HP Integrity Virtual Machines 4.3: Installation, Configuration, and Administration

HP Part Number: 5900-1712
Published: April 2011
Edition: 10.0

Contents

Figures

Tables

About This Document

This document describes how to install and configure the HP Integrity Virtual Machines (Integrity VM) product, and how to create and install virtual machines and guest operating systems.

For recent updates to the product documentation, see the *HP Integrity Virtual Machines 4.3: Release Notes*.

Intended Audience

This document is intended for system and network administrators responsible for installing, configuring, and managing Integrity VM 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 on their virtual machines.

New and Changed Information in This Edition

This manual supersedes the manual of the same title for HP Integrity Virtual Machines Version 4.2. For more information about the new version of the product, see the *HP Integrity Virtual Machines 4.3: Release Notes*.

Typographic Conventions

This document uses the following typographic conventions.

Book Title	Title of a book or other document.
Linked Title	Title that is a hyperlink to a book or other document.
http://www.hp.com	A website address that is a hyperlink to the site.
`Command`	Command name or qualified command phrase.
`user input`	Commands and other text that you type.
`computer output`	Text displayed by the computer.
Enter	The name of a keyboard key. Note that **Return** and **Enter** both refer to the same key. A sequence such as **Ctrl+A** indicates that you must hold down the key labeled **Ctrl** while pressing the **A** key.
term	Defined use of an important word or phrase.
`variable`	The name of an environment variable, for example `PATH` or `errno`.
`value`	A value that you may replace in a command or function, or information in a display that represents several possible values.
find(1)	HP-UX manpage. In this example, “find” is the manpage name and “1” is the manpage section.

NOTE: Examples captured from software may display software versions that differ from the actual released product.

Product Naming Conventions

Table 1 defines the naming conventions for the versions of the HP-UX operating system.

Table 1 HP-UX Versions

Version Number	Version Name
HP-UX 11i v2	HP-UX 11.23
HP-UX 11i v2 (0505)	HP-UX 11i v2 May 2005 release
HP-UX 11i v2 (0609)	HP-UX 11i v2 September 2006 release
HP-UX 11i v3	HP-UX 11.31

Table 2 defines the naming conventions for the versions of the Integrity VM product.

Table 2 Integrity VM Versions

Version Number	Version Name
Integrity VM A.01.20	HP Integrity Virtual Machines version 1.2
Integrity VM A.02.00	HP Integrity Virtual Machines version 2.0
Integrity VM A.03.00	HP Integrity Virtual Machines version 3.0
Integrity VM A.03.50	HP Integrity Virtual Machines version 3.5
Integrity VM B.04.00	HP Integrity Virtual Machines version 4.0
Integrity VM B.04.10	HP Integrity Virtual Machines version 4.1
Integrity VM B.04.20	HP Integrity Virtual Machines version 4.2
Integrity VM B.04.20.05	HP Integrity Virtual Machines version 4.2.5
Integrity VM B.04.30	HP Integrity Virtual Machines version 4.3

Document Organization

This manual consists of the following chapters:

- Chapter 1: "Introduction" (page 17) describes the Integrity VM software and related products for managing Integrity servers from HP.
- Chapter 2: "Installing Integrity VM" (page 31) describes how to install the Integrity VM product.
- Chapter 3: "Creating Virtual Machines" (page 49) describes how to create virtual machines.
- Chapter 4: "Creating HP-UX Guests" (page 75) describes how to create HP-UX guests
- Chapter 6: "Creating Virtual Storage Devices" (page 85) describes how to create virtual storage devices.
- Chapter 7: "Creating Virtual Networks" (page 125) describes how to create virtual networks.
- Chapter 8: "Managing Guests" (page 145) describes how to start, stop, and manage virtual machines and resources.
- Chapter 9: "Migrating Virtual Machines" (page 177) describes how to migrate guests to other VM Host systems.
- Chapter 10: "Using HP Serviceguard with Integrity VM" (page 203) describes how to set up Serviceguard to manage your guests.
- Chapter 11: "Reporting Problems with Integrity VM" (page 227) describes how to solve virtual machine problems.
- Appendix A:"Rolling Back to the Previously Installed Version of Integrity VM" (page 229) describes how to rollback Integrity VM to the previous version.
- Appendix B: "Sample Script for Adding Multiple Devices" (page 235) provides a sample script for adding multiple storage devices to a guest.

- Manpages: "Integrity VM Manpages" (page 241) lists the HP-UX manpages provided with the HP Integrity VM software.
- Glossary : "Glossary" (page 371) defines important terms used in the Integrity VM documentation.

Related Information

You can download the latest version of this document from Business Support Center Manuals. The following related documents can also be downloaded from the same site:

- *HP Integrity Virtual Machines Release Notes*
- *Ignite-UX Reference*
- *Ignite-UX Installation Booting* White Paper
- *Using Ignite-UX with Integrity VM* White Paper
- *HP-UX Installation and Update Guide*
- *HP-UX Reference*
- *Managing Serviceguard*
- *HP Auto Port Aggregation (APA) Support Guide*
- *Using HP-UX VLANS*
- *HP Integrity Virtual Machines Manager 4.1 Software: User Guide*
- *HP Integrity Virtual Machines Manager 4.1 Software: Release Notes*
- *HP Insight Dynamics –VSE 6.2: Integrity CMS Installation and Configuration Guide*
- *HP Insight Global Workload Manager 6.2 Software: User Guide*
- *HP Insight Virtualization Manager 6.2 With Logical Server Managment: User Guide*
- *HP Insight Capacity Advisor 6.2 Software: User Guide*
- *HP Integrity Essentials Global Workload Manager: Workload Management for HP Integrity Virtual Machines* (a white paper)

The website Business Support Center Manuals also includes technical papers about using virtual machines.

For a time-limited evaluation version of Integrity VM, search software.hp.com.

Publishing History

Manufacturing Part Number	Supported Operating Systems	Supported Versions	Document Edition Number	Publication Date
T2767-90004	HP-UX	11i v2	1.0	October 2005
T2767-90024	HP-UX	11i v2	2.0	October 2006
T2767-90067	HP-UX	11i v2	3.0	April 2007
T2767–90105	HP-UX	11i v2	4.0	December 2007
T2767–90141	HP-UX	11i v3	5.0	September 2008
T7267–90141	HP-UX	11i v3	6.0	October 2008
T2767–90180	HP-UX	11i v3	7.0	December 2009
T2767–90206	HP-UX	11i v3	8.0	March 2010

Manufacturing Part Number	Supported Operating Systems	Supported Versions	Document Edition Number	Publication Date
T2767–91006	HP-UX	11i v3	9.0	March 2011
5900–1712	HP-UX	11iv3	10.0	April 2011

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 VM Host system. Information from Insight Remote Support running on virtual machines should not be used to determine the hardware state.

HP Encourages Your Comments

HP encourages your comments concerning this document. We are truly committed to providing documentation that meets your needs.

Please submit comments to:

http://www.hp.com/bizsupport/feedback/ww/webfeedback.html

Please include the document title, manufacturing part number, and any comment, error found, or suggestion for improvement you have concerning this document.

1 Introduction

This chapter describes the Integrity VM product, including the following topics:

- “About HP Integrity Virtual Machines”
- “New Features and Enhancements in This Release”
- “Using AVIO with Integrity VM”
- “Running Applications in the Integrity VM Environment”
- “Related Products”
- “Using the Integrity VM Documentation”
- “Using This Manual”

1.1 About HP Integrity Virtual Machines

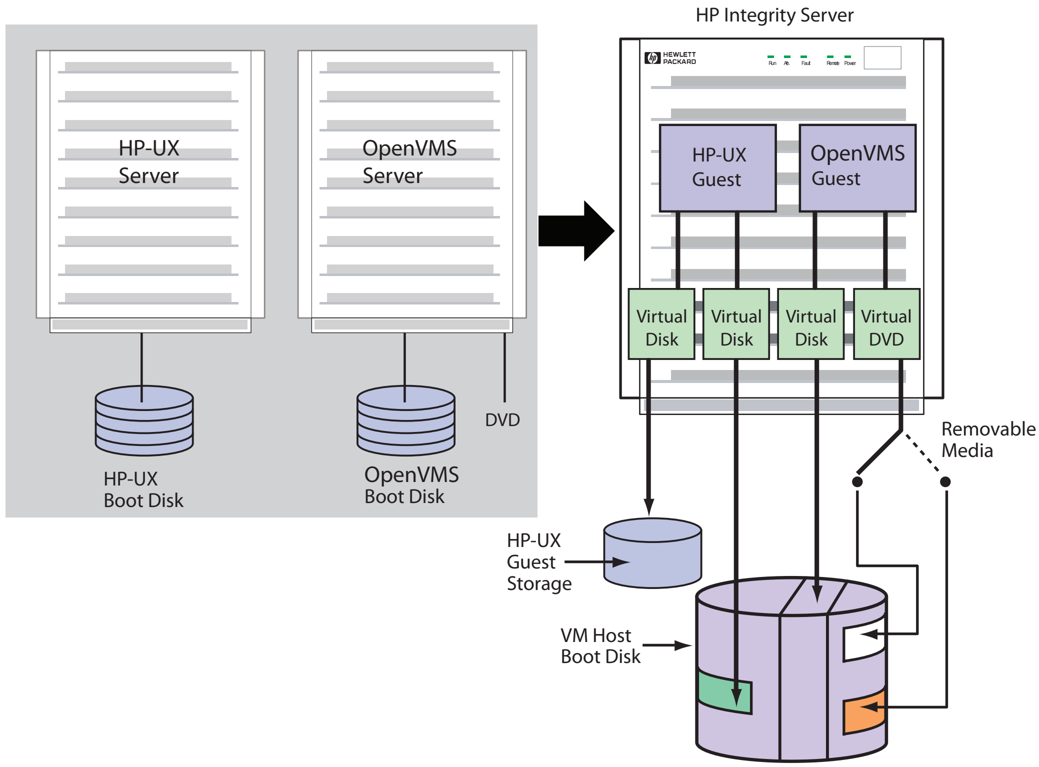
Integrity Virtual Machines is a soft partitioning and virtualization technology that provides operating system isolation, with sub-CPU allocation granularity and shared I/O. Integrity VM can be installed on an Integrity server, Integrity server blade, or hardware partition (nPartition) running HP-UX.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

- VM Host
- Virtual machines (also called guests)

The VM Host 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. Integrity VM provides a small guest software package that aids in local management of the guest's virtual machine.

Figure 1 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 VM Host. Each virtual machine is isolated from the others. The VM Host administrator allocates virtual resources to the guest. The guest accesses the number of CPUs that the VM Host 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 VM Host system has sufficient physical CPUs for it. Figure 1 illustrates how an HP-UX system and an OpenVMS system can be consolidated on a single Integrity server. The HP-UX boot disk is consolidated onto the same storage device as the VM Host boot disk and the OpenVMS guest storage. The OpenVMS guest also has access to removable media (CD/DVD) that can be redefined as necessary.

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 VM Host'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.

1.2 New Features and Enhancements in This Release

The features in the following list have been included in this release of Integrity VM:

- HP-UX 11i v3 VM Host — HP-UX 11i v3 1103
- HP-UX 11i v3 guests — HP-UX 11i v3 1003 through 1103 (all Integrity servers)
- HP-UX 11i v2 guests — HP-UX 11i v2 0712 to 0806 (all Integrity servers)
- HP OpenVMS V8.4 guests
- NVRAM Edit Utility — Displays, creates, edits and removes EFI variables in NVRAM files from a VM Host.
- Integrity VM virtual iLO Remote Console — Allows access to the guest console by logging into a specific IP address.
- `hpvmhostgdev` script utility
- Serviceguard 11.20
- 16 virtual CPUs
- 128 GB guest memory
- Multiple Accelerated Virtual Input/Output (AVIO) LAN localnet support — Allows the creation of multiple instances of localnet vswitches.
- Guest VLAN support for AVIO localnet
- 256 virtual AVIO storage devices
- `hpvmsar` enhancements
- Support for NFS guest back stores for root, swap, and dump. See Section 6.2.2.4 (page 114).
- Hyperthreading no longer needs to be turned off in the Integrity VM Host before installing or using Integrity VM. It is no longer necessary to specify:

  `/usr/sbin/setboot -m off`

- Support for Fibre Channel over Ethernet (FCoE) using the next generation converged infrastructure (CNA) hardware.

  NOTE: Integrity VM V4.3 supports HP NC551m Dual Port FlexFabric 10Gb Converged Network Adapter for FCoE and networking. For information about CNA, see *HP Converged Network Adapters (CNA)* at the following BSC website: HP Adapter and I/O Card documentation

- Support for networking HBA's AM225A, AM232A, and AM233A

The features in the following list have been rolled-into Integrity VM V4.3 from the patch kits that were provided after Integrity V4.2.5 was released.

- Support for ID-VSE Logical Server Management 6.2
- Support for HP OpenVMS V8.4 guests

1.3 Integrity VM Media

The HP Integrity Virtual Machines V4.3 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 Integrity VM, select the optional software bundle for HP Integrity Virtual Machines (T2767CC) prior to installing or updating HP-UX.

The 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

NOTE: The Online VM Migration feature is provided as a separate product (T8718AC) on the HP-UX 11i v3 Application Software (AR) DVD, as well as in the VSE-OE and DC-OE.

1.4 Using AVIO with Integrity VM

AVIO is supported by multiple guest operating systems and is optional based on a properly configured VM Host and changes to the guest operating systems. For AVIO support details, see the *HP Integrity Virtual Machines 4.3: Release Notes* and the AVIO product documentation. You must install and configure kernel drivers on both the VM Host and the guest systems. The guest configuration file and the `hpvmstatus` 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 VM Host and guests, for example, both from the OE or both from the same Web Release (for example, WEB0803).

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
```

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 host driver allows simultaneous access to a vswitch from guests configured with or without AVIO guest drivers for their virtual network interface cards (vNICs). Existing guests continue to operate correctly and their configurations remain unchanged. Guests configured to use a common vswitch are allowed to share VLANs that are in use by guests with or without AVIO configured. Ports remain distinct for each running guest 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 Integrity Virtual Machines 4.3: 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 guest to 256. The dynamic addition and deletion of AVIO storage devices is supported, and the following backing-store options are supported:

- Disk (such as `/dev/rdisk/disk1`)
- Null (for DVD devices only)
- File
- Volume (lv)
- Tape
- Burner
- Changer

Null for non-DVD devices is not supported.

NOTE: In general, AVIO devices (that is, `avio_lan`, `aviolan`, `avio_stor`, or `aviostor`) are configured and managed the same as VIO devices (that is `lan` and `scsi`). However, some additional Integrity VM command options and limitations on backing devices are noted throughout this document.

With HP-UX 11i v3, the AVIO storage guest driver can receive events asynchronously from the VM Host 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 guest 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 guest driver automatically detects the new device without the manual scan. The guest automatically detects any modification of the underlying backing storage.

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

Guest 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 Integrity VM server 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 guest must be modified using the `hpvmmodify` command to reference the new disk in place of the old disk.

1.5 Running Applications in the Integrity VM Environment

The VM Host system runs the Integrity VM software, which is responsible for allocating processor and memory resources to the running guests. The VM Host system can run physical resource, performance, and software management and monitoring tools. To allow the VM Host to allocate resources to the virtual machines, do not run end-user applications, such as database software, on the VM Host system. Instead, run them on virtual machines.

Typical software you can run on the VM Host system includes the following:

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

  NOTE: Integrity VM Version 4.3 is included in the HP-UX VSE-OE, as well as the in the HP-UX DC-OE. You can install Integrity VM from the OE and run it on the VM Host system. For HP-UX guests, you must purchase separate HP-UX 11i v3 licenses.

  For information about the software that is required on the VM Host system, see Chapter 2 (page 31).

- 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 guests are running, do not run other applications on the VM Host system. Example of applications that should not be run on the host are: Oracle, Workload Manager (WLM), HP SIM, and so forth. Integrity VM installation modifies kernel parameters, making the system unsuitable for running applications.

A guest running on a virtual machine 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 on a guest includes the following:

- HP-UX 11i V2 or 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 guest OS. Operating system patches and hardware restrictions apply to guests.

Do not run the following types of applications on a guest:

- Integrity VM software
- Utility pricing tools (run on the VM Host)
- Capacity planning tools (run on the VM Host)
- Applications that require direct access to physical hardware (for example, disaster-tolerant solutions)
- Multipath storage solutions
- SAN Management tools and applications that require access to serial interfaces (Integrity VM virtualizes SCSI and Ethernet devices only)
- Auto port aggregation (APA)

You must purchase licenses for any software you run on a virtual machine, 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 Integrity VM software and how to create guests to run on the VM Host system.

1.6 Do Not Install Applications on the VM Host System

When you install Integrity VM, HP-UX kernel parameters are changed to accommodate the virtual machine environment. This makes the system unsuitable for running any other applications. Regardless of whether guests are configured and running, the VM Host system is not configured to allow applications to share system resources. You can run system management utilities and Serviceguard, as documented in the *HP Integrity Virtual Machines Installation, Configuration, and Administration* manual.

- **Using backup solutions for VM Host and guest backups**

  Backup solutions such as HP Data Protector or Veritas NetBackup can be used on both the VM Host system and the guest systems. Consult the support matrix of such products for supported versions. Install the backup (client) agents on the VM Host and the guests. HP highly recommends that the /var and /opt directories, in addition to the standard locations, be backed up regularly on the VM Host system. Do not use the VM Host system as a backup server. For more information, see the *HP-UX 11i v2 Installation and Update Guide*.

- **Using HP GlancePlus/iX to monitor guests**

  You can use Glance on the VM Host to monitor guest data, but recorded measurements can be misleading. Glance receives the CPU accounting information from the guest kernel. Because the VM Host can take the guest processor away (for example, when a hardware interrupt occurs), the time spent running other guests is reported for the state that the guest 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 VM Host or in a guest; however, certain measurements might not apply in a particular context or report limited results. For example, measuring CPU utilization on the VM Host reports all the time spent running in guests as "system time"; to receive "user time" or "nice time" for a given guest, you must run Glance in that guest. Similarly, memory-related faults, or system calls for a guest, are not visible from Glance running in the VM Host, from Glance running in that guest. Glance also offers a number of virtualization-related measurements. Note, Glance refers to virtual machines as logical systems.

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

  If you use gWLM within Insight Dynamics — VSE to manage virtual machines, when you upgrade the VM Host, make sure the gWLM agent on that host is running gWLM A.02.50 or greater. Also, the managing Insight Dynamics — VSE Central Management Station (CMS) must be running A.02.50 or greater, as described in the *HP Insight Dynamics —VSE 6.0: Integrity CMS Installation and Configuration Guide*. To upgrade the VM Host, use the following procedure:

  1. Remove the gWLM agent using the following command:

     ```
     # swremove gWLM-Agent
     ```

  2. Upgrade Integrity VM as described in the *HP Integrity Virtual Machines 4.3: Installation, Configuration, and Administration* manual.
  3. Upgrade the gWLM agent, as described in the *HP Insight Dynamics — VSE 6.2: Integrity CMS Installation and Configuration Guide*.

  If you install the current version of Integrity VM without upgrading to gWLM A.02.50 or later, and then attempt to use gWLM within Insight Dynamics — VSE to manage virtual machines, the following error is reported:

