must be directed to the guest administrator or VSP administrator.

9.3.2 Storage use cases

This subsection describes ways to use the vPar/VM storage commands.

9.3.2.1 Adding virtual storage devices

A VSP administrator adds or attaches vPar/VM storage using the `hpvmstatus` and `hpvmmodify` commands. Virtual storage devices can be added or attached while the vPar/VM is powered on or off. A new virtual storage adapter can be added only when the vPar/VM is off. The virtual storage adapter can have up to 128 AVIO devices total (the number of virtual and attached devices.)

The process to add or attach a virtual storage device to a guest is as follows:

1. Based on the all vPar/VM storage considerations, choose a storage device to add.
2. Based on the device type, set up and configure the VSP to form a valid resource statement. This includes accounting VSP resources to avoid future storage conflicts.
3. Use the valid resource statement with the `hpvmmodify` command to add or attach the vPar/VM storage device.

The resource statement for adding an vPar/VM storage device does not require virtual hardware addressing. If the PCI bus, slot and target numbers are not specified, vPars and Integrity VM automatically chooses the first position available for the device. For example:

```
# hpvmmodify -P myvmm -a disk:avio_stor::disk:/dev/rdisk/disk7
# hpvmstatus -P myvmm
..
[Storage Interface Details]
...
disk avio_stor 0 1 0 0 0 disk /dev/rdisk/disk5
disk avio_stor 0 1 0 1 0 disk /dev/rdisk/disk7
```

To add an AVIO storage device with whole disk as the backing store, specify the following:

```
host# hpvmmodify -P guest1 -a disk:avio_stor:0,5,0:disk:/dev/rdisk/disk11
```

NOTE: You can achieve higher guests performance for HP-UX 11i v3 guests by configuring as many AVIO storage adapters as the number of virtual CPUs in the guest. The `pcibus`, `pcislot`, and `aviotgt` portions need to be explicitly specified for each device. For example, a resource statement for a 4–vCPU guest takes the following form:

```
-a disk:avio_stor:1,0,0:disk:/dev/rdisk/disk1
-a disk:avio_stor:1,1,0:disk:/dev/rdisk/disk2
-a disk:avio_stor:1,2,0:disk:/dev/rdisk/disk3
-a disk:avio_stor:1,4,0:disk:/dev/rdisk/disk4
```

9.3.2.2 Deleting storage devices

A VSP administrator deletes or detaches vPar/VM storage using the `hpvmstatus` and `hpvmmodify` commands. vPar/VM storage devices can be deleted or detached while the virtual machine is powered on or off. An Integrity VM storage adapter can only be removed when the virtual machine is off. The vPar/VM storage adapter is automatically removed when the last vPar/VM storage device connected to the adapter is removed.

The process to delete or detach a virtual storage device from a vPar/VM is as follows:

1. Use the `hpvmstatus` command to locate the resource to verify whether the vPar/VM is powered on. If the vPar/VM is on, consult with the guest administrator to obtain permission to remove the resource before proceeding.
2. Use the `hpvmmodify` command to delete or detach the resource.
3. Verify that the VSP resource is no longer being used by the vPar/VM.

The resource statement for deleting an vPar/VM storage device does not require virtual hardware addressing. For example:

```
# hpvmstatus -P myvmm
...
[Storage Interface Details]
...
disk avio_stor 0 1 0 0 0 disk /dev/rdisk/disk5
disk avio_stor 0 1 0 1 0 disk /dev/rdisk/disk7
disk avio_stor 0 1 0 2 0 disk /dev/rdisk/disk9
disk avio_stor 0 5 0 0 0 disk /dev/rdisk/disk11
# hpvmmodify -P myvmm -d disk:avio_stor::disk:/dev/rdisk/disk7
# hpvmstatus -P myvmm
...
[Storage Interface Details]
```

```
disk avio_stor 0 1 0 0 0 disk /dev/rdisk/disk5
disk avio_stor 0 1 0 2 0 disk /dev/rdisk/disk9
```

To delete an AVIO storage device, specify the following:

```
host# hpvmmodify -P guest1 -d disk:avio_stor:0,5,0:disk:/dev/rdisk/disk11
```

9.3.2.3 Modifying storage devices

The VSP administrator or the guest administrator can modify an vPar/VM storage device. The VSP administrator can use the `hpvmstatus` and `hpvmmodify` commands to change the virtual media of virtual devices. The guest administrator uses the virtual console to change the virtual media of virtual DVDs. All attached devices are modified using physical VSP access.

When the VSP administrator uses the `hpvmstatus` and `hpvmmodify` commands to modify the virtual media of a virtual device, the operation is seen by the guest OS as a whole-disk replacement or a DVD removable media event, depending on the device type.

The process for modifying the virtual media of a virtual device is as follows:

1. Use the `hpvmstatus` command to locate the virtual device resource to modify and to see if the virtual machine is powered on. If the vPar/VM is on, consult with the guest administrator to before proceeding to replace the virtual media.
2. Based on the vPar/VM storage considerations, choose a new virtual media type to add.
3. Based on the virtual media type, set up and configure the VSP to form a valid VSP storage specification. Take into account the other demands on VSP resources to avoid vPar/VM storage conflicts.
4. Use the VSP storage specification with the `hpvmmodify` command to modify the virtual device resource.
5. Verify that the old VSP resource is no longer in use by a vPar/VM.
6. When run on an active vPar/VM and with a storage device managed by `avio_stor` HBA, the vPar/VM guest needs to run the `gvsdmgr` command prior to using the modified backing store. For information about the `gvsdmgr` utility, see the HP-UX *gvsdmgr*(1M) manpage.

The resource statement for modifying a virtual device requires virtual hardware addressing (see Section 9.2.2.1 (page 122)). For example:

```
# hpvmstatus -P myvmm
...
[Storage Interface Details]
...
disk scsi 0 1 0 0 0 disk /dev/rdisk/disk5
disk scsi 0 1 0 1 0 disk /dev/rdisk/disk7
disk scsi 0 1 0 2 0 disk /dev/rdisk/disk9
# hpvmmodify -P myvmm -m disk:scsi:0,1,1:lv:/dev/rdisk/disk2
# hpvmstatus -P myvmm
...
[Storage Interface Details]
...
disk scsi 0 1 0 0 0 disk /dev/rdisk/disk5
disk scsi 0 1 0 1 0  lv  /dev/rdisk/disk2
disk scsi 0 1 0 2 0 disk /dev/rdisk/disk9
```

To complete a DVD ejection and insertion, follow the virtual console menus. However, new media selections might require the help of the VSP administrator. Changes through the virtual console are not saved across guest OS reboots

If the VSP administrator sets up a Virtual DVD for the vPar/VM, the virtual console eject and insert command unlock and lock the physical VSP CD/DVD drive. The `eject` command changes the Virtual DVD into a Virtual NullDVD in the vPar/VM, unlocking the VSP CD/DVD drive in the process. The physical media in the VSP CD/DVD drive can then be changed by the VSP administrator or the guest administrator if access is permitted. Once the media has been changed, the `insert` command can be used to change the Virtual NullDVD back into a Virtual DVD, locking the VSP CD/DVD drive and making the newly loaded media now accessible by the vPar/VM. For example:

```
# diskinfo /dev/rdisk/disk7
AVIO describe of /dev/rdisk/disk7:
             vendor: HP
         product id: Virtual DVD
               type: CD-ROM
               size: 665600 Kbytes
   bytes per sector: 2048
vMP> ej

                Ejectable Guest Devices
Num     Hw-path         (Bus,Slot,Tgt)  Gdev    Pstore  Path
-------------------------------------------------------------------------
[1]     0/0/1/0.7.0     (0,1,7)         dvd     disk    /dev/rdisk/disk7

Enter menu item number or [Q] to Quit: 1
Confirm eject action
     G - Go
     F - Force

Enter menu item or [Q] to Quit: G
vMP> co
# diskinfo /dev/rdisk/disk7
SCSI describe of /dev/rdisk/disk7:
             vendor: HP
         product id: Virtual NullDVD
               type: CD-ROM
               size: 0 Kbytes
   bytes per sector: 0

vMP>
```

After inserting a new disk on the VSP CD/DVD drive, enter the following:

```
vMP> in
Insertable Guest Devices
Num     Hw-path         (Bus,Slot,Tgt)  Gdev
-----------------------------------------
[1]     0/0/1/0.7.0     (0,1,7)         dvd

Enter menu item number or [Q] to Quit: 1
Insertable File Backing Stores
Num     File
--------------------
[1]     /dev/rdisk/disk7

Enter menu item number or [Q] to Quit: 1
Confirm insertion action
     G - Go
     F - Force

Enter menu item or [Q] to Quit: G
vMP> co
# diskinfo /dev/rdisk/disk7
SCSI describe of /dev/rdisk/disk7:
             vendor: HP
         product id: Virtual DVD
               type: CD-ROM
               size: 4300800 Kbytes
   bytes per sector: 2048
```

To modify an existing AVIO storage backing store, specify the following:

```
host# hpvmmodify -P guest1 -m disk:avio_stor:0,5,0:disk/dev/rdisk/disk11
```

In this command, avio_stor indicates the “from” adapter and the “bus,dev” specification indicates the bus and device list of storage targets to convert.

For information about AVIO support, see the *HP-UX vPars and Integrity VM V6.1 Release Notes* at http://www.hp.com/go/hpux-hpvm-docs.

Prior to running the `hpvmmodify` command shown previously, devices unsupported by AVIO need to be moved to a new HBA, using the `hpvmmodify` delete and add operations.

NOTE: Guest operating systems, applications, or configuration files sensitive to device names or hardware paths need to be repaired after the move. Because HP-UX 11i v3 supports the agile device naming model, 11i v3 guest applications using agile device names are not affected as long as they are configured with disk backing stores.

If the VSP administrator sets up a Virtual FileDVD for the vPar/VM, the virtual console options to eject and insert are used to select among the ISO files provided in the file directory for the Virtual FileDVD. The `eject` command changes the Virtual FileDVD into a Virtual NullDVD device. ISO files can be added to or removed from the file system directory for the Virtual FileDVD by the VSP administrator. Once this ISO file directory is updated, use an `insert` command to view all the newly available ISO files in the directory and to choose one to be used for a new Virtual FileDVD. It is not necessary to change the file directory between each eject and insert operation. The guest administrator can change the ISO files provided in the file directory without any VSP administrator interaction. For example:

```
# diskinfo /dev/rdisk/disk0
SCSI describe of /dev/rdisk/disk0:
             vendor: HP
         product id: Virtual FileDVD
               type: CD-ROM
               size: 665600 Kbytes
   bytes per sector: 2048
vMP> ej
                Ejectable Guest Devices
Num     Hw-path         (Bus,Slot,Tgt)  Gdev    Pstore  Path
------------------------------------------------------------------------------
[1]     0/0/1/0.7.0     (0,1,7)         dvd     file    /var/opt/hpvm/ISO-images/hpux/IOTdisc

Enter menu item number or [Q] to Quit: 1

Confirm eject action
     G - Go
     F - Force

Enter menu item or [Q] to Quit: G
vMP> co
vm # diskinfo /dev/rdisk/disk0
SCSI describe of /dev/rdisk/disk0:
             vendor: HP
         product id: Virtual NullDVD
               type: CD-ROM
               size: 0 Kbytes
   bytes per sector: 0

vMP> in
                Insertable Guest Devices
Num     Hw-path         (Bus,Slot,Tgt)  Gdev
------------------------------------------
[1]     0/0/1/0.7.0     (0,1,7)         dvd

Enter menu item number or [Q] to Quit: 1
                Insertable File Backing Stores
Num     File
--------------------
[1]     0505-FOE.iso
[2]     0512-FOE.iso
[3]     0603-FOE-D1.iso
[4]     0603-FOE-D2.iso
[5]     IOTdisc

Enter menu item number or [Q] to Quit: 1
Confirm insertion action
     G - Go
     F - Force

Enter menu item or [Q] to Quit: G
vMP> co
# diskinfo /dev/rdisk/disk0
```

```
SCSI describe of /dev/rdisk/disk0:
             vendor: HP
         product id: Virtual FileDVD
               type: CD-ROM
               size: 3686144 Kbytes
   bytes per sector: 2048
```

For attached devices, modifications are made physically on the device. The guest OS supplies commands for loading and unloading tapes using media changers. But loading new media into the media changer, changing tapes in standalone drives, and changing discs with CD/DVD burners are accomplished manually. This process requires cooperation between the VSP administrator and the guest administrator.

10 Creating virtual networks

You can allocate virtual network devices or virtual network interface cards (vNICs) to the vPar or VM when you create the them with the `hpvmcreate` command or when you modify an existing vPar/VM using the `hpvmmodify` command, as described in Chapter 7 and "Creating virtual partitions" (page 59). Virtual network interface cards are added using the same option that is used to add storage devices, but the format of the argument to the command option is different. To add a vNIC to a guest, use the following command option:

`-a network:adaptertype:bus,device,mac-addr:vswitch:vswitch-name:portid:portnumber`

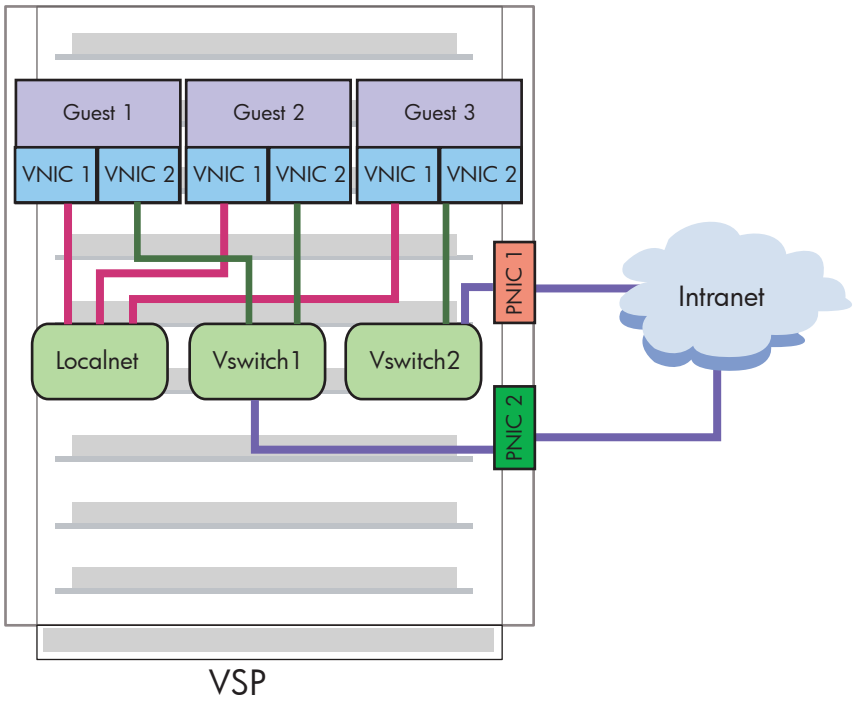
However, before you can allocate the vswitch to the vPar/VM, you must create the virtual switch (vswitch) using the `hpvmnet` (or `vparnet` command) command. This chapter describes how to create and manage vswitches, including the following topics and tasks:

- Introduction to virtual network configuration
- Creating and managing vswitches
- Managing vNICs
- Configuring VLANs
- Using DIO functionality
- Troubleshooting network problems

10.1 Introduction to virtual network configuration

The guest virtual network configuration provides flexibility in network configuration, allowing you to provide high availability, performance, and security to the vPars/VMs running on the VSP. The basic virtual network configuration is illustrated in Figure 10.

Figure 10 Virtual Network Configuration



The virtual network configuration consists of the following components:

- VSP physical network interface card (pNIC) — the physical network adapter, which may be configured with Auto Port Aggregation (APA). (For more information about APA, see the *HP Auto Port Aggregation (APA) Support Guide*.)

  NOTE: Trunking software such as APA is supported only on the VSP, not on the guest.

  APA can be configured on the VSP to provide a highly available LAN for the vswitch (APA in active/passive mode) or to increase the bandwidth of the vswitch LAN (APA active/active mode). Before you stop APA, use the `hpvmnet -h` command to halt the vswitch. If you do not halt the vswitch first, the `hpvmnet` command reports an incorrect MAC address for the vswitch.

- Guest virtual network interface card (vNIC) — the virtual network adapter, as recognized by the guest operating system.
- Virtual switch (vswitch) — the virtual network switch maintained by the VSP that is associated with a pNIC and can be allocated to one or more guests.

  Vswitches must not be connected to network devices that are set to promiscuous mode. Do not run applications like `tcpdump` on the VSP on interfaces that are used for virtual switches.

Using redundant pNICs and APA, you can ensure high availability of the guest networks and provide greater capacity for the VSP system running many guests with network-intensive applications.

You can configure HP-UX VLANs for the guests. VLANs isolates broadcast and multicast traffic by determining which targets should receive that traffic, thereby making better use of switch and end-station resources. With VLANs, broadcasts and multicasts go only to the intended nodes in the VLAN.

10.2 Creating and managing vswitches

The following sections describe how to create, modify, delete, and manage vswitches.

10.2.1 Creating vswitches

To allow guests to access network devices, you must create vswitches on the VSP. This section describes how to create a vswitch and verify that it has started.

To create vswitches, use the `hpvmnet` command. The following is the basic format of the `hpvmnet` command to create a vswitch:

`hpvmnet -c -S` *`vswitch-name`* `-n` *`nic-id`*

This command format includes the following options:

- `-c` indicates the creation of a vswitch.
- **`-S`** *`vswitch-name`* specifies the name of the vswitch.
- `-n` *`nic-id`* specifies the network interface on the VSP that the new vswitch will use. For example, `–n 0` indicates `lan0`. Network interfaces are displayed by the `lanscan` command. If you do not include the `-n` option, a local vswitch is created, as described in Section 10.2.1.1 (page 147).

The `hpvmnet` command also allows you to display and manage the vswitches on the VSP. Table 21 describes the options to the `hpvmnet` command.

Table 21 Options to the `hpvmnet` Command

Option	Description
`-b`	Boots a vswitch. The vswitch must be booted before it can accept network traffic. All vswitches are booted automatically when Integrity VM is started.
`-c`	Creates a new vswitch.
`-h`	Halts one or all vswitches. You are asked to confirm this action.
`-F`	Omits the confirmation dialog before halting, deleting, or rebooting the vswitch. This option is intended for use by scripts and other noninteractive applications (Force mode). **NOTE:** The `-F` option is deprecated in Integrity VM commands; this option should be used only at the direction of HP Support.
`-d`	Deletes a virtual switch. You are asked to confirm this action.
`-n` *`nic-id`*	Specifies the network interface on the VSP that the new vswitch will use. For example, to associate a vswitch to `lan0`, enter `-n 0`.
`-p` *`n`*	Specifies the port number. To display information about all ports, enter `–p all`.
`-Q`	Specifies the command function should proceed without asking for confirmation. By default, the command requires confirmation, and does not proceed without it.
`-r`	Restarts the vswitch information.
`-s` *`vswitch_number`*	Retrieves statistics of the vswitch specified by its number..
`-S` *`vswitch_name`*	Specifies the name of the virtual switch. The vswitch name can be up to 64 characters and must be unique on the VSP.
`-u portid:`*`portnum`*`:vlanid:[`*`vlanid`*` | none]`	Configures the port *`portnum`* on the virtual switch so that it is isolated to the VLAN specified by *`vlanid`*. See Section 10.4: "Configuring VLANs" (page 152) for more information.
`-i`	Enables list of VLAN ids on the list of ports. Specifying *`all`* allows you to enable all VLANs at once.
`-A`	Displays information about vswitches in verbose mode. If you specify the vswitch using either the `-S` or `-s` options, network counters are included in the display.
`-o`	Disables the list of VLAN ids on the list of ports. Specifying *`all`* disables all VLANs at once.
`-Z`	Used with the `-A` option, clears statistics after retrieving them.
`-M`	Displays verbose resource information in a machine-readable format. The `-V`, `-M`, and `-X` options are mutually exclusive.
`-X`	Displays verbose resource information in XML format.
`-V`	Enables verbose mode, displaying information detailed information about one or all vswitches. The `-V`, `-M`, and `-X` options are mutually exclusive.

Table 21 Options to the `hpvmnet` Command *(continued)*

Option	Description
`-v`	Displays the version number of the `hpvmnet` command in addition to the vswitch information.
`-C`	Changes the specified vswitch. If used with the `-N` option, the changes are made to the cloned vswitch. You must include either the `-S` or `-s` option.
`-N` *new-vswitch-name*	Creates a new vswitch based on the existing vswitch. For *new_vswitch_name*, specify the unique name of the new virtual switch. The name of the vswitch can be up to 64 characters. You must include either the `-S` or `-s` option.

NOTE: When working with vPars, you can also use the `vparnet` command. See the `vparnet(1M)` manpage.

The following command creates a virtual switch called `clan1` that is associated with `lan1`. The second `hpvmnet` command displays information about the `clan1` vswitch.

```
# hpvmnet -c -S clan1 -n 1
# hpvmnet

Name     Number State   Mode       PPA    MAC Address    IP Address
======== ====== ======= ========== ====== ============== ===============
localnet      1 Up      Shared            N/A            N/A
myswitch      2 Up      Shared            N/A            N/A
clan1         5 Down    Shared     lan1
```

The physical point of attachment (PPA) for `clan1` is `1`. Two vswitches (`localnet` and `lan0`) communicate over the localnet.

To boot a vswitch, enter the `hpvmnet` command with the `-b` option. For example, to boot the vswitch named `clan1`, enter the following command:

```
# hpvmnet -S clan1 -b
# hpvmnet -v

Name     Number State   Mode       PPA    MAC Address    IP Address
======== ====== ======= ========== ====== ============== ===============
localnet      1 Up      Shared            N/A            N/A
myswitch      2 Up      Shared            N/A            N/A
clan1         5 Up      Shared     lan1   0x00306e3977ab
```

Note that `clan1` is associated with the network interface on the VSP that has MAC address `0x00306e3977ab` (this is not the MAC address of any virtual machine connected to this vswitch).

For information about connecting vswitches to guests, see Chapter 7. For information about modifying virtual networks, see Section 10.3.1 (page 151).

You can create multiple vswitches associated with the same host physical NIC. However, you cannot boot (`hpvmnet –b`) more than one of them at the same time.

NOTE: The Cisco switch for HP BladeSystem c-Class Server Blades has a protocol error that causes it to respond to every MAC address. Because MAC addresses are unique, Integrity VM checks that the generated guest virtual MAC address is unique. If one of these bad switches is on your network, Integrity VM's check will fail.

The `hpvmcreate` command might fail with messages like the following:

```
hpvmcreate:  WARNING (host): Failed after 3 attempts.
hpvmcreate:  WARNING (host): Unable to create Ethernet MAC Address.
```

Similarly, the `hpvmstart` command might fail with messages like the following:

```
# hpvmstart -P vm2
HPVM guest vm2 configuration problems:
Warning 1 on itme nic1: Guest MAC address for switch nic1 is in use.
```

Cisco Systems, Inc. released a fix for the Cisco Catalyst Blade Switch 3020 in December 2006, which is available from the Cisco Systems website:

http://cco.cisco.com

It is also available from the HP website:

http://www.hp.com

From the HP website, select Software & Driver downloads and search for switch cisco 3020. The minimum required firmware version is 12.2(35) SE.

10.2.1.1 Local networks

Virtual network communication may be limited to virtual machines on the VSP system through the use of vswitches that are not connected to a physical NIC. A virtual network such as this is called a local virtual network or simply a local network (`localnet`). To create a local network, a vswitch must first be created using `hpvmnet` without the `-n` option, so that it is not connected to the physical network. For example, to create a local network vswitch named `clan0`, enter the following commands:

```
# hpvmnet -c -S clan0
# hpvmnet -b -S clan0
```

All vNICs connected to that vswitch will then be on the same local network. The VSP does not communicate on local networks.

The following command adds a vNIC to the guest `host1`, which can be used to communicate with any virtual machine connected to the `localnet` vswitch.

```
# hpvmmodify -P host1 -a network:lan::vswitch:clan0
```

During startup of the Integrity VM software, a default vswitch, localnet, is created and booted. The localnet vswitch can be added to a guest, which allows communication between any other guest using the localnet vswitch. For example:

```
# hpvmmodify -P compass1 -a network:lan::vswitch:localnet
```

10.2.2 Changing vswitches

You can use the `-C` option to change the physical network interface card (pNIC) the guest has in use. For example, enter the `lanscan` command, as follows:

```
# lanscan
Hardware Station        Crd Hdw   Net-Interface  NM  MAC       HP-DLPI DLPI
Path     Address        In# State NamePPA        ID  Type      Support Mjr#
0/0/3/0  0x00306E4A93E6 0   UP    lan0 snap0     1   ETHER     Yes     119
0/1/2/0  0x00306E4A92EF 1   UP    lan1 snap1     2   ETHER     Yes     119

# hpvmnet
Name     Number State   Mode      NamePPA  MAC Address    IP Address
```

```
======== ====== ======= ========= ======== ============== ===============
localnet      1 Up      Shared              N/A            N/A
hostnet     296 Up      Shared    lan0      0x00306e4a93e6
```

If `lan0` goes down, enter the following command to swap to use `lan1`:

```
# hpvmnet -C -S hostnet -n 1
# hpvmnet
Name     Number State   Mode      NamePPA  MAC Address    IP Address
======== ====== ======= ========= ======== ============== ===============
localnet      1 Up      Shared              N/A            N/A
hostnet     296 Up      Shared    lan1      0x00306e4a92ef
```

10.2.3 Cloning vswitches

Using the `-N` option with the `-C` option creates a new vswitch based on the changed vswitch information. For example, the following command sequence displays the current vswitch (`vmvlan`), modifies the vswitch to specify connection to `lan1`, and creates a new vswitch named `clnvlan`. The final command displays information about the new vswitch.

```
# hpvmnet -S vmvlan
Name     Number State   Mode      NamePPA  MAC Address    IP Address
======== ====== ======= ========= ======== ============== ===============
vmvlan       13 Up      Shared    lan900   0x00306e4bc7bf

[Port Configuration Details]
Port    Port         Untagged Number of    Active VM
Number  state        VLANID   Reserved VMs
======= ============ ======== ============ ============
1       Reserved     none     1
2       Reserved     20       1
3       Reserved     none     1
# hpvmnet -C -S vmvlan -n 1 -N clnvlan
# hpvmnet -S clnvlan
Name     Number State   Mode      NamePPA  MAC Address    IP Address
======== ====== ======= ========= ======== ============== ===============
clnvlan     320 Down    Shared    lan1

[Port Configuration Details]
Port    Port         Untagged Number of    Active VM
Number  state        VLANID   Reserved VMs
======= ============ ======== ============ ============
2       Available    20       0
```

Note that only the configured VLAN port identification data is copied to the new vswitch. Use this `hpvmnet` command option when you have a vswitch with numerous VLAN ports. This process makes it unnecessary to reenter all the port data for each new vswitch.

10.2.4 Deleting vswitches

To delete a vswitch, first stop the vswitch using the `-h` option to the `hpvmnet` command. Then delete the vswitch using the `-d` option to the `hpvmnet` command. For example, the following command shows the error that prevents you from deleting an active vswitch (`clan1`):

```
# hpvmnet -S clan1 -d

hpvmnet: The vswitch is currently active
hpvmnet: Unable to continue
```

The following example uses the `hpvmnet` command to halt the vswitch and then to delete it. Both commands require you to confirm the action..

```
# hpvmnet -S clan1 -h

hpvmnet: Halt the vswitch 'clan1'? [n/y]: y

# hpvmnet -S clan1 -d
```

```
hpvmnet: Remove the vswitch 'clan1'? [n/y] y
```

The default command function (if you press **Enter**) is to not perform the function of the command. To perform the command function, enter **y**.

In the case of commands where a confirmation is required, such as the `hpvmnet -h` command, you can include the –Q option to override the confirmation process. This is useful in scripts and processes that are not interactive. For example, to stop a vswitch (`clan1`) without requiring confirmation from the user, enter the following commands:

```
# hpvmnet
Name     Number State   Mode      NamePPA  MAC Address    IP Address
======== ====== ======= ========= ======== ============== ===============
localnet      1 Up      Shared              N/A            N/A
clan1         2 Up      Shared    lan0      0x00306e39f70b
# hpvmnet -S clan1 -h -Q
# hpvmnet
Name     Number State   Mode      NamePPA  MAC Address    IP Address
======== ====== ======= ========= ======== ============== ===============
localnet      1 Up      Shared              N/A            N/A
clan1         2 Down    Shared    lan0
```

When an active vswitch is deleted, the VSP automatically determines that the vswitch is gone. When the vswitch is recreated, the guest network automatically becomes functional again.

10.2.5 Recreating vswitches

To change the vswitch to use another pNIC on the VSP (for example, to change from `lan0` to `lan1`), follow this procedure:

1. Delete the vswitch that was associated with `lan0`. For example:

   ```
   # hpvmnet -S myswitch -h -Q
   # hpvmnet -S myswitch -d
   ```

2. Create a new vswitch associated with `lan1`. For example:

   ```
   # hpvmnet -S myswitch -c -n 1
   ```

3. Add a new vNIC to your guest using the new vswitch. For example:

   ```
   # hpvmmodify -P guestname -a network:lan:,,:vswitch:myswitch
   ```

10.2.6 Starting vswitches

Virtual switches (vswitches) start automatically when the VSP system is started. You can start the vswitch manually using the –b option to the `hpvmnet` command. For example, the following command boots the vswitch named `clan1`:

```
# hpvmnet -S clan1 -b
```

You must restart a vswitch after the following events:

- The MAC address corresponding to the LAN number being used by the virtual switch is changed on the VSP (either by swapping the network adapter associated with the vswitch or associating the vswitch with a different network adapter).
- The way the network adapter accepts and passes on packets to the next network layer is changed. This can occur as a result of using the `ifconfig` or `lanadmin` command to set CKO on or off.
- If you use the `hpvmmodify` command to change the adapter type for a virtual NIC (vswitch port).

10.2.7 Halting vswitches

Use the `hpvmnet -h` command to halt a vswitch. For example:

```
# hpvmnet -S clan1 -h
hpvmnet: Halt the vswitch 'clan1'? [n]: y
```

Auto Port Aggregation (APA) can be configured on the VSP to provide a highly available LAN for the vswitch (APA in active/passive mode) or to increase the bandwidth of the vswitch LAN (APA active/active mode). Before you stop APA, halt the vswitches associated with it. If you do not bring down the vswitch first, the `hpvmnet` command reports an incorrect MAC address for the vswitch.

10.2.8 Restarting vswitches

It is necessary to restart the vswitch when:

- You replace the physical network card associated with the vswitch.
- You change a VSP IP address associated with the vswitch's network interface card.
- You change the network interface characteristics on the VSP; for example, by using the `nwmgr` command to change checksum offloading (CKO).

When you restart a vswitch, it is not necessary to restart the guests using the vswitch.

10.2.9 Guest AVIO interface behavior

The following list describes the guest AVIO interface behavior when guest boots while vswitch is down or resetting:

- If you boot a guest while the vswitch is not up, AVIO interfaces associated with the vswitch might not be claimed in the guest. For example, this might occur if the guest is booted prior to booting the vswitch or if the corresponding network interface on the VSP is not cabled. If you encounter this problem, first fix the vswitch state (that is, ensure that `hpvmnet` displays its state as Up), and then execute the `ioscan` command in the guest. These actions will claim the AVIO interfaces.
- If the vswitch is in an unstable state while the guest is booting, guest AVIO interfaces might fail initialization and move to the DOWN state (as displayed by the `lanscan` command). When this occurs, first ensure that the vswitch enters a stable state, then reset the guest interface using `nwmgr`.