  ```
  Error acquiring workload management lock. Look in the the file
  /var/opt/gwlm/gwlmagent.log.0 on hostname for more details.
  ```

  NOTE: You can use gWLM on the VM Host, but to manage iCAP resources. gWLM is the workload management solution for managing resource allocation to virtual machines.

- **Using the HP Integrity Virtual Machines Manager (VMMgr)**

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

  - HP System Management Homepage (SMH).

    For more information about using Integrity Virtual Machines Manager under SMH, see the *HP Integrity Virtual Machines Manager Version 4.0 Getting Started Guide*.

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

    For more information about Insight Dynamics — VSE, see the *HP Insight Dynamics 6.0 Getting Start Guide* on the BSC website: http://bizsupport2.austin.hp.com/bc/docs/support/SupportManual/c02048567/c02048567.pdf

  If you have installed the HP Integrity Virtual Machines Manager software, you must upgrade it to a version that supports this version of Integrity VM.

  To use Integrity Virtual Machines Manager, you must install the VMProvider bundle that is provided with Integrity VM. If you upgrade Integrity VM, be sure to keep the VMProvider up-to-date also. If the VMProvider version does not match the Integrity VM version, the Integrity Virtual Machines Manager will not work properly.

  HP Integrity Virtual Machines Manager (VMMgr) Version 4.0 can be used with Integrity VM V4.2.5. Earlier versions of VMMgr (V3.0 and earlier) do not install or run on an HP-UX 11i v3 VM Host and cannot be used with Integrity VM V4.2.5. Users upgrading to Integrity VM V4.2.5 must upgrade to VMMgr V4.0 or preferably VMMgr V6.1. In addition, users of HP Insight Dynamics Virtual Software Environment software versions earlier than V4.0 must upgrade to HP Insight Dynamics — VSE V4.1 (which contains VMMgr V4.0).

  VMMgr V4.0 is designed to run on HP-UX 11i v3 with Integrity VM V4.3 or earlier. It cannot use or display any new features in Integrity VM V4.3, but does support new features in HP-UX 11i v3, including the new agile addressing hardware paths and device files for storage devices. This means the following:

- The VMMgr product V4.0 allows the display and setting of the maximum and minimum setting of vCPUS and entitlement caps and the display of the graceful stop timeout. VMMgr V4.0 also uses the currently set entitlement caps to validate input on the Modify > Virtual Machine Entitlement dialog.
- V4.0 does not display any information about MSE group membership for the VM Host, nor support the initiation of either an offline or online guest migration from one VM Host to another. If a virtual machine is running on one member of an MSE group, it will be displayed as stopped on a different member of the MSE group without indicating that the virtual machine is not currently runnable on that host. It would permit a user to attempt to start the virtual machine. However, in this case, the underlying Integrity VM V4.3 `hpvmstart` command that VMMgr executes would issue an error message indicating that the virtual machine is not runnable on the current VM Host.

  Also, if a virtual machine that is defined on 2 different VM Hosts in an MSE group has the same Integrity VM UUID on both hosts, HP Insight Dynamics — VSE V4.1 and V6.0 might not display the virtual machine as contained in the VM Host where it is currently running in the Visualization Manager component. It might appear in Visualization Manager as if the virtual machine is running on one of the other VM Hosts where it is defined.
- Beginning with VMMgr V4.0, you can display persistent device files (and their associated LUN hardware paths) as backing device choices when adding a storage device.

  The VMMgr product V4.0 correctly displays devices by either legacy and agile specifications on the storage tab, depending on which was used when the virtual storage device was added to the virtual machine, and will correctly map those virtual devices to their respective physical backing devices. In addition, VMMgr V4.0 adds a new simplified display on the storage tab that eases tracing the connections from the virtual machine's virtual device to the physical backing device. The original, more complex, display is also available by selecting the "Show host bus adapter" checkbox on the storage tab. Persistent device files and LUN hardware paths are displayed in the physical storage device representations (boxes).

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

  You can use HP Instant Capacity solutions on the VM Host system as you would on any other HP-UX system.

1.7 Attributes That Can Be Changed Dynamically

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

- CPU
  - Adding or removing guest vCPU entitlement. The default is uncapped mode. In uncapped mode, this is also automatic based on overall “free” entitlement.
  - Enabling or disabling vCPUs from within a guest.
- Memory
  - Adding or removing the memory in use by a guest. This can also be made automatic with AMR (Automatic Memory Reallocation) based on overall "free" memory.
- Network
  - Adding or removing virtual switches (vswitches) on the VM Host.
- Storage
  - Adding or removing storage to or from a guest.

    NOTE: Depending on the type of storage being used, there may be additional steps required. See Section 6.2.1.5 (page 103)
- Migration
  - Migrating a guest online.

NOTE: Before you add or remove memory, networking, or storage from a guest, be sure you know if further action is required on the guest. For example, if you add storage to an OpenVMS guest, you must issue the following command on the guest, `SYSMAN IO AUTO`.

1.8 Related Products

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

- HP-UX operating system — Integrity VM runs on HP-UX 11i v3 Integrity systems on the VM Host. For more information, see the *HP-UX 11i v3 Installation and Update Guide*.
- HP WBEM Services for HP-UX — Many related products, such as VM Manager and gWLM, require the VM Host system be running HP WBEM Services.
- HP Insight Dynamics — VSE (ID-VSE) for Integrity— A graphical user interface for managing HP Integrity central managed systems (CMS). Runs under HP Systems Insight Manager. For more information, see the *HP Insight Dynamics 6.0 Getting Started Guide*.
- HP Insight Global Workload Manager (gWLM) — As part of HP Insight Dynamics — VSE (ID-VSE) for Integrity, 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 the version of Integrity VM, as described in the *HP Integrity Virtual Machines Release Notes*.
- HP Integrity Virtual Machines Manager — A graphical user interface for creating and managing HP Integrity virtual machines. Runs under either HP System Management Homepage (HP SMH) or HP Systems Insight Manager (HP SIM) as part of the HP Insight Dynamics — VSE (ID-VSE) Integrity. For more information, see the *HP Integrity Virtual Machines Manager 4.1 Software: User Guide*.

- HP Integrity VM Providers — To manage Integrity VM guests with VM Manager, gWLM, or any ID-VSE components, install the appropriate provider software from the operating system media or the Integrity VM guest management software kit.
- VERITAS Volume Manager— A data storage solution product that can be used to manage the physical disks on the VM Host. For more information, see the *VERITAS Volume Manager Administrator's Guide*.
- HP Auto Port Aggregation (APA) — A network switch that allows you to manage multiple network interfaces, which can be allocated to guests. For more information, see the *HP Auto Port Aggregation (APA) Support 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, and Chapter 10 (page 203), in this manual.

1.9 Using the Integrity VM Documentation

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

1.9.1 Integrity VM Manpages

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

- *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.
- *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 VM Host devices available for virtual machine access.
- *hpvmhostrdev*(1M) — Manages virtual machine access to devices used by the Integrity VM Host system.
- *hpvminfo*(1M) — Describes how to get information about the VM Host.
- *hpvmmigrate*(1M) — Describes how to migrate active guests and offline virtual machines from one VM Host 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 VM Host.
- *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 VM Host devices available for virtual machine access.

- *hpvmsar*(1M) — Displays performance information about one or several guests on the same host.
- *hpvmsg_move*(1M) — Describes how to initiate an online migration (move) of a virtual machine that has been associated with a Serviceguard package.
- *hpvmsg_package*(1M) — Assists the user with developing and managing Serviceguard package configurations.
- *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.
- *hpvmupgrade*(1M) — Examines the current Integrity VM server system to determine whether any virtual machines will have difficulty booting after the upgrade to the next Integrity VM version.
- *p2vassist*(1M) — Describes how to move applications from one server to another.

On the HP-UX guest, the following manpages are provided:

- *hpvmcollect*(1M) — Describes how to collect virtual machine statistics.
- *hpvmdevinfo*(1M) — Reports about storage for a virtual machine.
- *hpvminfo*(1M) — Describes how to get information about the VM Host.
- *hpvmmgmt*(1M) — Describes how to manage dynamic memory from the guest.
- *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.

1.9.2 Help Files

The virtual machine console is a special interface for managing guests. To start the virtual console after you create a guest, enter the `hpvmconsole` command and specify the guest name. For help using the virtual console, enter the `HE` command. For more information about the virtual console, see Section 8.7 (page 157).

1.10 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 3 describes each chapter in this manual.

Table 3 Chapters in this Manual

Chapter	Read if...
Chapter 1: "Introduction" (page 17)	You are new to HP Integrity Virtual Machines.
Chapter 2: "Installing Integrity VM" (page 31)	You are installing the HP Integrity Virtual Machines product.
Chapter 3: "Creating Virtual Machines" (page 49)	You are setting up new virtual machines on your VM Host system.
Chapter 4: "Creating HP-UX Guests" (page 75)	You are creating virtual machines that will run the HP-UX operating system.

Table 3 Chapters in this Manual *(continued)*

Chapter	Read if...
Chapter 6: “Creating Virtual Storage Devices” (page 85)	You need to make changes to the storage devices used by the VM Host or virtual machines.
Chapter 7: “Creating Virtual Networks” (page 125)	You need to make changes to the network devices on the VM Host system or to the virtual network devices used by the virtual machines.
Chapter 8: “Managing Guests” (page 145)	You need to manage existing virtual machines and resources.
Chapter 9: “Migrating Virtual Machines” (page 177)	You need to move virtual machines or active guests from one system to another.
Chapter 10: “Using HP Serviceguard with Integrity VM” (page 203)	You need to set up Serviceguard to manage your VM Host system or your virtual machines.
Chapter 11: “Reporting Problems with Integrity VM” (page 227)	You encounter problems while creating or using virtual machines.
	You plan to roll back to the previous version of Integrity VM.
Appendix B: “Sample Script for Adding Multiple Devices” (page 235)	You want to specify multiple storage devices at one time for a guest.
Integrity VM Manpages	You need to understand how to use an Integrity VM command. This section represents the Integrity VM manpages, which are available online using the `man` command. For example: `# man hpvminfo`
“Glossary” (page 371)	You do not understand the definition of a term used in the Integrity VM product documentation.

This manual and the *HP Integrity Virtual Machines 4.3: Release Notes* are available on the Instant Information DVD or may be viewed, downloaded, and printed from the web.

2 Installing Integrity VM

This chapter describes how to install the Integrity VM software and how to prepare the VM Host environment for guests. It includes the following topics:

- "Installation Requirements"
- "Installation Procedure for New Installations"
- "Upgrading the VM Host from Previous Versions of Integrity VM V3.X to Integrity VM V4.3"
- "Verifying the Installation of Integrity VM"
- "Removing Integrity VM"
- "Reserving VM Host Devices"
- "Troubleshooting Installation Problems"

2.1 Installation Requirements

To prepare your VM Host 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 Integrity Virtual Machine Release Notes*, which are available on the product media. The most up-to-date release notes are available on http://www.hp.com/go/virtualization-manuals.

2.1.1 VM Host System Requirements

You must install Integrity VM Version 4.3 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 VM Host system (such as disks, network bandwidth, memory, and processing power), are shared by the VM Host 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 VM Host, such as the VM Host operating system boot disk.

Table 4 describes the minimum configuration requirements for installing Integrity VM on the VM Host system.

Table 4 Requirements for Installing Integrity VM V4.3

Resource	Description
Computer	An Integrity server
Operating system	HP-UX 11i v3 March 2011 running on an Integrity server, as well as any appropriate software patches (see the *HP Integrity Virtual Machines 4.3: Release Notes*). The license for Integrity VM includes the license for running the HP-UX 11i v3 Base Operating Environment (BOE) on the VM Host 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).

Table 4 Requirements for Installing Integrity VM V4.3 *(continued)*

Resource	Description
Disk storage	Sufficient disk space for the following: • The VM Host operating system (see the *HP-UX 11i v3 Installation and Update Guide*) • The VM Host software (50 MB) • Disk space for each guest operating system, including swap space **NOTE:** The VM Host 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 6 (page 85).
Memory	Sufficient physical memory (RAM), including the following: • For the VM Host, 1250 MB + an additional 8.5% of the total physical memory • For the guests, the sum of the memory requirements of all of the guests that are booted simultaneously, plus an additional 8% of the memory used for all guests with less than 4GB of memory, plus an additional 8.3% of the memory used for all guests with more than 4GB of memory. • HP-UX 11i v2 May 2005 requires a minimum of 1 GB of memory, so a guest running HP-UX must be configured with at least that much memory. For example, for a VM Host with GB of memory and two VMs configured with 3 GB of memory each, the memory requirements would be calculated as follows: • 2.58 GB for the VM Host (1250 MB plus 8.5% of 16 GB) • 6.48 GB total guest requirement (108% of 6 GB) • Total requirements = 9.06 GB of memory This leaves 6.93 GB of memory for additional guests.
Network configuration	A configured and operational network. To allow guests network access, the VM Host must have at least one functioning network interface card (NIC). For more information about configuring network devices for virtual machines, see Chapter 7 (page 125).

NOTE: The VM Host overhead is valid for VM Hosts with base_pagesize=64.

2.1.2 Inhibitors to Installing Integrity VM

The following items block Integrity VM Version 4.3 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. Check for this product with the following command:

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

2.1.3 Bundle Names

The Integrity VM release contains the following software:

- `T2767CC` — B.04.30 Integrity VM
- `T8718AC` — B.04.30 Integrity VM Online Migration software bundle, provided as a separate product on the Application software (AR) DVD.
- `VMGuestLib` — B.04.30 Integrity VM Guest Support Libraries, installed as part of the HP-UX VSE-OE and required by Integrity VM
- `VMGuestSW` — B.04.30 Integrity VM Guest Support Software
- `VMProvider` — B.04.30 WBEM Provider for Integrity VM, which allows you to use the HP Integrity VM Manager to manage the VM Host
- `VMVirtProvider` — B.04.30 Integrity VM Virtualization Provider, which provides support for logical servers to manage Integrity VM Hosts and guests.
- `VMKernelSW` — B.04.20 Integrity VM Kernel Software
- `HP-UX 11i V3` — HP-UX 11i v3 Virtual Server Operating Environment [VSE-OE] that is provided for the VM Host system
- `PRM-Sw Krn` — Installed as part of the HP-UX VSE-OE and required by Integrity VM
- `B8465BA` — HP WBEM Services for HP-UX, installed as part of the VSE-OE and required by Integrity VM

2.1.4 Installing VMProvider

To install the VMProvider bundle with the Integrity VM software on the VM Host, enter the following command:

```
# swinstall -x autoreboot=true -s my.server.foo.com:/depot/path T2767CC vmProvider
```

To install the VMProvider bundle with the Integrity VM software on the HP-UX guest, see Section 9.3 (page147):

If you install the VMProvider on either the VM Host or a guest, make sure that the system is using HP WBEM Services A.02.00.10 or higher. For example:

```
# swlist WBEMServices
# Initializing...
# Contacting target "alien2"...
#
# Target:  alien2:/
#

# WBEMServices                              A.02.00.11     WBEM Services CORE
 Product
  WBEMServices.WBEM-CORE                    A.02.00.11     WBEM Services CORE
 Fileset for hp Integrity servers
  WBEMServices.WBEM-CORE-COM                A.02.00.11     WBEM Services COM
 Fileset for hp Integrity servers and hp 9000 servers
  WBEMServices.WBEM-MAN                     A.02.00.11     WBEM Services MAN
 Fileset
  WBEMServices.WBEM-MX                      A.02.00.11     WBEM Services MX
 fileset
```

2.1.5 Integrity VM Requires HP WBEM Services on the VM Host

The version of HP WBEM Services for HP-UX must be A.02.00 or later. Integrity VM fails to install if the version of WBEM Services on your VM Host is not A.02.00 or later.

2.2 Installation Procedure for New Installations

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 4.3, see Section 2.3 (page 35). If you are updating Integrity VM V4.0 or later to Integrity V4.3, see Section 2.4.

Once you have read the product release notes and verified that you have met the system requirements as described in Section 2.1.1 (page 31), install the Integrity VM software as described in this section.

NOTE: Installing the 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 VM Host and path name that correspond to the software distribution depot that contains the T2767CC 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, the following command installs Integrity VM and online VM migration bundle:

   ```
   # swinstall -x autoreboot=true -s my.server.foo.com:/depot/path T2767CC T8718AC
   ```

   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 Integrity VM bundle (T2767CC) from the list presented by the GUI, and if included on the list, the online VM migration software bundle, T8718AC..

   The VM Host 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/backup` directory. If you revert to the older version of Integrity VM, you can use the backup configuration files to restore your VM Host and guest configurations.

   NOTE: You might receive a warning message during the `swinstall` session indicating that the AVIO bundles are not installed. The installation will continue and complete successfully. The AVIO bundles are optional if all your guests are using the HP-UX operating system and should install by default. If they did not, and you want to install them with the Integrity VM bundle T2767CC, go to the HP-UX 11i v3 1103 media and look for the following bundles to install on the VM Host, HostAVIOStor and HostAvioLan.

   ---

3. Unmount and remove any installation media. The VM Host system automatically reboots, if necessary.
4. Once the Integrity VM software is installed and running, the VM Host 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 Integrity VM guest management software is installed in the `/opt/hpvm/guest-images` directory.
- The HP Serviceguard for Integrity VM Toolkit is installed in the `/opt/cmcluster/toolkit/hpvm` 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 guests using the `hpvmcreate` command, as described in Chapter 3 (page 49).

Integrity VM installation modifies certain kernel parameters. For this reason, you cannot install any other applications besides Integrity VM on the VM Host system, regardless of whether guests are running or not. Table 5 lists the kernel parameters that are modified when you install Integrity VM.

Table 5 Kernel Parameters

Parameter	Default Value	Modified Value
`filecache_max`	50%	1%
`filecache_min`	5%	1%
`lcpu_attr`	0	0
`lockable_mem_pct`	90%	99%
`maxdsiz_64bit`	4294967296	34359738368

NOTE: Integrity VM Version 4.2 and later support both 4KB (default) and 64KB settings for the base_pagesize tunable. In previous versions, installation of Integrity VM set the base_pagesize tunable to 64KB, because of the significant improvements in memory efficiency that resulted. HP still recommends setting the base_pagesize tunable to 64K for this reason. However, there are several defects that can potentially hamper software installation and operation when HP-UX is configured with 64K base_pagesize. For more details, see the white paper on this topic, *Tunable Base Page Size* available from:

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

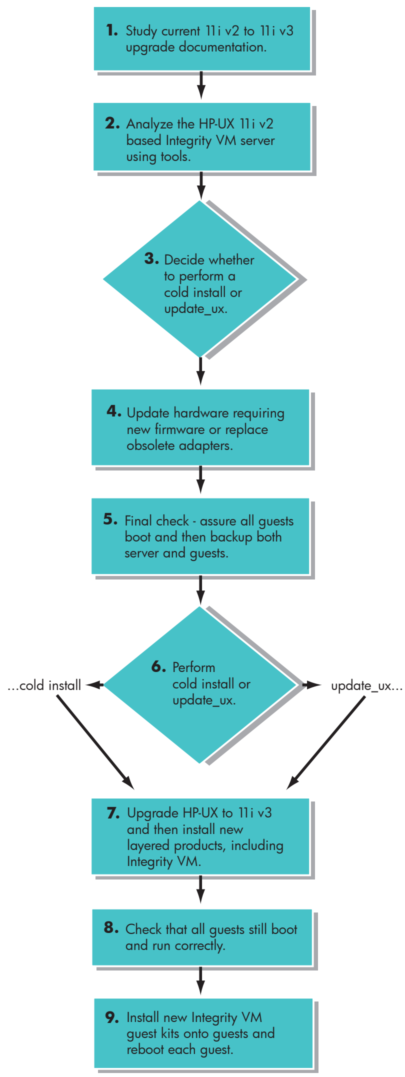
2.3 Upgrading the VM Host from Previous Versions of Integrity VM V3.X to Integrity VM V4.3

Integrity VM Version 4.3 VM Host 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 4.3 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 upgraded the VM Host from Integrity VM V4.0 or later to Integrity VM V4.3, see Section 2.4.

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 VM Host to make sure that all configured virtual machines (guests) boot and run after completing the upgrade to 11i v3.

Figure 2 provides a flowchart of the upgrade procedure from 11i v2 to 11i v3.

Figure 2 Upgrade Procedure



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

1. Offline migrate all virtual machines to another VM Host, 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

2.3.1 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 V4.3 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.

2.3.2 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`) and the Integrity VM `hpvmupgrade` tool 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.

The `hpvmupgrade` tool is provided with Integrity VM V4.3. Use one of the following ways to obtain this tool:

- Download VMGuestLib from the http://software.hp.com website and follow the instructions documented there.
- Install VMGuestLib from HP-UX 11i v2 AR media, September 2008 or later.

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.

2.3.2.1 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.

2.3.2.2 Run the Integrity VM Upgrade Tool

The Integrity VM upgrade tool, `hpvmupgrade`, can be run on either an 11i v2 or an 11i v3 system. This tool focuses on analyzing guest configurations for problems that might cause the guest not to boot when running on an 11i v3 Integrity VM server. Run this utility during the upgrade analysis stage.

To run this utility, use the following command:

```
# hpvmupgrade -e
```

It produces the following log file:

`/var/opt/hpvm/common/hpvmupgrade.`*`current_date_and_time`*

This log file contains a section for each guest configured on the server and displays the following message types:

- Warning messages indicate problems that can cause a guest booting problem.
- Error messages indicate problems that can cause a guest problem. For example, if a guest's virtual disk is backed by a file or device that does not exist on the 11i v2 system, a warning is issued because the problem is likely to be the same on the 11i v3 system. If, however, a guest is using a device associated with a multipath solution that is no longer supported on 11i v3, an error is issued.

The Integrity VM upgrade tool is focused on the devices that are used to back guest virtual devices. Each guest configuration is queried for its virtual backing storage. The guests device list is then compared to known multipath solutions, AutoPath, Secure Path, PowerPath PVLinks, or Veritas DMP devices to detect any dependencies.

Because the 11i v3 storage stack supports native multipath access to devices through the agile device names, the common solution for old 11i v2 multipath solutions is to remove them and change the applications to reference the new agile devices.

The upgrade tool also examines guest devices for volume backing storage that was used with multipath devices for physical storage. If this dependency is found, it is flagged in the log file. This tool also verifies that the 11i v2 Integrity VM server is at V3.0 or V3.5.

2.3.2.3 Determine HP-UX 11i v3 Memory and System Disk Requirements

Integrity VM V4.3 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 V4.3. 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

---

2.3.2.4 Determine Version Requirements for HP-UX OE and Integrity VM

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 V4.3 release. HP recommends that all virtual machines (guests) be upgraded with Integrity VM Version 4.3 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 8.6 (page 156).

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 2.6 (page 46)).

The reasons for exceptions to this behavior, such as when guests do not boot after the upgrade, are defined in the following documents. The 11i v3 Update 2 release that Integrity VM V4.2 requires has changed its OE packaging. For information about the new 11i v3 OEs, see these documents:

- Customer Support Letter Describing OE Licensing
- New HP-UX 11i v3 Operating Environments

The following are the new 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 Integrity VM (T2767CC).

- 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 Integrity VM (T2767CC).

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

Table 6 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 (vmmgr) Version 3.0 or earlier before upgrading to Integrity VM Version 4.3. After installing Integrity VM V4.3, install Integrity VM Manager V3.5 or later.

The latest available version of Integrity VM Manager is V4.1. You can use Integrity VM Manager V4.1 with Integrity VM V4.3 and any earlier versions that are still supported. However, Integrity VM Manager V4.1 is not guaranteed to support new features provided by Integrity VM V4.3.

NOTE: If you are upgrading an Integrity VM Host 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*.

2.3.3 Decide Whether to Perform a Cold-Install or an Update

The preferred method for upgrading an HP-UX 11i v2 based Integrity VM Host to an 11i v3 based VM Host 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 VM Host.

NOTE: HP recommends a full back up of both the Integrity VM Host 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 and the `hpvmupgrade -e` utility.

2.3.4 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/

2.3.5 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 VM Host to be functionally equivalent to the 11i v2 Integrity VM Host configuration. Integrity VM V4.3 provides the `hpvmdevtranslate` utility to assist in mapping the legacy devices used by guests on the 11i v2 VM Host 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 VM Host 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 VM Host and mark them as reserved disks.

   `# hpvmdevmgmt -a rdev:device_name`
2. Back up and collect all relevant configuration from the 11i v2 VM Host.
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 VM Host.
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 2.1.2 (page 32) 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 Integrity VM V4.3 (T2767CC), 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 VM Host.
13. Install Integrity VM Version 4.3 to the 11i v3 VM Host.

   NOTE: See Section 2.1.2 (page 32) 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 VM Host.
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 VM Host configuration until the host is functionally equivalent to the former 11i v2 Integrity VM Host.

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 T2767CC
```

   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 T2767CC
```

6. Remove any blocking layered products that might block the Integrity VM installation. See Section 2.1.2 (page 32) 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 Integrity VM V4.3 (T2767CC), 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 4.3 to the 11i v3 VM Host.
10. Update non-OE applications from the Application media using the `swinstall` command.

   For example, if you plan to install vmmgr, switch to the AR disk and specify the following:

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

11. Create the recovery image.

2.3.6 Verifying Guests after Installing Layered Products

Follow these steps after installing layered products:

1. Use the hpvmupgrade utility to see whether any guests have configuration problems.
2. Start and stop each guest, one at a time, and make sure that they boot to their OS.
3. Use the guest troubleshooting section, Chapter 11 (page 227), to resolve guest booting problems.
4. Upgrade each guest with the new guest kit.
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 VM Host, 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 VM Host 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
```

2.3.7 Upgrade Troubleshooting Issues

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

- Mass storage issues

  The Integrity VM V4.3 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.

- Network issues

  Make sure there are no incompatibilities at this time.

- 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

2.4 Updating Previous Versions of the Integrity VM Host to Integrity VM V4.3

This section describes the process of updating a previous version of the Integrity VM Host to Integrity VM V4.3.

Once you have read the product release notes and verified that you have met the system requirements as described in Section 2.1.1, install the Integrity VM software as described in this section.

NOTE: Installing the 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 VM Host and path name that correspond to the software distribution depot that contains the T2767CC 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 T2767CC
   ```

   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 Integrity VM bundle (T2767CC) from the list presented by the GUI.

   The VM Host 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/hpvmbackup` directory. If you revert to the older version of Integrity VM, you can use the backup configuration files to restore your VM Host and guest configurations.

   NOTE: You might receive a warning message during the `swinstall` session indicating that the AVIO bundles are not installed. The installation will continue and complete successfully. The AVIO bundles are optional if all your guests are using the HP-UX operating system and should install by default. If they did not, and you want to install them with the Integrity VM bundle T2767CC, go to the HP-UX 11i v3 1103 media and look for the following bundles to install on the VM Host, HostAVIOStor and HostAvioLan.

3. Unmount and remove any installation media. The VM Host system automatically reboots, if necessary.

4. Once the Integrity VM software is installed and running, the VM Host 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.
   ```

2.5 Verifying the Installation of Integrity VM

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

```
# hpvminfo
hpvminfo: Running on an HPVM host.
```

To see exactly what versions of specific bundles are installed, enter the `swlist` command. (Specific version numbers on your installation may not match the examples in this manual.)

```
# swlist | grep -i "integrity vm"
  T2767CC               B.04.30     Integrity VM
  VMGuestLib            B.04.30     Integrity VM Guest Support Libraries
```

```
VMGuestSW              B.04.30     Integrity VM Guest Support Software
VMKernelSW             B.04.20     Integrity VM Kernel Software
```

When you install Integrity VM, the file `/etc/rc.config.d/hpvmconf` is created to record the product configuration. Note, HPVMNETINTVL has been removed from this file.

2.6 Removing Integrity VM

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

- VMProvider (if installed)
- T2767CC
- VMGuestLib
- VMGuestSW
- VMKernelSW (reboots the system)
- HostAvioLan
- HostAVIOStor
- GuestAvioLan
- GuestAVIOStor

To remove these bundles, enter the following commands:

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

Guests are not affected by this procedure. To remove guests, see the procedures in Section 3.8 (page 72).

2.7 Reserving VM Host Devices

You can protect the storage and network resources used by the VM Host against usage and corruption by virtual machines by marking the VM Host devices as restricted devices. For example, you can reserve the disk storage on which the VM Host 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 8.12 (page 171).

2.8 Troubleshooting Installation Problems

If the installation verification fails, report the problem using the procedures described in Chapter 11 (page 227). Some problems encountered in the process of installing Integrity VM are described in the following sections.

2.8.1 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.

2.8.2 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.

3 Creating Virtual Machines

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

3.1 Legacy and Agile Device Names

Integrity VM Version 4.3 supports the usage of both legacy and agile devices in guest configurations. Use of agile device names for configuring storage devices is highly recommended as it provides multipathing benefits. You can use the `hpvmmodify -P` *`guest`* `-m` *`rsrc`* command to change the existing legacy device to its agile device equivalent. The `ioscan -m dsf` command displays the mapping of the agile devices to legacy devices for the specific device special file `dsf` and vice versa. For example:

```
# ioscan -m dsf /dev/rdisk/disk2
```

Or:

```
# ioscan -m dsf /dev/rdsk/c12t0d0
```

NOTE: The `rmsf -L` command should not be executed on an Integrity VM server until all guest devices names have been changed to agile device names.

The following commands are helpful for reconfiguring guests to use agile device names:

Table 7 Commands for Reconfiguring Guests to Agile Device Names

Command	Description
`insf`	By default, creates both persistent and legacy DSFs for new devices.
`insf -L`	Restores legacy DSFs and legacy configuration information.
`rmsf -L`	Aids in migration by removing all legacy DSFs and legacy configuration information.
`ioscan —m dsf`	Maps persistent DSFs to their equivalent legacy DSFs and vice versa.
`ioscan -N`	Prints persistent DSFs when used with the `-n` option to list DSFs.
`io_redirect_dsf`	Associates a new disk with an existing set of DSFs. This is used when replacing an internal disk or a disk in a JBOD.
`hpvmmodify -P guest —m rsrc`	Modifies backing devices for guest devices.

3.2 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 machines.
- `hpvmmigrate`, which moves virtual machines from one system to another.
- `hpvmmodify`, which modifies existing virtual machines.

All of these commands accept the same options for specifying virtual machine characteristics. Table 8 describes each characteristic and command option.

Table 8 Characteristics of an Integrity Virtual Machine

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 3.2.1 (page 51)
Operating system type	If you do not specify the operating system type, it is set to UNKNOWN.	`-O` *os_type [:version]*	Section 3.2.2 (page 51)
Virtual CPUs (vCPUs)	If you omit this option when you create the virtual machine, the default is one vCPU.	`-c` *number_vcpus*	Section 3.2.3 (page 51)
CPU entitlement	If you omit this option when you create the virtual machine, the default is 10%.	`-e`*percent [:max_percent]* `-E` *cycles [:max_cycles]*	Section 3.2.4 (page 52)
Memory	If you omit this option when you create the virtual machine, the default is 2 GB.	`-r` *amount*	Section 3.2.5 (page 52)
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 3.2.7 (page 53)
Virtual machine label	If you omit this option, the virtual machine has no label.	`-l`*vm_label*	Section 3.2.8 (page 57)
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 3.2.9 (page 57)
Dynamic memory	If you omit the option, dynamic memory is not enabled for the guest.	`-x` *keyword=parameter*	Section 3.2.10 (page 58)
Group with administrator or operator privileges	If you omit this option, no group accounts have `admin` or `oper` privileges.	`-g` `[+]`*group*`[:`*admin* | *oper*`]`	Section 8.5 (page 154)
User with administrator or operator privileges	If you omit this option, no user accounts have `admin` or `oper` privileges.	`-u` `[+]`*user*`[:`*admin* | *oper*`]`	Section 8.5 (page 154)

3.2.1 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 VM Host, 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.

3.2.2 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 one of the following case-insensitive values:

- `hpux`

  For specific information about installing HP-UX guests, see Chapter 4 (page 75).

- `openvms`

  For specific information about installing OpenVMS guests, see the OpenVMS V8.4 documentation at OpenVMS Documentation.

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.

3.2.3 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 CPUs on the VM Host 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 CPUs, as this can cause undesirable behavior. The following command specifies the number of virtual CPUs this virtual machine has

```
# 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 VM Host, 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 VM Host. However, disabled virtual CPUs still appear on the VM Host, for example in the `hpvmsar` command.

3.2.4 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 VM Host 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 VM Host 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 VM Host system, the virtual machine can receive more than its entitlement. When there is contention for processing resources (on a VM Host 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 VM Host system. For more information, see *HP Integrity Essentials Global Workload Manager Administrator's Guide*.

3.2.5 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 VM Host 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 VM Host 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 VM Host for its operating system and its administrative functions. For more information about the memory requirements of the VM Host, see Section 2.1 (page 31).

Guest memory allocation can be viewed and allocated dynamically (that is, without stopping the guest) by using dynamic memory parameters, as described in Section 8.10 (page 161).

3.2.6 Automatic Cell Balancing

When creating a guest, Integrity VM determines the best fitting locality domain for the new guest when the VM Host is predominantly Cell Local Memory(CLM) or the guest has the `sched_preference` flag set to cell with the `hpvmmodify`, `hpvmcreate` or `hpvmclone` command. Integrity VM uses this setting as a guide for run-time scheduler planning as well as guest boot time CPU and memory binding. The `hpvmstatus -C` command provides a list of guests with their memory type.

If you do not use CLM at all, then all the guests use Interleaved Memory (ILM). If however, CLM is set, every `hpvmstart` command checks whether Integrity VM chooses cell or interleaved for this particular guest, and if cell is chosen, which cell it is. For example, you have an 8 GB VM Host configured with 75 percent CLM and 25 percent ILM. With two cells, each contributes 3 GB to cell local and 1 GB to interleaved. On boot, the operating system takes 1GB of the interleaved memory. If each guest takes 1 GB to start, the breakdown looks like this:

- guest 1: cell 0 (2 GB CLM left)
- guest 2: cell 1 (2 GB CLM left)
- guest 3: cell 0 (1 GB CLM left)
- guest 4: cell 1 (1 GB CLM left)
- guest 5: cell 0 (no CLM left)
- guest 6: cell 1 (no CLM left)
- guest 7: interleaved (no memory left at all)

The general trend is for CLM if any cell has at least as much free space as the available ILM.

3.2.7 Virtual Devices

Use the `-a` option to allocate virtual network switches and virtual storage devices to the virtual machine. The VM Host 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 direct 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.

3.2.7.1 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 VM Host 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.

Each guest can have two different types of LAN network devices, VIO and AVIO. For VIO guest networks, a vswitch functions just like a physical network interface card (pNIC), accepting network traffic from one or more virtual machines and directing network traffic to all of its ports. A vswitch without the backing of a host physical network card can be used by VIO guest devices for communication among VIO guest devices registered with the same vswitch. This type of vswitch is typically referred to as `localnet`. For more information, see Section 7.2.1.1 (page 129).

Unlike VIO guest networks, 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 VM Host. 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 7.2.1 (page 126).

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`* can be either `lan` or `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 VM Host or on different VM Host 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.

If you configure an HP-UX guest and an OpenVMS guest with the same vswitch, the network communication between these guests fail. This problem will be fixed in a future version of OpenVMS.

To workaround this problem, configure the HP-UX guest and the OpenVMS guest with different vswitches.

For more information about creating and managing VLANs on virtual switches, see Section 7.4 (page 134).

3.2.7.2 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 VM Host system (backing stores). The VM Host system must have sufficient physical storage for the VM Host 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`

  NOTE: DVD burners are not currently supported on OpenVMS guests.

- *adapter-type* can be `scsi` or `avio_stor` on an HP-UX guest.
- *hardware-address* or `pcibus, pcislot,scsitgt` (optional) specifies the virtual device PCI bus number, PCI slot number, and SCSI 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 VM Host. Specify it using the following format:

  `storage``:`*`location`*

  Where *`storage`* is one of the following:

  - `disk`
  - `lv`
  - `file`

   - `null`
   - `attach`

   And *`location`* is a VM Host system file.

For complete information about constructing storage specifications for virtual machines, see Section 6.2.2.1 (page 104).

The type of VM Host backing store can affect the performance of the virtual machine. Use the `ioscan` command to obtain information about the current device configuration on the VM Host 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 6 (page 85).

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 VM Host 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:scsi::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 rescan for devices on the virtual machine.

Some devices should be restricted to use by the VM Host 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 8.12.2.4 (page 174).

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 11.

NOTE: MPT virtual storage devices (VIO) have a maximum SCSI queue depth of 8. This is sufficient for the default SCSI queue depth of all guest types that have applied their guest kits. Increasing SCSI queue depths beyond the defaults might result in some I/O failures due to exhaustion of guest I/O retries.

If the VM Host storage used by a virtual MPT adapter is slow due to hardware problems or heavy I/O loads, the following HP-UX spinlock timeout might occur inside an HP-UX guest:

```
==============  EVENT  ============================ = Event #0 is CT_PANIC on CPU #2; = p crash_event_t
0xe000000100385000 = p rpb_t 0xe0000001011e4ff0 ==============  EVENT  ============================
RR0=0x00800831  RR1=0x00000831  RR2=0x02cd0031  RR3=0x02450031
RR4=0x03450031  RR5=0x00ffff31  RR6=0x07ff8031  RR7=0x00dead31
                BSP                    SP                    IP
0xe000000200024810 0xe0000002000434f0 0xe000000001f3d7c0 panic+0x410
0xe0000002000247b8 0xe000000200043500 0xe000000001fa8320 too_much_time+0x380
                                      pdk_spinlock.c:1619 wait_for_lock_spinner(inlined)
0xe0000002000246b8 0xe000000200043500 0xe000000000de4c80 wait_for_lock+0x670 0xe000000200024680 0xe000000200043570
 0xe000000000d7d820 spinlock+0xe0
0xe000000200024628 0xe000000200043570 0xe000000000c61de0 fw_lock_acq+0x70
0xe0000002000245b8 0xe000000200043570 0xe000000000c61be0 sal_proc_real+0x100
0xe000000200024588 0xe000000200043570 0xe0000000012ef860 cec_cfg_inX+0xd0 0xe000000200024560 0xe000000200043570
 0xe000000001306650 cec_cfg_in16+0x30
```

```
0xe000000200024538 0xe000000200043570 0xe00000000158d0a0 gh2p_rd_cfg_w+0x60
0xe000000200024508 0xe000000200043570 0xe0000000021d3f10 wsio_cfg_in16+0x70
               BSP                 SP                 IP
0xe000000200024460 0xe000000200043570 0xe00000012f7313a0 mpt:mpt_handle_chip_fault+0xe0
0xe0000002000243e8 0xe000000200043580 0xe00000012f730fe0 mpt:mpt_ch_task_mgmt+0x540
0xe0000002000243a8 0xe0000002000437b0 0xe00000012f7a0e20 mpt:$cold_mpt_io_active+0x540
0xe000000200024388 0xe0000002000437b0 0xe00000012f76b9b0 mpt:mpt_handle_address_reply+0x210
0xe0000002000242c8 0xe0000002000437b0 0xe00000012f76cd00 mpt:mpt_isr+0xa00
               BSP                 SP                 IP
0xe000000200024298 0xe0000002000437c0 0xe000000000eb2ca0 sapic_interrupt+0x60
0xe0000002000241b8 0xe0000002000437c0 0xe000000000eb1e20 external_interrupt+0x4b0 0xe000000200024190
0xe0000002000437f0 0xe000000001d9a780 bubbleup+0x880
+-------------  TRAP  ----------------------------
|  External Interrupt in KERNEL mode
|    IIP=0xe000000000d7d910:1
|  p struct save_state 0xdead31.0xe000000200043800
+-------------  TRAP  ----------------------------
0xe000000200024170 0xe000000200043b90 0xe000000000d7d911 spinunlock+0x51
                                        pm_swtch.c:3829 idle_drop_spu_state_locks(inlined)
0xe000000200024000 0xe000000200043b90 0xe000000000e6dc10 idle+0x1a50
```

Converting the guest's virtual MPT adapters to AVIO storage adapters prevents this spinlock timeout.

3.2.8 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 ("").

3.2.9 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 VM Host system is booted, and when the Integrity VM software is stopped and restarted on a running VM Host. For example, when you upgrade Integrity VM to a new version on a running system, the software is started automatically. The VM Host 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 VM Host. 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 VM Host 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 VM Host 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.

3.2.10 Specifying Dynamic Memory Parameters

Specifies whether the new virtual machine will use 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`*

NOTE: Dynamic memory is supported only on HP-UX guests.

For more information about using dynamic memory for guests, see Section 8.10 (page 161).

3.2.11 Configuration Limits

Table 9 lists the configuration limits for Integrity VM Version 4.3.

Table 9 Configuration Limits

Description	Support
# vCPUs/VM — Maximum (Integrity VM V4.3 Max vCPU = 16)	min (#pCPUs, Max vCPU)
# vCPUs/pCPU — Maximum	20
# VMs per VM Host — Maximum	256
# pCPUs in VM Host	HP-UX limit
Memory per VM — Minimum (11i v2 HP-UX)	1 GB
Memory per VM — Minimum (11i v3 HP-UX)	1.5 GB
Memory per VM — Maximum (HP-UX)	128 GB
# virtual SCSI devices / VM — Maximum	256 AVIO + 30 VIO

Table 9 Configuration Limits *(continued)*

Description	Support
# virtual NICs / VM — Maximum	62
# virtual switches — Maximum	50
# virtual NICs / vswitch	511
# virtual AVIO storage devices / VM — Maximum	128
# file backing store devices / VM — Maximum	30
# virtual AVIO storage devices	VM Maximum 256
# file backing store devices	VM Maximum 30
Maximum size of backing store for VIO (disk, lvol, file)	< 2TB
Maximum size of backing store for AVIO (disk, lvol, file)	> 2TB

3.2.12 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 VM Host 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

3.2.13 Default Guest Settings for HP-UX and OpenVMS

Table 10 lists the default guest settings for HP-UX, OpenVMS 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 10 Guest Default Settings

	HP-UX Guest Default Settings	OpenVMS Guest Default Settings	Unknown Guest Operating System Default Settings
Maximum CPUs	16	8	16
Default CPUs	1	1	1
Default memory	2 GB	2 GB	2 GB
Minimum memory	512 MB[1]	512 MB	32 MB
Maximum memory	128 GB	64 GB	128 GB
Default reserved memory	64 MB	64 MB	64 MB
Minimum reserved memory	32 MB	32 MB	32 MB
Maximum reserved memory	128 GB	64 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.

3.3 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 11 describes the options you can use with the `hpvmcreate` command.

Table 11 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 11 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 8.10 (page 161). 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 9 (page 177)
`-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 11 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 VM Host is a Serviceguard cluster node. • `SG-`*`pkgname`* indicates that the VM Host is a Serviceguard package. • `GWLM` indicates that the VM Host 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. You must also specify the `–L` option.
`-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.

3.3.1 Example of Virtual Machine Creation

To create a virtual machine named `host1`, enter the following command:

```
# hpvmcreate -P host1
```

This command creates a virtual machine named `host1` 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 #  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
host1                   12 UNKNOWN Off            1     0     0    2 GB        0
```

The `host1` virtual machine has been assigned virtual machine number 12, has been created with an UNKNOWN operating system type, one vCPU, no storage devices, no network devices, and 2 GB of memory. The `Runsysid` column indicates the VM Host that runs the virtual machine in a Serviceguard cluster. If the virtual machine runs on the local VM Host, or if Serviceguard is not configured, the `Runsysid` is zero. For more information about running virtual machines under Serviceguard, see Chapter 10: "Using HP Serviceguard with Integrity VM" (page 203).