10.3 Managing vNICs

After you create the vswitch, you can allocate it to one or more virtual machines for use by guest operating systems and applications. To create a vNIC for a virtual machine, enter one of the following commands:

- To create a new virtual machine with one vswitch:

  ```
  # hpvmcreate -P vm-name -a network:adapter-type:[hardware-address]:vswitch:vswitch-name
  ```

- To create a new virtual machine based on the configuration of an existing virtual machine:

  ```
  # hpvmclone -P vm-name -N clone-vm-name –a network:adapter-type:[hardware-address]:vswitch:vswitch-name
  ```

  The vNIC specified with this command is added to the new virtual machine.

- To modify an existing virtual machine:

  ```
  # hpvmmodify -P vm-name –a network:adapter-type:[hardware-address]:vswitch:vswitch-name
  ```

  The `–a` option adds the specified vNIC to the virtual machine.

NOTE: If you modify a vNIC from `lan` to `avio_lan`, or `avio_lan` to `lan`, you must restart the vswitch.

As with virtual storage devices, use the `-a` *`rsrc`* option to associate a guest virtual network device with a vswitch. Before you use this option to associate the virtual network device with a vswitch, create the vswitch using the `hpvmnet` command. The format of the *`rsrc`* parameter for network devices is:

network:*adapter-type*:[*hardware-address*]:vswitch:*vswitch-name*

The guest virtual network device information consists of the following fields, separated by colons:

- network
- *adapter-type*, which can be either lan or avio_lan
- [*hardware-address*] (optional), formatted as *bus*,*device*,*mac-addr*. If you do not specify the hardware address, or a portion of it, the information is generated for you. HP recommends allowing Integrity VM to generate the hardware address. The hardware address consists of the following information:
  - *bus* (virtual network device PCI bus number)
  - *device* (virtual network device PCI slot number)
  - *mac-addr* (the virtual network device MAC address) in either of the following formats: 0xaabbcc001122 or aa-bb-cc-00-11-22. The MAC address that you enter is checked to make sure it does not conflict with any of the VSP's physical network adapter MAC addresses.
- *vswitch*

  The virtual switch information is formatted as vswitch:*vswitch-name* (where *vswitch-name* is the name assigned to the virtual network switch when you create it using the hpvmnet command)

10.3.1 Adding vNICs

You can define a vNIC for a guest using the hpvmmodify command. For example, the following command adds a vNIC to the guest named host1.

```
# hpvmmodify -P host1 -a network:lan:0,0,0x00306E39F70B:vswitch:clan1
```

The guest configuration file /var/opt/hpvm/guests/*guestname*/vmm_config.current contains an entry for each guest virtual network device. When the guest is booted (through the hpvmstart or hpvmconsole command), the guest LAN is configured as specified in the LAN entry in the guest configuration file. For example:

```
.
.
.
# Virtual Network Devices
#
lan(0,0).0x00306E39F70B = switch(clan1).4
.
.
.
```

The localnet vswitch can be used as a local network ,and vNICs can be specified for a guest. For example:

```
# hpvmmodify -P host1 -a network:lan::vswitch:clan0
```

NOTE: Never modify the guest configuration files directly. Always use the Integrity VM commands to modify virtual devices and virtual machines. Failure to follow this procedure can result in unexpected problems when guests are started.

The virtual network entry in the guest configuration file includes the guest information on the left side of the equal sign (=), and VSP information on the right. The data about the guest LAN example includes the following information:

`lan(0,0)`	Bus 0 and device number 0 indicate the guest LAN hardware path.
`0xEEEE4077E7EB`	Guest virtual MAC address.
`switch(clan1)`	The vswitch name is `clan1`.
`4`	The VLAN port number is 4.

Entering the `lanscan` command on the guest `host1` results in the following:

```
# lanscan

Hardware Station         Crd Hdw   Net-Interface  NM  MAC       HP-DLPI DLPI
Path     Address         In# State NamePPA        ID  Type      Support Mjr#
0/0/3/0  0xEEEE4077E7EB 0   UP    lan0 snap0     1   ETHER     Yes     119
0/1/2/0  0x00306E3977AB 1   UP    lan1 snap1     2   ETHER     Yes     119
0/4/1/0  0x00306E4CE96E 2   UP    lan2 snap2     3   ETHER     Yes     119
```

NOTE: Do not include the hardware address (for example, bus, device, mac-addr) with the `hpvmmodify` command, because Integrity VM picks an available pcibus, pcislot and generates a random MAC address.

The hardware path from the output of `lanscan` on the guest matches the path in the guest configuration file. The `Station Address` in the `lanscan` output also matches the guest virtual MAC address in the guest configuration file.

10.3.2 Removing vNICs

To remove a vNIC from a virtual machine's configuration, first stop the guest using the `hpvmstop` command. Then use the `-d` option to the `hpvmmodify` command. The `-d` option allows you to specify the vswitch and the vNIC information. The following is the syntax of the `hpvmmodify -d` command:

```
hpvmmodify -P vm-name -d network:adapter-type:[hardware-address]:vswitch:vswitch-name
```

After making this change, start the guest using the `hpvmstart` command.

10.4 Configuring VLANs

A local area network (LAN) defines a broadcast domain in which bridges and switches connect all end nodes. Broadcasts are received by every node on the LAN, but not by nodes outside the LAN.

A virtual LAN (VLAN) defines logical connectivity instead of the physical connectivity defined by a LAN. A VLAN provides a way to partition a LAN logically such that the broadcast domain for a VLAN is limited to the nodes and switches that are members of the VLAN.

VLANs provide the following benefits:

- Enhanced security through traffic isolation within nodes that are VLAN members
- Bandwidth preservation, limiting the broadcast domain to a VLAN instead of the entire LAN
- Enhanced manageability for node migrations and network topology changes

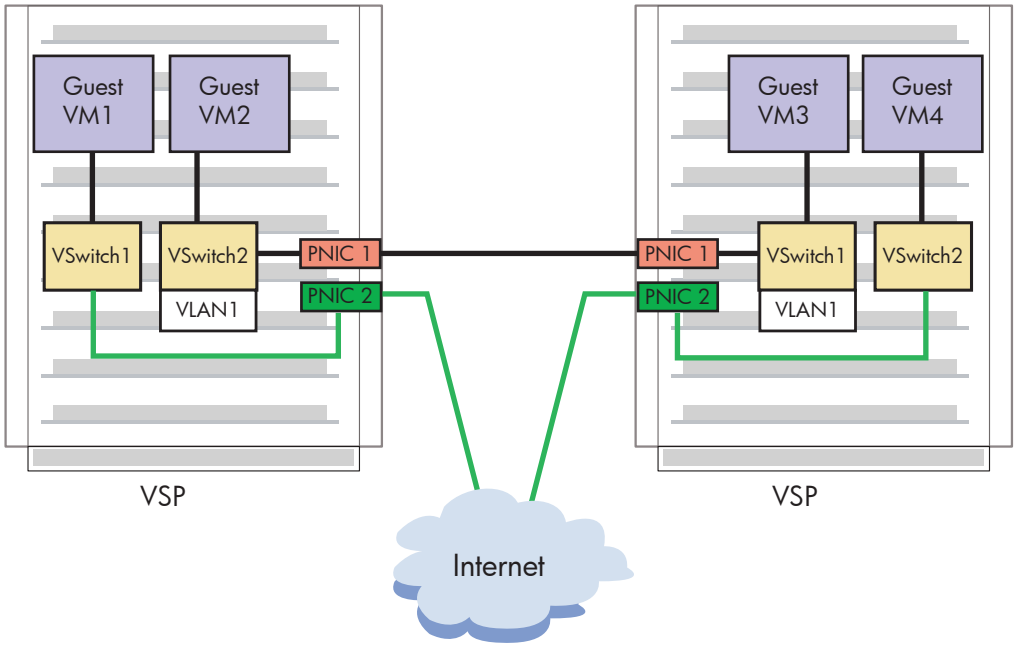
The following sections describe the Port-based VLANs, Guest-based VLANs, and VLAN-backed vswitch features.

NOTE: All three features are supported on the accelerated virtual I/O (AVIO) network. Only the Port-Based VLAN feature is supported in virtual I/O and AVIO networks.

10.4.1 Port-Based VLANs

Figure 11 illustrates a basic virtual machine VLAN that allows guests on different VSP systems to communicate.

Figure 11 Integrity VM VLAN Configuration Example



A vNIC on a guest is associated with a port on the vswitch and all network communication to and from the guest passes through this vswitch port. You can configure VLAN rules on the individual ports of the vswitch, similar to most physical switches. Each VLAN is identified by a VLAN identifier (VLAN ID). The VLAN ID is a number in the range 0-4094. A port on the vswitch can be assigned a VLAN ID that identifies the VLAN to which the port (and, therefore, the guest vNIC using that port) belongs.

Ports on a vswitch that are configured for the same VLAN ID can communicate with each other. Ports on a vswitch that are configured for different VLAN IDs are isolated from each other. Ports on a vswitch that do not have any VLAN ID assigned cannot communicate with ports that have a VLAN ID assigned, but they can communicate with other ports that have no VLAN ID assigned. VLAN port IDs for a vswitch can range from 0–511.

The emulation of the virtual network I/O card is based on the Intel I8254X family. Thus, the virtual network card (vNIC) is presented to the guest operating system as PCI-X 1000Base-T with the speed of 1 Gb regardless of the physical network interface card backing the vswitch. This emulation could lead to an incorrect calculation of vNIC performance by some network performance applications on the guest.

To accurately calculate vNIC performance, take into consideration the speed of the backing device on the Integrity VSP.

If the guest has to communicate with the VSP or outside the VSP over a VLAN, additional configuration is necessary. For communication to the VSP, configure a VLAN interface on the VSP interface for that vswitch. This VLAN interface should have the same VLAN ID as the guest port. For information about configuring VLANs on the VSP, see the *Using HP-UX VLANs* manual. Do not use the `hpvmnet` command to create a virtual switch that is associated with a VLAN port on the VSP (that is, a LAN created with `lanadmin -V`). This “nested VLAN” configuration is not supported.

Frames arriving at the vswitch from a guest can be “tagged” by the vswitch. Tagging consists of inserting the VLAN ID information into the MAC header before forwarding the frame on. Tagged frames destined for a guest are always stripped of the tag information in the frame before being forwarded. For Integrity VM, only tag-unaware guests are supported.

To configure a VLAN, follow this procedure:

1. Create and start the vswitch. For example, to create and boot vswitch `vmlan4` on `lan1`, enter the following command:

   ```
   # hpvmnet -c -S vmlan4 -n 1
   # hpvmnet -b -S vmlan4
   ```

2. Use the `hpvmnet` command with the `–u` option to create the port and assign it a VLAN ID. For example, to create ports 1 and 2 for VLAN 100, enter the following command:

   ```
   # hpvmnet -S vmlan4 -u portid:1:vlanid:100
   # hpvmnet -S vmlan4 -u portid:2:vlanid:100
   ```

3. Add the vswitch ports to the guest configuration using the `hpvmmodify` command. For example, to add the new VLAN ports to guests `vm1` and `vm2`, enter the following command:

   ```
   # hpvmmodify -P vm1 -a network:lan::vswitch:vmlan4:portid:1
   # hpvmmodify -P vm2 -a network:lan::vswitch:vmlan4:portid:2
   ```

The following command shows the resulting configuration:

```
# hpvmnet -S vmlan4
  Name     Number State   Mode      PPA    MAC Address    IP Address
  ======== ====== ======= ========= ====== ============== ===============
  vmlan4        2 Up      Shared    lan4   0x00127942fce3 192.1.2.205
  [Port Configuration Details]
  Port    Port         Untagged Number of    Active VM
  Number  state        VLANID   Reserved VMs
  ======= ============ ======== ============ ============
  1       Active       100      2            vm1
  2       Active       100      1            vm2
  3       Active       none     2            vm1
  4       Active       none     1            vm2
```

The two virtual machines, `vm1` and `vm2`, have access to the virtual switch `vmlan4` and are active on VLAN 100. Specifically, port 1 (guest `vm1`) and port 2 (guest `vm2`) can communicate with each other. Port 1 (guest `vm1`) and port 4 (guest `vm2`) cannot communicate with each other.

The `hpvmnet` command displays the following information about the VLAN ports:

- Port number.
- State of the port. Table 22 describes the possible VLAN port states:

  Table 22 VLAN Port States

State	Description
`Active`	The port is active and is allocated to a running guest. No other guests with the same vNIC with the same vswitch and port can start
`Down`	The port is inactive and is allocated to a running guest. No other guests with the same vNIC with the same vswitch and port can start.
`Reserved`	At least one guest reserved the port for its vNIC, but no guest that uses the port is running.
`Available`	No guest reserved the port for its vNIC. When a VLAN is configured on the port, that port is displayed as Available. If no VLAN is configured, the port is not displayed at all.

- The untagged VLAN ID number (if any)
- The number of virtual machines that have access to the VLAN
- The names of virtual machines that are up and that have access to the VLAN

10.4.1.1 Cloning guests with VLAN information

If you use the `hpvmclone` command to clone guests, the operation automatically assigns new port numbers for new guests. To assign the same port number to the new guest, use the `–S` option, as follows:

```
# hpvmclone -P vm1 -N vmclone1 -S
```

This command creates a new guest (`vmclone1`) based on the existing guest `vm1`, and preserves the vswitch port number so that the new guest will have access to the same VLANs as the existing guest.

10.4.1.2 Displaying VLAN information

You can display the vswitches and ports on a vswitch used by a guest using the `hpvmstatus` command. For example, to display the network information about the guest named `vm1`, enter the following command:

```
# hpvmstatus -P vm1

.
.
.
[Network Interface Details]
  Interface Adaptor    Name/Num   PortNum Bus Dev Ftn Mac Address
  ========= ========== ========== ======= === === === ==============
  vswitch   lan        localnet   1         0   1   0 de-19-57-23-74-bd
  vswitch   lan        localnet   2         0   2   0 7a-fb-4e-68-4f-5f
  vswitch   lan        vmlan4     1         0   4   0 -e8-c6-fa-b5-bc
  vswitch   lan        vmlan4     2         0   5   0 fa-18-82-9f-1a-95
  vswitch   lan        vmlan900   1         0   6   0 86-81-0b-6d-52-36
  vswitch   lan        vmlan900   2         0   7   0 6a-b9-cf-06-02-94
.
.
.
```

The preceding example shows the Network Interface Details portion of the `hpvmstatus` display. In the list of network interfaces, note that each virtual network connection is associated with either port 1 or port 2 of several vswitches. The vswitch named vmlan4 is associated with Bus/Dev/Ftn 0/4/0 on port 1, and with 0/5/0 on port 2.

To disable a VLAN, use the following command:

```
# hpvmnet -S vswitch-name -u portid:portnum:vlanid:none
```

To display information about a specific VLAN port, include the `-p` option to the `hpvmnet` command. For example, display VLAN information for port 2 on the vswitch named `vmlan4`, enter the following command:

```
# hpvmnet -S vmlan4 -p 2
  Vswitch Name            :  vmlan4
  Max Number of Ports     :  512
  Port Number             :  2
    Port State            :  Active
    Active VM             :  vm1
    Untagged VlanId       :  100
    Reserved VMs          :  vm1
    Adaptor               :  avio_lan
    Tagged VlanId         :  none
```

To view the all the VLANs defined on the vswitch named `vlan4`, enter the following command:

```
# hpvmnet -S vmlan4 -p all
  Vswitch Name            :  vmlan4
  Max Number of Ports     :  512
  Configured Ports        :  4
  Port Number             :  1
    Port State            :  Active
```

```
    Active VM             :  vm1
    Untagged VlanId       :  none
    Reserved VMs          :  vm1
    Adaptor               :  avio_lan
    Tagged VlandID        :  none
Port Number             :  2
    Port State            :  Active
    Active VM             :  vm1
    Untagged VlanId       :  100
    Reserved VMs          :  vm1
    Adaptor               :  avio_lan
    Tagged VlanID         :  none
Port Number             :  3
    Port State            :  Active
    Active VM             :  vm2
    Untagged VlanId       :  none
    Reserved VMs          :  vm2
    Adaptor               :  avio_lan
    Tagged VlanId         :  none
Port Number            :  4
    Port State            :  Active
    Active VM             :  vm2
    Untagged VlanId       :  100
    Reserved VMs          :  vm2
    Adaptor               :  avio_lan
    Tagged VlanID         :  none
```

10.4.2 Guest-Based VLANs (AVIO)

To use guest-based VLANs, you must first enable the tagged VLAN IDs of the GBVs on the vswitch port. To enable the tagged VLAN IDs, use the `hpvmnet -S <vsw> -i` command. To disable the VLAN IDs, use the `hpvmnet -o` command option.

On a vswitch port, you cannot use a VLAN ID as both an untagged VLAN ID and a tagged VLAN ID at the same time. That is, a VLAN ID used with the `hpvmnet -u` command option cannot be used with the `hpvmnet -i` option.

Guest-based VLANs are supported with HP–UX 11i v3 guests only.

10.4.3 Configuring VLANs on virtual switches

The VLAN-backed vswitch feature (VBVsw) enables a virtual switch (vswitch) to be backed by a physical network device with HP-UX VLAN (IEE 802.1Q) configured. The feature allows this type of vswitch to function just like a vswitch that is bound to a physical interface or an aggregate. Each VLAN backing the vswitch can be considered as a single network even though it is a discrete logical LAN being managed by the VSP.

On the VSP, multiple VLAN interfaces can be configured on a guest LAN backed by VBVsw type vswitch is created, the network traffic delivered to and from the guest is filtered using the VLAN ID. Guest LANs backed to the same vswitch that has VLAN configured share the same VLAN ID. Thus, these guest LANs can communicate with each other as if they were on the same physical network.

For information about VLANs on HP-UX, see the *HP-UX VLAN Administrator's Guide for HP-UX 11i v3* and *Planning and Implementing VLANs with HP-UX* manual.

10.4.3.1 Creating and managing a vswitch with a VLAN interface

To illustrate how to create and manage a vswitch with a VLAN interface, assume that your system has physical and aggregate interfaces as shown by the following format:

```
Name/          Interface Station          Sub-   Interface      Related
ClassInstance  State     Address        system   Type           Interface
============== ========= ============== ======== ============== =========
```

```
lan0           UP        0x0017A4AB5461 igelan   1000Base-T
lan1           UP        0x0017A4AB5460 igelan   1000Base-T
lan2           UP        0x001A4B06E90A iether   1000Base-T
lan3           UP        0x001A4B06E90B iether   1000Base-T     lan900
lan900         UP        0x001A4B06E90B hp_apa   hp_apa
lan901         DOWN      0x000000000000 hp_apa   hp_apa
lan902         DOWN      0x000000000000 hp_apa   hp_apa
lan903         DOWN      0x000000000000 hp_apa   hp_apa
lan904         DOWN      0x000000000000 hp_apa   hp_apa
```

To configure a PPA of the VLAN interface (VPPA) with a VLAN ID = 20 on the lan900 aggregate, enter the following:

```
# nwmgr -a -S vlan -A vlanid=20, ppa=900
VLAN interface lan5000 successfully configured.
lan5000 current values:
   VPPA = 5000
   Related PPA = 900
   VLAN ID = 20
   VLAN Name = UNNAMED
   Priority = 0
   Priority Override Level = CONF_PRI
   ToS = 0
   ToS Override Level = IP_HEADER

VLAN          Related    VLAN  Pri  Pri         ToS  Tos         Name
Interface     Interface  ID         Override        Override
Name                               Level           Level
============= =========  ===== ==== ========== ==== ========== ================
lan5000       lan900     20    0    CONF_PRI   0    IP_HEADER  UNNAMED
```

To create, boot and display a vswitch bound to VLAN lan5000, enter the following:

```
# hpvmnet -c -S vs5020 -n 5000
# hpvmnet -b -S vs5020
# hpvmnet -S vs5020
Name     Number State   Mode      NamePPA  MAC Address    IPv4 Address
======== ====== ======= ========= ======== ============== ===============
vs5020       18 Up      Shared    lan5000   0x001a4b06e90b

[Port Configuration Details]
Port    Port         Port     Untagged Number of    Active VM    Tagged
Number  State        Adaptor  VLANID   Reserved VMs               VLANIDs
======= ============ ======== ======== ============ ============ =============
1       Reserved     avio_lan none     2                         none
2       Reserved     avio_lan none     1                         none
3       Active       avio_lan none     1            u03          none
```

To enable the VLAN-backed vswitch (VBVsw) feature, HP-UX PHNE_40215 or a superseding patch is required on the VSP. This patch is available as an individual patch or as part of "FEATURE11i" bundle. To verify that the patch is installed, enter the following:

```
# swlist -l product |  grep PHNE_40215
PHNE_40215            1.0            LAN cumulative patch
```

The `dlpi_max_ub_promisc` kernel tunable needs to be set to when using a VBVsw type vswitch. Otherwise, attempting to boot the vswitch fails with the following error message from the `hpvmnet` command:

```
# hpvmnet -b -S vs5000
hpvmnetd: setup_downlink: promisc failed, recv_ack:
promisc_phys: UNIX error - Device busy, errno 5
```

To set the kernel tunable, enter the following:

```
# kctune dlpi_max_ub_promisc=
```

10.4.4 Configuring VLANs on physical switches

When communicating with a remote VSP or guest over the network, you might need to configure VLANs on the physical switches. The physical switch ports that are used must be configured specifically to allow the relevant VLANs. If the remote host is VLAN aware, You must configure

VLAN interfaces on the host for the relevant VLANs. Use the `lanadmin` command to configure VLANs on a remote HP-UX host. For example, to configure a VLAN interface with VLAN ID 100 on `lan4`, enter the following command:

```
# lanadmin -V create vlanid 100 4
Successfully configured
lan5000: vlanid 100 name UNNAMED pri 0 tos 0 tos_override IP_HEADER pri_override CONF_PRI ppa 4
```

10.5 Using direct I/O networking

The following commands provide direct I/O networking for vPars and VMs:

- The `hpvmhwmgmt` command — allows you to:
  - List direct I/O capable functions on the VSP:

    ```
    # hpvmhwmgmt -p dio -l
    ```

    NOTE: Assignment level is displayed as the output to this command:

    – function: Function Level Assignment (FLA)

    Each function may be added or deleted individually to or from the DIO pool.
    Each function may be added or deleted individually to vPars/VMs.
    Each function may be used individually by vPars/VMs and the VSP.

    – device: Device Level Assignment (DLA)

    The entire device is added or deleted to or from the DIO pool when one function of the device is specified.
    Each function may be added or deleted individually to vPars/VMs.
    Only one vPar/VM at a time can use functions that are part of the same device.

  - Add a function to the direct I/O pool:

    ```
    # hpvmhwmgmt -p dio -a hwpath [-l label]
    ```

    NOTE: You cannot add a function if it is in use by the VSP or restricted for VSP use. Labels are optional and are used for offline migration.

  - Delete a function from the direct I/O pool:

    ```
    # hpvmhwmgmt -p dio -d hwpath
    ```

  - Modify a label:

    ```
    # hpvmhwmgmt -p dio -m hwpath -l label
    ```

  - Delete a label:

    ```
    # hpvmhwmgmt -p dio -m hwpath -L none
    ```

    NOTE: The `hpvmdevmgmt –a/m/d` command blocks any attempt to add, modify, or delete the Label attribute.

- The `hpvmmodify` command — allows you to:
  - Add a direct I/O function to a vPar or VM:

    ```
    # hpvmmodify -P vm -a lan:dio:[b,d,macaddr]:hwpath:hwpath
    ```

    NOTE: The function must already be in the direct I/O pool.

  - Delete a direct I/O function from a vPar or VM:

    ```
    # hpvmmodify -P vm -d lan:dio:[b,d,macaddr]:hwpath:hwpath
    ```

  - Replace a direct I/O function in a vPar or VM:

```
# hpvmmodify -P vpar -m lan:dio:b,d,macAddr:hwpath:new-hwpath
```

- Modify the MAC address:

```
# hpvmmodify -P vpar -m lan:dio:b,d,new-macAddr:hwpath:hwpath
```

- The `hpvmstatus` command — allows you to:
  - See vPar and VM configurations. The direct I/O network functions are included in the #NETs count.

    ```
    # hpvmstatus
    ```

  - See specific vPar or VM I/O details:

    ```
    # hpvmstatus -P vm -d
    ```

    NOTE: There are no new switches specific to direct I/O for the `hpvmstatus` command.

- The `hpvmstart` command — allows you to:
  - Start a vPar or VM with direct I/O:

    ```
    # hpvmstart -P vm
    ```

    NOTE: Two vPars or two VMs cannot start if they are using the same direct I/O function. Also, two vPars or two VMs cannot start if they are using the same device level assignment (DLA) device.

- The `hpvmstop` command — allows you to:
  - Stop a vPar or VM that is using direct I/O:

    ```
    # hpvmstop -P vpar
    ```

    NOTE: There are no new switches specific to direct I/O for the `hpvmstart` or `hpvmstop` commands.

To map direct I/O devices between the VSP and the vPars or VMs:

- From the VSP:

  ```
  # hpvmdevinfo -P vm
  ```

- From the vPar or VM:

  ```
  # hpvmdevinfo
  ```

To restrict DIO-capable devices to the VSP, use the following command:

`hpvmdevmgmt -a rdev:`*`hwpath`*

NOTE: Note: if the *`hwpath`* is for a DLA function, all functions will be added.

The *`hwpath`* must be assigned to the VSP in order for it to be restricted for VSP use. If the *`hwpath`* is already in use by a vPar/VMt, the `-a` add will fail.

HP recommends that administrators manually restrict all functions that are used by the VSP for VSP networking, because that is not currently done automatically. HP also recommends that administrators manually restrict all functions that are assigned to vPars and VMs for use with AVIO, to avoid conflicts at vPar/VM boot time, because those functions will not appear to be in use until the vPars and VMs are booted.

To view the NICs that support DIO on a VSP, use the command: `vparhwmgmt -l -p dio` (or `hpvmhwmgmt -l -p dio`), which shows the DIO supported cards and the assignment level they support (device or function):

```
# vparhwmgmt -l -p dio
                                                    Assignment
```

```
H/W Path       Class   Owner Description                Level      Label
-------------- ------- ----- ------------------------- ---------- -------------
0/0/0/3/0/0/0  lan     host  HP  PCIe 2-p 10GbE Built- device
0/0/0/3/0/0/1  lan     host  HP  PCIe 2-p 10GbE Built- device
0/0/0/3/0/0/2  lan     host  HP  PCIe 2-p 10GbE Built- device
0/0/0/3/0/0/3  lan     host  HP  PCIe 2-p 10GbE Built- device
0/0/0/3/0/0/4  lan     host  HP  PCIe 2-p 10GbE Built- device
0/0/0/3/0/0/5  lan     host  HP  PCIe 2-p 10GbE Built- device
0/0/0/3/0/0/6  lan     host  HP  PCIe 2-p 10GbE Built- device
0/0/0/3/0/0/7  lan     host  HP  PCIe 2-p 10GbE Built- device
0/0/0/4/0/0/0  lan     host  HP  PCIe 2-p 10GbE Built- device
0/0/0/4/0/0/1  lan     host  HP  PCIe 2-p 10GbE Built- device
0/0/0/4/0/0/2  lan     host  HP  PCIe 2-p 10GbE Built- device
0/0/0/4/0/0/3  lan     host  HP  PCIe 2-p 10GbE Built- device
0/0/0/4/0/0/4  lan     host  HP  PCIe 2-p 10GbE Built- device
0/0/0/4/0/0/5  lan     host  HP  PCIe 2-p 10GbE Built- device
0/0/0/4/0/0/6  lan     host  HP  PCIe 2-p 10GbE Built- device
0/0/0/4/0/0/7  lan     host  HP  PCIe 2-p 10GbE Built- device
```

Use the `hpvmhwmgmt -p dio –a` *`path`* command to assign the card/function to the DIO pool. For DLA cards, you can use the path of any port on the card. All functions of the card are assigned to the DIO pool. Once the function/device is added to the DIO pool, `hpvmhwmgmt` command shows the owner as `hpvm` and not `host`.