3.4 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 12 describes the options to the `hpvmstart` command.

Table 12 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  Runsysid
==================== ===== ======= ========= ====== ===== ===== ======= ========
config1                  1 HPUX    Off            1     5     1  512 MB        0
config2                  2 HPUX    Off            1     7     1    1 GB        0
guest1                   5 OPENVMS On (OS)        1     5     1    1 GB        0
host1                   13 UNKNOWN On (EFI)       1     0     0    2 GB        0
```

For more information about using the `hpvmstatus` command, see Chapter 8 (page 145).

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/rdsk/c6t9d0 took 32 seconds to open.
```

3.5 Changing Virtual Machine Configurations

You can create a virtual machine with characteristics that the VM Host 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 VM Host 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 VM Host 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 13 describes the options you can use on the `hpvmmodify` command.

Table 13 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.

Table 13 Options to the `hpvmmodify` Command *(continued)*

Option	Description
`-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 VM Host. `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`
	`openvms`

`-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 VM Host 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.
`-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).

Table 13 Options to the `hpvmmodify` 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 VM Host is a Serviceguard cluster node. • `SG-pkgname` indicates that the VM Host is a Serviceguard package. • `GWLM` indicates that the VM Host 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.
`-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` • `online_migration` For more information about dynamic memory, see Section 8.10 (page 161).

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.

The `hpvmmodify` command generated no warnings, so the VM Host 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 scsi burner:

  resource: -a burner:scsi:[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 4.

3.6 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 8 (page 50) 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 14 describes the options you can use with the hpvmclone command.

Table 14 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 VM Host system.

`-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 VM Host 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`
	 `openvms`

`-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 6. For information about forming a virtual network device specification, see Chapter 7.
`-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 6. For information about forming a virtual network device specification, see Chapter 7.

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

Option	Description
`-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 6. For information about forming a virtual network device specification, see Chapter 7.
`-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 VM Host system.
`-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 7.4 (page 134).
`-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.

Table 14 Options to the `hpvmclone` 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`* • `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 8.10 (page 161). To specify the serial number of the new virtual machine, enter `serial_number={new | same}`
`-C`	Provides information about the memory type for Host and guests: cell local memory, interleaved, or none.

For example, to clone the virtual machine named `host1`, to create a new virtual machine named `clone1`, enter the following commands. First display the current guest status on the VM Host:

```
# 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 VM Host.

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 85).

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 6.

3.7 Stopping Virtual Machines

To stop a running virtual machine, use the `hpvmstop` command. You must confirm this command. Table 15 describes the options to the `hpvmstop` command:

Table 15 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.
`-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  Runsysid
==================== ===== ======= ========= ====== ===== ===== ======= ========
config1                  1 HPUX    Off            1     5     1  512 MB        0
config2                  2 HPUX    Off            1     7     1    1 GB        0
guest1                   5 OPENVMS On (OS)        1     5     1    1 GB        0
host1                   12 UNKNOWN Off            1     0     0    2 GB        0
```

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 VM Host, 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 VM Host. Do not stop a guest by killing the `hpvmapp` process.

3.8 Removing Virtual Machines

To remove a virtual machine from the VM Host, use the `hpvmremove` command. By default, you are required to confirm this action. Table 16 describes the options to the `hpvmremove` command.

Table 16 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  Runsysid
==================== ===== ======= ========= ====== ===== ===== ======= ========
config1                  1 HPUX    Off            1     5     1  512 MB        0
config2                  2 HPUX    Off            1     7     1    1 GB        0
guest1                   5 OPENVMS On (OS)        1     5     1    1 GB        0
```

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 VM Host's pool of available resources. (Any guest operating system and application data on the VM Host 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
```

3.9 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 11.

The following section describes a problem that might be encountered during virtual machine creation.

3.9.1 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.

4 Creating HP-UX Guests

To create HP-UX guests, install the HP-UX operating system on the 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 HP-UX Guest Management Software"
- "Troubleshooting HP-UX Guest Creation"

4.1 Installing the HP-UX Guest Operating System

You can install either HP-UX 11i v2 operating system or the HP-UX 11i v3 operating system as a guest OS, See the *HP Integrity Virtual Machines 4.3: Release Notes*, *Chapter 11 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 VM Host 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 OPENVMS 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:scsi::0,0,0
```

3. If necessary, restart the guest to remove the DVD from the guest configuration.

NOTE: You might receive a warning message during the guest installation process indicating that the AVIO bundles are not installed. The installation will continue and complete successfully. The AVIO bundles are optional and install by default. If they do not, and if you want to install them with the Integrity VM bundle T2767CC, go to the HP-UX 11i v3 1103 media and look for the following bundles to install on the HP-UX guest, GuestAVIOStor and GuestAvioLan.

After you install the AVIO bundles, upgrade your guest configurations to use AVIO.

NOTE: After installing the HP-UX operating system as a guest OS, HP recommends that you also install the PHKL_38623 and PHKL_38762 patches on the guest. These patches, which are included with the feature11i 1103 bundle, increase the accuracy of cycle-based accounting in the guest.

4.2 Do Not Create Golden Images of the VM Host for Guest Installation

Do not use the VM Host 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 VM Host software completely removed. To do so, remove both the Integrity VM bundle (T2767CC) and the VMKernelSW bundle:

```
# swremove -x autoreboot=true T2767CC VMKernelSW
```

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 T2767CC VMKernelSW
# Initializing...
# Contacting target "foo"...
ERROR:   Software "T2767CC" was not found on host "foo:/".
ERROR:   Software "VMKernelSW" was not found on host "foo:/".
```

For more information about using Ignite-UX golden images, see the *Ignite-UX Administration Guide*.

4.3 Golden Images of Systems with HPVM-Guest Installed Must have Integrity VM Device Drivers Configured

Golden images of systems with HPVM-Guest bundle installed must explicitly configure the dynamic memory device driver. If your golden system has the HPVM-Guest bundle installed, be sure the dynamic memory device driver is configured correctly in the associated golden image's configuration file. To achieve this, add the line

```
set_kernel += “module hpvmdynmem loaded”
```

after the “init_sw_sel” stanza in the golden image's configuration file. Failure to do so might render the dynamic memory control inoperable when the golden image is installed on a virtual machine.

4.4 Installing HP-UX Guest Management Software

After you install the HP-UX operating system on the virtual machine, install the Integrity VM guest management software. The guest management software includes:

- Operating system patches to optimize virtual machine operation
- Integrity VM management tools, including `hpvmcollect` and `hpvminfo` commands
- The VM Provider, which allows you to use the VM Manager to manage the guest.

To install guest management software on an HP-UX guest, select the appropriate version of HP-UX:

- `11iv2` indicates HP-UX 11.23.

  For HP-UX 11.23 guests, the guest depot file is:

/opt/hpvm/guest-images/hpux/11iv2/hpvm_guest_depot.11iv2.sd

- 11iv3 indicates HP-UX 11.31.

  For HP-UX 11.31guests, the guest depot file is:

  /opt/hpvm/guest-images/hpux/11iv3/hpvm_guest_depot.11iv3.sd

Complete instructions for installing the guest management software are in the README.txt file in /opt/hpvm/guest-images/hpux directory.

HP-UX guests reboot as part of the guest management software installation process.

4.5 Troubleshooting HP-UX Guest Creation

The following section describes a problem that might occur during HP-UX guest installation.

4.5.1 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.

5 Creating HP OpenVMS Guests

The following sections contain the release notes specific to OpenVMS guests.

For information about installing OpenVMS guests, see the *HP OpenVMS V8.4 for Integrity Servers Upgrade and Installation Guide* at the following website: OpenVMS Documentation.

5.1 Minimum Processor Requirement for OpenVMS Guests

OpenVMS guests are supported on Integrity VM Host systems with Intel® Itanium® 2 9000/9100 Series processors or later processors that OpenVMS supports natively.

The following example shows the results of creating or starting a guest on a processor that does not support OpenVMS:

```
# hpvmstart -P g1
HPVM guest g1 configuration problems:
    Warning 1: OpenVMS guests not supported on this host's CPU.
These problems may prevent HPVM guest g1 from starting.
```

If you do not specify the OpenVMS operating system type, `-O` **`openvms`**, when booting an OpenVMS ISO installation media, you receive the following:

```
# hpvmconsole -P g1
[g1] vMP> co
fs2:
fs2:\> \efi\boot\bootia64
................................................................................
.....................................................
%VMS_LOADER-W-Unable to identify primary console device.
> Select a primary console from the Boot Configuration Menu.
> Attempting to continue with a default console device.

*** VM shutdown ***
*** OpenVMS not supported ***
```

5.2 Minimum VM Host Page Size

The OpenVMS guest might have problems booting if one or more of the following occurs:

- The VM Host is under memory pressure due to frequent allocations and freeing large amounts of memory.
- The VM Host has just enough physical memory to support the guest's requirements and the VM Host's base_pagesize is set to 4K.

OpenVMS expects a guest pagesize of 8K, and the boot processing can have issues loading an in-memory disk used during the boot process. If either of the following situations occur, setting the VM Host's base_pagesize to 64K or setting the guest's preferred pagesize to 8K should resolve the problem:

- The following message is written to the VM Host's `/var/opt/hpvm/common/hpvm_mon_log` `file`:

  ```
  #  WARNING: Host memory is fragmented.
  #           Reducing VHPT preferred page size from 64K to K.
  #           This may result in degraded performance for this virtual machine.
  ```

  Where # is a guest vm number assigned by hpvmdvr.

- Depending on how fragmented and how small the VM Host pagesizes are, the following OpenVMS error message and text appear on the guest’s console:

  ```
  %SYSBOOT-F-LDFAIL, unable to load SYS$PUBLIC_VECTORS.EXE, status = 00000044


  ************************************************************
  * Exception taken before exception handler has been loaded!  *
  ```

```
* Unable to take crashdump.                                  *
**************************************************************

* Exception Frame Display: *
…
```

Crash dump information follows this output.

Use one of the following solutions to fix either of these issues:

- Set the VM Host base_pagesize = 64K (See the base_pagesize(5) manpage for details of determining and setting the VM Host's base_pagesize.)
- Set the guest preferred pagesize to 8K:

  `# hpvmmodify -P vm-name -x tunables=ptsz=13`

5.3 Guest Device Placement and Adapter Limitation

The OpenVMS guest utilizes the PKDRIVER SCSI port driver and the DKDRIVER SCSI class driver. A guest configuration uses a UNIX-like algorithm to place storage devices on an AVIO storage adapter starting with entry “0,0” and continuing through “7,7”, where each AVIO storage adapter can have up to 128 targets. The default device placement algorithm places 14 devices on each adapter, in order. Thus, when the guest creation (`hpvmcreate`) or modification (`hpvmmodify`) utility attempts to add a device (using the `–a` option), it adds, in order, from “0,0,0” through “0,0,14” before creating a new adapter and starting over again with “0,1,0” through “0,1,14”. You can specify usage of any target value from 0 through 127.

Each Integrity VM adapter corresponds to the OpenVMS controller letters A through Z. The exact mapping depends on the order of devices found in the guest configuration file. Typically, “0,0” maps to PKA/DKA, while “0,1” maps to PKB/DKB, and so on. The following are known restrictions:

- The guest target value is a pure number assigned as the unit number of the device. The guest target value is not the same as a hardware SCSI target, where the disk in the 0th slot would be DKA0, the disk in the 1st slot would be DKA100, the disk in the 2nd slot would be DKA200, and so forth. Furthermore, it is not possible to assign the ‘lun’ number for the storage devices, such as DKA101. Placing a device at “0,0,1” results in the guest device “DKA1:”.
- unique DKDRIVER SCSI class adapters can be created. OpenVMS identifys them as DKAn: through DKPn:, where ‘n’ is the target value. Attempts to use a 17th adapter results in a port class PKQ0: being created with no corresponding DKQn: devices. The Integrity VM guest creation (`hpvmcreate`) and modification (`hpvmmodify`) utilities are not aware of this restriction.

You can add devices dynamically by using the `hpvmmodify` command on the VM Host to add the storage in conjunction with the OpenVMS command `sysman io autoconfigure all` on the guest to find the storage.

5.4 OpenVMS System Dump Analyzer (SDA)

Using the OpenVMS SDA command `CLUE CONFIG` the first time results in a `CLUE-W-NOSYMBIOS, cannot access SYMBIOS table` warning. Subsequent `CLUE CONFIG` commands display incorrect data. Additionally, using `EXAM/PHYS FE000` to view the guests SMBIOS data results in the message `%SDA-E-NOREAD, unable to access location 00000000.000FE000`. This issue seems to affect the SDA utility.

5.5 Formatting SYSTEM UUID

The VM Host (`hpvmstatus`) command and the guest (F$GETSYI(“system_uuid)) output the UUID in a different order, due to operating system conventions. To convert the OpenVMS version to the VM Host version, use the following DCL code:

```
$! Format and print a UUID
$uuid = f$getsyi("system_uuid")
$len=f$len(uuid)
$if (len .eq. 32)
$then
$    part1 = f$fao("!AS",f$ext(24,8,uuid))
$    part2 = f$fao("!AS",f$ext(20,4,uuid))
$    part3 = f$fao("!AS",f$ext(16,4,uuid))
$    part4 = f$fao("!AS!AS",f$ext(14,2,uuid),f$ext(12,2,uuid))
$    part5 = f$fao("!AS!AS!AS!AS!AS!AS", -
               f$ext(10,2,uuid),f$ext(8,2,uuid), -
               f$ext(6,2,uuid),f$ext(4,2,uuid), -
               f$ext(2,2,uuid),f$ext(0,2,uuid))
$    pr_uuid = f$edit("''part1'-''part2'-''part3'-''part4'-''part5'", -
                       "lowercase")
$
$    write sys$output "SYSTEM_UUID=''pr_uuid'"
```

5.6 OpenVMS Guest Notes

The following notes apply to OpenVMS guests using Integrity VM V4.3:

- Packaging OpenVMS guests as Serviceguard packages is not supported.
- OpenVMS does not support checksum offloading. Turning on CKO on the physical device will cause failure of communications in the following scenarios:
  - VM Host to OpenVMS guest and OpenVMS guest to VM Host
  - OpenVMS guest to a non-OpenVMS guest on the same VM Host

  To workaround this problem, turn off the Checksum offloading in the VM Host interface driver and restart the associated virtual switch. For example:
  - Turn off the CKO on PPA 4 by entering this command on the VM Host:

    `# nwmgr -s -A tx_cko=off –c lan4`
  - Restart the virtual switch by entering this command on the VM Host:

    `# hpvmnet -r -S switch1`

6 Creating Virtual Storage Devices

This chapter describes what Integrity VM storage is, how to configure it, and how to use it. The topics included in this chapter are:

- "Introduction to Integrity VM Storage"
- "Configuring Integrity VM Storage"
- "Using Integrity VM Storage"

6.1 Introduction to Integrity VM Storage

The way you configure and manage Integrity VM storage affects the way virtual machines perform. To get the most benefit from using virtual machines, learn how Integrity VM makes storage devices available to virtual machines. The following sections describe:

- "Integrity VM Storage Goals"
- "Integrity VM Storage Architectures"
- "Integrity VM Storage Implementations"

6.1.1 Integrity VM Storage Goals

To successfully configure and manage virtual storage, it is helpful to understand the basic goals of the Integrity VM storage subsystem, including:

- "Storage Utilization"
- "Storage Availability"
- "Storage Performance"
- "Storage Security"
- "Storage Configurability"

6.1.1.1 Storage Utilization

The main purpose of Integrity VM is to increase system resource utilization on Integrity servers. The Integrity VM storage subsystem meets this goal by permitting multiple virtual machines to share a variety of physical storage adapters and devices that are available on an Integrity server. Furthermore, the Integrity VM storage subsystem allows for a single storage LUN on the VM Host to be carved up into smaller entities that can be used as separate individual disks or DVDs on the virtual platform.

6.1.1.2 Storage Availability

Like HP Integrity servers, it is expected that virtual machines will have several different storage device types available for use. The Integrity 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 for a guest OS to interact with. Because a guest OS only accesses Integrity VM virtual hardware, the guest OS can use physical hardware that it does not support on an Integrity server.

6.1.1.3 Storage Performance

Each release of the Integrity VM storage subsystem strives to improve performance. Performance is improved in each release by lowering costs of virtualization, exploiting new features in the VM Host, and tuning operating systems for the virtual platform. At the same time, Integrity VM provides more virtualization choices to VM Host administrators, so that they can find the best balance between virtualization and performance to meet their needs.

6.1.1.4 Storage Security

To avoid problems while supporting multiple virtual machines on one physical machine, Integrity VM isolates each virtual machine. Using Integrity VM commands, the VM Host administrator determines the physical storage resources that each virtual machine can access. This storage isolation is maintained by the Integrity VM storage subsystem through DMA boundary checks on each virtual machine I/O operation, thereby ensuring that one virtual machine does not access the memory of another.

6.1.1.5 Storage Configurability

VM Host administrators expect the virtual machines to be as easily configurable as HP Integrity servers. The Integrity VM storage subsystem allows for easy changes of the storage devices through Integrity VM commands. Using these commands, the VM Host administrator dynamically adds, deletes, and modifies storage devices on virtual machines. Guest administrators can change some storage, limited in scope by the VM Host administrator, using the virtual console.

6.1.2 Integrity VM Storage Architectures

To provide the flexibility required to meet a variety of data center needs, the Integrity VM storage subsystem consists of two storage architectures, shared I/O and attached I/O.

6.1.2.1 Shared I/O

The shared I/O architecture is a means by which a virtual machine accesses an entirely virtualized storage subsystem provided by Integrity VM. The Integrity VM storage subsystem emulates real hardware to the virtual machine while interacting with the VM Host to complete the virtual machine I/O operation to the VM Host storage entity. This abstraction provides the ability of a VM Host administrator to share physical VM Host 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 VM Host LUN into smaller parts, like logical volumes, or files. Each of these sub-LUN VM Host entities can then be used as media for separate virtual storage devices. Virtual machines access the virtual storage devices as real storage devices, with no knowledge that the virtual storage media is actually a sub-LUN VM Host entity.

The way the virtual storage media is accessed by the Integrity VM storage subsystem allows virtual machines to share physical VM Host storage adapters. All virtual storage media is accessed through user-defined interfaces on the VM Host. The VM Host maintains complete control of the physical hardware and handles the virtual machine 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 VM Host, virtual machine 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 VM Host LUN. It only has to be able to support the virtualized version of it. Thus by using shared I/O, a virtual machine can run with any physical hardware that is supported by the VM Host.

Finally, all virtual machine 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 Integrity VM storage subsystem calls to handle communication of virtual machine 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 virtual machine does not know the difference.

6.1.2.2 Attached I/O

Attached I/O allows a virtual machine to access to a VM Host LUN directly. In this architecture, the Integrity VM storage subsystem attaches a LUN on the VM Host 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 VM Host and all the storage connected to those adapters may be shared among virtual machines. In attached I/O, only the storage adapter is virtualized. Therefore, only the VM Host physical storage adapters may be shared. At least one LUN, the attached LUN, cannot be shared. It is owned and solely controlled by the virtual machine it is attached to.

To provide the VM with complete control over attached devices, the Integrity VM storage subsystem interprets I/O requests from the guest device drivers into I/O requests that can be completed by the VM Host storage subsystem on the guest's behalf. In the process, the VM Host storage subsystem sends all the actual data and responses back the guest device drivers. With all this data, the guest device driver is in complete control over the device. As such, the guest OS must have built-in support for the attached VM Host 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.

6.1.3 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), have the following benefit over attached devices configured with VIO (`scsi` adapter):

- Allow sharing of tapes, changers, and burners among multiple guests and host
- Support of USB 2.0 DVD burners
- Improved performance

The resource specifier for attached devices using AVIO (`avio_stor` adapter) is different from the VIO (`scsi` adapter) resource specifier. See Section 6.1.3.1 (page 87) for the new syntax.