NOTE: If you use the `–L label` option when adding a DLA card to the DIO pool only the function (path) that was specified in the command line will be labeled, other ports of the DLA card will need to be labeled individually.

```
# hpvmhwmgmt -p dio -a 0/0/0/4/0/0/1 -L DLA1
# hpvmhwmgmt -l -p dio
                                                          Assignment
H/W Path       Class   Owner Description                Level      Label
-------------- ------- ----- ------------------------- ---------- -------------
0/0/0/3/0/0/0  lan     host  HP  PCIe 2-p 10GbE Built- device
…
0/0/0/4/0/0/0  lan     hpvm  HP  PCIe 2-p 10GbE Built- device
0/0/0/4/0/0/1  lan     hpvm  HP  PCIe 2-p 10GbE Built- device     DLA1
0/0/0/4/0/0/2  lan     hpvm  HP  PCIe 2-p 10GbE Built- device
0/0/0/4/0/0/3  lan     hpvm  HP  PCIe 2-p 10GbE Built- device
0/0/0/4/0/0/4  lan     hpvm  HP  PCIe 2-p 10GbE Built- device
0/0/0/4/0/0/5  lan     hpvm  HP  PCIe 2-p 10GbE Built- device
0/0/0/4/0/0/6  lan     hpvm  HP  PCIe 2-p 10GbE Built- device
0/0/0/4/0/0/7  lan     hpvm  HP  PCIe 2-p 10GbE Built- device


# hpvmhwmgmt -p dio -m 0/0/0/4/0/0/7 -L DLA1.1
# hpvmhwmgmt -p dio -l | grep DLA1
0/0/0/4/0/0/1  lan     hpvm  HP  PCIe 2-p 10GbE Built- device     DLA1
0/0/0/4/0/0/7  lan     hpvm  HP  PCIe 2-p 10GbE Built- device     DLA1.1
```

When a DIO device has been added to the DIO pool, `ioscan` shows the device as being claimed by the `hpvmdio` device:

```
# ioscan -funC hpvmdio
Class     I  H/W Path       Driver  S/W State   H/W Type     Description
=============================================================================
hpvmdio   0  0/0/0/4/0/0/0  hpvmdio   CLAIMED     INTERFACE    HP  PCIe 2-p 10GbE Built-in  FLEX-10
                           /dev/hpvmdio0
….
```

You cannot add a function to the pool if it is in use by the VSP:

```
# hpvmnet
Name                    Number State   Mode      NamePPA MAC Address    IPv4 Address
====================== ====== ======= ========= ======= ============== ===============
localnet                     1 Up      Shared            N/A            N/A
hpnet                        2 Up      Shared    lan0    0x1cc1de40d040 15.43.212.199
priv_net                     3 Up      Shared    lan1    0x1cc1de40d044

# hpvmhwmgmt -l -p dio | grep 0/0/0/3/0/0/7
0/0/0/3/0/0/7  lan     host  HP  PCIe 2-p 10GbE Built- device
```

```
# hpvmhwmgmt -p dio -a 0/0/0/3/0/0/7
hpvmhwmgmt: Sibling path '0/0/0/3/0/0/0' (lan0) is being used as vswitch 'hpnet'.
hpvmhwmgmt: Sibling path '0/0/0/3/0/0/1' (lan1) is being used as vswitch 'priv_net'.
hpvmhwmgmt: Lan devices used as vswitches cannot be added to the DIO pool.
hpvmhwmgmt: Unable to manage dio pool resource.
```

Use the `vparstatus –A` command to view the functions available in the DIO pool:

```
# vparstatus -A | grep dio
        lan:dio::hwpath:0/0/0/4/0/0/0
        lan:dio::hwpath:0/0/0/4/0/0/1
        lan:dio::hwpath:0/0/0/4/0/0/2
        lan:dio::hwpath:0/0/0/4/0/0/3
        lan:dio::hwpath:0/0/0/4/0/0/4
        lan:dio::hwpath:0/0/0/4/0/0/5
        lan:dio::hwpath:0/0/0/4/0/0/6
        lan:dio::hwpath:0/0/0/4/0/0/7
```

Use the `hpvmmodify` command or `vparmodify` command to add the dio device to an existing guest:

```
# vparmodify -p atcvpar2 -a lan:dio::hwpath:0/0/0/4/0/0/0
```

If you attempt to add a function of a DLA device when another vPar/VM has been assigned a function on that same DLA device **and** has `resources_reserved` set to true, the add will fail:

```
# vparmodify -p newatcvpar1 -a lan:dio::hwpath:0/0/0/4/0/0/1
vPar/VM newatcvpar1 configuration problems:
    Error 1: The sibling DLA function: '0/0/0/4/0/0/0' of function: '0/0/0/4/0/0/1' is in use by another guest.
vparmodify: Unable to modify the vPar.
```

Setting the `resources_reserved` flag on the vPar/VM to false allows you to add the function to the vPar/VM:

```
# vparmodify -p atcvpar2 -x resources_reserved=false
# vparmodify -p newatcvpar1 -a lan:dio::hwpath:0/0/0/4/0/0/1

# vparstatus -v -p atcvpar2 | grep dio
        lan:dio:0,6,0x7e06f5393261:hwpath:0/0/0/4/0/0/0
# vparstatus -v -p newatcvpar1 | grep dio
        lan:dio:0,4,0xca7e0c0d0e96:hwpath:0/0/0/4/0/0/1
```

However, only one of these vPars will be able to boot at one time:

```
# vparboot -p atcvpar2
(C) Copyright 2000 - 2012 Hewlett-Packard Development Company, L.P.
UsrDirectAdd: hw_path="0/0/0/4/0/0/0"  MAC=0x7e06f5393261.
…

# vparstatus
[Virtual Partition]
Num Name                        RunState      State
=== ==========================  ============  =========
  2 atcvpar2                    EFI           Active
  1 newatcvpar1                 DOWN          Inactive

…


# vparboot -p newatcvpar1
vPar/VM newatcvpar1 configuration problems:
    Error 1: The sibling DLA function: '0/0/0/4/0/0/0' of function: '0/0/0/4/0/0/1' is in use by another guest.
vparboot: Unable to continue.
```

For the syntax and complete list of options for these commands, see the appropriate manpages.

Troubleshooting DIO

If you are unable to add a DIO function or device to the DIO pool that is not in use by the VSP or already in the DIO pool, check the CRA log file `/var/adm/cra.log`. When `hpvmhwmgmt -p dio –a …` is executed, a Critical Resources Analysis (CRA) Report is generated and might provide clues as to why the function/device cannot be added to the pool. For example, Serviceguard might own the interface:

```
DETAILED REPORT: Analyzed following hardware paths to detect any
usages in the system:
```

```
0/0/0/4/0/0/0 (lan2)
0/0/0/4/0/0/1 (lan3)


DATA CRITICAL RESULTS:

        Interface lan2: COMMAND cmnetd  PID 2907

        Interface lan2: COMMAND cmnetd  PID 2907
        Service-Guard(SG) Usage:
            The interfaces listed below are being used by SG:
                lan2
```

Use the `hvmdevinfo` command to display the hardware device mapping between vPar/VM and the VSP. You can run this command on the VSP or the vPar/VM:

VSP:

```
# hpvmdevinfo
Virtual Machine Name    Device Type     Bus,Device,Target       Backing Store Type      Host Device Name
  Virtual Machine Device Name
====================    ===========     ====================    ==================      ================
  ============================
atcvpar2                disk            [0,0,0]                 disk                    /dev/rdisk/disk13
  /dev/rdisk/disk3
atcvpar2                disk            [0,0,2]                 disk                    /dev/rdisk/disk21
  /dev/rdisk/disk5
atcvpar2                hba             [0,5]                   npiv                    /dev/fcd0
  /dev/gvsd2
atcvpar2                lan             [0,6,0x7E06F5393261]    hwpath                  0/0/0/4/0/0/0
  0/0/0/6/0 (lan3)
```

vPar/VM:

```
# hpvmdevinfo
Device Type     Bus,Device,Target       Backing Store Type      Host Device Name        Virtual Machine Device
 Name
===========     =================       ==================      ================        ============================
disk            [0,0,0]                 disk                    /dev/rdisk/disk13       /dev/rdisk/disk3
disk            [0,0,2]                 disk                    /dev/rdisk/disk21       /dev/rdisk/disk5
hba             [0,5]                   npiv                    /dev/fcd0               /dev/gvsd2
lan             [0,6]                   hwpath                  0/0/0/4/0/0/0           0/0/0/6/0 (lan3)
```

For current limitations, see the *HP-UX vPars and Integrity VM V6.1 Release Notes* on the BSC website at http://www.hp.com/gp/hpux-hpvm-docs

10.6 Troubleshooting network problems

This section describes some commonly encountered problems using virtual networks.

- **Do not kill hpvmnetd**

  Do not use the `kill` command to remove the `hpvmnetd` process. If you do, the following error message indicates that the `hpvmnet` daemon has been killed:

  ```
  hpvmnetd: Switch 0000564d4c414e31 already exists
  ```

  If the `hpvmnetd` process is removed, vswitches do not work properly.

- AVIO LAN devices not claimed by guest with DOWN vswitch at boot time.

  In addition to running `ioscan`, you must re-run the necessary network startup scripts, so that IP addresses can be reconfigured on the network interface cards (NICs). For example:

  ```
  /sbin/rc2.d/S340net start
  /sbin/rc2.d/S340net-ipv6 start
  ```

10.6.1 Redefining pNICs for HP-UX guests

Changing the hardware address of a vswitch has the same effect as moving a network adapter from one hardware slot to another on an HP Integrity system. Similar to other HP-UX systems, the guest file `/etc/rc.config.d/netconf` must be modified so that `INTERFACE_NAME[0]` reflects the new LAN PPA assigned by the HP-UX network driver on the first guest reboot after the modification. At this first reboot, the LAN interfaces configuration fails, as follows:

```
Configure LAN interfaces ..........................              . FAIL
*
```

When the guest is running, you can use the `lanscan` command to identify the new LAN PPA and to modify `netconf`. For example:

```
# lanscan
Hardware Station        Crd Hdw   Net-Interface  NM  MAC       HP-DLPI DLPI
Path     Address        In# State NamePPA        ID  Type      Support Mjr#
0/0/5/0 0x02636C6E3030  1   UP    lan3 snap3     1   ETHER     Yes     119
```

In the preceding example, before the modification, the LAN PPA was `0`. The new LAN PPA on the first boot after the modification is `3`. Therefore, you must first bring the guest network down, then you must change the `INTERFACE_NAME[0]` from `lan0` to `lan3`. You can then use `/sbin/rc2.d/S340net` to restart the guest network. For example:

```
# /sbin/rc2.d/S340net stop
# ch_rc -a -p "INTERFACE_NAME[0] = "lan3"
# /sbin/rc2.d/S340net start
```

The guest network begins to function.

10.6.2 Troubleshooting VLAN problems

When VLANs are configured on the vswitch, the partitioned LAN must have its own set of network servers to service requests on the VLAN. For example, the VLAN's DNS server or a router setup on the VLAN should be set up on the VLAN. If guests start slowly or hang during starting, determine whether the guest network interface is on a VLAN, and whether the appropriate network services (like DNS) are set up and available on the VLAN. You might need to either set up the appropriate services on the VLAN, or disable some of these network services on the guest before booting up the guest on a VLAN.

When VLANs are configured on the vswitch and the guests are required to communicate over a VLAN with a remote node outside the VSP, you might need to set up the physical network appropriately for the VLAN. For information about configuring VLANs on the switches, see the product documentation for the physical network adapters.

If TCP/UDP applications have trouble communicating between a guest and the local VSP over a VLAN, it is possible that the host interface for the vswitch is checksum-offload capable. To resolve the problem, identify the interface used by the vswitch and run the following command on the VSP to disable the CKO feature, where 4 is the VSP interface as shown in the `hpvmnet` command output.

```
# lanadmin -X send_cko_off 4
Hardware TCP/UDP (IPv4) transmit checksum offload is currently disabled
```

Checksum offloading (CKO) is not supported. On most of the physical interfaces that are not of 10 Gigabyte type, CKO is turned off by default. Consult your interface card documentation for details.

Turning on CKO can cause host-to-guest connections as well as guest-to-host communication over a VLAN to fail. If you are receiving failures with host-to-guest connections or guest-to-host communication using a VLAN, ensure that the CKO is turned off in the host interface driver. If that does not fix the problem, reboot the vswitch.

To turn off the CKO on the VSP, identify the PPA of the network interface for the vswitch using the `hpvmnet` command. For example:

```
# hpvmnet

Name Number State Mode PPA MAC Address IP Address
======== ====== ======= ========= ====== ============== ===============
localnet 21 Up Shared N/A N/A
vmlan0 22 Up Shared lan0 0x00306ea72c0d 15.13.114.205
```

```
vmlan4 23 Up Shared lan4 0x00127942fce3 192.1.2.205
vmlan900 24 Up Shared lan900 0x00306e39815a 192.1.4.205
```

NOTE: The `lanadmin` command is deprecated and will be removed in a future HP-UX release. HP recommends that you use the `nwmgr` command to perform all network interface-related tasks.

The following table shows the `nwmgr` command that performs the same task as the `lanadmin` command:

Table 23 The `nwmgr` command

Task	Legacy Command	`nwmgr` Command

Check the status of the transmit CKO.	`# lanadmin -x cko 4`
	`Hardware TCP/UDP (IPv4) transmit checksum offload is currently enabled.`
	`Hardware TCP/UDP (IPv4) receive checksum offload is currently disabled.`
		`# nwmgr -g -A all -c lan4 | grep Checksum`
		`Transmit Checksum Offload=Off`
		`Receive Checksum Offload=On`

In this example, the VLANs are configured over the vswitch vmlan4. This vswitch is created on PPA 4 on the VSP.
Turn off CKO on PPA 4 by entering this command on the VSP.
	`# lanadmin -X send_cko_off 4`
	`Hardware TCP/UDP (IPv4) transmit checksum offload is currently disabled.`
		`# nwmgr -s -A tx_cko=off -c 4`
		`lan2 current values:`
		`Transmit Checksum Offload=Off`

10.6.3 Troubleshooting VLAN-Backed vswitches

To enable the VLAN-backed vswitch (VBVsw) feature, PHNE_40215 or a superseding patch is required to be installed on the VSP. This patch is available as an individual patch or as part of "FEATURE11i" bundle. To verify that the patch is installed, enter the following:

```
# swlist -l product |  grep PHNE_40215
PHNE_40215            1.0            LAN cumulative patch
```

The `dlpi_max_ub_promisc` kernel tunable needs to be set to when using a VBVsw type vswitch. Otherwise, attempting to boot the vswitch fails with the following error message from the `hpvmnet` command:

```
# hpvmnet -b -S vs5000
hpvmnetd: setup_downlink: promisc failed, recv_ack:
promisc_phys: UNIX error - Device busy, errno 5
```

To set the kernel tunable, enter the following:

```
# kctune dlpi_max_ub_promisc=
```

10.7 Other issues and notes

The following list provides networking information with the Integrity VM V6.1 release:

- If you modify the MAC address of an interface in the guest, the `hpvmstatus` command in the VSP does not display the current MAC address correctly. There is no fix or workaround for this problem at this time.
- Just as with physical devices on a network, for communication to occur uninterrupted between all stations on a LAN segment, the MTUs of all the systems on the LAN segment or VLAN must match, whether they are physical systems or guests. The VSP does not check for MTU mismatches for its guests.
- The `lanadmin` card specific options that are supported on igssn on the guest are:
  - `-x:speed,fctrl,cko,type,card_info,stats drv,vmtu,and drv_pr.`
  - `-X:drv_pr_on,drv_pr_off,stats clear`

11 Managing vPars/VMs

To manage a vPar/VM, connect to the vPar/VM using a remote connection and use the operating system administration procedures appropriate to the guest OS. vPars and Integrity VM provides utilities for managing vPars and virtual machines from the VSP and from inside the vPar/VM. This chapter describes how to manage vPars/VMs using Integrity VM commands and utilities, including the use of Logical Server Management (LSM) to manage VMs. The following topics are included in this chapter:

- Managing VMs with LSM
- Integrity VM Virtualization Provider
- Monitoring vPars/VMs
- Creating vPar/VM administrators and operators
- Installing VirtualBase
- Using the virtual console
- Using the virtual iLO Remote Console
- vPar/VM configuration files
- Dynamic memory for VMs
- Log files
- Managing the device database

11.1 Managing VMs with LSM

Integrity VM V6.1 fully supports HP Matrix OE Logical Server Management (LSM) Version 7.0. With this version, administrators can manage the full life-cycle of VMs with the following LSM operations: create, modify, delete, activate, deactivate, power on/off, import, move, and unmanage.

NOTE: The following steps must be run on the VSP before you can use LSM to create virtual machines and before you can use HP Insight Orchestration to provision VMs.

To create a VM using LSM, you must adhere to the following:

1. Create the appropriate size SLVM volume group (VG) for the device management database using LVM Version 7.0. For example:
   - Create the volume group using LVM Version 7.0:

     ```
     # vgcreate -V 7.0 -s 4m -S 100g /dev/slvm_v21 /dev/disk/disk61
     ```

   For information about creating SLVM volume groups, see the *SLVM Online Volume Reconfiguration* whitepaper at SLVM Online Volume Reconfiguration.
2. Add SLVM volume groups into the device database using the `hpvmdevmgmt` command. For each SLVM volume group you add to the device management database, set the device attribute *`VIRTPTYPE`* to *`container_volume_SLVM`*, with the `PRESERVE=YES` attribute setting. For example:

   ```
   # hpvmdevmgmt -a gdev:/dev/slvm_v22:attr:VIRTPTYPE=container_volume_SLVM,PRESERVE=YES
   ```

   For information about storage requirements for importing logical servers with backing storage from SLVM volume groups, see Section 11.1.1 (page 166)
3. Run `hpvmhostrdev –u` to add the underlying disks of the created SLVM volume groups into the device database as restricted devices.

NOTE: The SLVM volume groups must be in the activated mode before running the `hpvmhostrdev` script. For information about deactivated volume groups, see Section 11.1.2 (page 166).

4. Execute the `hpvmhostgdev -a` command to ensure that all devices are populated in the `gdev` database. The hpvmhostgdev command analyzes `disklist` and `lvlist` output and adds unused `gdevs` to the device database.

   NOTE: If you add new devices in the future, run the `hpvmhostgdev -a` script again. If you want to select the guest devices instead of adding all of them to the gdev database, create a list of unused disks and logical volumes with the `-l` option and pipe them to a file. Use the specified device-list file to add devices for guest use with the `-f` option.

   ```
   # hpvmhostgdev -l > devicelist
   # hpvmhostgdev -f devicelist
   ```

   For information about the `hpvmhostgdev` script, see the `hpvmhostgdev (1M)` manpage.

5. Managing VMs does not require them to be in a VM as a Serviceguard Package. However, if you plan to use clustered VMs, ensure that the VSP is properly configured with Serviceguard (11.19 or 11.20) and Shared Logical Volume Manager (SLVM).

   NOTE: For information about configuring Serviceguard and SLVM, see the *Using Serviceguard* manual on the BSC website.

   If you already have your VMs clustered in a VM as a Serviceguard Package, but prefer not to manage them this way, you can use the `cmdeployvpkg` Serviceguard command to properly deconfigure (delete) the package. For information about the `cmdeployvpkg` command, see the *HP Serviceguard Toolkit for Integrity Virtual Servers User Guide* at HP Serviceguard Toolkit for Integrity Virtual Servers.

11.1.1 Storage requirements for managing existing VMs with Logical Server Management

To use Logical Server Management (LSM) to manage vPars or virtual machines created outside of LSM, the VM backing storage needs to be the following:

- Whole LUNs — The supported LSM operations are: Import, Online Move, Power On, Power Off, and Unmanage.
- SLVM-based logical volumes (LVs) — The volume group (VG) type must be *`container_volume_SLVM`* in the vPar/VM device management database.

  The supported operations are: Import, Online Move, Power On, Power Off, Activate, and Deactivate, and Unmanage.

NOTE: For information about vpars and virtual machines created with LSM or HP Insight Orchestration using SLVM-based LVs, see Section 11.1 (page 165).

11.1.2 Storage for deactivated volume groups not protected by VM storage management

When an LVM volume group is deactivated, the storage (physical volumes) used by that storage is designated as unused by HP-UX system administration tools such as System Management Homepage (SMH). This is also true for Integrity VM storage management. As a result, these physical volumes are not automatically protected from use by virtual machines as virtual disks.

You can resolve this problem in one of two ways:

- If the volume group is to remain deactivated, the VSP administrator can manually add the physical volume as a restricted device with the `hpvmdevmgmt` command.
- Or, after activating the volume group, execute the `hpvmhostrdev` command, so that the VSP storage management database is updated accordingly.

An HP-UX system administrator can deactivate a volume group using the `vgchange` command. It can also be deactivated, if it is a shared LVM (SLVM) volume group, whenever the associated Serviceguard cluster is reconfigured, or the VSP system is rebooted. Take care to check that all SLVM volume groups are activated after a VSP reboot or Serviceguard cluster reconfiguration.

11.1.3 Managing existing VM with LSM

You can import existing VMs that are configured with whole LUNs, and perform the following LSM operations on these VMs: Online Move, Power On, Power Off, and Unmanage. All other operations are not supported with these imported VMs.

Integrity VSPs that are managing only VMs with whole LUNs do not need to be configured with Serviceguard and SLVM. If you plan to create new VMs on that VSP, follow the steps in Section 11.1 (page 165).

11.1.4 Managing VMs using gWLM

VMs configured with processing power specified in cycles instead of percentage are incompatible with gWLM A.02.50 and earlier versions.

If gWLM/Matrix OE produces an error message similar to the following, a VM is configured with the processing power specified in cycles:

```
A VM encountered with no size
```

This is apparent when using gWLM A.02.50 with Integrity VM A.03.00. You can correct the problem by modifying the guest and specifying processing power in percentage rather than CPU cycles. For example, to modify the guest named `compass1` to use 10% of the CPU processing power, enter the following command

```
# hpvmmodify -P compass1 -e 10
```

You must boot the guest to initiate this setting for gWLM.

Alternatively, upgrade gWLM to A.03.00 for use with Integrity VM A.03.00.

11.2 VM Virtualization Provider

The Integrity VM Virtualization Provider, used with the logical server feature in the Matrix OE, enables virtual to virtual migration with logical server management (LSM). A logical server is a set of configuration information that you create, activate, and move across physical and virtual machines. It contains the logical server definition and description, including the server computer resources (for example, the number of CPU cores and amount of memory), and the server connections to storage fabric and networks.

For information about LSM and VMM, see documentation on the Business Support Center website.

11.2.1 Adding and removing devices

vPars and Integrity VM adds devices not in use by the VSP automatically. You can add devices that are not automatically added by using the `hpvmdevmgmt gdev PRESERVE` attribute. The following device types require manual addition:

- File backed disks
- File backed DVDs
- VxVM volumes

The following examples show how to add various device types to the storage pool:

- File:

  ```
  # hpvmdevmgmt -a gdev:/var/opt/hpmv/ISO-images/hpux/112350GOLD.ISO:attr:PRESERVE=YES
  ```

- VxVM volume:

  ```
  # hpvmdevmgmt -a gdev:/dev/vx/rdisk/guestdg/vxvm_g2:attr:PRESERVE=YES
  ```

To remove a device from the storage pool, used the following command:

```
# hpvmdevmgmt -d gdev:/dev/rdisk/disk23
```

NOTE: Adding devices to the storage pool does not prevent them from being used by the HP-UX operating system or other Integrity VM commands.

The storage pool does not fully support lunpaths or directories. In addition, Virtual Machine Management (VMM), a layer between Integrity VM and LSM, has no way to insert or eject a DVD, because this is done from the virtual console.

11.2.2 Registering and unregistering a VM

A VM is registered when it is runnable, modifiable, and visible. When a VM is not registered, it is not visible to the graphical tools, such as LSM, and you cannot modify it or start it. When you register a virtual machine with VMM using the `hpvmmodify` command, the following attributes are set:

- `runnable_status=enabled`
- `modify_status=enabled`
- `visible_status=enabled`

VMM and LSM ensure that a virtual machine is registered (and, therefore, runnable) on only one VSP at a time.

When a virtual machine is unregistered, the following attributes are set:

- `runnable_status=disabled`
- `modify_status=disabled`
- `visible_status=disabled`

After a migration, the `hpvmmigrate` command sets the virtual machine on the source host as unregistered. The VM is marked not runnable, not visible, and not modifiable. The `hpvmstatus` command lists these attributes:

```
# hpvmstatus -P vmname -V
```

When the graphical tool queries the `register_status`, the value of `visible_status` is returned. If the VM is not visible, you cannot visualize it with the graphical tools, and therefore; you cannot modify it or run it.

You can set the `register_status` of a VM to enabled or disabled with the `hpvmmodify -x register_status` command.

CAUTION: HP does not recommend using the `-x register_status` option. Integrity VM commands ensure that the VM is registered only on one VSP at a time. Registering a VM on more than one VSP can lead to accidentally booting the VM on more than one VSP and could cause inconsistencies with the display of graphical tools. However, if you find that VM is not registered on any VSP, you can manually register it with the `hpvmmodify` command. For information on this command, see Table 16 (page 91).

11.2.3 Changes to the hpvmmodify command

The hpvmmodify -x command has been changed to allow changing the modify_status, and visible_status, and register_status attributes with the -x option, in addition to runnable_status.

```
# hpvmmodify -P vmname -x runnable_status={enabled|disabled}
# hpvmmodify -P vmname -x modify_status={enabled|disabled}
# hpvmmodify -P vmname -x visible_status={enabled|disabled}
# hpvmmodify -P vmname -x register_status={enabled|disabled}
```

- The runnable_status option, which already exists in Integrity VM, prevents a VM from being started.

CAUTION: HP does not recommend using the -x runnable_status option. Integrity VM ensures that the VM is runnable only on one VSP at a time. Marking a VM runnable on more than one VSP can lead to accidentally booting the VM on more than one VSP.

- The modify_status option of a VM is listed in the hpvmstatus -V output. If modify_status=disabled, you cannot modify a VM except to set modify_status=enabled.

CAUTION: HP does not recommend using the -x modify_status option, except with extreme caution. If modify_status is disabled, the VM is most likely running on another VSP. Any modification made to this VM's configuration will be lost when it is migrated back to this VSP.

- You can enable or disable visible_status with the hpvmmodify command. When a VM has the visible_status option set to disabled, the graphical tools will not display the VM.

CAUTION: HP does not recommend using the -x visible_status option, except with extreme caution. Use of this option may cause inconsistencies with the display of graphical tools and has no effect on the command-line output.

- If a VM is not registered on any VSP, you can manually register it with the hpvmmodify -x register_status=enabled command.

CAUTION: HP does not recommend using the -x register_status option. Integrity VM commands ensure that the VM is registered only on one VSP at a time. Registering a VM on more than one VSP can lead to accidentally booting the VM on more than one VSP, which could cause inconsistencies with the display of graphical tools.

The hpvmmodify command does not allow modification to guests marked modify_status=disabled. When the modify_status=disabled attribute is set, the only change allowed is to set the modify_status=enabled attribute. When the hpvmmigrate command sets the guest to the NR state (runnable_status=disabled), it now also sets the modify_status=disabled and visible_status=disabled attributes. Likewise, when the hpvmmigrate command sets the guest to be runnable, it now also sets the modify_status=enabled and visible_status=enabled attributes.

11.2.4 Cannot distinguish between JBOD and Remote SAN with Device Check

If your Integrity VM server has local JBOD disks configured, they appear as disks that are SAN-resident in the Virtualization Provider making them available for guests. If your guest configurations require only SAN-resident disks, the JBOD disks, set them as restricted disks in the Integrity VM device database.

The following example sets the device /dev/rdisk/disk100 as a restricted device:

```
# hpvmdevmgmt -a rdev:/dev/rdisk/disk100
```

11.2.5 Unpresenting SAN devices to Integrity VSPs

Unpresenting SAN devices that were configured to be used by guests causes the guest to fail to start. If SAN devices must be unpresented, guests configured to use those devices should be reconfigured to no longer require them. After unpresenting a device special file, remove it from the Integrity VSP using the following command:

```
# rmsf  -a  device_special_file
```

The device special file can be derived from the wwid_string, obtained from the SAN appliance, as follows:

```
# scsimgr -p get_attr -a wwid -a device_file current all_lun | grep wwid_string
```

11.2.6 Changes to the hpvmstatus command

The Runsysid and Rmt Host columns of the hpvmstatus command output have been renamed to allow for the display of additional information. The hpvmstatus command now displays the virtual machine type, as follows:

The following example shows the output of the hpvmstatus command:

```
# hpvmstatus
Virtual Machine Name VM #  Type OS Type State     #VCPUs #Devs #Nets Memory
==================== ===== ==== ======= ========= ====== ===== ===== =======
vPar0002                 2 VP   HPUX    Off            3     0     0 2048 MB
guest1                   3 SH   UNKNOWN Off            4     0     0   10 GB
ux1                      1 SH   HPUX    Off            4     2     1    3 GB
```

The hpvmstatus -V option has been modified to display the new attributes, after the "Runnable status" and associated attributes.

```
Graceful stop timeout   : 30
Runnable status         : Disabled
Not runnable setby      : Migrate
Not runnable reason     : Guest has been migrated to host colonial6.
Modify status           : Disabled
Not modify setby        : Migrate
Not modify reason       : Guest has been migrated to host colonial6.
Visible status          : Disabled
Not visible setby       : Migrate
Not visible reason     : Guest has been migrated to host colonial6.When these attributes are enabled the string
 "Enabled" will be displayed.
```

If you need to parse the output of the hpvmstatus command, use the -M option, which provides output in a machine-readable format. The hpvmstatus manpage explains the -M option:

-M displays verbose attribute and resource information in machine-readable format including information about migrating virtual machines.

11.3 Monitoring guests

To display information about all the vPars/VM configured on the VSP, enter the hpvmstatus command.

```
# hpvmstatus
[Virtual Machines]
Virtual Machine Name VM #  OS Type State    #VCPUs #Devs #Nets Memory  Runsysid
==================== ===== ======= ======== ====== ===== ===== ======= ========
config1                  1 HPUX    Off           1     5     1  512 MB         0
config2                  2 HPUX    On (OS)       1     7     1    1 GB         0
guest1                   5 HPUX    Off           1     5     1    1 GB         0
```

The vPar/VM status is displayed in the State column and indicates whether the virtual machine is powered off or on. When the vPar/VM is on, the status also includes one of the following:

- `EFI` indicates the vPar/VM is running normally in EFI.
- `OS` indicates the vPar/VM is running normally in the operating system.
- `ATTN!` indicates the guest is not responding to interrupts.

Table 24 describes the options to the `hpvmstatus` command.

Table 24 Options to the `hpvmstatus` Command

Option	Description
`-v`	Displays the version of the Integrity VM product that is running on the VSP.
`-V`	Displays detailed information about the specified virtual machine or about all the virtual machines if you do not specify one using either the `-p` or `-P` option.
`-M`	Specifies the display output should be in machine-readable format.
`-X`	Specifies the display output should be in XML format.
`-P` *vm-name*	Specifies the name of the virtual machine for which to display information.
`-p` *vm-number*	Specifies the number of the virtual machine for which to display information.
`-D`	Displays the resource allocation of the specified virtual machine. You must include either the `-p` option or the `-P` option.
`-e`	Displays the event log for the VSP or the specified virtual machine. The event log records all changes to virtual machine configurations.
`-r`	Displays the memory and virtual CPU resource allocation for the virtual machines (or for the specified virtual machine if you use the `-p` option or the `-P` option). This option displays the entitlement and virtual CPUs parameters configured for the virtual machine and the current usage of those resources.
`-d`	Displays the devices allocated to the virtual machine you specify using either the `-p` option or the `-P` option.
`-S`	Displays the scheduler mode for the VSP. `CAPPED` indicates that gWLM is managing the node. NORMAL indicates that the node is not being managed by gWLM.
`-s`	Displays the current VSP resources.
`-m`	If Serviceguard is installed, displays information about the multiple-server environment.
`-R`	Displays the resource reservations settings of the virtual machines.
`-L`	Displays the changes from the current configuration.
`-i`	When used with the `-P` option, prints statistics collected by the monitor.
`-C`	Displays whether the guests prefer cell local memory (clm), interleaved memory (ilm), or none.
`-A`	Displays the guest configuration differences between the next start and the last start guest configurations.