With VIO (*scsi* adapter type), USB CD/DVD devices are not supported for use as attachable media. AVIO (*avio_stor* adapter type) supports USB 2.0 DVD burners.

To identify USB CD/DVD devices, use the `ioscan -fun` command.

NOTE: Because Integrity VM may 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:scsi::disk:/dev/rdisk/disk5
# hpvmcreate -P guest -a dvd:scsi::null:/dev/rdisk/disk5
# hpvmmodify -P guest -a dvd:scsi::disk:/dev/rdisk/disk5
# hpvmmodify -P guest -a dvd:scsi::null:/dev/rdisk/disk5
```

6.1.3.1 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`

In contrast, the resource specifier for the same case with the `scsi` adapter looks like this:

`tape:scsi:0,4,0:attach:/dev/pt/pt_tape1`

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 VM Host) access to the attached device path is denied.

To know which lunpath is being used by a guest when the `scsi` (VIO) adapter is used, use the `scsimgr` command with the `get_info` option specifying the tape device special file. Look for the line "LUN path used when policy is path_lockdown" to see the chosen lunpath hardware path.

6.1.3.2 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*. The DSF for attached devices with the `scsi` adapter is /dev/pt/pt_tape1.

6.1.3.3 Sharing an Attached Device

Attached devices can be shared among multiple guests in a VM Host using a single physical HBA port (initiator) or multiple physical HBA ports (initiators) in the VM Host. 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 VM Host using a single HBA port (initiator). It has one lunpath through initiator (0/5/0/0/0/0). Device `tape6` is connected to the VM Host 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 Integrity VM- generated alias for the hardware path. The Integrity 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 guest:

```
# 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
```

6.1.3.3.1 Sharing Conflicts

You cannot share a device when a mixture of VIO and AVIO usage is specified; however, you can configure two guests with one guest using a device with VIO access and the other guest using a device with AVIO access.:

```
# hpvmmodify -P guest1 -a tape:scsi::attach:/dev/pt/pt_tape6
# hpvmmodify -P guest2 -a tape:avio_stor::attach_path:0/4/1/0.0x500104f0004732d9.0x0
```

Any attempt to boot both guests at the same time results in the second guest being blocked from booting. Setting the value of either `SHARE` or `SHARE_LUNPATHS` attributes to YES has no effect on this restriction:

```
# hpvmstatus
[Virtual Machines]
  Virtual Machine Name VM #  OS Type State     #VCPUs #Devs #Nets Memory  Runsysid
  ==================== ===== ======= ========= ====== ===== ===== ======= ========
  guest1                   2 UNKNOWN On (EFI)       1     1     0  100 MB        0
  guest2                   3 UNKNOWN Off            1     1     0  100 MB        0

# hpvmstart -P guest2
hpvmstart: ERROR (host): Device: '/dev/rtape/tape6_BESTn' is in use by another guest.  Lunpath cannot be used
at the same time that the device special file is in use by a guest.
  HPVM guest guest2 configuration problems:
      Warning 1 on item 0/4/1/0.0x500104f0004732d9.0x0: Device file '0/4/1/0.0x500104f0004732d9.0x0' in use by
 another guest.
  These problems may prevent HPVM guest guest2 from starting.
  hpvmstart: Unable to continue.
```

The following example shows that two guests are configured with the same tape device, each using a different hardware path, where both hardware paths use the same initiator:

```
# hpvmstatus -d -P guest1 | grep tape
tape:avio_stor:0,0,0:attach_path:0/4/1/0.0x500104f0004732d9.0x0
# hpvmstatus -d -P guest2 | grep tape
tape:avio_stor:0,0,0:attach_path:0/4/1/0.0x500104f0004732da.0x0
```

List devices configured using tape 6 (Note that SHARE_LUNPATHS=YES.):

```
# hpvmdevmgmt -l all | grep lunpath | grep tape6

/dev/rtape/tape6_BESTn,/dev/pt/pt_tape6:CONFIG=gdev,EXIST=YES,SHARE=NO,DEVTYPE=ATTACH,SHARE_LUNPATHS=YES:
lunpath6,lunpath5:0x01.0x00.0x03.0x500104f0004732d8

 0/4/1/0.0x500104f0004732d9.0x0,lunpath3:CONFIG=gdev,EXIST=YES,SHARE=NO,DEVTYPE=ATTACHPATHLUN,AGILE_DSF=
/dev/rtape/tape6_BESTn:guest1:0x01.0x00.0x03.0x500104f0004732d8_lunpath3

 0/4/1/0.0x500104f0004732da.0x0,lunpath5:CONFIG=gdev,EXIST=YES,SHARE=NO,DEVTYPE=ATTACHPATHLUN,AGILE_DSF=
/dev/rtape/tape6_BESTn:guest2:0x01.0x00.0x03.0x500104f0004732d8_lunpath5
```

The following example shows that one guest is running:

```
# hpvmstatus | grep guest
guest1                   2 UNKNOWN On (EFI)        1     1     0  100 MB        0
guest2                   3 UNKNOWN Off             1     1     0  100 MB        0
#
```

The following example shows that the second guest cannot be started:

```
# hpvmstart -P guest2
hpvmstart: ERROR (host): Device: '0/4/1/0.0x500104f0004732da.0x0' associated with guest: 'guest2' conflicts
with device:
'0/4/1/0.0x500104f0004732d9.0x0' associated with guest: 'guest1'.  Both guests cannot be booted at the same
time because both
devices use the same initiator, even though SHARE_LUNPATHS is set to YES.
  HPVM guest guest2 configuration problems:
      Warning 1 on item 0/4/1/0.0x500104f0004732da.0x0: Device file '0/4/1/0.0x500104f0004732da.0x0' in use by
 another guest.
  These problems may prevent HPVM guest guest2 from starting.
  hpvmstart: Unable to continue.
```

NOTE: After the user adds more than one attached lunpath to a particular device to one or more guests, each device database lunpath entry has an AGILE_DSF attribute that points to the agile-named parent DSF. If the user wants to allow guests to boot at the same time when they use different lunpaths to the same device, `SHARE_LUNPATHS=YES` is set on the entry to which that `AGILE_DSF` points.

Two different guests can use different lunpaths to the same device at the same time (`SHARE_LUNPATHS=YES`) if the paths use DIFFERENT initiators; however, Integrity VM blocks the use of different guests using different lunpaths at the same time if they use the SAME initiator. `SHARE_LUNPATHS=YES` can still be set when potential initiator-sharing conflicts exist. The `hpvmstart` command blocks the guests from booting when the same-initiator conflict (of using different paths) exists.

NOTE: If the guest operating system version is prior to HP-UX 11i v3 0809 (that is, 11i v3 0709 or 0803), a guest LUN might not appear after a delete and add. For example, this issue might occur with the following sequence of events::

1. In the guest, execute rmsf for the lun path (lunpath Class in ioscan display).
2. In the VM Host, delete the lun entry from the guest's configuration using the `hpvmmodify` command.
3. In the VM Host, add the lun entry back to the guest's configuration using the `hpvmmodify` command.
4. In the guest ioscan, the lun (which was deleted and added back) does not appear.

Perform the following workaround:

1. In the guest, rmsf the target path (tgtpath Class in ioscan display) corresponding to the lun path.
2. In the guest, perform an ioscan.

6.1.3.4 VIO to AVIO Migration with Attached Devices

Integrity VM does not support automatic migration of a virtual *scsi* (VIO) adapter containing attached devices to *avio_stor* (AVIO) adapter. To do the migration manually, follow these steps:

1. Shut down the guest.
2. Identify the virtual HBA that needs to be migrated using the `hpvmstatus` command.
3. Make note of the attached devices under the virtual HBA.
4. Delete the attached devices under the virtual HBA using the `hpvmmodify` command.
5. Migrate the remaining non-attached devices on the adapter by running the `hpvmmodify` command with the `-m hba` option.
6. Find the adapter specific device part of the resource specifier. See Section 7.1.3.1 (page 95).
7. Add previously deleted attached devices using the `hpvmmodify` command.

For example, to migrate from VIO to AVIO for guest1, do the following:

```
# hpvmstatus -P guest1 -d
:
[Storage Interface Details]
disk:scsi:0,2,0:lv:/dev/vg00/rlvol1
disk:scsi:0,2,1:disk:/dev/rdisk/disk28
tape:scsi:0,4,0:attach:/dev/pt/pt_tape1
```

```
disk:scsi:0,4,1:lv:/dev/vg01/rlvol1
disk:scsi:0,4,2:lv:/dev/vg01/rlvol2
disk:scsi:0,4,3:lv:/dev/vg01/rlvol3
disk:scsi:0,4,4:disk:/dev/rdisk/disk20
dvd:scsi:0,4,5:file:/fdev/0x500000e012afb8d2/file2
:

# hpvmmodify -P guest1 -d tape:scsi:0,4,0:attach:/dev/pt/pt_tape1
# hpvmmodify -P guest1 -m hba:avio_stor:0,4
# ioscan -m lun /dev/pt/pt_tape1
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

# hpvmmodify -P guest1 -a tape:avio_stor:0,4:attach_path:0/7/1/1.0x500104f00048b29e.0x0
```

To migrate from AVIO back to VIO for guest1, do the following:

```
# hpvmstatus -P guest1 -d
:
[Storage Interface Details]
disk:scsi:0,2,0:lv:/dev/vg00/rlvol1
disk:scsi:0,2,1:disk:/dev/rdisk/disk28
tape:avio_stor:0,4,0:attach_path:0/7/1/1.0x500104f00048b29e.0x0
disk:avio_stor:0,4,1:lv:/dev/vg01/rlvol1
disk:avio_stor:0,4,2:lv:/dev/vg01/rlvol2
disk:avio_stor:0,4,3:lv:/dev/vg01/rlvol3
disk:avio_stor:0,4,4:disk:/dev/rdisk/disk20
dvd:avio_stor:0,4,5:file:/fdev/0x500000e012afb8d2/file2
:
# hpvmmodify -P guest1 -d tape:avio_stor:0,4,0:attach_path:0/7/1/1.0x500104f00048b29e.0x0
# hpvmmodify -P guest1 -m hba:scsi:0,4
# 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
# hpvmmodify -P guest1 -a  tape:scsi:0,4:attach:/dev/pt/pt_tape1
```

6.1.3.5 Limitations

Although SCSI devices appear to a guest as Ultra320 SCSI controllers claimed by the MPT driver, this is an emulation. There are several differences from using a real device. Specifically:

- You cannot upload or download firmware for emulated devices.
- Although HP-UX commands such as *mptutil*(1M) and *mptconfig*(1M) do not fail when run in a guest, they do not always return the same information as they would when referencing a physical device.
- The EFI `drvcfg` command does not fail when run in a guest, but it returns no useful data.

The AVIO attached devices feature has the following limitations:

- Integrity VM does not support the `hpvmmodify -m hba` option when attached devices are present under the adapter. The `hpvmmodify -m hba` option allows you to change the adapter type from *scsi* to *avio_stor* and vice versa. Remove the attached devices from the adapter prior to issuing the command. For information about manual migration, see Section 6.1.3.4 (page 91).
- If you attempt to modify a storage adapter from `scsi` to `avio_stor` on a port for a running guest, the `hpvmmodify` command allows the change, but the change lasts until the next guest startup. In addition, the `hpvmnet` command displays incorrect port information for the currently running guest until the guest is stopped and restarted.
- If the GuestAVIOStor bundle is not installed on the HP-UX guest, any configured AVIO Stor HBAs will not be claimed in the guest, and the LUNs configured under the AVIO Stor HBAs will not be accessible. If the LUN is a boot disk, boot will fail with a panic indicating missing drivers.
- If a backing store is not responsive due to device errors, it might take up to 30 seconds on an HP-UX 11i v2 guest to report a failure. For example, the `diskinfo` command might fail after 30 seconds:

```
# timex diskinfo /dev/rsdk/c1t14d0:
diskinfo: can't open /dev/rdsk/c1t14d0: No such device or address
    real     0m30.26s
    user     0m0.00s
    sys      0m0.01s
```

This delay is caused by the retry of failed commands from the nonresponding backing store. There is currently no workaround.

- Devices configured under AVIO Stor HBA for a guest cannot be deleted (using the `hpvmmodify` command) if the guest is at EFI.

  Stop the guest using the `hpvmstop` command and retry the `hpvmmodify` command.

  Devices configured under AVIO Stor HBA for an HP-UX 11i v3 guest cannot be deleted (using the `hpvmmodify` command) if the guest is online.

  Run `ioscan -kfNC tgtpath` or `ioscan -kfNC lunpath` from the guest to obtain the `tgtpath` or `lunpath` H/W Path for the device to be deleted. Remove the device by using `rmsf -H` of the `lunpath` or `tgtpath` H/W Path from the guest and retry the `hpvmmodify` command from the host.

6.1.3.6 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 VM Host either at the EFI console or at the HP-UX operating system.

The following example shows the output of `hpvmstatus` from the Integrity VM Host:

```
# 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/rdsk/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 VM Host `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/rdsk/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/rdsk/c0t6d1
   Tue Oct  2 13:35:32 2007

   Lun DSF                                            : /dev/rdsk/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 VM Host `hpvmstatus` display using the `ioscan` utility:

```
# ioscan -m dsf /dev/rdsk/c0t6d1
Persistent DSF Legacy DSF(s)
========================================
/dev/rdisk/disk22 /dev/rdsk/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
0/0/2/0.0x16.0x0
/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
              0/0/2/0.0x16.0x0
```

6.1.3.7 Multi-Guest Boot Conflict with DVDs

When one guest has a DVD device configured as a VIO DVD and another guest has the DVD configured as an AVIO burner, both guests should not boot at the same time, but under certain circumstances, the boot-block check fails. If the AVIO guest boots first, the boot-block check fails and allows the VIO guest to boot when the AVIO guest is booted. In addition, if the VIO DVD is in the state where the backing store type is `null`, the boot-block check fails. This occurs in either one of the following cases:

- The device is first added to the guest and the backing store is set to `null`.
- The device is added to a guest with the disk backing store, which is later modified to be `null`. This can happen either by an `hpvmmodify` command specified by the user, or when a DVD is ejected from the guest console. The eject sets the backing store to `null`.

NOTE: The boot-block check works properly if the VIO guest boots first, and any attempt to boot the AVIO guest is blocked.

Setting the `SHARE` attribute to `YES` does not fix the problem, because it is invalid to allow simultaneous VIO and AVIO access to the same device. `SHARE=YES` is valid only in the context of multiple guests using the same access method to the device (VIO and AVIO, but not both at the same time).

6.1.3.8 Minimum Required Software Depot Versions

Table 7–1 lists the minimum required software depot versions for AVIO attached devices.

Table 17 Required Depot Versions

Product	HP-UX Version	Software Depot Version
Integrity VM	11i v3	B.04.30.00 or B.04.20.05 (with the required patches). For more information, see the online version of the *HP Integrity Virtual Machines Release Notes* for your specific release.
HostAVIOStor	11i v3	B.11.31.1103 for Integrity VM B.04.30.00 B.11.31.0910.01 for Integrity VM B.04.20.05
GuestAVIOStor	11i v3	B.11.31.0906
GuestAVIOStor	11i v2	B.11.23.0903

6.1.3.9 Patch Dependency

Table 7–2 lists the patch dependencies for the AVIO attached devices features.

Table 18 Patch Dependencies for AVIO Attached Devices

Patch Number	HP-UX Version	VM Host	Guest	Notes
PHKL_38604	11i v3	Yes	Yes	Hard[1] dependency for guest, and soft[2] dependency for VM Host.
PHKL_38605	11i v3	Yes	No	Soft dependency on VM Host.
PHKL_38750	11i v3	Yes	Yes	Recommended patch.

1 Enforced during `swinstall`.

2 Required only if attached devices are configured. No enforcement using `swinstall`.

6.1.3.10 Error Messages

This section lists possible VM Host and guest error messages and their description.

- VM Host error messages
  - Access error on a shared attached device

    The VM Host'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 VM Host 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 VM Host 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
    ```

6.1.4 Integrity VM Storage Implementations

This section describes the implementations of the Integrity VM storage architectures.

6.1.4.1 Integrity VM Storage Adapters

Integrity VM provides two types of virtual PCI storage adapters to process virtual storage I/O requests:

- Virtual parallel SCSI MPT adapter
- Accelerated Virtual I/O (AVIO) adapter

All supported guest operating systems contain native MPT SCSI adapter drivers that communicate with this PCI register emulation. All virtual and attachable devices can be used with this single virtual storage adapter.

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 VM Host 11i v3 storage stack features to provide better storage manageability in the guest. VxVM is also supported as an AVIO backing store. HP recommends the use of AVIO for the supported guest operating systems, guest devices and host backing stores. See the AVIO documentation for details.

6.1.4.2 Sample Script for Adding Multiple Devices at Once

To add 256 AVIO storage devices to a guest, 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 235).

6.1.4.3 Integrity VM Storage Devices

Integrity VM supports a variety of virtual and attachable devices. Disk and DVD-ROM devices support several virtual media types (see Section 6.1.4.3.1 (page 97)). Physical tapes, media changers, and CD/DVD burners are attachable; they can be used to perform data backups directly from a virtual machine (see Section 6.1.4.3.2 (page 97)).

6.1.4.3.1 Virtual Devices

Integrity VM supports the following virtual disk types:

Virtual Disk Type	Backing Storage Device	For more information, see...
Virtual Disk	VM Host disk	"Virtual Disks" (page 106)
Virtual LvDisk	VM Host LVM or VxVM logical volume.	Section 6.2.2.3.2 (page 107)
Virtual FileDisk	VM Host VxFS file	Section 6.2.2.3.3 (page 109)

The following virtual DVD-ROM types are supported:

Virtual DVD Type	Backing Storage Device	Described in ...
Virtual DVD	Disk in a VM Host physical DVD drive	Section 6.2.2.3.4 (page 109)
Virtual FileDVD	ISO file on a VM Host VxFS file system	Section 6.2.2.3.6 (page 111)
Virtual NullDVD (empty)	VM Host physical DVD drive or VxFS directory	Section 6.2.2.3.3 (page 109)

6.1.4.3.2 Attached Devices

Integrity VM supports a suite of attached devices to complete data backups from a virtual machine. Integrity VM attaches these devices using a special Integrity VM pass-through driver. With this pass-through driver, virtual machine I/O requests are interpreted by Integrity VM and sent through the virtual storage subsystem to the physical device. The virtual storage subsystem sends device responses to the Integrity VM pass-through driver, which sends the responses to the virtual machine. Because the virtual machine 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 virtual machine 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.

6.2 Configuring Integrity VM Storage

This section describes how to plan and set up Integrity VM storage, including the following topics:

- "Integrity VM Storage Considerations"
- "Setting Up Virtual Storage"

6.2.1 Integrity VM Storage Considerations

When you configure storage for a virtual machine, consider the following:

- VM storage supportability
- Storage performance
- VM storage multipath solutions
- VM storage management
- VM storage changes
- Virtual storage setup time

The following sections explain each of these considerations.

6.2.1.1 VM Storage Supportability

Before you configure virtual machine storage, make sure the VM Host storage can be supported by the virtual machine.

- All VM Host storage available for use by a VM must meet support requirements for the Integrity server and OS version that comprise the VM Host. If the physical storage is not supported by the VM Host, it is not supported for use by a virtual machine.
- All VM Host storage available for use by a VM must be connected with a supported adapter and driver type. See the *HP Integrity Virtual Machines 4.3: 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 guest. Use the `ioscan` command to display the VM Host storage that is connected to adapters and drivers.
- Any VM Host attachable devices available for use by a guest 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 guest.