For example, to see detailed information about the `host1` virtual machine, enter the following command:

```
# hpvmstatus -V -P host1
[Virtual Machine Details]
Virtual Machine Name     : host1
Virtual Machine UUID     : e4f786d4-14ad-11e1-b600-0017a4776014
Virtual Machine ID       : 1
Virtual Machine Label    :
VM's Model Name          : server Integrity Virtual Machine
VM's Serial Number       : VM01147000
VM's Config Version      : 6.10.0
VM's Config Label        : HPVM B.06.10 LR ccipf debug Wed Dec 07 2011 12h07m02s PST
Virtual Machine Type     : Shared
Has reserved resources   : No
Configuration is active  : Yes
Operating System         : HPUX
```

```
OS Version Number       :
State                   : Off
Start type              : Manual
Console type            : vt100-plus
Guest's hostname        :
Guest's vNIC IP Preference      :
Guest's IPv4 address    :
EFI location            : /opt/hpvm/guest-images/common/efi
Pattern File location   : /opt/hpvm/guest-images/common/patterns.vmmpat
vPar/VM revision        : 1
Running on serverid     : 0
Running on pid          : 0
Application controllers : NONE
Distributed             : 0
Effective serverid      : 0
Graceful stop timeout   : 30
Runnable status         : Runnable
Modify status           : Modify
Visible status          : Visible

[Online Migration Details]
Online migration status : Enabled
Init phase timeout      : 90 seconds
Copy phase timeout      : Infinite
I/O quiesce phase timeout: 15 seconds
Frozen phase timeout    : 60 seconds

[Suspend/Resume Details]
Suspend status          : Enabled

[Remote Console]
Remote Console not configured

[Authorized Administrators]
Oper Groups             :
Admin Groups            :
Oper Users              :
Admin Users             :


[Virtual CPU Details]
Number Virtual CPUs     : 4
Minimum Virtual CPUs    : 1
Maximum Virtual CPUs    : 16
Percent Entitlement     : 100.0%
Maximum Entitlement     : 100.0%


[Memory Details]
Total memory            : 3 GB
Minimum memory limit    : 512 MB
Maximum memory limit    : 128 GB
Reserved memory         : 64 MB
Minimum reserved limit  : 32 MB
Maximum reserved limit  : 128 GB
VHPT Size               : 1 MB

[Dynamic Memory Information]
    NOTE: Dynamic data unavailable, configured values only
Type                    : driver
Minimum memory          : 512 MB
Target memory           : 3072 MB
Memory entitlement      : Not specified
Maximum memory          : 3072 MB

[Storage Interface Details]
Device type             : disk
Adapter type            : avio_stor
Ioscan format           : 0/0/0/0.0.0
Bus                     : 0
Device                  : 0
Function                : 0
```

```
Target                  : 0
Lun                     : 0
Physical Storage type   : disk
Physical Device         : /dev/rdisk/disk5

Device type             : disk
Adapter type            : avio_stor
Ioscan format           : 0/0/0/0.1.0
Bus                     : 0
Device                  : 0
Function                : 0
Target                  : 1
Lun                     : 0
Physical Storage type   : disk
Physical Device         : /dev/rdisk/disk4


[Network Interface Details]
Interface               : vswitch
Adapter type            : avio_lan
Backing                 : sitelan
Vswitch Port            : 1
Bus                     : 0
Device                  : 1
Function                : 0
Mac Address             : ea-2a-7b-81-8e-19

[Direct I/O Interface Details]

[Misc Interface Details]
Device type             : serial
Adapter type            : com1
Physical Storage type   : tty
Physical Device         : console
```

To display the VSP system resource, use the `-s` option to the `hpvmstatus` command. For example:

```
# hpvmstatus -s
                [HPVM Server System Resources]

         vPar/VM types supported by this VSP = Shared
        Processor speed = 1596 Mhz
        Total physical memory = 16278 Mbytes
        Total number of operable system cores = 8
        CPU cores allocated for VSP = 1
```

Specific display output from some Integrity VM tools, such as the `hpvmstatus` command, is subject to occasional changes of form and content. Program scripts should always use machine-readable output options (for example, `hpvmstatus -M`) whenever available to avoid future script maintenance.

11.4 Monitoring Integrity VM performance

Guest and VSP performance information is displayed by the VSP command `hpvmsar`. One of the displays in `hpvmsar` can be shown in a GUI-type format with four different styles. For information about these styles, see `hpvmsar` manpage. Note that some `hpvmsar` commands can be run only on HP-UX guests.

Table 25 Options to the `hpvmsar` Command

Option	Display Description
`-a`	Default Guest & Host Cpu usage Display in text or GUI modes for all running guests
`-A`	Default Guest & Host Cpu usage Display in text or GUI modes for all guests whether they are running or stopped
`-D`	Host to Guest Storage Utilization Display

Table 25 Options to the `hpvmsar` Command *(continued)*

Option	Display Description
`-F`	Integrity VM core Memory Metrics Display
`-G`	Guest Dynamic Memory, Swap, Paging Display
`-H`	Host Memory, Swap, Paging Display
`-I`	Guest Interrupt Display
`-N`	Guest AVIO Network traffic by vswitch Display
`-S`	Vswitch AVIO Network traffic by Port Display

11.5 Creating guest administrators and operators

Integrity VM provides secure access to guest machine consoles. When you create the virtual machine, you can specify groups and user accounts to have administration or operator privileges on that guest. These users are allowed to log in to the VSP under their own user accounts and to use the `hpvmconsole` command to perform system administration tasks on the guest virtual machine.

A captive virtual console account is a special-purpose user account created on the VSP for each guest administrator or operator. These types of user accounts use the `/opt/hpvm/bin/hpvmconsole` directory for a shell, and the desired guest's per-guest directory for a home directory. For virtual console access, the account also requires a password, and access to its associated guest.

Before you create the virtual machine, use the `useradd` command to create user accounts for virtual console access. For example, the following command adds the user account `testme1`:

```
# useradd -r no -g users -s /opt/hpvm/bin/hpvmconsole \
 -c "Console access to guest 'testme'" \
 -d /var/opt/hpvm/guests/testme \
 testme1
```

Do not use the `hpvmsys` group for user accounts. This group is used for security isolation between components of Integrity VM.

These types of console users are specified as either `admin` (guest administrators) or `oper` (guest operators). Guest operators can access to the virtual machine console, shut down and reboot the guest, display system status, transfer control to another guest operator or administrator, and set system identification. The guest administrator has all these capabilities, as well as the ability to use the virutal console `say` commands (restricted to use by HP field support specialists).

You can specify guest administrators and operators using the `hpvmcreate`, `hpvmmodify`, `hpvmmigrate`, and `hpvmclone` commands. To assign administrator and operator privileges to a user group, include the `-g` option. To assign administrator and operator privileges to a specific user, use the `-u` option.

NOTE: Console users cannot use the `su` command to change from one privilege level to another. Per-user checks are based on login account identifiers, not on UUIDs.

The following command creates the virtual machine named `testme` with the adminstrator named `testme1`:

```
# hpvmcreate -P testme -u testme1:admin
```

Guest operators and administrators need access to the `hpvmconsole` command to control the virtual machine. If you do not want the same users to have access to the VSP, you can restrict use of the `hpvmconsole` command to guest console access only by creating a restricted account for that purpose. To do so, follow these steps:

1. Using the useradd command, set up an /etc/passwd entry for each guest on the VSP. The user name of the account must be the same as the guest name and must have no more than 8 characters. For example:

   ```
   # useradd -d /var/opt/hpvm/guests/host1 \
   -c 'host1 console' -s /opt/hpvm/bin/hpvmconsole host1
   ```

   This example uses the following options:

   - The -d option specifies the home directory for the host1 account.
   - The -c option specifies a comment text string that describes the account.
   - The -s option specifies the path for the shell of the new account.

2. Use the passwd command to set a password for the account. For example:

   ```
   # passwd host1
   ```

3. Use the hpvmmodify command to provide the user with guest administration privileges:

   ```
   #hpvmmodify -P winguest1 -u host1:admin
   ```

A guest administrator can now access the host1 virtual console by using the ssh command or telnet command on the VSP and logging in to the host1 account. The guest administrator cannot use the su command.

NOTE: For security reasons, HP strongly recommends that you do not include /opt/hpvm/bin/hpvmconsole, the virtual console image, in /etc/shells. Doing so opens two security vulnerabilities:

- It allows ftp access to the account.
- It allows a general user to select the image with the chsh command.

The following is an example session of remote access to the host1 virtual console on the VSP myhost:

```
# telnet host1

Trying .xx.yy.zz...
Connected to host1.rose.com.
Escape character is '^]'.

HP-UX host B.11.31 U ia64 (ta)

login: guest1
Password:
Please wait...checking for disk quotas


   MP MAIN MENU

         CO: Console
         CM: Command Menu
         CL: Console Log
         SL: Show Event Logs
         VM: Virtual Machine Menu
         HE: Main Help Menu
          X: Exit Connection

[host1] vMP>
```

The virtual console interface displays raw characters for the CL and CO commands, including the guest's attempts to query the console terminal for its type and characteristics. As a result, the terminal answers those queries, which can cause the terminal setup communication to interfere with the virtual console commands. Interactive users can clear the screen. However, this situation can be a problem for noninteractive or scripted use of the console.

11.5.1 Administrator account names

The virtual console administrator name can be any valid HP-UX login name. To continue accessing the virtual console, existing guest console accounts must be added to the authorization list for the associated guest with the `usermod` command. This allows multiple accounts to map to the guest, and requires the account names to be valid HP-UX login strings.

Authorization of access to the virtual console is determined by the guest configuration file (set using the `-u` and `-g` options to the `hpvmcreate`, `hpvmmodify`, and `hpvmclone` commands). This controlled access allows you to temporarily block access by using the `hpvmmodify` command to change the virtual console administrator account name.

11.5.2 vPars/VM user accounts

The configuration for captive `hpvmconsole` guest user accounts supports additional access controls and configurations. This change requires that the guest user accounts have the correct home directory. To ensure that the user continues to have administrative console access, use the following command:

```
# hpvmmodify -P compass1 -u compass1:admin
```

11.6 Using the virtual console

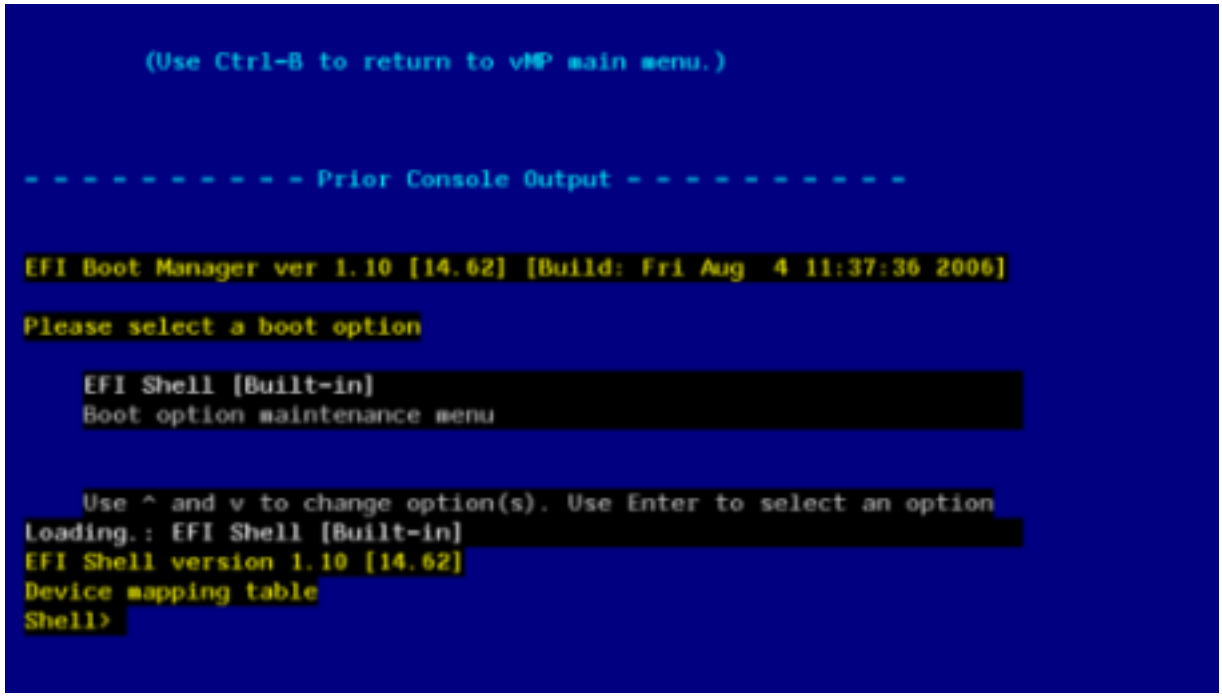
Each vPar/VM has its own virtual console from which to power on or off the vPar/VM, boot the guest operating system or shut it down, and so on. The `hpvmconsole` command connects to the virtual console of a specified vPar/VM.

To start the virtual console for the guest named `host1`, enter the following command:

```
# hpvmconsole -P host1


   vMP MAIN MENU

         CO: Console
         CM: Command Menu
         CL: Console Log
         SL: Show Event Logs
         VM: Virtual Machine Menu
         HE: Main Help Menu
          X: Exit Connection

[host1] vMP>
```

To return to the virtual console when the display is in the EFI, press **Ctr/B**. Use the `co` command to open the virtual console. For example:

```
[host1] vMP> co
```



You can pass a command to the vPar/VM console using the –c option to the `hpvmconsole` command. For example, to start a virtual machine named `host1`, enter the following command:

```
# hpvmconsole -P host1 -c "pc -on"
```

Table 26 lists the options to the `hpvmconsole` command.

Table 26 Options to the `hpvmconsole` Command

Option	Description
`-P` *vm-name*	Specifies the name of the virtual machine console to open.
`-p` *vm-number*	Specifies the number of the virtual machine console to open.
`-c` *command*	Specifies a machine console command to run on the virtual machine.
`-e` *echar*	Specifies an alternate interrupt character. The default interrupt character is **Ctrl/B**, unless the session is on the VSP's `/dev/console`, in which case, use the **Ctrl/X**.
`-f`	Follows the console output after reaching EOF on standard input. Used for scripting.
`-i`	Interacts with the console. Used for scripting.
`-q`	Makes scripted operations less verbose.

To get information about using the virtual console, enter the `HE` command. For example:

```
[host1] vMP> he
==== vMP Help: Main Menu ================================== (Admin) ============

HPVM B.06.10  ccipf opt Thu Dec 01 2011
(C) Copyright 2000 - 2011 Hewlett-Packard Development Company, L.P.

        Virtual Management Processor (vMP) Help System

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

For more information about using the `hpvmconsole` command, see *hpvmconsole*(1M).

11.7 Using the virtual iLO Remote Console

The vPars and Integrity VM virtual iLO Remote Console feature allows you access to the guest console by logging into a specific IP address. You can assign each guest a virtual iLO Remote Console IP Address with which the end user can connect using either telnet or Secure Shell (SSH). After login authentication, the guest console is immediately available. The user no longer needs to know the VSP machine IP address or guest name. They need to know only the virtual iLO Remote Console IP Address. The virtual iLO Remote Console IP stays the same even after an Online VM Migration. There is also no need to manually run any command, like the `hpvmconsole` command.

The following section describes:

- Configuring a virtual iLO Remote Console
- Choosing the virtual iLO Remote Console IP address
- Deleting a virtual iLO Remote Console
- Getting the virtual iLO Remote Console settings of a guest

11.7.1 Configuring, deleting, and obtaining status of a virtual iLO Remote Console

You can assign a virtual iLO Remote Console IP Address when you create, modify, or clone a guest, using the `hpvmcreate`, `hpvmmodify`, or `hpvmclone` commands:

- `hpvmclone -P` *`guestname`* `-K` *`Remote-Console-IP-Address`* `-L` *`Remote-Console-Mask`*
- `hpvmcreate -P` *`guestname`* `-K` *`Remote-Console-IP-Address`* `-L` *`Remote-Console-Mask`*
- `hpvmmodify -P` *`guestname`* `-K` *`Remote-Console-IP-Address`* `-L` *`Remote-Console-Mask`*

For example:

```
# hpvmmodify -P guestname -K 16.92.81.68 -L 255.255.252.0
```

NOTE: Only IPv4 addresses are supported, not IPv6.

The virtual iLO Remote Console IP address must be unique and different from both the Host IP address and the Guest IP address. The virtual iLO Remote Console IP address does not need to be configured in advance. When the virtual iLO Remote Console is created, Integrity VM automatically creates an alias interface for the IP address. For example, if you create the virtual iLO Remote Console:

```
# hpvmmodify -P guestname -K 16.92.81.68 -L 255.255.252.0
```

Integrity VM configures the IP alias in a similar manner as if you specified the `ifconfig` command:

```
"ifconfig lan0:274485572 16.92.81.68 netmask 255.255.252.0"
```

To see the alias interface that Integrity VM creates, run the `netstat` command:

```
# netstat -rn
Routing tables
Destination           Gateway           Flags Refs Interface  Pmtu
127.0.0.1             127.0.0.1         UH    0    lo0       32808
16.92.81.68           16.92.81.68       UH    0    lan1:274485572 32808
16.92.80.101          16.92.80.101      UH    0    lan1      32808
127.0.0.0             127.0.0.1         U     0    lo0       32808
default               16.92.80.101      U     0    lan1       1500
```

To delete a virtual iLO Remote Console, specify 0 as the IP address. For example:

```
# hpvmmodify -P guestname -K 0
```

To obtain the virtual iLO Remote Console settings of a guest, use the `hpvmstatus` command. For example:

```
# hpvmstatus -P guestname
....
[Remote Console]
Remote Console Ip Address:    16 .92.81.68
Remote Console Net Mask:        255.255.252.0
```

When users connect to the virtual iLO Remote Console IP Address, they must log in using the standard telnet or ssh system authentication. After authenticating, they receive immediate access to the guest console:

```
# ssh -l guest1admin 16.92.81.68
Password:
   vMP MAIN MENU

         CO: Console
         CM: Command Menu
         CL: Console Log
         SL: Show Event Logs
         VM: Virtual Machine Menu
         HE: Main Help Menu
          X: Exit Connection

[guest1] vMP>
```

The username used to access and log into the virtual iLO Remote Console must have Guest Administrator/Operator privileges. The following example creates a guest administrator name **guest1admin** for the guest **guest1**. The `hpvmmodify -u` option is used to grant the guest administrator privilege:

```
# useradd -d /var/opt/hpvm/guests/guest1 -c 'guest1 console' guest1admin
# passwd guest1admin
# hpvmmodify -P guest1 -u guest1admin:admin
# hpvmmodify -P pqsvm53  -K  xxx.xxx.xxx.xxx  -L  xxx.xxx.xxx.xxx
# telnet xxx.xxx.xxx.xxx
```

For more information, see Section 11.5 (page 174)

When a guest is migrated to another VSP using Online VM Migration (OVMM), the Integrity VM virtual iLO virtual iLO Remote Console is also migrated to the new VSP. Before migration, the virtual iLO Remote Console process is running on only the source VSP. After migration, the virtual iLO Remote Console process is stopped on the source VSP. Any client that was connected to that virtual iLO Remote Console is disconnected. A new virtual iLO Remote Console process is started on the target VSP. New client connections to the virtual iLO Remote Console IP address are now sent to the virtual iLO Remote Console process on the new VSP.

11.7.2 Integrity VM virtual iLO Remote Console limitations

The virtual iLO Remote Console feature has four limitations:

- The telnet method of connecting is not supported for the virtual iLO Remote Console by default, only Secure Shell.

  To add telnet support for virtual iLO Remote Console, you must install two additional HP-UX enhancement patches, one for telnetd and one for the login (/usr/bin/login) command. If you try to telnet to the virtual iLO Remote Console without these patches, an error message is sent to the telnet client, and the connection is closed.

  Install the following patches on the VSP:

  - PHCO_41595
  - PHNE_41452

- The virtual iLO Remote Console's SSH server host keys can change.

  When an SSH client connects to an SSH server, the client downloads the server's host keys and keeps a local copy (usually in a file such as `~/.ssh/known_hosts`). On subsequent

connections, the SSH client verifies that the host key sent by the server matches the local copy. If the keys do not match, the SSH client prints an error message.

The virtual iLO Remote Console uses the host system's SSH server host keys. If the guest is migrated to another host system (using Online VM Migration), these host keys will change. When an end user does an SSH connection, they will receive an error message. The end user must manually delete the local copy of the host key. For additional information, see the `ssh(1)` manpage .

- Guest Administrator accounts are not migrated during Online VM Migration (OVMM).

  Any guest administrator accounts residing on the source VSP system are not automatically migrated to the target VSP system during Online VM Migration (OVMM). You must manually add any guest administrator accounts to the target VSP system, using the same `useradd` commands performed on the source system. For information about creating guest administrator and operator accounts, see (page 174).

- The virtual iLO Remote Console does not support rlogin connections.

11.8 Guest configuration files

When the guest is created, the VSP creates the guest configuration file `/var/opt/hpvm/guests/`*`guestname`*.

Integrity VM creates up to three guest configuration files:

- The `vmm_config.current` file contains the current guest configuration currently set.
- The `vmm_config.prev` file contains the last known guest configuration settings.
- The `vmm_config.next` file contains the configuration settings that have changed since the guest was started. To initiate these changes, you must reboot the guest.

Never modify the guest configuration files manually. Always use the appropriate Integrity VM command (`hpvmmodify` or `hpvmdevmgmt`) to modify guest configuration parameters. Directly modifying the guest configuration files can cause guests to fail in unexpected ways.

11.9 Dynamic memory

Dynamic memory is an optional feature of Integrity VM that allows you to change the amount of physical memory in use by a virtual machine without rebooting the virtual machine.

NOTE: Dynamic memory is not available with vPars at this time.

An example of this feature allows a VM that is a Serviceguard node to be used as a standby server for multiple Serviceguard packages. When a package fails over to the VM, the VM memory can be changed to suit the requirements of the package before, during, and after the failover process.

To use dynamic memory, the VM must have the VirtualBase software installed, as described in Section 8.3 (page 104). Note the management software is installed automatically with HP-UX 11i v3 March 2012 VMs. If you are using VMs with HP-UX 11i v3 September 2011, you need to install the VirtualBase software.

11.9.1 Managing dynamic memory from the VSP

On the VSP, the dynamic memory software is included with Integrity VM. Manage dynamic memory on the VSP using the `-x` option with the `hpvmcreate`, `hpvmmodify`, or `hpvmclone` command. The `-x` option associates a variety of configuration parameters with the guest, including dynamic memory and network management for the guests. Table 27 provides a complete list of `-x` keywords used for dynamic memory.

Table 27 Dynamic Memory Control Command Options

Keyword Value Pair	Description
`dynamic_memory_control={1|0}`	Specifies whether a sufficiently privileged user on the guest (such as `root`) can change the dynamic memory values while the guest is running. To disable guest-side dynamic memory control, specify 0 (zero). If the guest is not active, the only effect is the modification of the guest configuration file. On the running guest, the change takes effect immediately.
`ram_dyn_type={none|any|driver}`	Specifies the type of dynamic memory control for the guest. When this configuration parameter is set to `none`, dynamic memory is disabled. This is the default setting. If your guest is running with dynamic memory enabled and you set this value to `none`, the guest configuration file is modified to remove all dynamic memory ranges and control information. When this configuration parameter is set to `any`, the next boot of the guest determines whether or not dynamic memory is enabled on the guest. If the dynamic memory driver is loaded, the value of this parameter is changed to `driver`. If no drivers are loaded or found, the value is not changed. When this configuration parameter is set to `driver`, guest dynamic memory controls and ranges are functional. Depending on the current or default settings, messages might be displayed indicating a resetting of the dynamic memory range values to match the current memory range settings. If you change the available guest memory value (using the `-r` option), the dynamic memory values are validated for range and modified.
`ram_dyn_min=`*`amount`*	Specifies the minimum amount of memory that can be dynamically allocated to the guest. The `ram_dyn_min` value must be greater than the minimum memory (displayed by the `hpvmstatus` command) and less than the `ram_dyn_max` value.
`ram_dyn_max=`*`amount`*	Specifies the maximum amount of memory that can be dynamically allocated to the guest. The value of `ram_dyn_max` must be greater than the value of`ram_dyn_min`.
`ram_dyn_target_start=`*`amount`*	Specifies the amount of memory that the dynamic memory driver attempts to access when the guest starts. The value of the `ram_dyn_target_start` must be greater than the `ram_dyn_min` parameter and less than or equal to the `ram_dyn_max` parameter. When the guest starts, it initially has access to the guest memory size (specified by the `-r` option), then the dynamic memory driver reduces the memory to the value of the `ram_dyn_target_start` parameter. The ram_dyn_entitlement and amr_enable options must be set to enable adjustments.
`ram_dyn_entitlement=`*`amount`*	Specifies the minimum guaranteed amount of memory.
`amr_enable={0|1}`	Specifies whether adjustments can be made.
`amr_chunk_size=`*`amount`*	Specifies the increment amount for changes in memory size (default is 256 MB). Larger values result in faster memory size growth.
`ram_target={0|start|`*`amount`*`}`	Sets the current memory size for the guest. The `ram_target` keyword is valid on the `hpvmmodify` and `hpvmmgmt` commands only. When you specify 0 (zero), the dynamic memory driver reduces the memory on the guest to a comfortable minimum without forcing guest memory to be paged out. This minimum value changes over time as the guest's operating needs change. When you specify `start`, the guest dynamic memory size grows to the allocated value specified using the `-r` option. This parameter is dynamic and can be used only on an active guest.

11.9.1.1 Configuring a virtual machine to use dynamic memory

By default, dynamic memory is not enabled. To configure a virtual machine to use dynamic memory, enter the `hpvmcreate`, `hpvmmodify`, or `hpvmclone` command. Include the following `-x` option to set initial values:

```
-x ram_dyn_type = any | driver
-x ram_dyn_min = minimum size for memory size changes
-x ram_dyn_max = maximum size for memory size changes
```

You can configure a virtual machine to reduce its memory size early in a boot process, making the virtual machine available but maintaining lower memory overhead on the VSP system. Use the following `-x` option to enable this feature:

```
-x ram_dyn_target_start = memory size after boot
```

You can supply several dynamic memory keywords on the same command line. For example, to enable dynamic memory and to configure the guest named host1 to reduce its size early in the boot process, enter the following command:

```
# hpvmmodify -P host1 -r 6G \
-x ram_dyn_type=any \
-x ram_dyn_min=1222M \
-x ram_dyn_max=6G \
-x ram_dyn_target_start=2G
```

This command specifies the following values:

- The virtual machine memory size is set to 6 GB.
- Dynamic memory is enabled using any dynamic memory support available.
- The minimum amount of memory that the virtual machine can be reduced to is 1222 MB.
- The maximum amount of memory that the virtual machine can be increased to is 6 GB.
- The memory size to reduce to after it boots is 2 GB.

If the virtual machine is running when the dynamic memory feature is configured for the first time, the virtual machine must be rebooted for the configuration changes to take effect.

11.9.1.2 Viewing dynamic memory on the VSP

Dynamic memory parameters and status are displayed for each guest using the standard Integrity VM commands. For example, for the guest named `host1`, the `hpvmstatus` command displays the following information about dynamic memory:

```
# hpvmstatus -V -P host1
.
.
.
[Dynamic Memory Information]
  Type                    : driver
  Minimum memory          : 1222 MB
  Target memory           : 2103 MB
  Maximum memory          : 6144 MB
  Current memory          : 2103 MB
  Comfortable minimum     : 27 MB
  Boot memory             : 6135 MB
  Free memory             : 125 MB
  Available memory        : 286 MB
  Memory pressure         : 0
  Memory chunksize        : 65536 KB
  Driver Mode(s)          : STARTED ENABLED
.
.
.
```

Table 28 describes the dynamic memory characteristics displayed by the `hpvmstatus` and `hpvmmgmt` commands.

Table 28 Dynamic Memory Characteristics

Characteristic	Setting	Description
Type	`none`	No dynamic memory support
	`any`	Dynamic memory is configured on the host, but the dynamic memory subsystem on the guest has not started and reported the implementation type.
	`driver`	Dynamic memory is implemented in a driver and does not use Guest OS Online Add/Delete features.
	`OLAD`	Dynamic memory is implemented using Guest OS Online Add/Delete features.
Minimum memory	*value*`M` (for megabytes) or *value*`G` (for gigabytes)	The lower bounds for `ram_target` and `ram_dyn_target_start`.
Target memory	*value*`M` (for megabytes) or *value*`G` (for gigabytes)	The target memory size of the guest, set using `ram_target` or `ram_dyn_target_start`.
Maximum memory	*value*`M` (for megabytes) or *value*`G` (for gigabytes)	The upper bounds for `ram_target` and `ram_dyn_target_start`.
Current memory	*value*`M` (for megabytes) or *value*`G` (for gigabytes)	The current memory size of the guest (normally equal to target memory).
Comfortable minimum	*value*`M` (for megabytes) or *value*`G` (for gigabytes)	A value for `ram_target` which can be used to reduce the guest memory but allow it sufficient memory resources to continue running a minimal workload.
Boot memory	*value*`M` (for megabytes) or *value*`G` (for gigabytes)	Size of physical memory in the virtual machine presented to the guest OS.
Free memory	*value*`M` (for megabytes) or *value*`G` (for gigabytes)	Amount of free memory in the guest.
Available memory	*value*`M` (for megabytes) or *value*`G` (for gigabytes)	Amount of memory in the guest allocated by user processes but not locked. This memory is available for paging.
Memory pressure	*value*	A value between 0 and 100 used as an indicator of memory deficit and paging. The higher the number the longer the system has been in a memory deficit. A memory pressure value approaching 100 usually means the system is hung.
Memory chunksize	*value*	The allocation chunk size used by dynamic memory when increasing and descreasing guest memory (as described in Section 11.9.3.4 (page 186).
Driver mode(s)	`started`	Dynamic memory can change guest memory size.
	`enabled`	Control that overrides `started`.
	`guestctl`	Guest-side control is enabled.

The following example displays active usage of the VSP and guest(s) dynamic memory usage values, along with the guest memory utilization. The guest's current swapping and paging and translation address memory misses per second are included. For a description of each column displayed, see the `hpvmsar` manpage. The dash (-) in the example indicates the guest named `ux2` is not currently booted.