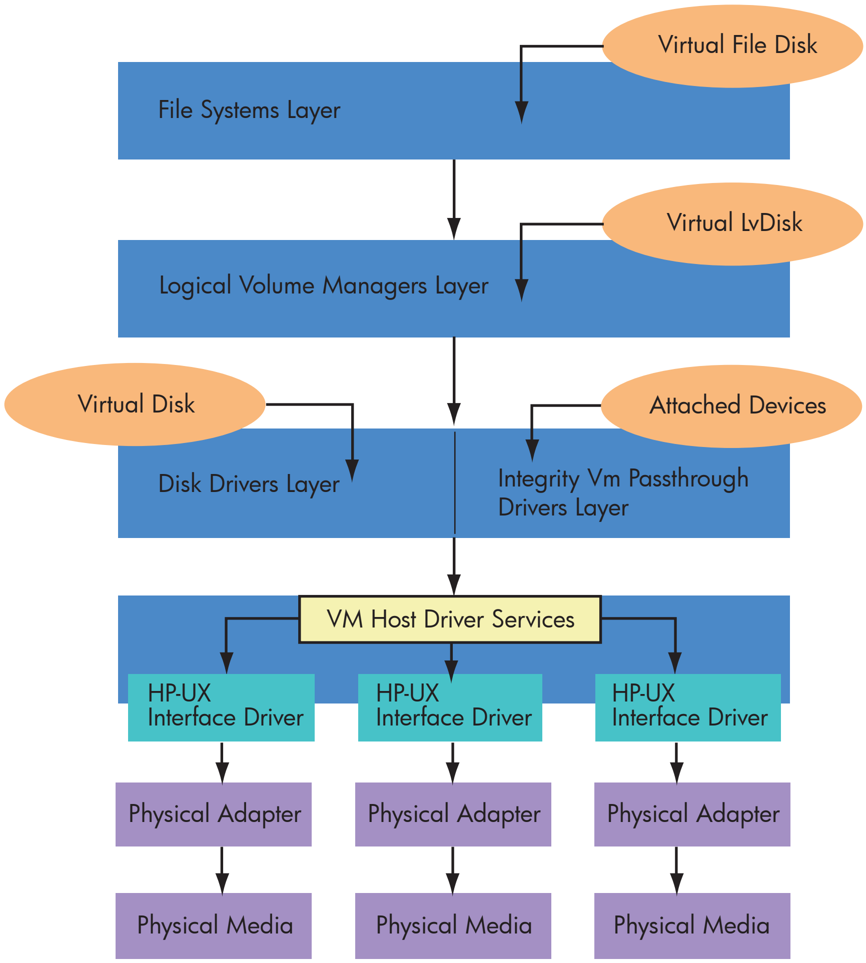
6.2.1.2 Performance of Virtual Devices

To meet the performance requirements of applications running in guests, consider the potential performance of each type of Integrity VM storage device.

Different types of virtual media have different effects on the performance of the virtual device because they communicate differently with the VM Host to complete virtual machine I/O operations.

To understand the effect of the virtual device type on potential performance, consider the Integrity VM storage I/O stack illustrated in Figure 3.

Figure 3 Integrity VM 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 VM Host physical storage device. The longer the path is, the longer it takes for virtual I/O to be completed. As shown in Figure 3, 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 VM Host I/O stack, the slower it operates.

The simplified I/O stack in Figure 3 does not completely illustrate all the choices that can affect the performance:

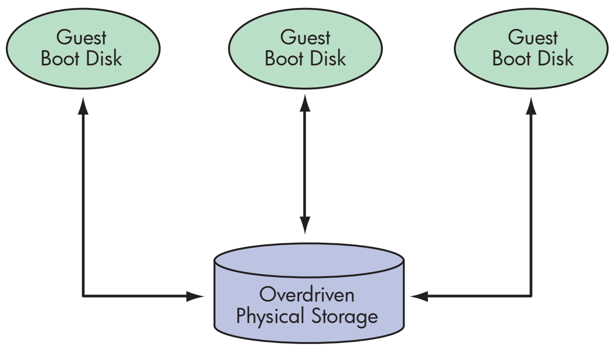
- Performance of different software layers differs.
- The interfaces to each software layer are different, allowing Integrity VM different ways to send I/O through the layers. For example, whole disks can achieve higher throughput rates than logical volumes and file systems.
- The I/O layer might have 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 VM Host, see the 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 Integrity VM, 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 4 shows a common mistake of overdriving physical storage with multiple guest OS boot disks, which are often I/O intensive.

Figure 4 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 VM Host, 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 3, all virtual I/O goes through a general VM Host I/O services layer that routes the virtual I/O to the correct VM Host 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 VM Host. For help with selecting a multipath solution for a virtual media type, see Section 6.2.1.3 (page 100).

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.

6.2.1.3 VM Storage Multipath Solutions

Integrity VM virtual devices support the built-in multipathing of the HP-UX 11i v3 VM Host, which is enabled by default to provide improved performance, load-balancing, and higher availability for 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 VM Host for the following reasons:

- The VM Host 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 virtual machine I/O with which it is competing on the VM Host (see Figure 3).
- Running a multipath solution in a virtual machine 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 VM Host. Errors in

communication to the virtual media are properly emulated as media errors sent to the guest OS, not as path failures.

- The VM Host does not return specific errors to Integrity VM for hardware path failures. Integrity VM does not detect such events and does not pass them to the virtual machine.

For supported multipathing configurations, see the *HP Integrity Virtual Machines 4.3: Release Notes*.

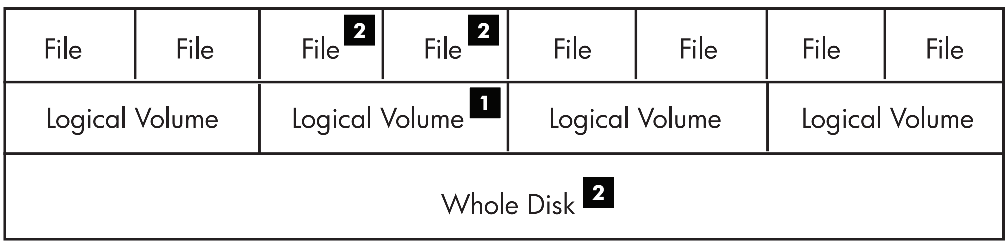
6.2.1.4 VM Storage Management

Before you decide how to divide VM Host storage, consider the impact on the management of the storage subsystem.

A VM Host administrator manages VM storage to make sure virtual media is allocated safely. This begins with understanding the VM Host I/O stack and knowing from where the virtual media is being allocated.

Figure 5 shows an example of a VM Host I/O stack as it applies to a single LUN.

Figure 5 Sub-LUN Storage Allocation Example



The virtual machine is allocated a logical volume from the LUN for a Virtual LvDisk.

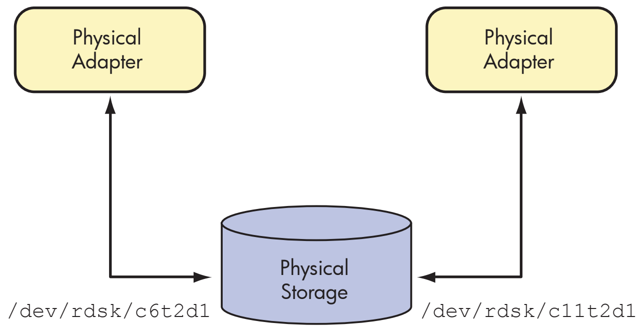
- The logical volume that has been allocated is labeled **1**.
- The parts of the disk that cannot be allocated are labeled **2**.

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.

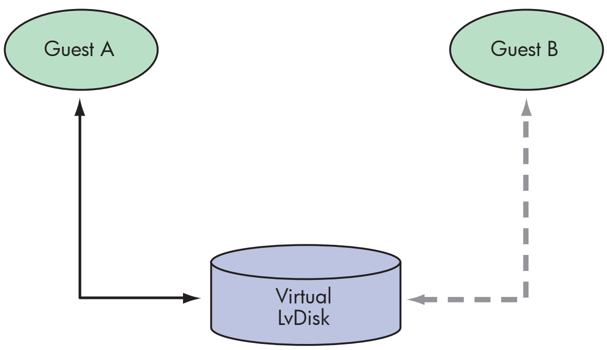
Beyond avoiding sub-LUN collisions, whole LUN collisions also need to be avoided. The same storage resource, virtual or attached, cannot be specified more than once to the same virtual machine. HP-UX 11i v3 supports both legacy per-path device files (for example, `/dev/rdsk/c6t2d0`) and agile non-path specific device files (for example, `/dev/rdisk/disk`). As shown in Figure 6, 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 6 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 7 shows a Virtual LvDisk being shared across virtual machines, which is not supported.

Figure 7 Bad Virtual Device Allocation



As these examples illustrate, it is important to know where storage is allocated from to avoid data corruption with virtual machines or even the VM Host. 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 VM Host administrators can see exactly which virtual machines are using each VM Host 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 VM Host 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 VM Host 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 VM Host system is rebooted. Take care to check that all SLVM volume groups are activated after a VM Host reboot or Serviceguard cluster reconfiguration.

6.2.1.5 VM Storage Changes

Depending on how you set up storage for a virtual machine, 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 VM Host file system commands.

No changes to any virtual media can take place on the VM Host 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 virtual machine is allocated virtual media for a virtual device, that virtual machine owns that media and can access it any time. VM Host 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 VM Host device is attached to the virtual machine, it is controlled and owned by that virtual machine. Modifications to the attached device, like changing a tape, can be done physically without detaching the device from the guest. However, such changes might need to be coordinated with the VM Host administrator, especially if the guest administrator has no physical access to the device attached to the virtual machine.

All types of virtual storage devices can be added and removed dynamically from virtual machines. 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 virtual machine when you add additional devices.

6.2.1.6 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 VM Host 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 8.12 (page 171)).

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 VM Host, use the `hpvmdevmgmt` command to quickly create the device files. Otherwise, see *scsi_ctl* for information about creating passthrough device files using *mknod*.

6.2.2 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 7), or virtual storage devices.

This section describes how to enter resource statements for use with the `hpvmcreate` command (described in Chapter 3) and the `hpvmmodify` command (described in Chapter 8). The resource statement specifies the virtual storage device that will be seen by the virtual machine and how it maps to the physical storage device on the VM Host.

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 virtual machine (see Section 6.2.2.1 (page 104))
- *`VM-Host-storage-specification`* defines where and how the virtual machine storage is supplied on the VM Host (see Section 6.2.2.2 (page 105))

For examples of how to construct resource statements, see Section 6.2.2.3 (page 105).

6.2.2.1 VM Guest Storage Specification

All virtual storage is addressed from virtual PCI buses. The Integrity VM virtual platform contains 8 PCI buses. Each PCI bus has 8 slots into which virtual PCI adapters can be placed. One such adapter, simply called `scsi`, is an emulated single-ported parallel SCSI MPT storage adapter that can be used to connect 15 SCSI target devices to a guest. An AVIO storage adapter is also supported. The AVIO storage adapter supports up to 128 devices per adapter (and host) and provides higher performance and guest storage manageability.

A VM Host administrator specifies this SCSI MPT adapter using the following:

`device``:scsi:`*`pcibus,pcislot,scsitgt`*

where:

- *`device`* is one of the following: `disk`, `dvd`, `tape`, `changer`, or `burner`
- *`pcibus`* is an integer from 0-6.

  The virtual AVIO and MPT adapters are 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 MPT storage adapter supports only a single channel.
- *`scsitgt`* is an integer from 0-14 for virtual SCSI (15 is reserved for the virtual SCSI adapter) and 0–127 for AVIO. All supported storage device types can share the same virtual SCSI MPT or AVIO adapter by specifying the same PCI bus and slot numbers. A virtual SCSI MPT or AVIO adapter can be added only to a virtual machine if it has a device connected to it.

  Unlike real parallel SCSI bus, there is no arbitration on virtual SCSI buses. The SCSI target IDs for the virtual devices must be unique.The virtual SCSI MPT adapter takes target ID 15 for itself, leaving 0-14 for SCSI targets.

  All SCSI targets connected to a 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 VM Host system files. The physical LUN number of an attached device has no impact. All virtual and attached SCSI LUN numbers are implicitly zero and therefore not specified.

  All supported storage device types can share the same virtual SCSI MPT adapter. Up to 15 storage devices can be added to the same SCSI MPT adapter by specifying the same PCI bus and slot numbers.

A virtual SCSI MPT adapter can be added only to a virtual machine 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 VM Host. Tapes, changers, and burners are physical VM Host devices. For these attached devices, the physical SCSI IDs do not determine their place on the virtual bus.

6.2.2.2 VM Host Storage Specification

Each VM storage device is backed by some VM Host storage entity. A VM Host entity is defined on the VM Host with a system file, which is used by Integrity VM and the VM Host operating system in processing I/O to and from that storage entity.

A VM Host administrator specifies these storage entities using the following specification:

`storage``:`*`location`*

where:

- *`storage`* is one of the following: `disk`, `lv`, `file`, `null`, or `attach`.

  The selection of storage type defines what VM Host system files apply. For example, `lv` implies the use of logical volume character device files.

  For virtual devices, the selection of VM Host 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 VM Host storage entity can only be used for one VM device type at a time. For example, a VM Host CD/DVD drive cannot be used for a Virtual DVD and an attached burner at the same time.

- *`location`* is a VM Host system file.

  The file permissions on the VM Host system file are not honored by Integrity VM. VM device types that support write operations can still do so using a VM Host 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-. Attached devices do not consider file permissions when backing up data.

  More than one VM Host system file might point to the same VM Host storage entity. For example, if multiple paths to storage are present on the VM Host, more than one disk system file can point to the same disk. Different VM Host 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 VM storage resources. A given VM storage resource can only be specified once to a given virtual machine. Therefore, only one VM Host system file per VM Host storage entity can be provided to a virtual machine (see Section 6.2.1.4 (page 101)).

Not all virtual device types support all VM Host storage types (see Section 6.1.4 (page 96)). Complete VM storage resource statements are discussed in the next section.

6.2.2.3 VM Storage Resource Statements

This section provides information about formulating complete valid resource statements for Integrity VM storage devices.

To specify an Integrity VM storage device for a virtual machine, use a complete valid resource statement with the `hpvmcreate` or `hpvmmodify` command. The resource statement is a combination of the VM guest resource specification (described in Section 6.2.2.1 (page 104)) and the VM Host Storage Specification (described in Section 6.2.2.2 (page 105)). 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 virtual machine can have up to 30 VIO devices or 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 virtual machine more than once (see Section 6.2.1.4 (page 101)). Unless otherwise noted, storage resources, virtual or attached, cannot be simultaneously shared by virtual machines.

All multipath products for storage resources must run on the VM Host; multipath solutions are not supported in a virtual machine. All multipath solutions used on the VM Host must be in valid supported configurations before being used for Integrity VM storage resources (see Section 6.2.1.3 (page 100)).

The resource statements in the following subsections do not contain VM hardware addressing. The PCI bus, PCI slot, and SCSI target numbers are optional.

6.2.2.3.1 Virtual Disks

A Virtual Disk is an emulated SCSI disk whose virtual media comes from a VM Host disk LUN. The VM Host 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 virtual machines (except in certain cluster configurations, as indicated in this manual). Only one active virtual machine at time can be given a particular Virtual Disk resource. Virtual Disk resources can be changed dynamically among active virtual machines.

To prevent virtual media conflicts that can result in data corruption, a proper accounting of how the VM Host whole disks are allocated for use by Virtual Disks needs to be done, as described in Section 6.2.1.4 (page 101).

The agile Virtual Disk resource statement takes the form:

`disk:scsi::disk:`*`/dev/rdisk/diskX`*

where `/dev/rdsk/diskX` is an HP-UX `esdisk` character device file.

The legacy Virtual Disk resource statement takes the form:

`disk:scsi::disk:`*`/dev/rdsk/cXtYdZ`*

`where /dev/rdsk/cXtYdZ` is an HP-UX `sdisk` character device file. Use of agile device names for configuring storage devices is highly recommended as it provides multipathing benefits.

These device files can be located for a VM Host 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 VM Host 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/rdsk/disk
```

6.2.2.3.2 Virtual LvDisks

A Virtual LvDisk is an emulated SCSI disk whose virtual media is provided by a raw VM Host logical volume. To specify a VM Host 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 virtual machines. Only one active virtual machine at time can be given a particular Virtual LvDisk resource. Virtual LvDisk resources can be changed dynamically between active virtual machines (see Section 6.3 (page 115)).

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 VM Host and guest , use only raw logical volumes that contain no VM Host file systems and are not currently mounted on the VM Host.

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 6.2.1.4 (page 101).

If you are using LVM, the Virtual LvDisk resource statement takes the following form:

`disk:scsi::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:scsi::lv:/dev/lvrackA/rdisk2`.

To use VxVM, the Virtual LvDisk resource statement takes the following form:

`disk:scsi::lv:`*`/dev/vx/rdsk/dg_name/v_name`*

where *`/dev/vx/rdsk/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:scsi::lv:/dev/vx/rdsk/VxvmTest1/vxvm_2`. For information about adapters that support VxVM, see the *HP Integrity Virtual Machines 4.3: Release Notes*.

For information about multipath solutions for Virtual LvDisks, see Section 6.2.1.3 (page 100).

6.2.2.3.3 Virtual FileDisks

A Virtual FileDisk is an emulated SCSI disk whose virtual media comes from a VM Host file. The VM Host file is specified using the absolute pathname to the file. The file can be on a VxFS file system locally mounted on the VM Host. NFS file systems are not supported for Virtual FileDisks.

Virtual FileDisks cannot be shared simultaneously across active virtual machiness. Only one active virtual machine can be given a particular Virtual FileDisk resource at a time. Virtual FileDisk resources can be changed dynamically between active virtual machines (see Section 6.3 (page 115)).

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 virtual machine. 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 6.2.1.4 (page 101).

The Virtual FileDisk resource statement takes the following form:

`disk:scsi::file:/pathname/file`

where the `/pathname/file` specifies the VM Host 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 VM Host 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:scsi::file:/fdev/frackA/disk1`.

Multipath options for a Virtual FileDisk device are discussed in Section 6.2.1.3 (page 100).

NOTE: Each Integrity VM guest can support a maximum of 30 Virtual FileDisks (combined limit for both AVIO and VIO).

6.2.2.3.4 Virtual DVDs

A Virtual DVD is an emulated SCSI DVD-ROM with virtual media that comes from a disc inside of a CD/DVD drive on the VM Host. The VM Host 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 VM Host CD/DVD drives prohibits them from being shared across active virtual machines. Thus only one active virtual machine at time should be given a particular Virtual DVD resource. Virtual DVD resources can be changed dynamically between active virtual machines (see Section 6.3 (page 115)).

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 virtual machine, make sure you know which virtual machine currently owns the device before you load a CD/DVD. This information can be found on the VM Host with the `hpvmstatus` commands.

The agile Virtual DVD resource statement takes the following form:

`dvd:scsi::disk:/dev/rdisk/disk#`

where `/dev/rdisk/disk#` is an HP-UX `esdisk` character device file for a VM Host CD/DVD drive. The legacy Virtual DVD resource statement takes the form `dvd:scsi::disk:/dev/rdsk/cXtYdZ`, where `/dev/rdsk/cXtYdZ` is an HP-UX `sdisk` character device file for a VM Host CD/DVD drive.

Typically, the HP-UX `esdisk` and `sdisk` character files will already be created before booting the VM Host. 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/rdsk/disk7
SCSI describe of /dev/rdsk/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:scsi::disk:/dev/rdisk/disk7`.

For a virtual machine to recognize a Virtual DVD, physical media must be present inside the VM Host CD/DVD drive. If media is not added at virtual machine start time, it can be inserted into the VM Host CD/DVD drive after the virtual machine is already up. A rescan by the guest OS picks up the new media and adds the Virtual DVD to the virtual machine.

If for some reason the VM Host Administrator requires control of the VM Host CD/DVD drive claimed by a virtual machine but has no media for the VM Host CD/DVD drive, then a Virtual NullDVD should be specified (see Section 6.2.2.3.6 (page 111)). Physical media can then be inserted into the VM Host CD/DVD drive and become virtual media for a Virtual DVD using the `hpvmmodify` or the virtual console's insert command (see Section 6.3.1.3 (page 117)).

After the Virtual DVD is in the virtual machine, the VM Host CD/DVD drive is locked. The VM Host CD/DVD drive is automatically unlocked when the virtual machine is shut down. The VM Host CD/DVD can also be changed while the virtual machine is up using the virtual console's `eject` command. Once ejected, the Virtual DVD turns into a Virtual NullDVD and the VM Host CD/DVD drive unlocks. After you place physical media in the VM Host's CD/DVD drive, use the virtual console's `insert` command to turn a Virtual NullDVD back to a Virtual DVD, relocking the VM Host CD/DVD drive.

Most physical VM Host CD/DVD devices on HP Integrity servers have only one path to them. As such, no multipath software is available on the VM Host for them.

6.2.2.3.5 Virtual FileDVDs

A Virtual FileDVD is an emulated SCSI DVD with virtual media that comes from a VM Host ISO file. The VM Host 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 VM Host. NFS file systems are not supported for Virtual FileDVDs.

The Virtual FileDVD resource statement takes the following form:

`dvd:scsi::file:`*`/pathname/file.ISO`*

where the *`/pathname/file.ISO`* specifies the VM Host ISO file to use as virtual media.

A VM Host 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 6.3.2.3 (page 120)). 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:scsi::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 6.2.1.3 (page 100) 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 6.2.1.4 (page 101)). 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.

6.2.2.3.6 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 VM Host CD/DVD drive or VM Host 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 6.2.2.3.4 (page 109)) or a Virtual FileDVD (see Section 6.2.2.3.5 (page 110)) 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 VM Host administrator gives to the guest administrator. Virtual DVD changes can be initiated from the virtual console (see Section 6.3.1.3 (page 117)). All virtual DVD changes by the guest administrator are constrainted by the actions of the VM Host administrator.

If the VM Host administrator gives access to the guest administrator to load and unload physical media on the VM Host CD/DVD drive, the Virtual NullDVD can be set up with the following form of the resource specification:

`dvd:scsi::null:`*`/dev/rdisk/disk#`*

where *`/dev/rdisk/disk#`* is an HP-UX `esdisk` character device file that points to the VM Host CD/DVD drive. The legacy sdisk device file, `/dev/rdsk/cXtYdZ`, may also be used.

This is the same as setting up a Virtual DVD (see Section 6.2.2.3.4 (page 109)), except that the VM Host CD/DVD might not contain media. The media is expected to come from the guest administrator, who should have access to the VM Host 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:scsi::null:/dev/rdisk/disk7`.

If the VM Host administrator does not want to give access to the VM Host 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:scsi::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 6.2.2.3.5 (page 110)), 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:scsi::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 VM Host 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 6.2.1.3 (page 100)). 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 8.12 (page 171).

Virtual NullDVDs require no additional management beyond that required for the Virtual DVD (see Section 6.2.2.3.4 (page 109)) or Virtual FileDVD (see Section 6.2.2.3.5 (page 110)) types they become.

6.2.2.3.7 Attachable Devices

Integrity VM allows you to attach physical VM Host backup device types to virtual machines. The VM Host backup device types are tapes, media changers, and CD/DVD burners. These devices are specified on the VM Host using HP-UX agile `esctl` or legacy `sctl` device files. Use of the agile `esctl` device files are recommended, because they are per physical device not per path, as legacy `sctl` device files are. When using legacy `sctl` device files, do not specify a physical device more than once to a virtual machine through different hardware paths. Information about the use of legacy `sctl` device files cane be found throughout this manual.

The guest OS running on the virtual machine 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:scsi::attach:`*`/dev/pt/pt_tape#`*

- For media changers:

  `changer:scsi::attach:`*`/dev/pt/pt_autoch#`*

- For CD/DVD burners:

  `burner:scsi::attach:`*`/dev/pt/pt_disk#`*

  where `/dev/pt/pt_*` files are HP-UX `esctl` device files.

Attachable devices can be specified as *`scsi`* or 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 VM Host:

   ```
   # 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 SCSI 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 VM Host. Integrity VM does not require the drivers to be loaded on the VM Host 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 VM Host 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 VM Host drivers, proceed to step 4.