```
# hpvmsar -G -A
HP-UX witch4 B.11.31 U ia64    10/22/10

10:02:28   GUEST   GTOTMEM(MB) HDYNRCLM(MB)  GCURMEM(MB) GCURFREE(MB)        GSWAP        GPAGE GADDRTMISS/s
10:02:30   ux1            8186            0         8186         5956            0            0            0
           ux2               -            -            -            -            -            -            -
10:02:31   ux1            8186            0         8186         5956            0            0            0
           ux2               -            -            -            -            -            -            -
10:02:32   ux1            8186            0         8186         5956            0            0            0
           ux2               -            -            -            -            -            -            -
```

11.9.1.3 Modifying a virtual machine's memory size on the VSP

Once dynamic memory is configured, a virtual machine's memory size can be changed to any value between the minimum size (**`ram_dyn_min`**) and the maximum size (**`ram_dyn_max`**) in increments of the chunk size (64 MB). Use the following `-x` option to the `hpvmmodify` command to change the memory size:

```
# hpvmmodify -P host1 -x ram_target = new memory size
```

11.9.2 Managing dynamic memory from the guest

Dynamic memory management from the guest is disabled by default and must be enabled from the VSP. If the feature is not enabled, dynamic memory information can be displayed , but the memory size cannot be changed.

Use the `hpvmcreate`, `hpvmmodify`, or `hpvmclone` command and include the `-x dynamic_memory_control` option. Specify `1` as the argument to the option. For example, on the VSP system, enter the following command to enable dynamic memory control on the guest named `host1`:

```
# hpvmmodify -P host1 -x dynamic_memory_control=1
```

11.9.2.1 Viewing dynamic memory information from the guest

Use the `hpvmmgmt` command on the HP-UX guest to manage and view the dynamic memory information. This command is installed when you install the VirtualBase software, as described in Section 8.3 (page 104). Note, VirtualBase is installed automatically with HP-UX 11i v3 March 2012 VMs. If you are using VMs with the HP-UX 11i v3 September 2011 release, you need to install the VirtualBase software.

Table 29 describes the options to the `hpvmmgmt` command.

Table 29 Options to the `hpvmmgmt` Command

`-l` *`type`*	Specifies the type of data to list more information about. For *`type`*, enter `ram`.
`-l` *`type`* `-t` *`interval`*	Allows you to continually watch and check the dynamic ram values. For the *`interval`*, specify the number of seconds between fetches of live data.
`-t` *`interval`*	Allows the `hpvmmgmt` command to continuously refetch the requested type of data using the value specified for the *`interval`* parameter.
`-c`*`num`*	Specifies the number of virtual CPUs to enable on the guest.
`-v`	Displays the version number of the `hpvmmgmt` command.
`-V`	Displays detailed information (verbose mode) about the virtual machines.

Table 29 Options to the `hpvmmgmt` Command *(continued)*

`-M`	Displays verbose attribute and resource information in a machine-readable format.
`-X`	Displays verbose attribute and resource information in the XML format.
`-x ram_target={0 | start | amount}`	Specifies the guest RAM target, where: • 0 indicates the guest dynamic memory will be reduced to a comfortable minimum value. • `start` indicates the guest dynamic memory will be set back to the boot time value. • *`amount`* is a specific target memory size for the guest.

For example, on the guest, use the `hpvmmgmt` command to list the dynamic memory information. Enter the following command:

```
# hpvmmgmt -l ram

  [Dynamic Memory Information]
  ======================================
  Type                    : driver
  Current memory          : 6135 MB
  Target memory           : 6135 MB
  Comfortable minimum     : 27 MB
```

To display more information, include the `-V` option. For example:

```
# hpvmmgmt -V -l ram
  [Dynamic Memory Information]
  ======================================
  Type                    : driver
  Current memory          : 2103 MB
  Target memory           : 2103 MB
  Comfortable minimum     : 2423 MB
  Minimum memory          : 1222 MB
  Maximum memory          : 6144 MB
  Boot memory             : 6135 MB
  Free memory             : 124 MB
  Available memory        : 286 MB
  Memory pressure         : 12
  Memory chunksize        : 65536 KB
  Driver Mode(s): STARTED ENABLED GUESTCTL
```

11.9.2.2 Modifying a virtual machine's memory size from the guest

Once the dynamic memory feature is configured and enabled, a virtual machine's memory size can be changed to any value between the minimum size (ram_dyn_min) and the maximum size (ram_dyn_max) in increments of the chunk size (64 MB). Use the following `-x` option to the `hpvmmgmt` command:

```
# hpvmmgmt -x ram_target=memory size
```

For example, to change the guest memory size to 4 GB, enter the following command:

```
# hpvmmgmt -x ram_target=4096M
Attempting to increase memory from 2103 MB to 4096 MB.
Successfully began to change ram_target to 4096 MB.
```

11.9.3 Troubleshooting dynamic memory problems

This section describes how to solve problems in the use of dynamic memory.

11.9.3.1 Dynamic memory restrictions

Use of dynamic memory is subject to the following restrictions:

- The size of a virtual machine cannot be increased above its original boot size (as specified with the `-r` option).
- If the virtual machine memory has become fragmented, attempting to reduce the size of the virtual machine might fail or might take a very long time. If you cannot reduce the size of the virtual machine to the desired size, abort the operation by setting a new target size.
- Increasing the size of a virtual machine requires free memory on the VSP. If the VSP memory is insufficient, the operation might take a very long time to complete and might fail.
- If the values of `ram_target` and `ram_dyn_target_start` are not inside the values of `ram_dyn_min` and `ram_dyn_max`, a warning is issued.

11.9.3.2 VSP resource considerations

HP-UX supports “large pages, ” a memory management feature used to improve performance. Integrity VM takes advantage of this feature by ensuring that when a virtual machine starts, it allocates the largest size pages that are available. Once these pages are allocated and locked down, they cannot change size. This constraint minimizes fragmentation of large pages.

This feature limits the types of software you can run on a VSP system. If the VSP system supports an additional workload beyond the virtual machines, the large pages tend to fragment and performance of the newly started virtual machine might degrade.

Dynamic memory increases the possibility of VSP memory becoming fragmented. The current implementation of dynamic memory releases portions of the memory allocated to a virtual machine. These operations must be performed in large contiguous chunks; otherwise, the act of reducing the size of a virtual machine fragments the VSP memory allocated to it. This potential fragmentation is prevented by the software, which reduces a virtual machine's size in multiples of a minimum chunk size of 64 MB of physically contiguous memory. For more information, see Section 11.9.3.3 (page 186) and Section 11.9.3.5 (page 187).

11.9.3.3 VM resource considerations

During normal operation of a system that has a workload running on it, the large pages might become fragmented over time. This is true on the VSP as well as a virtual machine running the HP-UX operating system. If the virtual machine's memory is fragmented, the dynamic memory subsystem is unable to reduce the size of guest. This is due to the minimum chunk size used for the reduction. If dynamic memory cannot remove at least 64 MB of physically contiguous guest memory, no reduction in size takes place.

11.9.3.4 Specify sufficient VM memory

If you set the value of `ram_dyn_target_start` too small, the VM's guest operating system might hang or crash while booting. In this case, the VM does not have access to a sufficient amount of memory. As a rule, do not decrease the memory allocated to an HP-UX guest by more than 75% of its allocated memory size. Do not reduce the memory of a virtual machines configured with 2 GB of memory by more than 50%.

If the VM crashes while booting on the VSP, use the `hpvmmodify` command to increase the value of the `ram_dyn_target_start` parameter. For example, to increase the memory size for the VM named `host1`, enter the following command on the VSP:

```
# hpvmmodify -P host1 -x ram_dyn_target_start=2GB
```

After you set this parameter, reboot the VM.

If the VM hangs, on the VSP, use the `hpvmstatus` command to check the memory statistics on the VM. For example:

```
# hpvmstatus -V -P host1
.
.
.
[Dynamic Memory Information]
  Type                    : driver
  Minimum memory          : 1222 MB
  Target memory           : 2103 MB
  Maximum memory          : 6144 MB
  Current memory          : 2103 MB
  Comfortable minimum     : 27 MB
  Boot memory             : 6135 MB
  Free memory             : 0 MB
  Available memory        : 286 MB
  Memory pressure         : 100
  Memory chunksize        : 65536 KB
  Driver Mode(s)          : STARTED ENABLED
.
.
.
```

An indication of this problem is a small or zero amount of free memory and a large memory pressure value (100). If these indicators are present, use the `hpvmmodify` command on the VSP to increase the memory size of the VM. The VM then boots normally.

11.9.3.5 Actual memory allocations might differ

If you specify a value for the `ram_target` or `ram_dyn_target_start` parameter that results in a change in memory size that is not a multiple of 64 MB, the target value is reset.

For example, if you specify 6 GB of memory, the HP-UX guest actually has access to 6135 MB of memory. If you attempt to set the memory size to 2048 MB, the amount of memory actually removed is 4087 MB. This is not a multiple of 64 MB, so the target memory size is reset to 2103 MB.

11.9.3.6 Enable dynamic memory on the VM and on the VSP

The VirtualBase software must be installed on the VM before you can use dynamic memory parameters on the VSP system. For example, if the VirtualBase software is not installed, the `hpvmstatus` command displays the following:

```
# hpvmstatus -V -P host1
.
.
.
[Dynamic Memory Information]
    NOTE: Dynamic data unavailable, configured values only
Type                    : driver
Minimum memory          : 1024 MB
Target memory           : 2048 MB
Maximum memory          : 3072 MB
.
.
.
```

Note, VirtualBase is installed automatically with HP-UX 11i v3 March 2012 VMs. If you are using VMs with HP-UX 11i v3 September 2011, you need to install the VirtualBase.

If you attempt to modify the VM's dynamic memory from the VSP, the following errors are displayed:

```
# hpvmmodify -x ram_target=2048M -P host1

hpvmmodify: ERROR (host1): Query to dynamic memory driver failed: Function is not available.
hpvmmodify: Failed to set ram_target.
hpvmmodify: Unable to modify the guest.
```

If you attempt to modify the dynamic memory from the VM, the following errors occur:

```
# hpvmmgmt -V -l ram
Dynamic memory driver not found on guest.
```

```
hpvmmgmt: Unable to continue.
# hpvmmgmt -x ram_target=2048
Failed to open dynamic memory driver, error: No such device.
Failed to set dynamic value error: No such device
hpvmmgmt: Unable to continue.
```

For information about installing the VirtualBase software, see Section 8.3 (page 104).

11.9.3.7 Upgrade the VirtualBase software when upgrading Integrity VM

The dynamic memory software has two components: the VSP support and the HP-UX guest support. These two components must be at the same version level for dynamic memory to function. When you upgrade Integrity VM, you must also install the new VirtualBase kit on the guest. (You should also upgrade the guest operating system if it is no longer supported.) During this upgrade process, dynamic memory may not function.

If there is a version mismatch, a message is written to the VSP's `syslog` file (`/var/adm/syslog/syslog.log`) when the guest starts. For example:

```
vmunix: (hpvmdvr) Dynamic memory version mismatch Guest 5.
Please update the guest kit
```

This example indicates that the VirtualBase software kit on virtual machine number 5 is out of date. To determine which guest is number 5, use the `hpvmstatus` command. In the following example, guest 5 is named `dale`:

```
# hpvmstatus

Virtual Machine Name VM #  OS Type State     #VCPUs #Devs #Nets Memory  Runsysid
==================== ===== ======= ========= ====== ===== ===== ======= ========
chip                     1 HPUX    On (OS)        2     1     1    3 GB         0
dale                     5 HPUX    On (OS)        2     1     1    3 GB         0
```

Note, VirtualBase is installed automatically with HP-UX 11i v3 March 2012 VMs. If you are using VMs with HP-UX 11i v3 September 2011, you need to install VirtualBase.

For information about installing the VirtualBase software, see Section 8.3 (page 104).

11.9.4 Automatic memory reallocation

Automatic Memory Reallocation is an optional feature of Integrity VM that allows automated changes in the amount of physical memory in use by virtual machines based on memory load conditions. Automatic memory reallocation is available only on guests that support dynamic memory.

To use automatic memory reallocation, the VM must have the VirtualBase software installed, because this is required for dynamic memory. For vPar/VM VirtualBase software installation instructions, see Section 8.3 (page 104).

Note, VirtualBase is installed automatically with HP-UX 11i v3 March 2012 VMs. If you are using VMs with HP-UX 11i v3 September 2011, you need to install VirtualBase software.

11.9.4.1 Enabling automatic memory reallocation on the VSP

On the VSP, the automatic memory reallocation software is included with Integrity VM. The automatic memory reallocation daemon (hpvmamrd) is enabled by default. To disable automatic memory reallocation, the following line must be included in the `/etc/rc.config.d/hpvmconf file`: `HPVMAMRENABLE=0`. When `HPVMAMRENABLE=0` is not set in `hpvmconf`, `hpvmamrd` is automatically started and stopped when Integrity VM is started and stopped.

When running, `hpvmamrd` monitors the state of VMs that have been enabled for automatic memory reallocation. Every ten seconds, `hpvmamrd` examines the state of relevant VMs, and takes action within the parameters described in the next section. It also takes action when an attempt is made to boot a VM that requires more physical memory than is currently available.

11.9.4.2 Enabling automatic memory reallocation on a VM

By default, VMs are not enabled for automatic memory reallocation. Only VMs that support dynamic memory can use automatic memory reallocation. Use the following `-x` options to enable automatic memory reallocation on a VM:

`-x amr_enable`
`-x ram_dyn_entitlement=`*`minimum memory size in MB`*

This option is supported on running VMs. No error occurs if this is executed for a VM that does not support dynamic memory, but it is ignored. A VM that does not have a value for `ram_dyn_entitlement` is also ignored by automatic memory reallocation. A VM that has been enabled for automatic memory reallocation does not support manual dynamic memory operations from the VM. It does not support manual dynamic memory operations from the VSP that would cause the VM to shrink below its entitlement.

11.9.4.3 Viewing automatic memory reallocation

Automatic memory reallocation parameters and status are displayed for each VM using the standard Integrity VM commands. The `hpvmstatus` command displays the following information about automatic memory reallocation:

```
# hpvmstatus -r
[Virtual Machine Resource Entitlement]
[Virtual CPU entitlement]
                                                    Percent       Cumulative
Virtual Machine Name VM #  #VCPUs Entitlement Maximum   Usage            Usage
==================== ===== ====== =========== ======= ======= ================
guest0                   1      2       10.0%  100.0%    2.0%            237
guest1                   2      2       10.0%  100.0%    2.5%            28863

[Virtual Machine Memory Entitlement]
                          DynMem  Memory   DynMem  DynMem DynMem  Comfort Total    Free   Avail    Mem    AMR
     AMR
Virtual Machine Name  VM #   Min   Entitle   Max    Target Current   Min   Memory  Memory  Memory  Press  Chunk
   State
==================== ===== ======= ======= ======= ======= ======= ======= ======= ======= ======= ===== =======
 ========
guest0                   1   512MB     2GB     5GB  5114MB  5114MB  1722MB     5GB  3534MB   324MB   0
0B  DISABLED
guest1                   2     1GB     2GB     4GB  2106MB  2106MB  1594MB     4GB   801MB   282MB   0     400MB
  ENABLED
```

11.10 vPar/VM log files

Each vPar/VM has a log file named `/var/opt/hpvm/guests/`*`guestname`*`/log` on each VSP.

The VSP log files are stored as `/var/opt/hpvm/common/command.log` and `hpvm_mon_log`.

NOTE: A `Failed API access to local running guest.` message in the `command.log` is a notification that a communication attempt with the `hpvmapp` process has failed. This message is not an indication of a problem and can be ignored.

11.11 Managing the device database

vPar/VM cannot detect all potential backing store conflicts, and does not always prevent misconfigured vPars/VMs from booting. Conflicts can arise from the following:

- Specifying the same backing store for more than one virtual device.

  If you add `disk:avio_stor::disk:/dev/rdisk/disk2` for guest A, do not add the same device to another guest or to the list of VSP restricted devices.

- Specifying multiple backing store parameters that lead to the same physical storage.

  If the VSP has multiple paths to a storage device, like `/dev/rdisk/disk0 and /dev/rdisk/disk4`, only one path should be specified for a `disk:avio_stor` or `dvd:avio_stor` in guest A. The other path should not be used as a backing store by guest A or by any other guest or the VSP.

- Overlapping physical storage allocated for different backing store types.

  If a guest uses a logical volume (for example, `rlvol1`) as a backing store device, the disks used by the volume group on which the logical volume is made (for example, `/dev/vg01`) cannot be used as backing stores.

- Veritas VxVM DMP device files (files under `/dev/vx/rdmp/`) are not supported by Symantec for whole disk backing stores for virtual machines.

You can use the `ioscan` and `sam` commands to detect these conflicts. If you force guests configured with these conflicts to start, data corruption might occur.

On the VSP, do not extend a logical volume (LVM or VxVM) used as a backing store for a guest root disk. If you do this, the guest panics on its next reboot with the following error:

```
System panic: all VFS_MOUNTROOTs failed: Need DRIVERS.
```

The guest should be able to boot if the logical volume is reverted (using lvreduce in case of LVM) to its original size. If this fails, the guest root device has been corrupted, and the guest operating system must be reinstalled.

An AVIO logical volume backing store not used as a root disk can be extended while the guest is online. For HP-UX 11i v3 guests using AVIO, the guest is notified of the increased size of the backing store for logical volumes as well as raw disks, and the guest can take the appropriate actions to use the larger size.

For a SCSI logical volume used as a backing store for a guest data disk, you can extend the volume after removing it from the guest using the `hpvmmodify` command. After extending the volume, use the `hpvmmodify` command to add the volume to the guest. Do not modify a logical volume used as a backing store without first removing it from the guest.

After you extend the logical volume, use operating system commands on the guest to extend its file system.

NOTE: When you create a file system using the `sam` command on an HP-UX guest, do not initialize the disk. This option returns an error and the file system is not created.

11.11.1 The vPars/VM device database file

The vParVM device management stores vPar/VM device mapping information in the device database file (`/var/opt/hpvm/common/hpvm_mgmtdb`). This file is divided into three sections:

- The header, which states that the file cannot be hand edited.
- The restricted device section, which contains a list of host devices that guests are not allowed to access.
- The guest devices section, which contains devices, both storage and network, that guests are configured to use.

Do not edit the `hpvm_mgmtdb` file directly unless you are specifically advised to do so. Always use supported Integrity VM commands (such as `hpvmmodify` or `hpvmdevmgmt`) to modify virtual devices.

11.11.2 Using the `hpvmdevmgmt` command

To list and modify the devices used by the VSP and the vPars/VMs, use the `hpvmdevmgmt` command.

Table 30 describes the options to the `hpvmdevmgmt` command.

Table 30 Options to the `hpvmdevmgmt` Command

Option	Description
`-l {server|rdev|gdev}:`*`entry_name`*`:attr:`*`attr_name=attr_value`*	Lists an entry. To list all entries, enter the following command: `# hpvmdevmgmt -l all`
`-v`	Displays the version number of the `hpvmdevmgmt` output format. The version number is followed by the display specified by other options.
`-V`	Increases the amount of information displayed (verbose mode).
`-S` *`size filename`*	Creates a file for use as a virtual device. The size argument must end in either M for megabyte or G for gigabyte.
`-I`	Creates passthrough device files (for example, `/dev/rscsi`). Passthrough devices are used by attached devices, such as tape devices, media changers, and CD/DVD burners.
`-m {server|rdev|gdev}:`*`entry_name`*`[:attr:`*`attr_name=attr_value`*`]`	Modifies an existing attribute or adds the attribute if it does not already exist.
`-a {server|rdev|gdev}:`*`entry_name`*`[:attr:`*`attr_name=attr_value`*`]`	Adds an entry.
`-d {server|rdev|gdev}:`*`entry_name`*`[:`*`param`*`:`*`arg`*`]`	Deletes an entry.
`-d gdev_alias:/dev/rdisk/disk`*`nn`*	Deletes one alias if a device has multiple aliases defined.
`-n gdev:`*`oldentry_name`*`:`*`newentry_name0`*`[,`*`newentry_name1`*`]`	Replaces a device.
`-r`	Generates a report script that can be used after inspection to fix various device database problems.

For example, to display a list of the restricted devices, enter the following command:

```
# hpvmdevmgmt -l rdev
/dev/rdisk/disk4:CONFIG=rdev,EXIST=YES,DEVTYPE=DISK,
SHARE=NO::6005-08b4-0001-15d0-0001-2000-003a-0000
```

To make a device shareable among guests, enter the following command:

```
# hpvmdevmgmt -m gdev:/data/file.iso:attr:SHARE=YES
```

NOTE: Whenever you add a device that is going to be used in guest configurations to an Integrity VSP, run the `hpvmdevmgmt -I` command after adding the device to the host.

11.11.2.1 Sharing devices

With Integrity VM, you can allow devices to be specified as either shared or not shared. By default, vswitches are configured to be shared, and storage devices are configured to not be shared. As administrator, you can configure a storage device to be shared by multiple guests.

The `SHARE` attribute is checked only when booting a guest. If one guest is running with a nonshared device and another guest attempts to boot using that same device, the latter guest is blocked. If multiple guests need to share devices, then the `SHARE` attribute for those devices must be changed to `SHARE=YES` using the modify option (`-m`) with the `hpvmdevmgmt` command.

For example, to make the HP-UX iso.* images shareable so that two virtual machines (`host1` and `host2`) can use them to install at the same time, enter the following commands:

```
# hpvmdevmgmt -m gdev:/var/opt/hpvm/ISO-images/hpux/:attr:SHARE=YES
# hpvmmodify -P host1 -a dvd:avio_stor::null:/var/opt/hpvm/ISO-images/hpux/
# hpvmmodify -P host2 -a dvd:svio_stor::null:/var/opt/hpvm/ISO-images/hpux/
```

Virtual DVDs and virtual network devices can be shared. DVDs are not shareable unless you specify otherwise. Sharing of virtual devices or hardware backing stores must be carefully planned in order to prevent data corruption.

To restrict the vswitch named `myswitch` so that it is no longer shareable, enter the following command:

```
# hpvmdevmgmt -m gdev:myswitch:attr:SHARE=NO
```

This command restricts the vswitch called `myswitch` to be used by one guest only.

11.11.2.2 Replacing devices

If a backing storage device malfunctions, replace it by using the `hpvmdevmgmt -n` option. The `-n` option works for only guest devices. It replaces the existing device entry with the new device entry while keeping all the current guest dependents. Thus, each guest dependent is modified to replace the old device with the new one. If the device being replaced is a pNIC, use the `hpvmnet` command to halt and remove the current vswitches using that pNIC and recreate the same named vswitches using the new pNIC. This method allows guests to use the new pNIC through the old vswitch names without modifying the guests.

11.11.2.3 Deleting devices

A device entry can be deleted only if it has no dependents. If a device has dependents, those dependents must be removed before you delete the device. The `hpvmmodify` command that removes a device removes that guest as a dependent on that device.

If the guest cannot be modified, you can use the `hpvmdevmgmt -d` command to delete a dependent from a device. However, this command does not modify the guest that is dependent on the device. Use this method only if you can use the `hpvmmodify` command on the guests that are dependent on the device. The following example shows how to remove a guest as a dependent:

```
# hpvmdevmgmt -d gdev:entry_name:depend:depend_name
```

11.11.2.4 Restricting VSP devices

You must set up restricted devices to ensure that no guest uses devices that are reserved for use by the VSP, including the storage devices that the VSP uses to boot and run. This can also include a network LAN device to which the host requires exclusive access.

If a volume manager is used for host-specific file systems, then the restricted devices should include both the volume devices and the underlying special device files to protect both from guest access. For more information, see Chapter 9 (page 107).

You can also allow guests to access certain files while restricting them from accessing the device files that contain those files. You can add or delete restricted device entries to the Integrity VM device database.

For example, to add `/dev/rdisk/disk0` as a restricted device, enter the following command:

```
# hpvmdevmgmt -a rdev:/dev/rdisk/disk0
```

To delete the restricted device `/dev/rdisk/disk0`, enter the following command:

```
# hpvmdevmgmt -d rdev:/dev/rdisk/disk0
```

To add network `lan0` as a restricted device, enter the following command:

```
# hpvmdevmgmt -a rdev:lan0
```

If a guest's configuration file contains restricted devices, the guest does not start.

11.11.3 Inspect and edit the repair script

The `hpvmdevmgmt -r` report and repair-script function might identify one or more new pathnames for disks whose old pathnames no longer exist. The repair-script performs that reassignment using the `hpvmdevmgmt -n` command.

In general, you should inspect and edit the script before running it for the following reasons:

- All replace commands, `hpvmdevmgmt –n`, in the script are commented out. You must delete only the comment characters before only one of the `hpvmdevmgmt –n` commands for a particular device. Otherwise, subsequent `hpvmdevmgmt -n` commands for the same device will fail.
- If a legacy device name is replaced with another legacy device name, both the legacy device name and the agile device name are added. However, if the agile device name is used to replace a legacy device name, only the agile device name is used.

11.12 HP AVIO Stor EFI Driver enumeration policy

The default enumeration policy of the “HP AVIO Stor EFI Driver” is to enumerate boot LUNs. Use the `drvcfg` EFI utility to change the enumeration policy to do the following:

- Enumerate boot LUNs only. (Default policy)
- Enumerate all LUNs.

The enumeration policy can be set separately for SCSI (non-NPIV) LUNs and FC (NPIV) LUNs. Setting the policy to enumerate all LUNs (especially FC LUNs) might result in long guest boot times in configurations with a large number of LUNs. The delays might be noticed in the following cases:

- The EFI Boot Manager menu screen takes a long time to present itself.
- When entering the EFI shell, a long delay might occur before the device mappings are displayed and the EFI shell prompt is presented.

The following example shows the policy configuration dialog. In this example, the policy is being unchanged from the default policy.

```
Shell> drvcfg -s
HP AVIO Stor Driver Configuration
=================================
Warning: enumerating all SCSI or FC LUNs increases initialization times.

Enumerate all SCSI LUNs (Y/N)? [current setting: N]: N

Enumerate all FC LUNs (Y/N)? [current setting: N]: N


  Drv[2F]  Ctrl[ALL]  Lang[eng] - Options set.  Action Required is None
None
None

Shell>

Reset the guest for the change to take effect

   vMP MAIN MENU

         CO: Console
         CM: Command Menu
         CL: Console Log
         SL: Show Event Logs
         VM: Virtual Machine Menu
         HE: Main Help Menu
          X: Exit Connection

[g1] vMP> CM

        (Use Ctrl-B to return to vMP main menu.)

[g1] vMP:CM> RS

At next boot only boot LUN will be enumerated

    Use ^ and v to change option(s). Use Enter to select an option
Loading.: EFI Shell [Built-in]
EFI Shell version 1.10 [14.62]onsole - - - - - - - - - - -
Device mapping table
  fs0  : Acpi(PNP0A03,0)/Pci(0|0)/Scsi(Pun0,Lun0)/HD(Part1,SigBEC59C34-E6C8-11DB-8002-D6217B60E588)
```

```
  fs1  : Acpi(PNP0A03,0)/Pci(0|0)/Scsi(Pun0,Lun0)/HD(Part3,SigBEC59C70-E6C8-11DB-8004-D6217B60E588)
  blk0 : Acpi(PNP0A03,0)/Pci(0|0)/Scsi(Pun0,Lun0)
  blk1 : Acpi(PNP0A03,0)/Pci(0|0)/Scsi(Pun0,Lun0)/HD(Part1,SigBEC59C34-E6C8-11DB-8002-D6217B60E588)
  blk2 : Acpi(PNP0A03,0)/Pci(0|0)/Scsi(Pun0,Lun0)/HD(Part2,SigBEC59C52-E6C8-11DB-8003-D6217B60E588)
  blk3 : Acpi(PNP0A03,0)/Pci(0|0)/Scsi(Pun0,Lun0)/HD(Part3,SigBEC59C70-E6C8-11DB-8004-D6217B60E588)
startup.nsh> echo -off