If you are loading the VM Host drivers, the devices should show up in `ioscan` with device files after the VM Host 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 virtual machines. Only one active virtual machine can be given a particular attached device at a time. However, like virtual devices, attached devices can be attached and detached dynamically across active virtual machines (see Section 6.3 (page 115)). Also, as the device is being attached to a virtual machine, it cannot be opened by the VM Host 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 VM Host. No multipath products are supported in the virtual machine.

Manage attached devices to prevent the wrong virtual machines from viewing sensitive information. You can display which virtual machines are currently using attached devices using the `hpvmstatus` command.

6.2.2.4 NFS-Mounted Backing Stores for Root, Swap, and Dump

Integrity VM V4.3 support 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 may be removed or changed in future 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 VM Host 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 VM Host (NFS client) and the NFS server systems must reside in the same IP subnet.
- Online VM Migration is supported for guests using NFS-mounted backing stores. For OVMM to work successfully, both Integrity VM Hosts 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 VM Hosts 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 Integrity VM V4.3:

- 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 VM Host inside the NFS-mounted file system.

NOTE: Creating a guest's backing-store files on an NFS client system (that is, VM Host), 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.

6.3 Using Integrity VM Storage

The following sections describe the roles of individuals accessing virtual storage, the commands they use, and some examples of using Integrity VM storage.

6.3.1 Integrity VM Storage Roles

This section describes the roles that individuals play in working with Integrity VM storage. Each role has different responsibilities in using Integrity VM storage. The roles might be played by one or more individuals depending on security requirements and skill sets. The three roles are:

6.3.1.1 VM Host Administrator

The VM Host administrator role is an individual responsible for the proper configuration and maintenance of the VM Host for running virtual machines. As such, this person needs complete access to the VM Host 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 VM Host administrator uses the following commands to manage virtual machine storage devices:

Management Function	Integrity VM Command
Add, delete, manage, and modify virtual machine storage devices.	`hpvmmodify` (see Section 3.5 (page 64))
Display information about the storage devices for a virtual machine.	`hpvmstatus` (see Section 8.3 (page 151))

Once a resource is added or attached to a virtual machine and the virtual machine 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.

6.3.1.2 Creating Virtual Machine 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 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 VM Host 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 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 VM Host 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.

6.3.1.3 Guest Administrator

The 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 VM Host administrator to control the virtual machine. 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, Integrity VM storage requires you to:

- Install any specific guest OS patches required by Integrity VM for proper OS operation on the virtual platform.
- Review and understand any Integrity VM storage release notes that are specific to the guest OS.
- Work with the VM Host administrator to complete virtual storage changes, including managing attached VM Host 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 VM Host 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 VM Host 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 VM Host 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:scsi::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:scsi::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/rdsk/c2t0d0        /dev/rdsk/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 VM host. 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.

6.3.1.4 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 VM Host administrator.

6.3.2 Integrity VM Storage Use Cases

This subsection describes ways to use the Integrity VM storage commands.

6.3.2.1 Adding Virtual Storage Devices

A VM Host administrator adds or attaches Integrity VM storage using the `hpvmstatus` and `hpvmmodify` commands. Virtual storage devices can be added or attached while the virtual machine is powered on or off. A new virtual storage adapter can be added only when the virtual machine is off. The virtual storage adapter can have up to 30 VIO storage devices or 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 Integrity VM storage considerations, choose a storage device to add.
2. Based on the device type, set up and configure the VM Host to form a valid resource statement. This includes accounting VM Host resources to avoid future storage conflicts.
3. Use the valid resource statement with the `hpvmmodify` command to add or attach the Integrity VM storage device.

The resource statement for adding an Integrity VM storage device does not require virtual hardware addressing. If the PCI bus, slot and SCSI target numbers are not specified, Integrity VM automatically chooses the first position available for the device. For example:

```
# hpvmmodify -P myvmm -a disk:scsi::disk:/dev/rdisk/disk7
# 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
```

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 `scistgt` 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
```

6.3.2.2 Deleting VM Storage Devices

A VM Host administrator deletes or detaches Integrity VM storage using the `hpvmstatus` and `hpvmmodify` commands. Integrity 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 Integrity VM storage adapter is automatically removed when the last Integrity VM storage device connected to the adapter is removed.

The process to delete or detach a virtual storage device from a virtual machine is as follows:

1. Use the `hpvmstatus` command to locate the resource to verify whether the virtual machine is powered on. If the virtual machine 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 VM Host resource is no longer being used by the virtual machine.

The resource statement for deleting an Integrity VM storage device does not require virtual hardware addressing. 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
disk avio_stor 0 5 0 0 0 disk /dev/rdisk/disk11
# hpvmmodify -P myvmm -d disk:scsi::disk:/dev/rdisk/disk7
# hpvmstatus -P myvmm
...
[Storage Interface Details]

disk scsi 0 1 0 0 0 disk /dev/rdisk/disk5
disk scsi 0 1 0 2 0 disk /dev/rdsk/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
```

6.3.2.3 Modifying VM Storage Devices

The VM Host administrator or the guest administrator can modify an Integrity VM storage device. The VM Host 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 VM Host access.

When the VM Host 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 virtual machine is on, consult with the guest administrator to before proceeding to replace the virtual media.
2. Based on the Integrity VM storage considerations, choose a new virtual media type to add.
3. Based on the virtual media type, set up and configure the VM Host to form a valid VM Host storage specification. Take into account the other demands on VM Host resources to avoid virtual machine storage conflicts.
4. Use the VM Host storage specification with the `hpvmmodify` command to modify the virtual device resource.
5. Verify that the old VM Host resource is no longer in use by a virtual machine.
6. When run on an active virtual machine and with a storage device managed by `avio_stor` HBA, the 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 6.2.2.1 (page 104)). 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/disk7
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 VM Host administrator. Changes through the virtual console are not saved across guest OS reboots

If the VM Host administrator sets up a Virtual DVD for the virtual machine, the virtual console eject and insert command unlock and lock the physical VM Host CD/DVD drive. The `eject` command changes the Virtual DVD into a Virtual NullDVD in the VM, unlocking the VM Host CD/DVD drive in the process. The physical media in the VM Host CD/DVD drive can then be changed by the VM Host 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 VM Host CD/DVD drive and making the newly loaded media now accessible by the virtual machine. For example:

```
# diskinfo /dev/rdisk/disk7
SCSI 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 VM Host 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.

To modify all targets on a SCSI storage adapter to be targets on an AVIO storage adapter, specify the following:

```
host# hpvmmodify -P guest1 -m hba:avio_stor:0,5
```

For information about AVIO support, see the *HP Integrity Virtual Machines 4.3: Release Notes*.

This command string modifies an existing SCSI device adapter with some number of device targets to use the AVIO device adapter. The command has the following restrictions:

- Each of the backing store devices under the SCSI HBA must be supported AVIO devices and adhere to storage type limitations.
- All targets to be converted must list the same device adapter type.
- The command can be executed only on a guest that is stopped.
- Multiple adapter entries can be placed on the same command line. For example:

  ```
  hpvmmodify -P guest1 -m disk:avio_stor:0,5 -m hba:avio_stor:0,6
  ```

  The modification can also be reversed to the original SCSI adapter:

  ```
  hpvmmodify -P guest1 -m disk:scsi:0,5
  ```

  The following example shows how to convert `scsi` to `avio_stor`:

  ```
  # hpvmstatus  -P guest1 -d
  ...
  [Virtual Machine Devices]

  [Storage Interface Details]
  disk:scsi::0,5,0:disk:/dev/rdisk/disk0
  disk:scsi::0,5,1:disk:/dev/rdisk/disk1
  disk:scsi::0,5,2:disk:/dev/rdisk/disk2
  disk:scsi::0,5,3:disk:/dev/rdisk/disk3
  disk:scsi::0,5,4:disk:/dev/rdisk/disk4
  ...

  # hpvmmodify -P guest1 -m hba:avio_stor:0,5
  # hpvmstatus -P guest1 -d
  ...
  [Virtual Machine Devices]

  [Storage Interface Details]
  disk:avio_stor:0,5,0:disk:/dev/rdisk/disk0
  disk:avio_stor:0,5,1:disk:/dev/rdisk/disk1
  ```

```
disk:avio_stor:0,5,2:disk:/dev/rdisk/disk2
disk:avio_stor:0,5,3:disk:/dev/rdisk/disk3
disk:avio_stor:0,5,4:disk:/dev/rdisk/disk4
...
```

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.

HP recommends the use of agile device names for guest configurations to avoid problems when VM Hosts are reconfigured.

If the VM Host administrator sets up a Virtual FileDVD for the virtual machine, 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 VM Host 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 VM Host 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 VM Host administrator and the guest administrator.

NOTE: HP-UX guest OS crash might fail to save a crash dump with the following console message:

```
Error: can't open first dump device /dev/dsk/c0t0d0. Dump aborted.
INIT[0]: OS_INIT ends. Resetting the system
```

This problem is caused by the HP-UX MPT dump driver in the guest waiting 1 second for a bus reset to finish. If there is an attached tape or changer sharing the same virtual MPT adapter as the guest boot disk, the reset time of that attached tape or changer might exceed the 1 second timeout, resulting in the error message above.

To avoid this problem on HP-UX 11i v2 0505–0706 or HP-UX 11i v3 guests, make sure the guest boot disk and any attached tape or changer do not share the same virtual MPT adapter. For information about how to specify bus and device numbers to place the tapes or changers on a separate MPT adapter than the guest boot disk, see the *hpvmresources* manpage.

For HP-UX 11i v2 0712, HP-UX 11i v3 0803, and all later supported HP-UX guest releases, make sure the HPVM-Guest depot is installed. The HPVM-Guest depot adjusts the MPT dump reset timeout allowing for tape and changers to be placed on the same virtual MPT adapter as the boot disk.

---

7 Creating Virtual Networks

You can allocate virtual network devices or virtual network interface cards (vNICs) to the guest when you create the guest with the `hpvmcreate` command or when you modify an existing guest using the `hpvmmodify` command, as described in Chapter 3. 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*

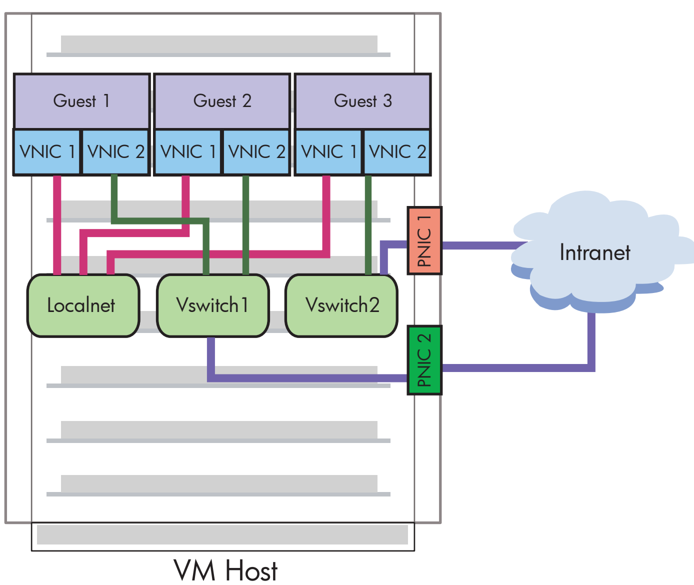
However, before you can allocate the vswitch to the guest, you must create the virtual switch (vswitch) using the `hpvmnet` command. This chapter describes how to create and manage vswitches, including the following topics and tasks:

- "Introduction to Virtual Network Configuration"
- "Creating and Managing vswitches"
- "Managing vNICs"
- "Configuring VLANs"
- "Troubleshooting Network Problems"

7.1 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 guests running on the VM Host. The basic virtual network configuration is illustrated in Figure 8.

Figure 8 Virtual Network Configuration



The virtual network configuration consists of the following components:

- VM Host 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 VM Host, not on the guest.

APA can be configured on the VM Host 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 VM Host 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 VM Host 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 VM Host 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.

7.2 Creating and Managing vswitches

The following sections describe how to create, modify, delete, and manage vswitches.

7.2.1 Creating vswitches

To allow guests to access network devices, you must create vswitches on the VM Host. 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 VM Host 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 7.2.1.1 (page 129).

The `hpvmnet` command also allows you to display and manage the vswitches on the VM Host. Table 19 describes the options to the `hpvmnet` command.

Table 19 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.

Table 19 Options to the `hpvmnet` Command *(continued)*

Option	Description
`-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 VM Host 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 is limited to 8 characters and must be unique on the VM Host.
`-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 7.4: “Configuring VLANs” (page 134) 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.
`-v`	Displays the version number of the `hpvmnet` command in addition to the vswitch information.

Table 19 Options to the hpvmnet Command *(continued)*

Option	Description
-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 is limited to eight characters. You must include either the -S or -s option.

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 VM Host 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 3. For information about modifying virtual networks, see Section 7.3.1 (page 133).

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.

7.2.1.1 Local Networks

Virtual network communication may be limited to virtual machines on the VM Host 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 VM Host 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 allowsOk communication between any other guest using the localnet vswitch. For example:

```
# hpvmmodify -P compass1 -a network:lan::vswitch:localnet
```

7.2.2 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
```

7.2.3 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.

7.2.4 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. The third command displays the current vswitches (without `clan1`).

```
# 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 VM Host automatically determines that the vswitch is gone. When the vswitch is recreated, the guest network automatically becomes functional again.

7.2.5 Recreating vswitches

To change the vswitch to use another pNIC on the VM Host (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
   ```

7.2.6 Starting vswitches

Virtual switches (vswitches) start automatically when the VM Host 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 VM Host (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 the 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).

7.2.7 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 VM Host 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.

7.2.8 Restarting Vswitches

It is necessary to restart the vswitch when:

- You replace the physical network card associated with the vswitch.
- You change a VM Host IP address associated with the vswitch's network interface card.
- You change the network interface characteristics on the VM Host; for example, by using the `nwmgr` command to change checksum offloading (CKO).
- You notice that there is no communication from an `avio_lan` interface to a `lan` interface after booting the guest(s) while the vswitch is down.

When you restart a vswitch, it is not necessary to restart the guests using the vswitch.

7.2.9 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 VM Host 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.
- After `ioscan` claims the AVIO devices in the guest, you might notice that the AVIO devices cannot communicate with another VIO guest interface configured on the same vswitch. When this occurs, invoke the `hpvmnet -r` option on the vswitch to restore connectivity.
- 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`.

7.3 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 VM Host'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)

7.3.1 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 VM Host 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.

7.3.2 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.

7.4 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

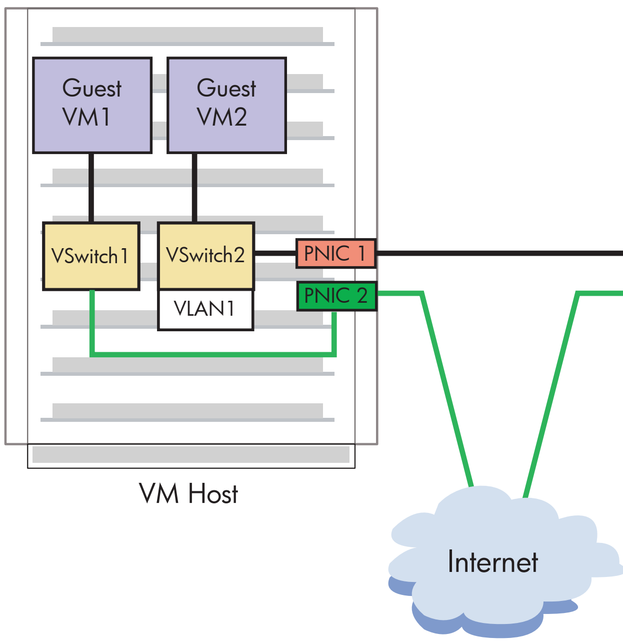
The following sections describe the Port-based VLANs, Guest-based VLANs, and VLAN-backed vswitch features.

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.

7.4.1 Port-Based VLANs

Figure 9 illustrates a basic virtual machine VLAN that allows guests on different VM Host systems to communicate.

Figure 9 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 VM Host.

If the guest has to communicate with the VM Host or outside the VM Host over a VLAN, additional configuration is necessary. For communication to the VM host, configure a VLAN interface on the VM host interface for that vswitch. This VLAN interface should have the same VLAN ID as the guest port. For information about configuring VLANs on the VM Host, 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 VM Host (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 20 describes the possible VLAN port states:

  Table 20 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

7.4.1.1 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.

7.4.1.2 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
```

7.4.2 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.

7.4.3 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 VM Host.

On the VM Host, 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.

NOTE: Guest VLANs and VLAN-backed vswitches are not supported with VIO.

OpenVMS guests that are created (`hpvmcreate`) or modified (hpvmmodify) to add and use a non-physical vswitch must use the same vswitch by name for each OpenVMS guest. The following scenarios fail:

- Attempting to start an OpenVMS guest that is using two different localnet vswitches on the same guest.
- Using a different localnet vswitch than an already active OpenVMS guest is using.

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.

7.4.3.1 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 VM Host. 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=
```

NOTE: AVIO LAN drivers on OpenVMS guests are VLAN tag-unaware and support the vswitch-based VLAN configuration. Use the following command to configure VLAN for an OpenVMS guest:

```
# hpvmnet -S vswitch-name -u portid:portnum:vlanid:vlandid
```

7.4.4 Configuring VLANs on Physical Switches

When communicating with a remote VM Host 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
```

7.5 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.

- TCP Segmentation Offload (TSO) is turned off by default in HP-UX. HP recommends that you leave it turned off on both the VM Host system and on HP-UX guests if you are using the VIO interface. This applies to both the virtual network interface cards in the guest and any physical network interface cards in the VM Host that are used by vswitches. When TSO is enabled, guest networks are interrupted.

  Note that this restriction applies to VIO interfaces.

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 comparable `nwmgr` command to use to replace the `lanadmin` command:

Task	Legacy Command	nwmgr Command
Verify whether TSO is turned on. *n* is the VM Host interface, as displayed by the `hpvmnet` command.	`#` **`lanadmin -x vmtu n`**	`#` **`nwmgr -g -A vmtu -c lann`**
Verify that TSO is on for lan0.	`#` **`lanadmin -x vmtu 0`** `Driver/Hardware supports TCP Segmentation Offload, Current VMTU = 32160`	`#` **`nwmgr -g -A vmtu -c lan0`** `lan0 current values: Virtual MTU = 32160 TCP Segmentation Offload is now disabled.`

Turn TSO off on lan0.	`#` **`lanadmin -X vmtu 0 0`**
	`Virtual MTU is set to 0`
		`#` **`nwmgr -s -A vmtu=0 -c lan0`**
		`lano current values:`
		`New Virtual MTU=0`

Allow the TSO attribute change along with the rest of the lan attributes to be perserved between system reboots by saving the current attributes to the lan configuration file.	Manually edit the configuration file in /etc/rc.config.d, or use SAM interface. See the *Ethernet Support Guide* for details.	`#` **`nwmgr -s -c lan0 -A all --sa -fr cu`**

- 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
  ```

7.5.1 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.