  setting hpux path(\EFI\HPUX)...
  type 'fs[x]:' where x is your bootdisk (0, 1, 2...)
  type 'hpux' to start hpux bootloader
```

12 Migrating virtual machines and vPars

There are several different forms of migration. With the `hpvmmigrate` command, you can move either an offline virtual partition or virtual machine, or a live, online virtual machine running a guest operating system and applications from a source VSP system to a target VSP system.

This chapter includes the following topics:

- Introduction to vPars and virtual machine migration
- Command line interface for online and offline migration
- VSP, vPars, and virtual machine configuration considerations
- Migrating from physical to virtual machines

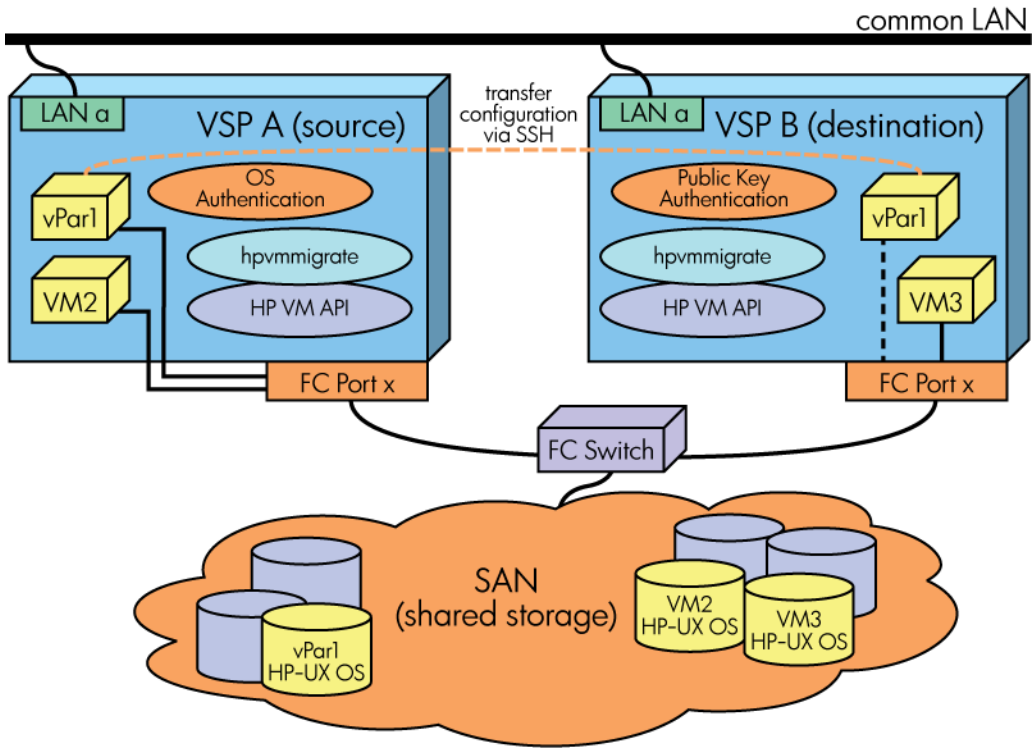
12.1 Introduction to migration

Virtual machines and virtual partitions can be moved from one VSP system to another in a variety of ways. vPars and Integrity VM V6.1 provides the following types of migration:

- To move the VM or vPar from one VSP system to another, use the `hpvmmigrate` command. (Note, the vPar can be moved offline only from one BL8x0c i2 to another or one Superdome 2 system to another.) The VM can be either a non-running VM guest or vPar configuration (offline migration) or a running VM guest (online migration). Online migration enables a running VM and its applications to be moved from one VSP to another without service interruption. All VM I/O connections to storage and networks remain active throughout the online migration, and the VM and all its applications continue operating without a reboot or application restart.
- To migrate a Serviceguard Packaged VM online, use the `cmmovepkg` command. For more information, see the `cmmovepkg(1M)` manpage or the *HP Serviceguard Toolkit for Integrity Virtual Servers User Guide* at HP Serviceguard Toolkit for Integrity Virtual Servers.

Figure 12 illustrates the process of moving a guest from Host A to Host B offline.

Figure 12 Symmetric VSPs Configured for Guest Migration



The basic virtual machine or vPar migration environment includes a source machine and a target machine. Both must be running vPars and Integrity VM V6.1 and must be able to run the guests. Both machines must conform to their operating system requirements and restrictions, and both must be able to provide the allocated resources to the guest. If the guest uses 2 GB of memory on one machine, it must be able to use that amount on the other machine. Similarly, if the source machine can provide a guest with four vCPUs, the target machine must also be able to provide them. To modify the virtual devices or network on the target host, use the `hpvmmodify` command.

To enable migration, all resources used by the guest must be configured symmetrically on both the source and target host. A symmetric configuration includes:

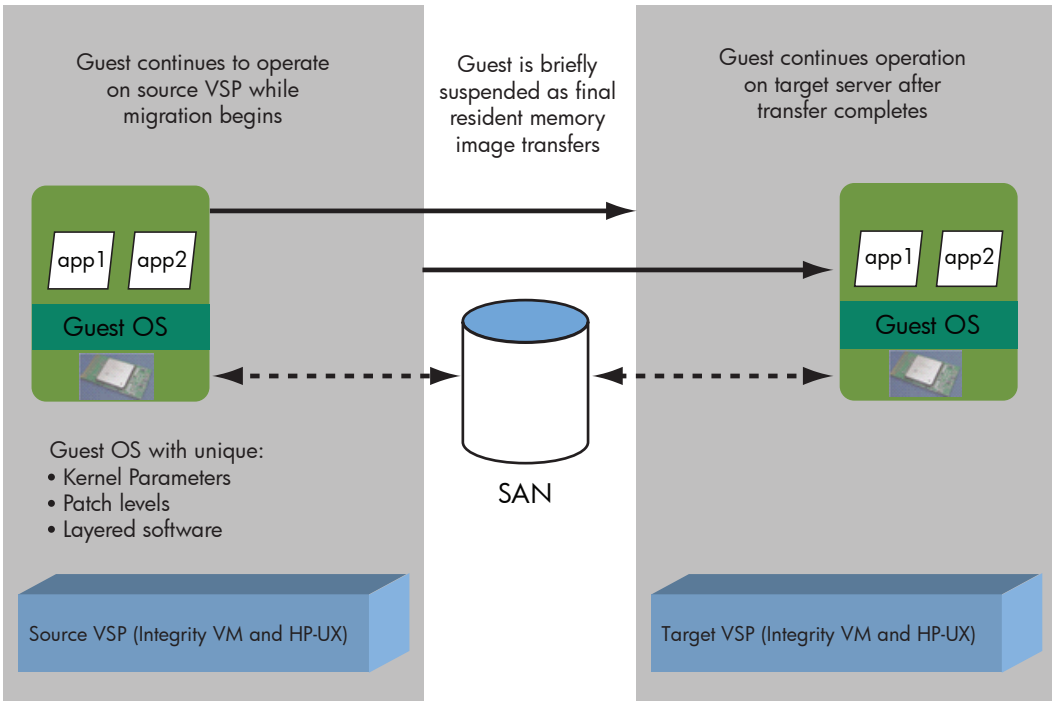
- A common local area network (LAN)
- Identical subnet and vswitch connectivity
- Common access for Storage Area Network (SAN) based storage
- Private, high-speed network connection (for Online VM Migration)

For guidelines about setting up storage for migrating virtual machines, see Section 12.3 (page 205).

If the HP Capacity Advisor is used on the virtual machine, collect utilization information before you migrate the virtual machine. The Capacity Advisor cannot continue to collect the utilization information for the virtual machine during the migration operation.

Figure 13 (page 196) illustrates moving a guest online from a source VSP to a target VSP.

Figure 13 Online VM Migration from Source to Target



12.1.1 Reasons for migrating an online VM

There are various reasons why you might want to migrate an online VM. Most can be summarized into four categories:

- Vacating a VSP system
- Targeting a particular VSP

- Balancing VSP workloads
- Optimizing physical resource utilization

With Online VM Migration, you can migrate all your VMs off of a VSP to one or more other VSPs without interrupting the workload activity on the virtual machine. A common reason to do this is for maintenance of the VSP system: hardware, firmware or software. You can configure the hardware that does not have hot-plug support. You can update the firmware, which requires the system to be shut down. You can also update software components that require a VSP reboot. A rolling upgrade of VSP software is possible by moving the running guests to another VSP, upgrading the VSP, then migrating the guests back. Being able to move VMs while keeping active applications online allows greater flexibility in scheduling maintenance or upgrades, and minimizes the impact of unpredictable maintenance. For example, you can move online VMs in response to predictive failure alerts without interrupting your applications.

You might also want to migrate an active VM workload to a particular VSP to take advantage of a particular resource or feature on that target VSP without losing application availability. If your current VSP resources become oversubscribed, you can migrate one or more of the VMs to other VSPs that have remaining capacity. Perhaps a potential target VSP has a large quantity of RAM, CPUs or I/O adapters, which might facilitate faster processing or greater I/O bandwidth while on that VSP. Another possibility is that certain VSPs have special devices that are needed only temporarily by VM workloads. Because Online VM Migration enables VMs to be migrated without interrupting their workloads, it is convenient and practical to migrate VMs temporarily to certain VSPs to take advantage of their particular resources and features when they are needed. This is especially true for workloads with well-understood cyclic resource requirements (for example, month-end processing).

You might want to segregate VMs to balance the workload on VSPs workloads. For example, you might want to separate VMs whose workloads peak simultaneously. Or, perhaps you want to group workloads together that have similar special resource requirements. For example, you would usually run your multi-threaded applications on a VSP that has several CPUs in order to maximize the effectiveness of multi-way virtual machines. Online VM Migration enables a new level of workload-to-resource alignment flexibility and agility -- you can segregate or combine your workloads as you wish, without any interruption in application availability.

The Online VM Migration feature enables you to optimize the physical resources being used by running VM. You can conveniently "park" idle, near-idle, or just currently less-critical VM workloads together on a smaller or less powerful machine. You can use the dynamic memory feature to reduce the amount of memory in use by the VMs and shrink CPU entitlements to more tightly packed VMs on a smaller VSP.

Table 31 provides the supported online migration paths for HP-UX guests:

Table 31 Online Forward Migration Paths

Integrity VM Version	Supported Forward Migration Path
Integrity VM V4.1	Integrity VM V4.1 or Integrity VM V4.2
Integrity VM V 4.2	Integrity VM V4.2, Integrity VM V4.2.5, or Integrity VM4.3, and Integrity VM 6.1
Integrity VM V4.2.5	Integrity VM V4.2.5 or Integrity VM V4.3, and Integrity VM 6.1
Integrity VM V4.3	Integrity VM V6.1

Online migration among Integrity servers is limited by the processor architecture. Online migration among servers with processor family 31 is supported regardless of the model number within the family. Migration among servers with processor family 32 and model numbers 0 or 1 is supported. Otherwise, online migration is supported among servers with identical processor family and model number.

To check if a guest can be migrated to the target VSP, use the `hpvmmigrate -s` option.

12.1.2 Reasons for migrating virtual machines or vPars offline

This sections lists reasons why you might want to migrate a virtual machine or vPar offline. For example:

- The vPar or VM might be stopped, so you need to move the configuration information offline.
- Migrating the virtual machine or vPar offline does not use the VSP resources (like memory and CPUs) on the source and target VSPs. (You can migrate vPars offline only.)
- The vPar or VM might have local storage, logical volumes or file-backed storage, which must be copied to the target VSP.
- The source and target VSPs might have different processor types that prevent online migration.
- The source VSP might be running a version of Integrity VM prior to Version 6.1, which does not support Online VM Migration.
- You can offline migrate vPars or VMs between different processor families.

Table 32 provides the migration path for offline migration:

Table 32 Offline Migration Paths

vPars and Integrity VM Version	Supported Offline Migration Path (Forward or Backward)
Integrity VM V3.5	Integrity VM V3.5
Integrity VM V4.0	Integrity VMV4.0
Integrity VM V4.1 or later	Integrity VM V4.1 or later
vPars V6.0	vPars V6.1

Offline migration of a vPar or VM with DIO functions assigned to it requires that each function is assigned a label using the `hpvmhwmgmt -L label` switch. (See `hpvmhwmgmt(1M)` for the command syntax.) Additionally, there must be at least one DIO-capable function with the same label on the target VSP for each DIO-capable function on the vPar or VM on the source VSP.

A label may contain up to 256 alphanumeric characters, including A-Z, a-z, 0-9, the dash (-), the underscore (_), and the period (.), except that it may not be the string "." or "..". Labels only apply to DIO functions that have been added to the DIO pool on the source and target VSPs with:

`hpvmhwmgmt -p dio -a` *`hwpath`*

If any DIO function in a vPar or VM has no label, offline migration will fail. There may be more functions with the same label available on the target VSP than are needed to do a one-to-one matching of DIO-functions on the target VSP, but there must be at least a one-to-one correspondence between each labeled function on the source vPar or VM and available DIO-capable functions on the target VSP.

HP recommends that labels be assigned to correspond to IP names, so that the network mapping on the source vPar or VM is preserved when the vPar/VM is migrated to the target VSP. That is not a requirement for offline migration to succeed, but failure to maintain the one-to-one correspondence with IP names might cause problems when the migrated vPar or VM is started.

Label-matching is independent of whether the labels are assigned to DLA or FLA functions. See “Using direct I/O networking” (page 19) for an explanation of DLA and FLA distinction. However, offline migration will attempt to do an exact match of like-for- like function types first.

12.2 Command line interface for online and offline migration

To migrate a virtual machine to another VSP, perform the following steps:

1. Set up SSH keys on both the source and target hosts, as described in Section 12.3.3 (page 210).
2. Present all SAN storage assigned to the virtual machine to the target VSP (if it is not already there).
3. If using offline migration and the guest is booted, stop the guest on the source host, using the `hpvmstop` or `hpvmconsole` command. You can also use the `hpvmmigrate -d` command to stop the guest during the migration. This has an advantage in that resource checks are made on the target before the guest is stopped on the source. However, for many cases, it is actually best to log into the guest and shut it down before starting an offline migration. This ensures that all guest data is properly flushed to the disks.

   For information about starting and stopping guests, see Chapter 11 (page 165).
4. On the source host, enter the `hpvmmigrate` command, as described in Section 12.2.1 (page 199). When migrating an online guest, there are several reasons why the migration might abort, leaving the guest running on the source host. Causes might include: insufficient resources on the target host, excessively busy VSPs, a slow network connection, or an extremely busy guest. If conditions like this exist, the attempted migration is aborted so the guest's workload can continue running on the source host. This is not a serious problem, because the migration can be re-attempted when conditions improve.
5. If migrating the guest offline, restart the guest on the target host using the `hpvmstart` or `hpvmconsole` command. You can also use the `hpvmmigrate -b` option with an offline migration to automatically restart the guest on the target.
6. If you do not use the `hpvmmigrate -D` option to remove the virtual machine configuration on the source VSP, it is marked `Not Runnable`, and it is configured with all its devices. This protects the storage from unintended use by Integrity VM commands.

   If you never intend to migrate the guest back to the source VSP, you can remove the virtual machine configuration with the `hpvmremove` command. Once the guest is removed from the VSP, you should unpresent the guest's SAN storage and remove the associated device special files (using the `rmsf` command). Or, if you cannot unpresent the storage, you should use the `hpvmdevmgmt -a rdev:`*`device`* command for each device to mark them restricted.

The `hpvmmigrate` command verifies that the target host has sufficient resources (such as memory, network switches and storage devices) for the guest to run. If the resources are insufficient or do not exist, or if other errors occur, the guest is not migrated to the target host.

After successfully migrating the guest, the `hpvmmigrate` command automatically disables the guest on the source host.

12.2.1 Using the `hpvmmigrate` command

Use the `hpvmmigrate` command to move an online guest or an offline virtual machine or vPar from a source VSP to a specified target VSP. Virtual machines and vPars can be migrated while OFF, and online guests can be migrated while ON and running. Use the `-o` option with virtual machines to migrate an online guest, which involves copying all the virtual machine's configuration information and transferring the active guest memory and virtual CPU state. Omit the `-o` option to migrate just the offline virtual machine's or vPar's configuration information, and optionally local disk contents, to the target VSP.

The resources that are defined in the virtual machine's configuration information are checked to determine whether the migrated virtual machine can boot on the target VSP. If there is a problem,

it is reported, and the virtual machine is not migrated. You can specify the `-F` option (force) to suppress the errors and force the virtual machine migration to the target VSP.

△ **CAUTION:** Use the `-F` option with caution, because some errors can prevent a virtual machine from working properly on the target VSP.

The `-F` option is deprecated in Integrity VM commands; this option should be used only at the direction of HP Support.

By default, Integrity VM retains the virtual machine configuration and marks it Not Runnable (NR) on the source VSP after it is migrated successfully to the target VSP. Run the `hpvmstatus` command to see that the state of the virtual machine is now `Off(NR)` on the source VSP and the guest is `On(OS)` on the target VSP. The guest is running on the target VSP and is, therefore, considered `Runnable`.

This mechanism allows the same virtual machine to be configured on multiple VSPs, while still preventing accidental booting of the same guest on multiple hosts simultaneously. Each virtual machine must be `Runnable` on only one VSP at any given time to prevent the possibility of two virtual machines using the same SAN storage at the same time. Use the `hpvmmodify` command, if necessary, to mark the virtual machine `Runnable` on only the VSP, and `Not Runnable` on all other hosts that know about that virtual machine configuration information.

NOTE: Mark a migrated virtual machine as `Runnable` only in rare circumstances and with great care. Inappropriate use can cause disk corruption.

When you invoke the `hpvmmigrate` command, you must specify the name of the guest to be migrated and the target VSP system.

Specify the guest using one of the following options:

- `-P` *`source-vm-name`* to specify the guest name
- `-p` *`source-vm_number`* to specify the virtual machine number

Specify the target host by including the `–h` option and specifying one of the following:

- Target host alias for the private, high-speed network connection
- Target host IP address of the private, high-speed network connection

---

NOTE: If you migrate a virtual machine or vPar that is being managed by Matrix OE, use Capacity Advisor to collect utilization data before you migrate the virtual machine. Otherwise, the utilization information for the VSP prior to the migration is lost.

Table 33 lists the options to the `hpvmmigrate` command.

NOTE: Online migration is supported only with VMs. You can use only offline migration with vPars.

Table 33 Options to the `hpvmmigrate` Command

Option	Description
`-A`	Attempts to abort an online vm migration.
`-b`	For offline migrations, causes `hpvmmigrate` command to boot the virtual machine on the target automatically after the migration process is complete. If the `-b` option is specified for an offline migration, all backing stores must be copied.
`-c` *`number-vcpus`*	For offline migrations, specifies the number of virtual CPUs for which this virtual machine will be configured on the target.

Table 33 Options to the `hpvmmigrate` Command *(continued)*

Option	Description
`-C`	For offline migrations, physically copies the storage device specified with the `-m` option to the target VSP during the migration process. If specified before the first `-m` option, it applies to all `-m` options that specify an appropriate type of storage. This might take a long time to complete if a large amount of storage needs to be copied.
`-d`	For offline migrations, causes `hpvmmigrate` to automatically shut down a running guest before migrating the virtual machine configuration to the target VSP. Consider migrating the guest online by using the `-o` option instead.
`-D`	Deletes the virtual machine from the source VSP after migrating the virtual machine to the target VSP system. If not specified, the virtual machine is marked `Not Runnable` on the source VSP after migration.
`-e` [*:max-percent*]	For offline migrations, specifies the percentage of CPU resources to which each of the virtual machines virtual CPUs is entitled. During peak system CPU load, the entitlement is the guaranteed minimum allocation of CPU resources for this virtual machine. The percent can be set to an integral value between 0 and 100. If the value specified is less than 5, the virtual machine is allocated the minimum percentage of 5%. The default is 10%. Integrity VM reserves processing power for essential system functions such as logging, networking, and file system daemons. The `-e` and the `-E` options are mutually exclusive.
`-E` [*:max-cycles*]	For offline migrations, specifies the virtual machine's CPU entitlement in CPU cycles. The cycles are expressed as an integer followed by one of these units: • M (megahertz) • G (gigahertz) If no letter is specified, the default unit is megahertz. The `-e` and the `-E` options are mutually exclusive.
`-F`	Forces the migration of a virtual machine, whether or not there are resource validation errors (such as resource conflict, resource nonexistence, and so forth). Use the `-F` option rarely and with caution. This option ignores all resource validation errors, including oversubscribing of resources. **NOTE:** These errors can prevent the virtual machine from booting on the target VSP. Any validation errors are logged in the Integrity VM command log. The `-F` option is deprecated in Integrity VM commands; this option should be used only at the direction of HP Support
`-h` *target-host-alias-or-IP-address*	Specifies the host alias or IP address of the target VSP machine to which the virtual machine is being migrated. The target machine must be a valid VSP and must be accessible by the source VSP. Almost all forms of the `hpvmmigrate` command require the `-h` option. For online migration, the parameter for the `-h` option should specify a private, dedicated, high-speed network link to the target VSP. If you specify a simple non-qualified host name, the `hpvmmigrate` command appends `-hpvm-migr` to the name and checks if a host alias has been defined for a private network corresponding to the simple name. Online guest migration does not check to ensure the link is private, but using a private network is important for efficient and secure online migrations and to preserve the bandwidth of the regular site network.
`-H`	Displays information about how to use the `hpvmmigrate` command.

Table 33 Options to the `hpvmmigrate` Command *(continued)*

Option	Description
`-k`	Creates the virtual machine configuration on the target VSP and marks it `Not Runnable`, but does not change the virtual machine on the source VSP. This is used primarily to distribute virtual machine configurations for Serviceguard.
`-l` *`new-vm-label`*	Specifies a descriptive label for the virtual machine, which can be useful in identifying a specific virtual machine in the `hpvmstatus` command verbose display. The label can contain up to 256 alphanumeric characters, including A-Z, a-z, 0-9, the dash (-), the underscore (_), and the period (.). To specify white space, the label must be quoted (" ").
`-m` *`rsrc-with-absolute-path`*	For offline migrations, specifies a resource of a virtual machine for copying, translation, and so on. This option can be specified more than once. for information about specifying virtual machine storage and network resources, see `hpvmresources(5)`.
`-n`	Quits after starting the migration in the background. If not specified, the `hpvmmigrate` command continues to run interactively and reports the migration status until the migration is complete.
`-N` *`new-vm-name`*	Specifies the new name for the virtual machine being migrated. The *`new-vm-name`* can be up to 256 alphanumeric characters, including A-Z, a-z, 0–9, the dash (-), the underscore character (_), and the period (.).The virtual machine name must not start with a dash (-). If the virtual machine name exists on the target VSP, the virtual machine must have the same UUID as the source virtual machine, and the virtual machine on the target must be marked `Not Runnable`.
`-o`	Specifies an online guest migration. To be compatible for online migrations, both the source and the target VSP must have the same processor family (as reported by the `machinfo` command). To maintain online guest network connectivity, a vswitch with the same name and connected to the same subnet must be configured on the target VSP. Also, only whole disk backing storage consisting of SAN LUNs, and null backing store DVD devices, are supported for online migration guest storage.
`-p` *`source-vm-number`*	Specifies the unique number of the virtual machine to be migrated. To display the *`source-vm-number`*, enter the `hpvmstatus` command. Most forms of the `hpvmmigrate` command require either the `-p` option or the `-P` option.
`-P` *`source-vm-name`*	Specifies the unique name of the virtual machine to be migrated. Most forms of the `hpvmmigrate` command require either the `-p` option or the `-P` option.
`-q`	Displays fewer informative messages. Some potential error conditions are still reported.
`-Q`	For online migrations, set non-interactive mode. Assume that the output device is not a terminal.
`-r` *`amount`*	For offline migration, specifies the amount of memory available to this virtual machine. The size is expressed as an integer, optionally followed by one of these letters: • M (megabytes) • G (gigabytes) If the letter is omitted, the default unit is megabytes.

Table 33 Options to the `hpvmmigrate` Command *(continued)*

Option	Description
`-s`	Indicates that the migration should not occur, but the `hpvmmigrate` command should check whether or not the migration is possible. Because virtual machines and their hosts are dynamic, a successful `-s` trial does not always guarantee a subsequent successful migration. The `hpvmmigrate` command with the `-o-s` and `-h` options (but without a `-p` or `-P` option) checks host connectivity, licensing, and CPU compatibility for online migration.
`-t`	For offline migration, translates the storage device names specified with the `-m` option by comparing WWIDs. To compare WWIDs, the storage resources must be present and available on both the source and the target VSPs. If you specify the `-t` option before the first `-m` option, the `-t` option applies to all `-m` options. The `-t` option overrides the `-T` option for storage resources specified with the -m option. Device translation is automatic for online migration.
`-T`	For offline migration, specifies not to translate devices.
`-v`	Displays the version of the `hpvmmigrate` command .
`-w`	For online migrations, bypasses all vswitch connectivity checks. Use the `-w` option only if you are certain that the source and target vswitches are connected to the same subnet; otherwise, your online guest will lose network connectivity after migrating.
`-Y`	Suppresses encyrption negotiations and sends guest memory data in the clear.
`-y`	Requires encryption negotiation and sends guest memory data with protection.

NOTE: You can online migrate VMs that are using logical volume backing stores, as long as you follow the configuration steps listed in Section 11.5 (page 200).

NOTE: Before enabling the guest on the source, check the target to ensure that the guest was not actually migrated there.

It is rare but possible that a guest is marked Not Runnable after a failed offline migration. If this occurs, use the following command to return the guest to the registered state:

```
# hpvmmodify -P guestname -x register_status=enabled
```

12.2.2 Examples of the `hpvmmigrate` command

The following example displays the version number of the `hpvmmigrate` command:

```
# hpvmmigrate -v
hpvmmigrate: Version B.06.10.00
```

Offline Migration Example

The following example shows how to migrate the guest named VPAR1, residing on the host named HostA, to the target host (HostB). On the system named HostA, enter the following command:

```
# hpvmmigrate -P VPAR1 -h HostB
```

This example specifies:

- The name of the vPar (`-P VPAR1`)
- The name of the target host (`-h HostB`)

Online Migration Example

The Online VM Migration feature is initiated with the `-o` option to the `hpvmmigrate` command. The following example migrates a guest to another VSP. The guest name is vm3. The target VSP

is called host2, and the target VSP's private network is called `host2-hpvm-migr` (that is, `host2-hpvm-migr` is an alias for the private network defined in `/etc/hosts`).

NOTE: The `hpvmmigrate` command does not check that you are using a private network to migrate your guest. Using a private network is important for security and to maintain the performance of your site's public network.

To migrate guest `vm3` to VSP `host2`, issue the following command:

```
# hpvmmigrate -o -P vm3 -h host2-hpvm-migr
```

The `hpvmmigrate` command displays status as various phases of migration completion. Output messages that are indented from the left margin are from the remote, target VSP.

To prevent data corruption on your guest's SAN storage, the Integrity VM software helps to prevent you from accidentally running the same guest on more than one VSP simultaneously. If the `hpvmmigrate -D` option is not specified, the guest is marked `Not Runnable (NR)` on the source VSP after online migration has finished. This prevents the virtual machine from booting on the original source VSP while it is running on the target VSP. If the `hpvmmigrate -D` option is used, unpresent the guest's SAN storage from the source VSP as soon as migration completes, thus avoiding accidental usage of the storage on that VSP.

12.2.3 Using the `hpvmstatus` command to see migration details

Use the `hpvmstatus` command to see the current state of all virtual machines on this VSP. Several states are related to Online VM Migration:

- `On (OS)` — The guest is on and running the guest operating system. It is considered `Runnable`.
- `Off (NR)` — The virtual machine is not booted and is `Not Runnable`.
- `On (MGS)` — The guest is on and running a guest operating system. It is the source of an online migration to another VSP.
- `On (MGT)` — The virtual machine is on, but not yet running a guest operating system. It is the target of an online migration from another VSP.

Use the `hpvmstatus -P` and `-V` options to get more detailed migration status about a particular virtual machine. If the guest is actively migrating, the `hpvmstatus` command shows the phase information about Online VM Migration phases.

12.2.4 Options to `hpvmmodify` command for online migration

Use the `hpvmmodify -x` option to change the online migration phase timeout values. See Section 12.2.4 (page 204) for a list of time-out phases.

Use the `hpvmmodify -x online_migration=disabled` option to prevent a particular virtual machine from migrating online. This is especially important if the guest is running software that is sensitive to external network monitoring with short timing intervals, such as Serviceguard.

NOTE: The Online VM Migration feature is not supported for a guest running as a Serviceguard node. Therefore, disable online migration for all guests that are Serviceguard nodes. For example:

```
# hpvmmodify -P sg_node1 -x online_migration=disabled
```

A transient network error might cause the `hpvmmigrate` command's vswitch connectivity check to report a failure. If the connectivity check fails, retry the migration by re-issuing the `hpvmmigrate` command.

If the `hpvmmigrate` command's network connectivity check continues to fail, verify the vswitch and network configuration, and test connectivity with the `nwmgr` command.

If the vswitch connectivity required by the guest on the target VSP is properly configured and verified, you can use the `hpvmmigrate -w` option to bypass vswitch connectivity checks.

The Online VM Migration feature is supported with Serviceguard packaged guests. For more details, see the *HP Serviceguard Toolkit for Integrity Virtual Servers User Guide* at HP Serviceguard Toolkit for Integrity Virtual Servers.

12.2.5 Using the `hpvminfo` command in the guest

The `hpvminfo` command is part of the Integrity VM guest kit and should be installed on all of your guests. With V6.1, if VirtualBase B.06.10 is installed on the guest, the guest kit does not need to be installed. Use the `hpvminfo -V` option to display information about the guest and the current VSP.

The following is a shell script using the `hpvminfo -M` option (for machine-readable output) that you can run on any Unix guest to show when an online migration has occurred. The script gets the guest name (G), and the current host (H1), and then begins an infinite loop testing and reporting if the host on which it is running has changed. Terminate the shell script with a ^C.

```
G=$(hpvminfo -M | awk -F : '{print $12;}')
H1=$(hpvminfo -M | awk -F : '{print $7;}')
echo $(date) $G: Current host is $H1
while true
do
   H2=$(hpvminfo -M | awk -F : '{print $7;}')
   if [ "$H1" != "$H2" ]; then H1=$H2; echo $(date) $G: host is now $H2; fi
done
```

The following is a sample output from this script:

```
Tue Aug 26 10:52:39 PDT 2008 vm6: Current host is host2
Tue Aug 26 10:53:36 PDT 2008 vm6: host is now host1
Tue Aug 26 10:54:28 PDT 2008 vm6: host is now host2
Tue Aug 26 10:55:19 PDT 2008 vm6: host is now host1
```

12.3 VSP and virtual machine configuration considerations

This section discusses the configuration information you need for a successful migration and how to chose which hosts and guests can participate in Online VM Migration. Effective migration of online guests among VSPs depends on proper configuration of the networks and storage connected to the VSP and used by the online guests. The `hpvmmigrate` command verifies that the source and target hosts provide the guest with symmetric accessibility to network and storage resources. If you set up the configuration properly on both hosts before you migrate the guest, the migration task is much easier and faster.

To migrate guests among a group of VSP servers, the VSPs require common access to storage devices, networks and virtual switch configurations. Pathnames to storage need not be identical, however the same LUNs assigned to a guest must be presented to both the source and the target VSPs. There must be equivalent access to guest storage and equivalent network reachability on both the source and the target VSPs. The network on the target VSP must be able to make all the same network connections that can be used by the guest on the source VSP.

A vswitch of the same name, connected to the same network must be available on the source and target VSP servers. The `hpvmmigrate` command does connectivity checking before migration. You can use the `hpvmmigrate -w` option to bypass the vswitch connectivity checks, but only use `-w` if you are certain that the source and target vswitches are connected to the same subnet. Otherwise, your guest will lose network connectivity after migrating.

For online migration, in addition to sharing the same LAN segment for normal guest connectivity, the VSPs should be connected with a private 1 GbE (or faster) network for efficient VSP–to-VSP communications and for secure guest memory transfer. Using NTP for time synchronization is strongly recommended on all VSPs and guests to maintain consistent time accuracy.

12.3.1 Using Network Time Protocol (NTP) in Integrity VM environments

Using NTP in Integrity VM environments is recommended to keep time-of-day clocks in sync and correct. Use `xntpd` on HP-UX to synchronize time use NTP.

NTP Configuration on a VSP

On each VSP, NTP should be configured just as it would be on any typical (non-virtual) system. In `/etc/ntp.conf`, specify a drift file and one or more high quality time servers:

```
driftfile /etc/ntp.drift
server <A-HIGH-QUALITY-TIME-SERVER> prefer # a preferred time source
server <ANOTHER-HIGH-QUALITY-TIME-SERVER>  # a backup time source
server <YET-ANOTHER-HIGH-QUALITY-TIME-SERVER>
```

The local clock should also be configured as a fall back if necessary:

```
server 127.127.1.0                  # use local clock as backup
fudge 127.127.1.0 stratum 10        # show poor quality
```

If you have a group of VSPs that you would like to synchronize, you can add "peer" references in the `/etc/ntp.conf` file for each of those associated VSPs, so they will do mutual synchronization:

```
peer <AN-ASSOCIATED-VM-HOST>
peer <ANOTHER-ASSOCIATED-VM-HOST>
peer <YET-ANOTHER-ASSOCIATED-VM-HOST>
```

After configuring the VSP's `/etc/ntp.conf` file, assuming the NTP is already enabled, (that is, the XNTPD variable in `/etc/rc.config.d/netdaemons` is set to 1, as in export XNTPD-1), you can execute `/sbin/init.d/xntpd start` to restart `xntpd` on the HP-UX VSP.

NTP Configuration on a VM Guest

Because NTP was not designed to run inside a virtual machine, using NTP on VM guests requires special configuration to be stable. Using a typical default NTP configuration on a VM guest might result in NTP instability and failure to synchronize, or in apparent lost time on the guest. To avoid these virtualization related NTP issues, each VM guest should get its time directly from the VSP. Also, VM guests should not serve time to any other systems.

You can monitor NTP status by using the `ntpq -p` command and noting the *`offset`* and the *`disp`* values. Ideally both values will be well under 100. For information about how to check NTP stability, see the *HP-UX Internet Services Administrators Guide*.

You can improve time stability on VM guests by tuning NTP to poll more frequently for time corrections. The default NTP values for the *`minpoll`* and *`maxpoll`* intervals are 6 (64 seconds) and 10 (1024 seconds) respectively. NTP adjusts the current polling interval depending on network quality and delays. A VM guest uses a virtual LAN that can cause NTP to set the polling value incorrectly. To help mitigate this issue use the *`minpoll`* and *`maxpoll`* directives in the `ntp.conf` file to change the polling intervals.

Start with *`minpoll`* at 4 (16 seconds) and `maxpoll` at 6 (64 seconds) and then reduce *`maxpoll`* towards 4 if necessary to force shorter polling intervals. HP recommends that a VM guest never be allowed to deliver time (allow guests to be a time consumers). Because a VM guest never delivers time, you do not need to configure the local clock (server 127.127.1.0) or an `ntp.drift` file. So, the `ntp.conf` file on a VM guest should be as simple as the single line:

```
server <VM-HOST-SERVER-NAME> minpoll 4 maxpoll 6
```

After configuring the guest's `/etc/ntp.conf` file, assuming NTP is already enabled (that is, the XNTPD variable in `/etc/rc.config.d/netdaemons` is set to 1, as in export XNTPD=1), you can run the following commands on an HP-UX guest to sync its time with the VSP and restart `xntpd`:

```
/sbin/init.d/xntpd stop
/usr/sbin/ntpdate -b <VM-HOST-SERVER-NAME>
/sbin/init.d/xntpd start
```

NOTE: For VM guests that are on a different subnet than the VSP, the VSP may not be the best source of time if there is another accurate time server available with less network latency. In the case of different subnets, measure latency from the guest to various time servers using the `ping` and `traceroute` commands to determine which potential time server has the least network latency. Using the VSP may be the best solution, but this depends on your local network topology and the relative network distance to alternate time servers. If using an alternate (non-VM-Host) time server appears best, it may be helpful for the alternate time server and the VSP to use each other for peer mutual time synchronization.

12.3.2 VSP requirements and setup

All the latest HP-UX patches required by Integrity VM, as well as any required Integrity VM patches for Integrity VM, should be installed. Consult the most recent HP-UX vPars and Integrity VM Release Notes, available from http://www.hp.com/go/hpux-hpvm-docs, for general vPars and Integrity VM installation details, including supported VSP operating system versions, patches, and other system requirements. Required patches are available on the http://www.itrc.hp.com website.

12.3.2.1 VSP processors for online migration

VSPs can be different Integrity server models with different numbers of processors, different I/O adapters and configurations, different amounts of memory, different firmware revisions, and so on. In particular, guests can migrate between radically different size, capacity, and power VSPs. However, for online migration, all the eligible VSP servers in a group must have equivalent architecture implementations. They must all report the same processor *family* output for the HP-UX command `machinfo -v`. Different processor frequencies and cache sizes are supported for Online VM Migration. Table 34 lists the recent Itanium processors showing different values for processor family:

Table 34 Itanium Processor Families

Family	Model	Series
31	0	Itanium 2
31	1	Itanium 2
31	2	Itanium 2
32	0	Itanium 9000
32	1	Itanium 9100
32	2	Itanium 9300

Look for identical processor `Family` as shown in the following example output from the `machinfo -v` command. (As more processors families and models are added, more specific capability requirements might be necessary.) The systems `host19` and `host20` in this example are compatible for migration, because they have the same processor `family (32)`.

```
# hostname
host19
# machinfo -v
CPU info:
 12 Intel(R) Itanium 2 9000 series processors (1.6 GHz, 24 MB)
```

```
            533 MT/s bus, CPU version C2
            24 logical processors (2 per socket)

            Vendor identification:        GenuineIntel
            Processor version info:       0x0000000020000704
                    Family 32, model 0, stepping 7
            Processor capabilities:       0x0000000000000005
                    Implements long branch
                    Implements -byte atomic operations
            . . .

# hostname
host20
# machinfo -v
CPU info:
  4 Intel(R) Itanium 2 9000 series processors (1.6 GHz, 24 MB)
            533 MT/s bus, CPU version C2
            8 logical processors (2 per socket)

            Vendor identification:        GenuineIntel
            Processor version info:       0x0000000020000704
                    Family 32, model 0, stepping 7
            Processor capabilities:       0x0000000000000005
                    Implements long branch
                    Implements -byte atomic operations
            . . .
```

12.3.2.2 Private network setup

Source and target VSP systems should be connected with a dedicated, high-speed private network. To use the private network during a migration, specify the name of the private network connection in the `hpvmmigrate -h` option. As a helpful convention, if you specify a simple non-qualified host name, the `hpvmmigrate` command appends `-hpvm-migr` to the name and checks if a host alias has been defined for a private network corresponding to the simple name. If so, that host-alias is used (that is, *`host`*`-hpvm-migr` is used instead of *`host`*.).

To set up a private network between two systems, identify which physical network interfaces are to be used for the private network. Then connect those ports to the same network switch, or cable them directly to each other with a cross-over cable if these two VSP systems are the only two systems that will migrate guests. Also, BladeSystems in the same enclosure can be connected directly together without an external switch or cable.

Assign private network IP addresses to those interfaces by editing `/etc/hosts`, `/etc/nsswitch.conf` and `/etc/rc.config.d/netconf` on each host. Private (non-routable) IP addresses in the range of 10.0.0.0 to 10.255.255.255 are good choices to use. (See the chapter on "Network Addressing in the current version of the *HP-UX LAN Administrator's Guide* for assistance with subnetworking configuration: HP-UX LAN Adminstrator's Guide).

In the following example, VSP system `host2` is using network interface lan3 as its private network to connect to VSP `host1`:

Address aliases from`/etc/hosts` on the host1 and host2 systems:

```
127.0.0.1       localhost          loopback
.17.81.141    host1           host1.alg.hp.com
.17.81.142    host2           host2.alg.hp.com
10.3.81.141     host1-hpvm-migr
10.3.81.142     host2-hpvm-migr
```

Excerpt from `/etc/nsswitch.conf` on the VSP systems:

```
hosts:    files dns
ipnodes:  files dns
```

Excerpt from `/etc/rc.config.d/netconf` on the `host2` system:

```
INTERFACE_NAME[3]=lan3
IP_ADDRESS[3]=10.3.81.142
SUBNET_MASK[3]=255.255.252.0
BROADCAST_ADDRESS[3]=""
INTERFACE_STATE[3]=""
DHCP_ENABLE[3]=0
INTERFACE_MODULES[3]=""
```

Example output from `netstat` on the `host2` VSP system:

```
# netstat -in
Name      Mtu  Network         Address         Ipkts ...
lan3      1500 10.3.80.0       10.3.81.142     1022313379 ...
lan0      1500 .17.80.0      .17.81.142    2420913 ...
lo0      32808 127.0.0.0       127.0.0.1       123762 ...
```

You can also use the `nwmgr` command to help verify the connection. The following example uses the `nwmgr` command on `host1` to get the Station Address (MAC):

```
# nwmgr

Name/          Interface Station          Sub-   Interface      Related
ClassInstance  State     Address        system   Type           Interface
============== ========= ============== ======== ============== =========
lan2           UP        0x001E0B5C0572 igelan   1000Base-SX
lan0           UP        0x001E0B5C05C0 igelan   1000Base-SX
lan1           DOWN      0x001E0B5C05C1 igelan   1000Base-SX
lan3           UP        0x001E0B5C0573 igelan   1000Base-SX
lan900         DOWN      0x000000000000 hp_apa   hp_apa
lan901         DOWN      0x000000000000 hp_apa   hp_apa
lan902         DOWN      0x000000000000 hp_apa   hp_apa
lan903         DOWN      0x000000000000 hp_apa   hp_apa
lan904         DOWN      0x000000000000 hp_apa   hp_apa
```

The following example on `host2` tests the connection to `host1`'s Station Address 0x001E0B5CO573:

```
# nwmgr --diag -A dest=0x001E0B5C0573 -c lan3
lan3: Link check succeeded.
```

Use the `ssh` and the `env` commands to check that the private network connection is working properly between two VSP systems, and that you are actually using the correct network interfaces. For example:

```
# ssh host1-hpvm-migr env | grep -i connection
SSH_CONNECTION=10.3.81.142 52215 10.3.81.141 22
```

NOTE: Because Integrity VM disables the TSO and CKO capabilities on the LAN interface's IP address (resulting in poorer than expected VM Host data-transfer performance) HP recommends that you dedicate a LAN interface solely for Online VM Migration data transfer to improve data transfer time. That is, to receive the best performance on host-to-remote data transfers on a LAN interface, do not configure a vswitch over it.

12.3.2.3 Conventions for using *target-hpvm-migr* names for private networks

If the name specified for the `hpvmmigrate -h` option is a simple `basename`, the `hpvmmigrate` command concatenates its conventional private network suffix `-hpvm-migr` to the basename and first checks if that name can be resolved. A simple `basename` is a reasonably short string with no specified domain hierarchy (for example, period (.) in the name). The simple `basename` cannot already contain the conventional suffix –`hpvm-migr` either. You should add the alias *target*-`hpvm-migr` to `/etc/hosts` that maps to the private IP network address for VSP target and modify `/etc/nsswitch.conf`, so lookups reference `/etc/host` before using DNS. (The resolution check is done by looking up the modified name with the `gethostbyname` function, so DNS is used if there is no alias in `/etc/hosts`.)

Because this is just a convention implemented local to each host, administrators can use it or not.. If this convention is configured correctly, both *target* and *target*-hpvm-migr resolve to the proper address. For example:

- hpvmmigrate -h host39 — Look up host39-hpvm-migr first, and if not found, look up host39.
- hpvmmigrate -h host39-hpvm-migr — Look up host39–hpvm-migr.
- hpvmmigrate -h host39.atl — Look up host39.atl.

Of course, *target*.fully.qualified.domain-name will not be modified.

By following this convention, defining an alias with suffix –hpvm-migr for the private network connections, you block use of the site network for online migrations in case someone accidentally specifies the target VSP's *hostname* for the hpvmmigrate -h option.

12.3.2.4 Using NTP on VSPs

Using NTP to synchronize clocks is strongly recommended for Online VM Migration environments. In addition to a typical NTP configuration, all the potential VSPs should use each other as mutual peer NTP servers to help maintain time consistency between hosts.

12.3.3 SSH setup between the VSPs

Only superusers can execute the hpvmmigrate command. The migration of a guest is controlled by a set of secure remote operations that must be enabled on both systems. The hpvmmigrate command requires HP-UX Secure Shell (SSH) to be set up on both the source and target host systems to provide a secure communication path between VSPs. SSH is installed on HP-UX systems by default. Passwords-based and host-based authentication are not supported. SSH security must be set up, so that superusers can use ssh commands between the source and target VSPs without requiring interactive passwords.

The hpvmmigrate command uses SSH public-key based authentication between the source and destination hosts. To enable secure communication between the source and target hosts, you must generate SSH keys on both systems. You need root privileges to generate and set up the SSH keys required for guest migration. The easiest way to do this is to use the secsetup script provided by Integrity VM.

Execute the following command on both the source and target hosts:

```
# /opt/hpvm/bin/secsetup -r otherhost
```

Instead of using secsetup, SSH keys can be generated manually on the systems by using the ssh-keygen command. The ssh-keygen command generates, manages, and converts authentication keys for SSH. For information about manual SSH key generation, see the ssh-keygen command HP-UX manpage.

12.3.3.1 Troubleshooting SSH key setup

If SSH is installed on both the source and the target system, you can run the ssh command on the source host to establish a connection to the target host without providing a password. This ability ensures that SSH keys are set up between the two hosts. If SSH keys are not set up properly, the hpvmmigrate command produces an error message indicating that the SSH setup needs to be checked.

If running the secsetup script does not work correctly, check the permissions on / to ensure that superusers have write permissions. For example,

```
# 11 -d /
drwxr-xr-x 20 root root 8192 Apr 29 06:25 /
```

If your VSP's root directory has different permissions than displayed in the previous example, use the chmod command to correct them.

```
# chmod 755 /
```

If a VSP is reinstalled at some point after using the `secsetup` script to configure SSH keys, you might receive warning messages from `ssh` commands about keys changed, or bad keys in your `known_hosts` file. In this case, use the `ssh-keygen -R` *`hostname`* command to remove obsolete keys from the *`known_hosts`* file, and then use the `secsetup` command again to configure new keys.

If you set up SSH security between VSPs before adding the conventional `–hpvm-migr` host alias to the `/etc/hosts` file and you do not run `secsetup` on the host-alias addresses, the `hpvmmigrate` command fails with the message, `Host key verification failed`, when it attempts to use the conventional host alias.

A workaround is to run SSH once manually (for example, `ssh -hpvm-migr` *`date`*) and enter **`yes`** to the question about whether or not you should continue. This action adds `-hpvm-migr` to the list of known hosts, and subsequent `hpvmmigrate` commands will find the proper host key.

12.3.3.2 Using a Third-Party SSH

The HP-UX native SSH is assumed. To use an incompatible SSH command with the `hpvmmigrate` command, make sure your version of SSH is set up for host-based authentication without requiring interactive passwords. Then set the SSHEXECPATH environment variable (in `/etc/rc.config.d/hpvmconf`) to invoke a command or shell script similar to the one provided in `alt_ssh_example`.

Customize `alt_ssh_example` script for use in your environment with your version of SSH to translate all the HP-UX SSH specific options to execute your alternate SSH command and to achieve similar behavior. The command, or shell script, must have permissions similar to a real `ssh` executable -- it should be writable only by the file owner. The `hpvmmigrate` command expects to use the HP-UX `ssh` command as in the following:

```
ssh -e none -o BatchMode=yes -T -x target-host-alias exec hpvmmigrate -#
```

See the `alt_ssh_example` comments for explanations of the `-e, -o, -T`, and `-x` options. With an alternate version of SSH, you might not need some of the HP-UX specific options; or, there may be different options that achieve the same effect; or, perhaps some alternate SSH configuration mechanism can be used eliminating the need for some of the HP-UX specific SSH options.

12.3.4 Virtual machine requirements and setup

Online VM Migration is supported on HP-UX 11i v2 and HP-UX 11i v3 guests. All memory sizes and virtual CPU configurations for the current version of Integrity VM are supported. As with all guest OS installations, the guest kit should be installed. With V6.1, if VirtualBase B.06.10 is installed on the guest, the guest kit does not need to be installed.

12.3.4.1 Setting online migration phase time-out values

Various things can cause and online migration to abort: insufficient resources on the target host, busy source or target hosts a slow private network connection, an excessively busy guest, and so on. When a migration aborts, the guest continues to run, unaffected, on the source VSP. Therefore, these are not serious errors. You can attempt the online migration again when the blocking conditions improve.

To protect the guest's workload, the online migration software limits the amount of time spent in each migration phase. The phases of an online migration are:

- Initialization phase — Establishes connections, various checks, starts the target guest, and so forth.
- Copy phase — Tracks writes to guest memory and copies all of guest memory.
- I/O quiesce phase — Queues new I/O requests and waits for outstanding I/O to complete.
- Frozen phase — Stops the virtual CPUs and copies modified memory and guest state.

For example, if a guest stops I/O to storage for too long, it could experience I/O errors and applications could fail or the operating system could crash. If a guest is frozen for too long, external network connections to the guest can time out and network connections can be dropped.

Network time-outs are especially troublesome for certain UDP applications that are not resilient enough to tolerate packets being delayed and dropped. If you run UDP applications that assume fast network packet turnaround, you might need to reduce the frozen phase time-out value, which might cause online migrations to abort more often. However, it will preserve the integrity of the network connections to the guest. The trade-off is that your migration might abort if conditions are not right for fast and efficient migrations.

If necessary, you can carefully adjust the following migration time outs with the `hpvmmodify -x` command:

- migrate_init_phase_timeout — Specifies the maximum number of seconds the online migration spends during the initialize phase of the migration. The default is 90 seconds.
- migrate_copy_phase_timeout — Specifies the maximum number of seconds the online migration spends during the full-copy phase. The default is infinite.
- migrate_io_quiesce_phase_timeout — Specifies the maximum number of seconds the migration spends during the quiesce phase. The default is 15 seconds.
- migrate_frozen_phase_timeout — Specifies the maximum number of seconds the migration spends during the freezing phase. The default is 60 seconds.

12.3.4.2 Migrations might time out and need to be restarted

To protect a guest's workload, the Online VM Migration feature has limits for the amount of time that a migrating guest can remain in various phases of a migration. There are several capacity and resource-related reasons an attempted online migration might time out and abort, leaving the guest running on the source host. Potential causes include:

- Insufficient resources on the target host
- Excessively busy VSPs
- A slow network connection
- An extremely busy guest

If conditions like these exist, the attempted migration is aborted , so the guest's workload can continue running on the source VSP. This is not a serious problem, because the guest continues to run on the source, and the migration can be re-attempted when conditions improve.

12.3.4.3 Guest storage device shareable attribute not propagated during online migration

The guest storage device shareable attribute is not propagated to the target VSP during an online migration. After the first guest that is configured to use the shared storage is online migrated to the target, enable the shared attribute for the device to avoid online migration failures for other guests that share the device. Use the `hpvmstatus` command to determine the device special filename of the shared device on the target and the `hpvmdevmgmt` command to mark the device shareable. For example:

```
hpvmstatus -P vm_name -d
hpvmdevmgmt -m gdev:/dev/rdisk/disknnn:attr:SHARE=YES
```

For online and offline migration, device special files (DSFs) assigned to virtual machines do not need to match on source and target VSPs. Do not physically rearrange controllers on the host systems to make the paths the same. This can lead to stale DSFs and stale entries in the Integrity VM device management database. The `hpvmmigrate` command converts from DSF on the source VSP to WWID and then DSF on the target VSP. Use `ioscan -C disk -P wwid` to see if the virtual machine's disks are presented to both VSPs. If you find stale DSFs and stale entries in your Integrity VM device management database, use the `insf -e` command and the `hpvmdevmgmt` command to repair the HP-UX VSP system.

Do not mark disks SHARE=YES for devices assigned to virtual machiness that will migrate (unless more than one virtual machine will share the storage on the same VSP). Marking a device SHARE=YES can lead to more than one virtual machine using the device at the same time and can lead to disk corruption.

12.3.4.4 Using NTP on the VM guests

Using NTP is strongly recommended for Online VM Migration environments. Each guest should include all potential VSPs as servers in its ntp.conf file so the current local VSP can be used as a time source. Whether migrating or not, guests should not be used as time servers. To maintain reliable time synchronization on a guest, it might be necessary to reduce the NTP polling interval, so the guest checks the time more frequently with the NTP server.

12.3.4.5 Marking a guest not runnable

On all VSPs that have a virtual machine configured, the virtual machine should be marked Runnable on only one VSP at a time. While migrating online guests, unexpected errors or guest resets or aborts should not cause your guest to be marked Runnable or Not Runnable incorrectly.

To verify the Runnable state of a virtual machine, use the hpvmstatus command to see that the guest is Runnable on only one VSP and Not Runnable on all other VSPs. If the Runnable state of a virtual machine is not correct on a VSP, use the hpvmmodify command to correct it.

For information about the hpvmmodifycommand and how to mark a guest Runnable or Not Runnable, see Section 12.2.4 (page 204).

To mark a guest Not Runnable, use the following command:

```
# hpvmmodify -P guestname -x runnable_status=disabled
```

To mark a guest Runnable, use the following command:

```
#hpvmmodify -P guestname -x runnable_status=enabled
```

⚠ **WARNING!** Be very careful when marking a guest Runnable when it was previously Not Runnable. Make sure this guest is Not Runnable and definitely not actually running on any other VSP.

12.3.5 Restrictions and limitations of online VM migration

Administrators should carefully configure certain aspects of VSPs and guests for online migration capability. Integration with automated workload placement, management and load balancing tools are not supported in this release. Only Integrity VM command-line interfaces are available in V6.1. More automated and more convenient management of distributed Integrity VM guests might follow in subsequent Integrity VM releases.

A dedicated high-speed network should not be on the data center, work site, company, or “public” LAN. Online migration can also swamp the network while a migration is in progress. Using the site's network for migration traffic would also create peaks of network activity that might affect network performance. Using a high-speed network is desirable to minimize guest memory transfer time and allows your guest to migrate smoothly.

Only whole disk backing storage consisting of SAN LUNs, and ejected file-backed DVDs, are supported for guest storage if you plan to migrate the guest online. File and logical volume backing storage are not supported for online guest migration.

Only one online migration to or from a VSP can be performed at a time. Also, be aware of the state of the guest while migrating it online. If the guest is in the On (EFI) state and no guest operating system is booted, the online migration fails with an error. If the guest is shutting down, restarting or crashing while migrating, the online migration aborts when the hpvmmigrate command can no longer communicate with the guest.

Online migration support among Integrity servers is limited by the processor architecture. Online migration among servers with processor family 31 is supported regardless of the model number within that family. Migration among servers with processor family 32 and model numbers 0 or 1 is supported. Otherwise, online migration is supported among servers with identical processor family and model number.

To check if a guest can be migrated to the target VSP, use the `-s` option to the `hpvmmigrate` command.

NOTE: Integrity VM supports SLVM backing storage for online migrations. For details about shared LVM (SLVM) storage, see Chapter 12 (page 195).

Veritas volumes are not supported for Online VM Migration.

13 Reporting problems with vPars and Integrity VM

Report vPars and Integrity VM defects through your support channel. Follow these instructions to collect data to submit with your problem report.

1. Run the `hpvmcollect` command to gather information about the guest before modifying any guest. Preserve the state of the VSP and the vPar/VM to best match the environment when the VSP failed.

   If multiple guests are running, run the `hpvmcollect` command for guest that was running at the time.
2. After the `hpvmcollect` archive is stored on the VSP, reboot the vPar/VM that caused the VSP to crash.
3. Run the `hpvmcollect` command on the guest again. Include this information in the `hpvmcollect` archive from the VSP.
4. Report the information through your support channel.

This chapter describes how to use the `hpvmcollect` command and how to investigate vPars and Integrity VM log files for information, including the following topics:

- "Collecting vPars and Integrity VM data"
- "Managing the size of the VMM driver log file"

13.1 Collecting vPars and Integrity VM data

You can use the `hpvmcollect` command on the VSP or on the vPar and VM to collect information that is useful in analyzing system problems. The options available for the `hpvmcollect` command on the VSP are different from those available on vPars/VMs. For information about using the `hpvmcollect` command, see one of the following sections:

- Using the `hpvmcollect` command the VSP: see Section 13.1.1 (page 215).
- Using the `hpvmcollect` command on vPars/VMss: see Section 13.1.2 (page 218).

13.1.1 Using the hpvmcollect command on the VSP

Table 35 describes the options to the `hpvmcollect` command on the VSP:

Table 35 Options to the `hpvmcollect` Command on the VSP

Option	Description
`-P` *vm-name*	Specifies the vPar/VM name, where *vm-name* is the name of the vPar or VM.
`-p` *vm-number*	Specifies the vPar/VM number, where *vm-number* is the number of the vPar orVM.
`-s` *host*	Specifies a VSP name to receive the archive, which is copied using the `scp` command. Verify that you can log in to the VSP without a password.
`-n` *crash-dump*	Specifies the number of crash dumps to copy to the archive. By default, the `hpvmcollect` command copies the latest crash dump directory (based on the bounds file). This option can be used only with the `-c` option.
`-d` *dir*	Specifies a target directory in which to create the `hpvmcollect_archive` directory.
`-b` *report-number*	Specifies the archive name with the specified label. If an archive with the same name exists, it is renamed by appending a time stamp to the original name before the new archive is created.

Table 35 Options to the `hpvmcollect` Command on the VSP *(continued)*

Option	Description
`-c`	Includes the latest crash dump directory in the archive. This option is used if the guest or the VSP fails or hangs.
`-f`	Forces an archive to be overwritten, if it exists, rather than renamed with an appended time stamp.
`-h`	Displays the help message for the `hpvmcollect` command.
`-l`	Leaves the collected information in a directory rather than in an archive file. The directory name follows the same naming convention as the archive name.
`-g`	Deletes old guest memory dump data as part of data collection.
`-a`	Selects all vPars/VMs on the VSP for inclusion in the collection. Valid only on the VSP.
`-r`*`directory`*	Specifies a remote target directory in which to store the collected archive, overriding the default of/crashes.Valid on both the VSP and the vPar/VM. The `-r` option is valid only with the `-s` option.

If the VSP hangs, generate a crash dump using the `TC` command on the VSP console. When the VSP crashes, it tries to dump a predefined set of memory pages into the crash dump area, including those that belong to Integrity VM. This is crucial to collecting a successful crash dump to analyze vPars and Integrity VM problems.

The `hpvmcollect` command is a shell script that can be run on either the VSP or the vPar/VM to gather system information, log files, Integrity VM logs, and configuration files for later analysis.

Because the `hpvmcollect` command collects generic vPars and Integrity VM and HP-UX operating system and system information, it might not collect all the information needed to analyze the source of the problem. Make sure that all the relevant information is included in the collection. For example, if the vPar/VM is running an Oracle® application, include the Oracle application log files and configuration.

By default, the `hpvmcollect` command creates a directory called `hpvmcollect_archive` in your current directory, and copies and collects all the vPars and Integrity VM and VSP information. For example, to gather information for a VM named `host1` on the VSP, enter the following command:

```
# hpvmcollect -P host1
```

This command creates a directory called `hpvmcollect_archive` in your current directory (if it does not already exist) and then collects information about the VSP crash dump. The information is then put into a `tar` file format (if there is a crash dump) or `tar.gz` file format (if there is no crash dump). Do not modify the guest configuration before running the `hpvmcollect` command.

If you do not want to archive the collection into `tar.gz` but simply want to examine the contents of the collection, use the `-l` option to leave the contents as they are.

If the VSP failed, use the `-c` option to collect crash dump files as well. Because the `-c` option collects the latest crash dump, use the `-n` option to specify a crash dump number.

Use the `-d` option to specify a different directory in which to store the `hpvmcollect_archive`.

For example, to collect information about `host1`, enter the following command:

```
# hpvmcollect -c -n 21 -d /tmp/hpvm_collect_archive -P host1
```

This command collects information about the guest called `host1` using crash dump number 21. The final archive is under `/tmp/hpvm_collect_archive` directory. The following is an example of `hpvmcollect` output on the VSP:

```
# hpvmcollect -P host1
```

```
HPVSP crash/log collection tool version B.06.10.00
Gathering info for post-mortem analysis of guest 'host1' on host

Collecting I/O configuration info ................................ OK
Collecting filesystem info ....................................... OK
Collecting system info ........................................... OK
Collecting lan info .............................................. OK
Running lanshow .................................................. NO
Collecting installed sw info ..................................... OK
Collecting command logs .......................................... OK
Collecting messages from vmm ..................................... OK
Collecting lv info ............................................... N/A
Collecting vgdisplay info ........................................ OK
Collecting vxprint info .......................................... OK
Collecting disk info ............................................. N/A
Collecting passthru disk info .................................... N/A
Collecting file backing store info ............................... N/A
Copying guest's log file ......................................... OK
Copying guest's tombstone file ................................... N/A
Copying guest's console log file ................................. OK
Copying hpvm configuration ....................................... OK
Copying hpvm control script ...................................... OK
Copying guest's config file ...................................... OK
Getting status of the guest ...................................... OK
Getting detailed status of the guest ............................. OK
Getting guest's entitlement ...................................... OK
Copying guest's config file change log ........................... OK
Copying guest VM crash image ..................................... OK
Copying host vmunix image ........................................ OK
Copying host hpvmmkimage image ................................... N/A
Copying VMM image ................................................ OK
Copying hpvmdvr image ............................................ OK
Copying hpvmntdvr image .......................................... OK
Copying NVRAM image .............................................. OK
Collecting IPMI logs ............................................. OK
Collecting crash dump ............................................ NO
Running crashinfo ................................................ NO
Collecting tombstone ............................................. NO
Collecting system message buffer ................................. OK
Collecting system syslogs ........................................ OK
Collecting measureware logs ...................................... OK

Finished with the collection

Tar archiving and compressing .................................... TGZ
Remote copying the archive  ...................................... NO

The collection is
"/tmp/host1/hpvmcollect/hpvmcollect_archive/test_Jan.28.12_095249EDT.tar.gz"
```

If the command results in an error message like the following, you are out of disk space in the current directory or in the directory you specified with the `-d` option:

```
msgcnt 10 vxfs: mesg 001: vx_nospace - /dev/vg00/lvol5 file system full(1 block extent)
Tar: end of tape
Tar: to continue, enter device/file name when ready or null string to quit.
```

Use a file system with enough free space for the archive, especially when you use the `-c` option.

Additional data collected by the `hpvmcollect` command includes log files (guest, Integrity VM, and VSP) as well as VSP system information, including output from the `ioscan`, `lanscan`, and `swlist` commands. The `hpvmcollect` command also collects information about devices used by the guest. Output from the `crashinfo` and `lanshow`commands are included, if available.

The `hpvmcollect` command records device information in the following files:

```
config/
     host.diskinfo
     host.fsinfo
     host.ioscan
     host.laninfo
     host.sysinfo
```

13.1.2 Using the hpvmcollect command on vPars/VMs

To use the hpvmcollect command on the vPar/VM, you must first install the vPar/VM VirtualBase software on the vPar/VM (if it is not already installed) as described in Section 8.3 (page 104).

Table 36 describes the options to the hpvmcollect command on the guest.

Table 36 Options to the hpvmcollect Command on Guests

Option	Description
-c	Includes the latest crash dump directory in the archive. This option is used if the vPar/VM or the VSP fails or hangs.
-f	Forces an archive to be overwritten, if it exists, rather than renamed with an appended time stamp.
-g	Deletes old vPar/VM t memory dump data as part of data collection.
-h	Displays the help message for the hpvmcollect command.
-l	Leaves the collected information in a directory rather than in an archive file. The directory name follows the same naming convention as the archive name.
-b *report-number*	Specifies the archive name with the specified label. If an archive with the same name exists, it is renamed by appending a time stamp to the original name before the new archive is created.
-d *dir*	Specifies a target directory in which to create the hpvmcollect_archive directory.
-n *crash-dump*	Specifies the number of crash dumps to copy to the archive. By default, the hpvmcollect command copies the latest crash dump directory (based on the bounds file). This option can be used only with the -c option.
-s *host*	Specifies a VSP name to receive the archive, which is copied using the scp command. Verify that you can log in to the VSP without a password.

When you use the hpvmcollect command on the vPar/VM, do not specify the vPar/VM name. By default, the vPar/VM name is used as an archive directory name. You can use the -d option to specify the archive name. The following is an example of the hpvmcollect when it is run on the VMhost1:

```
host1# hpvmcollect -c
HPVM guest crash/log collection tool version B.06.10.00
Gathering info for post-mortem analysis on guest (hostname 'host1')

Collecting I/O configuration info ................................. OK
Collecting filesystem info ........................................ OK
Collecting system info ............................................ OK
Collecting lan info ............................................... OK
Running lanshow ................................................... NO
Collecting installed sw info ...................................... OK
Collecting crash dump 1  .......................................... OK
Running crashinfo ................................................. NO
Collecting tombstone .............................................. N/A
Collecting system message buffer .................................. OK
Collecting system syslogs ......................................... OK
Collecting measureware log ........................................ N/A

Finished with the collection

Tar archiving and compressing ..................................... TAR
Remote copying the archive  ....................................... NO

The collection is
"//hpvmcollect_archive/host1_Sep.29.05_122453PST.tar"
```

13.2 Managing the size of the VMM driver log file

The monitor log file (`/var/opt/hpvm/common/hpvm_mon_log`) is limited in size to 1024 KB. When the log file grows larger than this, it is copied to a new file (`hpvm_mon_log.$time`), and an empty one is created for the new log. To allow this log file to increase to 102400 KB, include the following line in the `/etc/rc.config.d/hpvmconf` file:

```
VMMLOGSIZE=102400
```

After you make this change to the `hpvmconf` file, enter the following commands to determine the PID for the monitor log daemon and to kill it:

```
# cat /var/run/hpvmmonlogd.pid
5052
# kill -HUP 5052
```

On rare occasions, the monitor log might report warnings such as the following:

```
Warning: VCPUn not scheduled for x ms, command 0x0.
Warning: No recorder entry on VCPUn for x ms.
```

14 Support and other resources

14.1 Contacting HP

14.1.1 Before you contact HP

Be sure to have the following information available before you call contact HP:

- Technical support registration number (if applicable)
- Product serial number
- Product model name and number
- Product identification number
- Applicable error message
- Add-on boards or hardware
- Third-party hardware or software
- Operating system type and revision level

14.1.2 HP contact information

For the name of the nearest HP authorized reseller:

- See the Contact HP worldwide (in English) webpage (http://welcome.hp.com/country/us/en/wwcontact.html).

For HP technical support:

- In the United States, for contact options see the Contact HP United States webpage (http://welcome.hp.com/country/us/en/contact_us.html). To contact HP by phone:
  - Call 1-800-HP-INVENT (1-800-474-6836). This service is available 24 hours a day, 7 days a week. For continuous quality improvement, calls may be recorded or monitored.
  - If you have purchased a Care Pack (service upgrade), call 1-800-633-3600. For more information about Care Packs, see the HP website at HP Care Packs.
  - In other locations, see the Contact HP worldwide (in English) webpage (http://welcome.hp.com/country/us/en/wwcontact.html)

14.1.3 HP Insight remote support

HP strongly recommends that you install HP Insight Remote Support software to complete the installation or upgrade of your product and to enable enhanced delivery of your HP Warranty, HP Care Pack Service or HP contractual support agreement. HP Insight Remote Support supplements your monitoring, 24x7 to ensure maximum system availability by providing intelligent event diagnosis, and automatic, secure submission of hardware event notifications to HP, which will initiate a fast and accurate resolution, based on your product's service level. Notifications may be sent to your authorized HP Channel Partner for on-site service, if configured and available in your country. The software is available in two variants:

- HP Insight Remote Support Standard: This software supports server and storage devices and is optimized for environments with 1-50 servers. Ideal for customers who can benefit from proactive notification, but do not need proactive service delivery and integration with a management platform.
- HP Insight Remote Support Advanced: This software provides comprehensive remote monitoring and proactive service support for nearly all HP servers, storage, network, and SAN environments, plus selected non-HP servers that have a support obligation with HP. It is

integrated with HP Systems Insight Manager. A dedicated server is recommended to host both HP Systems Insight Manager and HP Insight Remote Support Advanced.

Details for both versions are available at:

http://www.hp.com/go/insightremotesupport

To download the software, go to Software Depot:

http://www.software.hp.com

Select Insight Remote Support from the menu on the right.

NOTE: HP recommends using Insight Remote Support on the VSP system. Information from Insight Remote Support running on virtual machines should not be used to determine the hardware state.

14.1.4 Subscription service

HP recommends that you register your product at the Subscriber's Choice for Business website: http://www.hp.com/country/us/en/contact_us.html. After registering, you will receive an email notification of product enhancements, new driver versions, firmware updates, and other product resources.

14.1.5 Documentation feedback

HP welcomes your feedback. To make comments and suggestions about product documentation, send a message to **docsfeedback@hp.com**. Include the document title and manufacturing part number. All submissions become the property of HP.

14.2 Related information

The following documents [and websites] provide related information:

Table 37 Documentation and its location

Documents	Website
HP-UX GUID Manager Administrator Guide	http://www.hp.com/go/hpux-vpars-docs and http://www.hp.com/go/insightdynamics-manuals
• *HP Integrity Virtual Server Manager User Guide* • *HP Integrity Virtual Server Manager Release Notes* **NOTE:** The HP Integrity Virtual Server Manager is the graphical user interface version of the command-line interface HP-UX vPars V6.1.	http://www.hp.com/go/matrixoe/docs and http://www.hp.com/go/insightdynamics-manuals
• *HP BladeSystem Onboard Administrator Command Line Interface User Guide Version 3.30* • *HP BladeSystem Onboard Administrator User Guide Version 3.30* • *HP BladeSystem c3000 Enclosure* (whitepaper) • *HP BladeSystem c7000 Enclosure technologies* (whitepaper)	http://www.hp.com/go/blades_enclosures-docs In the main page, click **HP BladeSystem c3000 Enclosures** or **HP BladeSystem c7000 Enclosures**.
Virtual Partitions documentation	www.hp.com/go/hpux-vpars-docs
HP-UX 11i v3 documentation	http://www.hp.com/go/hpux-core-docs In the main page, click **HP-UX 11i v3**.

14.3 Typographic conventions

This document uses the following typographical conventions:

`%`, `$`, or `#`	A percent sign represents the C shell system prompt. A dollar sign represents the system prompt for the Bourne, Korn, and POSIX shells. A number sign represents the superuser prompt.
audit(5)	A manpage. The manpage name is *audit*, and it is located in Section 5.
`Command`	A command name or qualified command phrase.
`Computer output`	Text displayed by the computer.
Ctrl+x	A key sequence. A sequence such as **Ctrl+x** indicates that you must hold down the key labeled **Ctrl** while you press another key or mouse button.
`ENVIRONMENT VARIABLE`	The name of an environment variable, for example, `PATH`.
ERROR NAME	The name of an error, usually returned in the `errno` variable.
Key	The name of a keyboard key. **Return** and **Enter** both refer to the same key.
Term	The defined use of an important word or phrase.
`User input`	Commands and other text that you type.
`Variable`	The name of a placeholder in a command, function, or other syntax display that you replace with an actual value.
[]	The contents are optional in syntax. If the contents are a list separated by `|`, you must choose one of the items.
{}	The contents are required in syntax. If the contents are a list separated by `|`, you must choose one of the items.
...	The preceding element can be repeated an arbitrary number of times.
□	Indicates the continuation of a code example.
|	Separates items in a list of choices.
WARNING	A warning calls attention to important information that if not understood or followed will result in personal injury or nonrecoverable system problems.
CAUTION	A caution calls attention to important information that if not understood or followed will result in data loss, data corruption, or damage to hardware or software.
IMPORTANT	This alert provides essential information to explain a concept or to complete a task
NOTE	A note contains additional information to emphasize or supplement important points of the main text.

A Rolling back to the previously installed version of Integrity VM

In the unlikely event that you need to roll back to a previous version of Integrity VM, this appendix provides the information needed to perform the rollback. The preferred method for rolling back to a previously installed version of Integrity VM is to restore the system image that was backed up before installing the current version of Integrity VM on the VSP. Because this is not always possible for all users the following method should work.

The VSP and guest configuration files are stored at `/var/opt/hpvm`. Because configuration files for newer versions of Integrity VM are not normally compatible for earlier versions of Integrity VM, a copy is made of the contents of `/var/opt/hpvm` to the `/var/opt/hpvm/backup` directory (except the ./guest-images and ./backups directories). If need be, it is possible to revert to the older version of Integrity VM using the backups directory and the following process:

1. Make sure you have the installation media for the version of Integrity VM that was installed before version B.06.10.
2. Stop Integrity VM (`/sbin/init.d/hpvm stop`)
3. Remove Integrity VM V6.1 software (This causes a system reboot.).

   ```
   # swremove -x autoreboot=true BB068AA VirtualBase
   ```

4. Move the `/var/opt/hpvm` area aside:

   ```
   # mv /var/opt/hpvm /var/opt/hpvm_6.1
   ```

5. Install the previously installed version of Integrity VM following the directions for installing Integrity VM in this manual for that version. This also causes a system reboot.
6. Once the system is back up, log in and stop Integrity VM (`/sbin/init.d/hpvm stop`).
7. Restore the previous Integrity VM environment:

   ```
   # cd /var/opt/hpvm_4.3/backups; tar -cpf - | cd /var/opt/hpvm; tar -xpf -
   ```

8. Start Integrity VM.

B Sample Script for Adding Multiple Devices

The following example provides a script that enables you to specify multiple storage devices at once for a guest.

```
#!/bin/ksh
# ----------------------------------------------------------------------------------------
# HP Integrity VM example script.
#
# SUMMARY:
#
#  Add disks to an Integrity VM (guest) in 'batch mode' with hpvmmodify, using AVIO.
#
# SYNOPSIS
#
# ./thisscript [-a] -P guestname -f disklistfile [-N #] [-n #] [-t #] [-qT] [-F flags]
#    or
# ./thisscript -h | -H
#
# DESCRIPTION
#
#  This is an example script of how to automate adding many disks to an
#  Integrity VM guest using hpvmmodify, adding them as AVIO storage resources,
#  adding them in 'batch mode', that is, adding multiple disks with a single
#  call to hpvmmodify. When adding many disks, adding them in 'batch mode'
#  provides a performance improvement over adding them one at a time (one
#  disk added per hpvmmodify call).
#
#  The disks to add are passed in as a filename that contains the list of
#  disks.  An example of how to generate this list is:
#
#   # hpvmhostgdev -u -l | grep /dev/rdisk > disklistfile
#
#  You may add all the disks in the disklistfile to a guest up to the supported
#  limit of 1024, or some lesser number of disks (see -N flag), starting with the
#  first disk in the disklistfile.
#
#  By default, this script adds 10 disks per hpvmmodify command.  You may
#  change the 'batch add' number with the -n flag.  The value of -n may be
#  any value between 1 and 1024.
#
#  Also by default, this script does not specify the virtual bus, device,
#  target (b,d,t) triple in the hpvmmodify resource string.  So the default
#  limit of disks that may be added is 945.  [The algorithm used by hpvmmodify
#  default b,d,t assignment imposes this limit.]
#
#  To add 946 to 1024 disks to a guest, hpvmmodify requires that the virtual
#  bus, device, target (b,d,t) triple be specified in the resource string of
#  the additional disks over 945.  This script provides an option, -t, that
#  causes the script to calculate and use explicit b,d,t values for all of
#  the disks.  The valid values for the -t option are 0 and 15-127.  See
#  below for more details on this option.
#
#  This script only adds disks to guests when you specify the -a flag.  If you
#  omit the -a flag, this script will only print the messages that show what the
#  hpvmmodify commands will be.  You may suppress the sample hpvmmodify command
#  messages with the -q flag.
#
#  This script will time the hpvmmodify command with the timex command if
#  you specify the -T command.  [Note: timex output goes to stderr.]
#
#  You may also specify other hpvmmodify command arguments by using the -F
#  option.  The options you chose should be specified as though you were
#  typing them yourself on a commandline, using "-<flag>" or "-<flag> <value>",
#  including the leading hypen ('-').  You must put the -F option value(s)
#  in double quotes for this script to include them in the hpvmmodify command..
#
#     WARNING: use the -F option at your own risk.  Also, you
#              must use -F option values that would work with
#              hpvmmodify if you were entering the command on
#              the commandline yourself.
#
# OPTIONS
#
#       -a                Add the disks (default is to only display what will be added)
#
#       -F "arg(s)"       Additional hpvmmodify options or flags (double quotes required)
#
#       -f disklistfile   File containing list of disks to add
#
#       -h                Print usage (help)
#       -H                Print usage (Help)
#
#       -N #              Number of devices to add from the disklistfile
#
#       -n #              Number of devices to add at one time (default: 10)
#
#       -P guestname      Name of Integrity VM (guest) to modify
#
```

```
#       -q                Quite mode - no display of hpvmmodify command that will run
#
#       -t targetmax      Max target value to use for -a disk:avio_stor:[b,d,targetmax]...
#                         Valid values:
#                           0       - special case: script will use full 0-127 range
#                          15...127 - script will use specified max
#                           1... 14 - not valid for this script, since 0-14 is
#                                     the normal default range for target values
#                                     if -t is not specified.
#
#       -T                Time the hpvmmodify add command with 'timex'
#
# EXAMPLES:
#
#  Add all the disks in file "disklistfile" using defaults
#  # ./thisscript -a -P guest -f disklistfile
#
#  Add all the disks in file "disklistfile" 20 disks at a time
#  # ./thisscript -a -P guest -f disklistfile -n 20
#
#  Add the first 50 disks in the disklistfile, 20 disks at a time
#  NOTE: this will result in 3 calls to hpvmmodify, to
#        add 20 disks, another 20, and then the final 10.
#  # ./thisscript -a -P guest -f disklistfile -N 50 -n 20
#
#  NOTE: all of the above examples do not specify b,d,t values in the
#        hpvmmodify resource string, so the default algorithm is used,
#        to add 15 targets, from 0...14, and then increment to the next
#        virtual adaptor (skipping 0,3).
#
#  The following examples will cause the script to calculate and use
#  explicit values for b,d,t in the hpvmodify resource string.
#
#  NOTE: Rules for specifying -t in this script:
#        0        Special case, means use 0...127
#        1...14   Invalid in this script, as this is part of the default
#                 range of 0...14
#        15...127 Use specified value as upper limit to target value before
#                 going to next virtual adaptor.
#
#  Add all disks in the file using the full range of target values 0...127:
#  # ./thisscript -a -P guest -f disklistfile -t 0
#
#  Add all disks in the file using a maximum target value of 30
#  # ./thisscript -a -P guest -f disklistfile -t 0
#
#
# ASSUMPTIONS AND LIMITATIONS
#
# - assume that the guest exists and may be modified
# - assume there are no storage devices assigned to the guest
# - assume the disks in the disklistfile are good
# - assume OK to add all disks as avio_stor
# - assume OK to add specified disks to the specified guest
# - limitation: 945 storage devices using default [b,d,t] values
# - limitation: 1024 max avio_stor storage devices
# - limitation: 127 max value for user specified target limit
# ------------------------------------------------------------------------------

#
# Script global variables
#
THISSCRIPT=$0
DFLTDISKLIMIT=945
MAXDISKCNT=1024
XNDEFAULT=10
BDT="" # default [b,d,t] setting

typeset -i BUS
typeset -i DEV
typeset -i TGT
typeset -i TGTMAX
typeset -i USERTGT
BUS=0
DEV=0
TGT=0
BUSMAX=7
DEVMAX=7
DEVSKIP=3
TGTMAX=127
USERTGT=0
WRKTGT=$TGTMAX

#
# function autobdt() - auto generates explicit b,d,t triples
#
function autobdt {
    # echo "autobdt() function not yet implemented"

    # use current BUS,DEV,TGT values

    BDT="$BUS,$DEV,$TGT"
```

```
    # setup BUS,DEV,TGT for next call
    TGT=$TGT+1
    if [ $TGT -gt $WRKTGT ]
    then
        TGT=0
        DEV=$DEV+1
    fi

    # Skip b,d of 0,3
    if [ $BUS -eq 0 ] && [ $DEV -eq $DEVSKIP ]
    then
        DEV=$DEV+1
    fi

    if [ $DEV -gt $DEVMAX ]
    then
        DEV=0
        BUS=$BUS+1
    fi

    if [ $BUS -gt $BUSMAX ]
    then
        # NOTE: should not be here, but error out just in case.
        echo "ERROR: Max supported bus value exceeded, no more room for another adaptor."
        exit 1
    fi

} # end autobdt()

#
# function usage() - prints help text
#
function usage {

echo "usage: $THISSCRIPT [[-a] [-F flags] -f disklistfile [-N #] [-n #] -P guestname [-q] [-T]] | [-H|-h]"
echo "       -a                Add the disks (default is to only display what will be added)"
echo "       -F \"arg(s)\"       Additional hpvmmodify options or flags (double quotes required)"
echo "       -f disklistfile   File containing list of disks to add"
echo "       -h                Print usage (help)"
echo "       -H                Print usage (Help)"
echo "       -N #              Number of devices to add from the disklistfile"
echo "       -n #              Number of devices to add at one time (default: $XNDEFAULT)"
echo "       -P guestname      Name of Integrity VM (guest) to modify"
echo "       -q                Quite mode - no display of hpvmmodify command that will run"
echo "       -t targetmax      Max target value to use for -a disk:avio_stor:[b,d,targetmax]..."
echo "                         Valid values:"
echo "                           0       - special case: script will use full 0-127 range"
echo "                          15...127 - script will use specified max"
echo "                           1... 14 - not valid for this script, since 0-14 is"
echo "                                     the normal default range for target values"
echo "                                     if -t is not specified."
echo "       -T                Time the hpvmmodify add command with 'timex'"

} # end usage()


#
# main() 'function'
#

# Command option verification variables
typeset -i a
typeset -i F
typeset -i f
typeset -i N
typeset -i n
typeset -i P
typeset -i q
typeset -i s
typeset -i T
typeset -i t
a=0
F=0
f=0
N=0
n=0
P=0
q=0
s=0
T=0
t=0

# Variables for cmd-line arguments
DISKLISTFILE=""
GUESTNAME=""
TIMECMD=""
FLAGS=""

typeset -i ADDFLAG
typeset -i AUTOBDT
typeset -i QUIET
typeset -i USERTGT
typeset -i USERDISKLIMIT
```

```
typeset -i XN
ADDFLAG=0
AUTOBDT=0
QUIET=0
USERDISKCNT=0
USERTGT=0
XN=$XNDEFAULT


#
# Get cmd line options
#
while getopts :aF:f:HhN:n:P:qTt: option
do
    case $option in
    a) # add flag - do actual call to hpvmmodify
        ADDFLAG=1
        a=$a+1
        ;;
    F) # hpvmmodify flags
        FLAGS=$OPTARG
        F=$F+1
        ;;
    f) # disklist file
        DISKLISTFILE=$OPTARG
        f=$f+1
        ;;
    H) # Help
        usage
        exit 0
        ;;
    h) # help
        usage
        exit 0
        ;;
    N) # number of disks to add from the disklistfile
        USERDISKCNT=$OPTARG
        N=$N+1
        ;;
    n) # number of disks to add at a time
        XN=$OPTARG
        n=$n+1
        ;;
    P) # guest name
        GUESTNAME=$OPTARG
        P=$P+1
        ;;
    q) # quiet mode
        QUIET=1
        q=$q+1
        ;;
    T) # time the add command
        TIMECMD="timex"
        T=$T+1
        ;;
    t) # target max
        USERTGT=$OPTARG
        AUTOBDT=1
        t=$t+1
        ;;
    ?) # error
        echo "ERROR: Error with option: $OPTARG (unknown option, or missing value"
        usage
        exit 1
        ;;
    esac
done

#
# Verify cmd line options
#
if [ $a -gt 1 ] || [ $F -gt 1 ] || [ $f -gt 1 ] || [ $N -gt 1 ] || [ $n -gt 1 ] || \
   [ $P -gt 1 ] || [ $q -gt 1 ] || [ $T -gt 1 ] || [ $t -gt 1 ]
then
    echo "ERROR: Duplicate arguments are not allowed."
    exit 1
fi

if [ $P -eq 0 ]
then
    echo "ERROR: '-P guestname' must be specified."
    exit 1
fi

if [ $f -eq 0 ]
then
    echo "ERROR: '-f disklistfile' must be specified."
    exit 1
fi

if [[ ! -f $DISKLISTFILE ]]
then
    echo "ERROR: Could not find disklist file: $DISKLISTFILE"
```

```
    exit 1
fi

if [ ! -s "$DISKLISTFILE" ]
then
    echo "ERROR: Disklist file: $DISKLISTFILE is a zero-length file."
    exit 1
fi

GUESTSTATUS="`hpvmstatus -P $GUESTNAME -M 2> /dev/null`"
if [ -z "$GUESTSTATUS" ]
then
    echo "ERROR: Could not find guest: $GUESTNAME"
    exit 1
fi

if [ $t -eq 1 ]
then
    if [ $USERTGT -gt 0 ] && [ $USERTGT -lt 15 ]
    then
        echo "ERROR: User specified target max (-t $USERTGT) must be 0 or in range 15...127."
        exit 1
    fi
    if [ $USERTGT -gt $TGTMAX ]
    then
        echo "ERROR: User specified target (-t $USERTGT) exceeds max value of $TGTMAX"
        exit 1
    fi

    if [ $USERTGT -ne 0 ]
    then
        WRKTGT=$USERTGT
    fi
fi

#
# Get disklist from file
#
DISKLIST="`cat $DISKLISTFILE`"

#
# Setup main loop variables
#
typeset -i DISKCNT
typeset -i FILEDISKCNT
FILEDISKCNT="`ls -1 $DISKLIST | wc -l`"
if [ $USERDISKCNT -eq 0 ]
then
    DISKCNT=$FILEDISKCNT
else
    if [ $USERDISKCNT -gt $FILEDISKCNT ]
    then
        echo "ERROR: -N value ($USERDISKCNT) is greater than number of disks in $DISKLISTFILE ($FILEDISKCNT)."

        exit 1
    else
        DISKCNT=$USERDISKCNT
    fi
fi

if [ $DISKCNT -gt $DFLTDISKLIMIT ] && [ $AUTOBDT -eq 0 ]
then
    echo "ERROR: Diskcount greater than $DFLTDISKLIMIT requires target max flag (-t) to be set."
    exit 1
fi

if [ $DISKCNT -gt $MAXDISKCNT ]
then
    DISKCNT=$MAXDISKCNT
    echo "INFO: Set diskcount to supported maximum \($MAXDISCOUNT\)."
fi

typeset -i CMDIDX
typeset -i DISKIDX
CMDIDX=0
DISKIDX=0
BASEMODCMD="hpvmmodify -P $GUESTNAME $FLAGS"

#
# Main Loop
#
if [ $ADDFLAG -eq 0 ]
then
    echo "INFO: Add flag (-a) was NOT specified (no disks will be added)."
fi

ADDCMD="$BASEMODCMD"
for DISK in $DISKLIST;
do
    if [ $AUTOBDT -eq 1 ]
    then
        autobdt
    fi
```

```
    ADDRSRC="-a disk:avio_stor:$BDT:disk:$DISK"
    ADDCMD="$ADDCMD $ADDRSRC"
    DISKIDX=$DISKIDX+1
    CMDIDX=$CMDIDX+1

    # Run hpvmmodify if at the add multiplier (-n) or at the last disk
    if [ $CMDIDX -eq $XN ] || [ $DISKIDX -eq $DISKCNT ]
    then
        # Do the hpvmmodify
        if [ $QUIET -eq 0 ]
        then
            echo "Calling: $TIMECMD $ADDCMD"
        fi

        if [ $ADDFLAG -eq 1 ] # check for -a flag
        then
            $TIMECMD $ADDCMD
            RETVAL=$?
            if [ $RETVAL -ne 0 ]
            then
                typeset -i FINALCNT
                FINALCNT=$DISKIDX-$XN
                echo "ERROR - hpvmmodify failed. (total disks added: $FINALCNT)"
                exit 1
            fi
        fi

        # In progress status ...
        echo "Subtotal of disks added: $DISKIDX"

        # Reset hpvmmodify cmd string
        ADDCMD="$BASEMODCMD"
        CMDIDX=0
    fi

    if [ $DISKIDX -eq $DISKCNT ]
    then
        # all done
 break;
    fi
done

if [ $ADDFLAG -eq 1 ]
then
    echo "All done (total disks addded: $DISKCNT)"
else
    echo "All done (Not in add mode: no disks added)"
fi
exit 0
```

Glossary

This glossary defines the terms and abbreviations as they are used in the Integrity VM product documentation.

Accelerated Virtual Input/Output See AVIO

adoptive node The cluster member where the package starts after it fails over.

APA Auto Port Aggregation. An HP-UX software product that creates link aggregates, often called "trunks," which provide a logical grouping of two or more physical ports into a single "fat pipe". This port arrangement provides more data bandwidth and higher reliability than would otherwise be available.

application A collection of processes that perform a specific function. In the context of virtual machine clusters, an application is any software running on the guest.

assignable resource The resources that you can designate to be assigned to a partition.

asymmetric Serviceguard configuration A cluster configuration in which the cluster nodes do not have access to the same physical storage and network devices.

autoboot A characteristic of a virtual machine whereby it is set to start whenever Integrity VM starts. Virtual machines can be set to either `auto` or `manual` boot using the `-B` option to the `hpvmcreate`, `hpvmmodify`, `hpvmmigrate`, or `hpvmclone` commands.

available resources Processors, memory, and I/O resources that are not assigned to a virtual machine. These resources are available to be used in new partitions or can be added to existing partitions.

AVIO Accelerated Virtual Input/Output. An I/O protocol that improves virtual I/O performance for network and storage devices used within the Integrity VM environment. The protocol also enables support for a greater number of virtual I/O devices per guest. Special drivers are required on both the VSP and guests. Participating guests must include a virtual I/O device configured to use the AVIO protocol.

backing store The physical device on the VSP that is allocated to guests, such as a network adapter, disk, or file.

Blade A board that contains CPUs and memory, and slots for C-class mezzanine cards, and onboard NICs. A blade is the equivalent of a cell in terms of being the unit of assignment for defining nPartitions.

BMC Baseboard Management Controller. The Management Processor (MP) console for Intel® Itanium systems.

boot virtual machines To load a virtual machine's operating system and start it. Once a virtual machine has been configured with an operating system, it is considered a guest, and is started automatically when Integrity VM starts, or manually using the `hpvmstart` command.
See also start virtual machines.

c3000 enclosure The HP BladeSystem c3000 enclosure works well in smaller data centers. A single c3000 enclosure is 6U high and can hold up to eight server, storage, or I/O option blades and up to four interconnect modules.

c7000 enclosure The HP BladeSystem c7000 enclosure is optimized for enterprise data centers. A single c7000 enclosure is 10U high and can hold up to 16 server, storage, or I/O option blades and up to eight interconnect modules.

captive virtual console account A special-purpose user account created on the VSP for each guest administrator or operator.

cell local memory See CLM

CLM Non-interleaved memory that can be quickly accessed by processors residing on the same socket as the memory. This is the same concept as SLM.

cluster Two or more systems configured together to host workloads. Users are unaware that more than one system is hosting the workload.

cluster member A cluster node that is actively participating in the Serviceguard cluster.

cluster node A system (VSP or guest) configured to be a part of a Serviceguard cluster.

Deconfigured The term used to describe the health of a resource that has been marked as unusable by the Health Repository. Such a resource will be excluded from partition activity.

dedicated device A pNIC or storage unit that is dedicated to a specific virtual machine. A dedicated device cannot be used by multiple virtual machines.

direct I/O networking The direct I/O networking feature allows virtual machines to directly control I/O devices.

distributed guests Guests that has been configured as a Serviceguard package.

EFI Extensible Firmware Interface. The boot firmware for all HP Integrity systems.

enclosure An HP BladeSystem c-Class enclosure holds ProLiant and Integrity server blades, storage blades, I/O option blades, interconnect modules (switches, pass-thru modules, and Virtual Connect modules), a NonStop passive signal midplane, a passive power backplane, power supplies, fans, and Onboard Administrator modules.

entitlement The amount of a system resource (for example, a processor) that is guaranteed to a virtual machine. The actual allocation of resources to the virtual machine can be greater or less than its entitlement, depending on the virtual machine's demand for processor resources and the overall system processor load.

event log Information about system events. An event log indicates what event has occurred, when and where it happened, and its severity (alert level). Event logs do not rely on normal I/O operation.

extensible firmware interface *See* EFI.

failover The operation that takes place when a primary service (network, storage, or CPU) fails, and the application continues operation on a secondary unit. In the case of Serviceguard virtual machines, the virtual machine can fail over to another cluster member. In case of a network failure, on a properly configured system the virtual machine can fail over to another LAN on the same cluster node.

guest The virtual machine running the guest OS and guest applications.

guest administrator The administrator of a virtual machine. A guest administrator can operate the virtual machine using the `hpvmconsole` command with action that can affect the specific guest only.

guest application A software application that runs on a guest.

guest application package A guest application that has been configured as a Serviceguard package.

guest console The virtual machine console that is started by the `hpvmconsole` command.

guest management software Software that is provided with Integrity VM that you install on the guest to ensure the guest is manageable by Integrity VM and other components of the Virtual Server Environment and HP Integrity Virtual Server Manager.

guest operator The administrator of the guest OS. This level of privilege gives complete control of the virtual machine but does not allow control of the other guests, the VSP, or the backing stores.

guest OS Guest operating system.

guest package A Serviceguard package that is an Integrity VM guest.

host
1. A system or partition that is running an instance of an operating system.
2. The physical machine that is the VSP for one or more virtual machines.

host administrator The system administrator. This level of privilege provides control of the VSP system and its resources, as well as creating and managing vPars/VMs.

host name The name of a system or partition that is running an OS instance.

host OS The operating system that is running on the host machine.

Ignite-UX The HP-UX Ignite server product. Used as a core build image to create or reload HP-UX servers.

ILM	Interleaved Memory. Is implemented as Partition Memory in HP Superdome 2, which includes Direct Access Partition Memory and Agent Access Partition Memory
Integrity Virtual Machines	The HP Integrity Virtual Machines product, which allows you to install and run multiple systems (virtual machines) on the same physical host system.
Integrity VM	*See* Integrity Virtual Machines..
ISSE	HP Instant Support Enterprise Edition. A secure remote support platform for business servers and storage devices.
localnet	A virtual switch created by default when Integrity VM is installed on a VSP. The local network created by this vswitch can be used for communications among guests but not for communication between the VSP and any guest or between any external system and a VM guest.
Machine check abort	See MCA.
MCA	Machine check abort
migration	The operation of stopping a Serviceguard package on one cluster member and then starting it on another cluster member. Migrating the package (for example, a virtual machine), can be useful in system management procedures and workload balancing. *See also* virtual machine migration..
multiserver environment	A Serviceguard cluster consisting of VSP systems.
N_Port ID Virtualization	See NPIV.
NIC	Network Interface Card. Also called “network adapter.”
nPartition	A partition that is assigned one or more blades and optionally zero or more I/O bays. An nPartition can run a single OS (either a standalone OS or an HPVM host), or an nPar can be sub-divided into customer-defined vPars. An HP Superdome 2 nPar works like an nPar on cellular servers.
NPIV	N_Port ID Virtualization. A Fibre Channel facility allowing multiple N_Port IDs to share a single physical N_Port.
NSPOF	No single point of failure. A configuration imperative that implies the use of redundancy and high availability to ensure that the failure of a single component does not impact the operations of the machine.
online VM migration	Enables a running guest and its applications to be moved from one VSP to another without service interruption.
OVMM	Online VM migration. See *online VM migration*.
package configuration script	A script that is customized for each virtual machine Serviceguard package and that contains specific variables and parameters, including logical volume definitions, for that virtual machine.
package control script	A script containing parameters that control how Serviceguard operates.
Partition Number	A unique numeric value assigned to a partition.
PMAN	Platform Manager. See *VSP*.
pNIC	Physical network interface card.
primary node	The cluster member on which a failed-over package was originally running.
redundancy	A method of providing high availability that uses multiple copies of storage or network units to ensure services are always available (for example, disk mirroring).
restricted device	A physical device that can be accessed only by the VSP system. For example, the VSP boot device should be a restricted device.
Serviceguard	Serviceguard allows you to create high-availability clusters of HP 9000 or HP Integrity servers. Serviceguard can be used to manage virtual machines as Serviceguard packages. A Serviceguard package groups application services (individual HP-UX processes) together and maintains them on multiple nodes in the cluster, making them available for failover.
Serviceguard node	A Serviceguard node, within the Integrity VM context, is a VSP. See *VSP*.

SGeRAC	Serviceguard extension for real application clusters.
SGeSAP	Serviceguard extension for SAP.
shared device	A virtual device that can be used by more than one virtual machine.
SLM	Non-interleaved memory that can be quickly accessed by processors residing on the same cell as the memory. This is the same concept as CLM.
socket local memory	See SLM
start virtual machines	To start a virtual machine that has been booted before. *See also* boot virtual machines.
storage unit	A file, DVD, disk, or logical volume on the VSP that is used by the virtual machines running on the VSP.
symmetric Serviceguard configuration	A cluster configuration in which the nodes share access to the same storage and network devices.
TOC	Transfer of control
Transfer of control	See TOC.
virtual console	The virtualized console of a virtual machine that emulates the functionality of the Management Processor interface for HP Integrity servers. Each virtual machine has its own virtual console from which the virtual machine can be powered on or off and booted or shut down, and from which the guest OS can be selected.
virtual device	An emulation of a physical device. This emulation, used as a device by a virtual machine, effectively maps a virtual device to an entity (for example, a DVD) on the VSP.
virtual machine	Virtual hardware system. Also called VM.
virtual machine application	The executable program on the VSP that manifests the individual virtual machine. The program communicates with the loadable drivers based on information in the guest-specific configuration file, and it instantiates the virtual machine.
virtual machine console	The user-mode application that provides console emulation for virtual machines. Each instance of the virtual machine console represents one console session for its associated virtual machine.
virtual machine host	*See* VSP.
Virtual Machine Manager (VMM)	The management application responsible for managing and configuring HP Integrity Virtual Machines.
virtual machine migration	Migration of a virtual machine from one VSP system to another by using the Integrity VM command `hpvmmigrate`. Do not use this command for virtual machine packages.
virtual machine package	A virtual machine that is configured as a Serviceguard package.
virtual network	A LAN that is shared by the virtual machines running on the same VSP or in the same Serviceguard cluster.
virtual switch	*See* vswitch.
Virtualization Services Platform	*See* VSP.
VM	See *Virtual machine*.
vNIC	Virtual network interface card (NIC). The network interface that is accessed by guest applications.
vPar	Virtual partition. A partition that is created and managed from the VSP. A vPar is assigned CPU cores, and memory.
VSP	Virtualization Services Platform. The management platform for creating and managing virtual partitions. Provides both command-line interface and graphical user interface for configuring and managing vPars.
vswitch	Virtual switch. A component in the guest virtual network. By associating the vswitch with a physical working LAN on the VSP, you provide the guest with the capability of communicating outside the localnet.

WBEM Web-Based Enterprise Management. A set of Web-based information services standards developed by the Distributed Management Task Force, Inc.A WBEM provider offers access to a resource. WBEM clients send requests to providers to get information about and access to the registered resources.

workload The collection of processes in a virtual machine.

Index

W