7.5.2 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 VM Host, 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 VM Host 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 VM Host to disable the CKO feature, where 4 is the VM Host 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 VM Host, 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:

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 VM Host.

The following table shows the `nwmgr` command that performs the task as the `lanadmin` command:

Task	Legacy Command	`nwmgr` Command

Turn off CKO on PPA 4 by entering this command on the VM Host.	`# 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`

7.5.3 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 VM Host. 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=
```

7.6 Other Issues and Notes

The following list provides networking information with the Integrity VM V4.3 release of which you should be aware :

- If you modify the MAC address of an interface in the guest, the `hpvmstatus` command in the VM Host 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 VM Host 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`

8 Managing Guests

To manage a guest, connect to the guest using a remote connection and use the operating system administration procedures appropriate to the guest OS. Integrity VM provides utilities for managing virtual machines from the VM Host and from inside the guest. This chapter describes how to manage guests using Integrity VM commands and utilities, including the use of ID-VSE Logical Server Management (LSM). The following topics are included in this chapter:

8.1 Managing Guests with LSM

Integrity VM fully supports ID-VSE Logical Server Management (LSM). With this version of Integrity VM, administrators can now 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 Integrity VM Host before you can use LSM to create Integrity VM guests and before you can use HP Insight Orchestration to provision VMs.

To create Integrity VM guests using ID-VSE LSM 6.2, you must adhere to the following:

1. Create the appropriate size SLVM volume group (VG) for the device management database using LVM Version 2.1 or later. For example:
   - Create the volume group using LVM Version 2.1:

     ```
     # vgcreate -V 2.1 -s 4m -S 100g /dev/slvm_v21 /dev/disk/disk61
     ```

   - Create the volume group using LVM Version 2.2:

     ```
     # vgcreate -V 2.2 -s 4m -S 70g /dev/slvm_v22 /dev/disk/disk70
     ```

   For information about creating SLVM volume groups, see the *HP Integrity Virtual Machines 4.2: Installation, Configuration, Administration* manual at HP Integrity Virtual Machines and Online VM Migration and 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 8.1.1 (page 146)

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 8.1.2 (page 146).

4. Execute the Integrity VM `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 Integrity VM 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 Integrity VM Host 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 HP Serviceguard with Integrity VM* section.

   If you already have your VMs clustered in a VM as a Serviceguard Package, but prefer not to manage them this way, run the following Serviceguard command to properly deconfigure (delete) the package:

   ```
   # hpvmsg_package -U -P package_guest_name
   ```

8.1.1 Storage Requirements for Managing Existing Integrity VM Guests with Logical Server Management

To use Logical Server Management (LSM) to manage virtual machines created outside of LSM, the guest 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 Integrity VM device management database.

  The supported operations are: Import, Online Move, Power On, Power Off, Activate, and Deactivate, and Unmanage.

NOTE: For information about virtual machines created with LSM or HP Insight Orchestration using SLVM-based LVs, see Section 8.1 (page 145).

8.1.2 Storage for Deactivated Volume Groups not Protected by Integrity 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 VM Host 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 VM Host 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 VM Host system is rebooted. Take care to check that all SLVM volume groups are activated after a VM Host reboot or Serviceguard cluster reconfiguration.

8.1.3 Managing Existing Integrity VM Guests With LSM

You can import existing Integrity VM guests 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 guests.

Integrity VM Hosts that are managing only VMs with whole LUNs do not need to be configured with Serviceguard and SLVM. If you plan to create new guests on that VM Host, follow the steps in Section 8.1 (page 145).

8.1.4 Managing Guests using gWLM

Guests configured with processing power specified in cycles instead of percentage are incompatible with gWLM A.02.50 and earlier versions.

If gWLM/Insight Dynamics — VSE produces an error message similar to the following, a guest 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.

8.2 Integrity VM Virtualization Provider

Integrity VM now supports an additional provider, the Integrity VM Virtualization Provider. The Integrity VM Virtualization Provider, used with the logical server feature in Insight Dynamics, 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.

8.2.1 Adding and Removing Devices

Integrity VM adds devices not in use by the VM Host 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/rdsk/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.

8.2.2 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 VM Host 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` command.

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 VM Host at a time. Registering a VM on more than one VM Host can lead to accidentally booting the VM on more than one VM Host and could cause inconsistencies with the display of graphical tools. However, if you find that VM is not registered on any VM Host, you can manually register it with the `hpvmmodify` command. For information on this command, see Section 9.1.3.

8.2.3 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 VM Host at a time. Marking a VM runnable on more than one VM Host can lead to accidentally booting the VM on more than one VM Host.

- 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 VM Host. Any modification made to this VM's configuration will be lost when it is migrated back to this VM Host.

- 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 VM Host, 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 VM Host at a time. Registering a VM on more than one VM Host can lead to accidentally booting the VM on more than one VM Host, 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.

8.2.4 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
```

8.2.5 Unpresenting SAN Devices to Integrity VM Hosts

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 VM Host 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
```

8.2.6 Changes to the `hpvmstatus` Command

The `Runsysid` column of the `hpvmstatus` command output has been renamed to "`Rmt Host`" to help with usability. Serviceguard-packaged VMs that are "`On`" that are running on another member of the cluster will have the state "`On (RMT)`" instead of simply "`On`". If a VM is not packaged in a Serviceguard cluster, the Rmt Host column displays a dash (-) instead of a zero (0).

The following example shows the output of the `hpvmstatus` command:

```
# hpvmstatus
[Virtual Machines]
Virtual Machine Name VM #  OS Type State     #VCPUs #Devs #Nets Memory  Rmt Host
==================== ===== ======= ========= ====== ===== ===== ======= ========
vm1                      1 HPUX    Off            2     2     2    6 GB        -
vm2                      2 HPUX    On (RMT)       4     1     1    2 GB        3
vm3                      3 HPUX    On (RMT)       2     1     1    1 GB        2
vm4                      4 HPUX    On (OS)        4     1     1    2 GB        1
vm5                      5 HPUX    On (RMT)       4     9     1    2 GB        3
```

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.

8.3 Monitoring Guests

To display information about all the virtual machines configured on the VM Host, 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 OPENVMS Off           1     5     1    1 GB        0
```

The virtual machine status is displayed in the `State` column and indicates whether the virtual machine is powered off or on. When the virtual machine is on, the status also includes one of the following:

- `EFI` indicates the virtual machine is running normally in EFI.
- `OS` indicates the virtual machine is running normally in the operating system.
- `ATTN!` indicates the guest is not responding to interrupts.

Table 21 describes the options to the `hpvmstatus` command.

Table 21 Options to the `hpvmstatus` Command

Option	Description
`-v`	Displays the version of the Integrity VM product that is running on the VM Host.
`-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 VM Host 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 VM Host. `CAPPED` indicates that gWLM is managing the node. NORMAL indicates that the node is not being managed by gWLM.
`-s`	Displays the current VM Host resources.
`-m`	If Serviceguard is installed, displays information about the multiple-server environment.
`-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    : 43d82eb8-ff27-11d9-a431-00306e39f70b
Virtual Machine ID      : 2
Virtual Machine Label   :
VM's Model Name         : server Integrity Virtual Machine
VM's Serial Number      : VM00530001
VM's Config Version     : 4.3.0
VM's Config Label       : HPVM B.04.30.00 BL02 clearcase opt Wed Jun 04 2008 05h
41m04s PST
Operating System        : HPUX
OS Version Number       :
State                   : On (OS)
Start type              : Manual
Console type            : vt100-plus
Guest's hostname        :
Guest's IP address      : .1.11.135
EFI location            : /opt/hpvm/guest-images/common/efi
Pattern File location   : /opt/hpvm/guest-images/common/patterns.vmmpat
Guest revision          : 1
Running on serverid     : 0
Running on pid          : 24447
Application controllers : NONE
Distributed             : 0

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

[Virtual CPU Details]
Number Virtual CPUs     : 1
Minimum Virtual CPUs    : 1
Maximum Virtual CPUs    : 4
Percent Entitlement     : 10.0%
Maximum Entitlement     : 100.0%

[Memory Details]
Total memory            : 2 GB
Minimum memory limit    : 512 MB
Maximum memory limit    : 64 GB
Reserved memory         : 64 MB
Minimum reserved limit  : 32 MB
Maximum reserved limit  : 64 GB
VHPT Size               : 1 MB

[Dynamic Memory Information]
Type                    : driver
Minimum memory          : 1024 MB
Target memory           : 2039 MB
Maximum memory          : 2048 MB
Current memory          : 2039 MB
Comfortable minimum     : 759 MB
Boot memory             : 2039 MB
Free memory             : 1148 MB
Available memory        : 215 MB
Memory pressure         : 0
Memory chunksize        : 65536 KB
Driver Mode(s)          : STARTED ENABLED GUESTCTL

[Storage Interface Details]
Guest Device type       : disk
Guest Adaptor type      : scsi
Ioscan format           : 0/0/1/0.0.0
Bus                     : 0
Device                  : 1
Function                : 0
```

```
Target                  : 0
Lun                     : 0
Physical Storage type   : disk
Physical Device         : /dev/rdisk/disk0


[Network Interface Details]
Physical Storage type   : vswitch
Guest Adaptor type      : lan
Backing                 : vswitch1
Vswitch Port            : 1
Ioscan format           : 0/0/0/0
Bus                     : 0
Device                  : 0
Function                : 0
Mac Address             : f6-92-cf-35-86-78


[Misc Interface Details]
Guest Device type       : serial
Guest Adaptor type      : com1
Physical Storage type   : tty
Physical Device         : console
```

To display the VM Host system resource, use the `-s` option to the `hpvmstatus` command. For example:

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

         Processor speed = 1400 Mhz
         Total physical memory = 12276 Mbytes
         Total number of processors = 2
         Available memory = 7367 Mbytes
         Available swap space = 4707 Mbytes
         Maximum vcpus for an HP-UX virtual machine = 2
         Maximum vcpus for an OpenVMS virtual machine = 2
         Available entitlement for a 1 way virtual machine = 1400 Mhz
         Available entitlement for a 2 way virtual machine = 1260 Mhz
```

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.

8.4 Monitoring Integrity VM Performance

Guest and VM Host performance information is displayed by the VM Host 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 22 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
`-F`	Integrity VM core Memory Metrics Display
`-G`	Guest Dynamic Memory, Swap, Paging Display
`-H`	Host Memory, Swap, Paging Display

Table 22 Options to the `hpvmsar` Command *(continued)*

Option	Display Description
`-I`	Guest Interrupt Display
`-N`	Guest AVIO Network traffic by vswitch Display
`-S`	Vswitch AVIO Network traffic by Port Display

8.5 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 VM Host 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 VM Host 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 `testme`:

```
# 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 VM Host, 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 VM Host. 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 VM Host 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 VM Host `myhost`:

```
# telnet host1

Trying .xx.yy.zz...
Connected to host1.rose.com.
Escape character is '^]'.

HP-UX host B.11.23 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.

8.5.1 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.

8.5.2 Guest User Accounts

The configuration for captive `hpvmconsole` guest user accounts has changed in Integrity VM Version 4.0 and later to support additional access controls and configurations. This change requires that the guest user accounts have the correct home directory. It is also necessary to list the console access account in the guest configuration file.

For example, using a guest named `compass1` (and therefore a user account named `compass1`), the home directory for user `compass1` must be `/var/opt/hpvm/guests/compass1`. To ensure that the user continues to have administrative console access, use the following command:

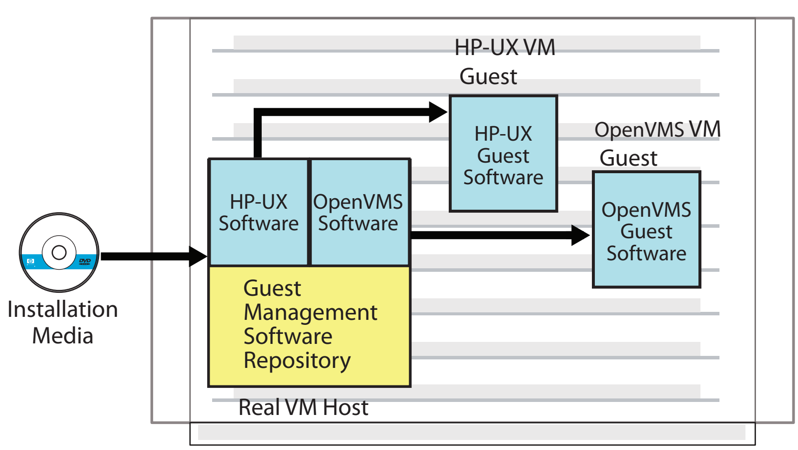
```
# hpvmmodify -P compass1 -u compass1:admin
```

8.6 Installing the Guest Management Software

After you install the guest OS, you must install special Integrity VM guest management software. Installing the guest management software on each guest provides patches for performance improvements and allows you to use Integrity VM command on the guest. Without the guest management software, you cannot use guest dynamic memory. And, with the latest guest management software, the guest can be managed by VM Manager and the ID-VSE products.

Figure 10 illustrates the process. Guest management software is copied onto the VM Host system when you install Integrity VM. When you upgrade to a new version of Integrity VM, the guest management software is also upgraded in the VM Host guest management software repository. You manually install the appropriate guest management software on each guest, depending on the guest OS.

Figure 10 Installing Guest Management Software



The guest management software is required on each guest. The guest management software ensures that guests have the required patches for optimum performance and manageability. It also enables the `hpvmmgmt`, `hpvmcollect`, and `hpvminfo` commands on the guests, as appropriate. (Not all commands run on all types of guests.) It also provides support for the VM Host's `hpvmsar` command.

The guest management software is stored on the VM Host system in the `/opt/hpvm/guest-images` directory. A subdirectory contains the guest management software for each type of guest operating system, as shown in the following example:

```
# cd /opt/hpvm/guest-images
# ls
common    hpux      openvms
```

To install guest management software on an HP-UX guest, select the appropriate version of HP-UX:

- 11iv2 indicates HP-UX 11.23.

  For HP-UX 11i v2 guests, the guest depot file:

  `/opt/hpvm/guest-images/hp-ux/11iv2/hpvm_guest_depot.11iv2.sd`

- 11iv3 indicates HP-UX 11.31.

  For HP-UX 11i v3 guests, the guest depot file is:

  `/opt/hpvm/guest-images/hpux/11iv3/hpvm_guest_depot.11iv3.sd`

Before installing the guest kit (bundle HPVM-Guest), 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 HPVM-Guest vmProvider
```

You might see the following warning on an Integrity VM guest installed on an HP-UX 11i v2 prior to December 2007:

```
WARNING: Recommended bundle GUESTAVIOLAN not installed on guest.
WARNING: Recommended bundle GUESTAVIOSTOR not installed on guest.
WARNING: The "checkinstall" script for "vmTools.VMTOOLS-RUN" had a
         warning (exit code "2"0. The script location was
         "/var/tmp/BAAa05960/catalog/vmTools/VMTOOLS-RUN/chekinstall".
       * This script had warnings but the execution of this fileset
         will still proceed.  Check the above output from the script
         for further details.
```

Each guest management software directory contains a `README` file that describes how to install the guest management software for that type of guest. For information about any additional software updates that you should also install on your guests, see the *HP Integrity Virtual Machines 4.3: Release Notes*.

8.7 Using the Virtual Console

Each virtual machine has its own virtual console from which to power on or off the virtual machine, boot the guest operating system or shut it down, and so on. The `hpvmconsole` command connects to the virtual console of a specified virtual machine.

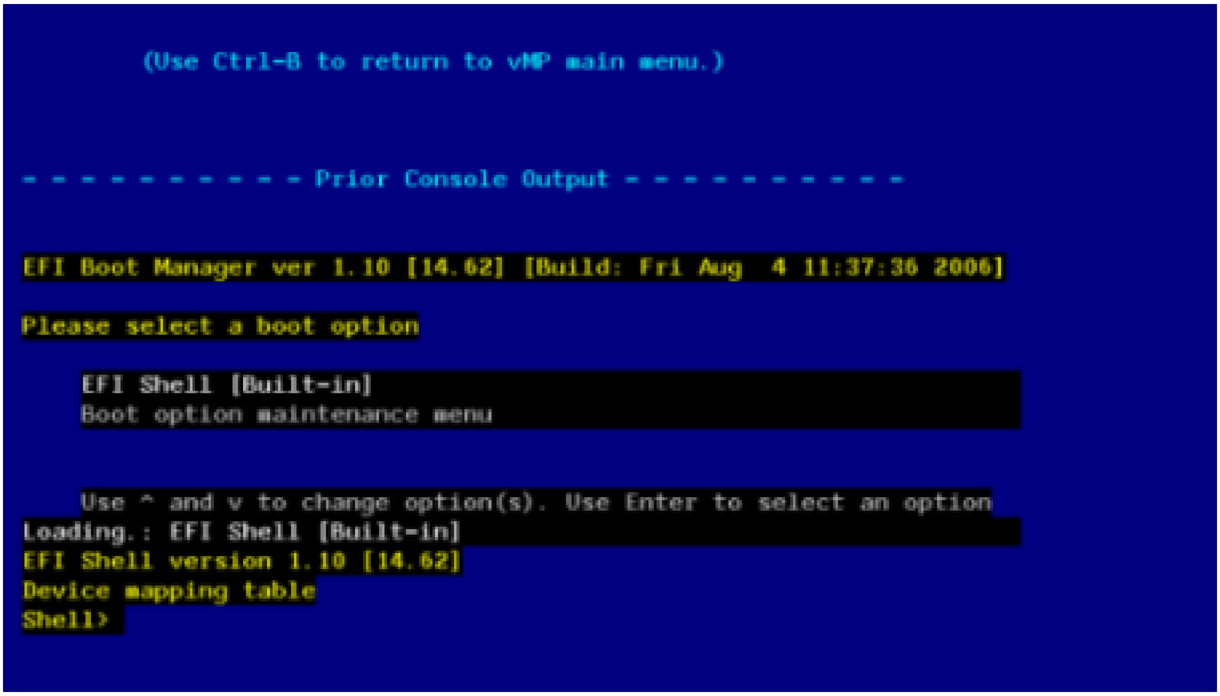
To start the virtual console for the guest named `host1`, enter the following command:

```
# 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 virtual machine 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 23 lists the options to the `hpvmconsole` command.

Table 23 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 VM Host'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.04.30  clearcase opt Wed Oct 7 2008 07h13m54s PST
(C) Copyright 2000 - 2008 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).

8.8 Using the virtual iLO Remote Console

The 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 VM Host 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

8.8.1 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 .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 .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 .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
.92.81.68          .92.81.68        UH    0    lan1:274485572 32808
.92.80.101         .92.80.101       UH    0    lan1       32808
```

```
127.0.0.0                127.0.0.1          U      0     lo0        32808
default                  .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:      .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 .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 8.5 (page 154)

When a guest is migrated to another VM Host using Online VM Migration (OVMM), the Integrity VM virtual iLO virtual iLO Remote Console is also migrated to the new VM Host. Before migration, the virtual iLO Remote Console process is running on only the source VM Host. After migration, the virtual iLO Remote Console process is stopped on the source VM Host. 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 VM Host. New client connections to the virtual iLO Remote Console IP address are now sent to the virtual iLO Remote Console process on the new VM Host.

8.8.2 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 VM Host:

    - 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 does 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 VM Host system are not automatically migrated to the target VM Host system during Online VM Migration (OVMM). You must manually add any guest administrator accounts to the target VM Host system, using the same `useradd` commands performed on the source system. For information about creating guest administrator and operator accounts, see (page 154).
- The virtual iLO Remote Console does not support rlogin connections.

8.9 Guest Configuration Files

When the guest is created, the VM Host 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.

8.10 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. In this release of Integrity VM, dynamic memory is available on HP-UX guests only.

An example of this feature allows a guest that is a Serviceguard node to be used as a standby server for multiple Serviceguard packages. When a package fails over to the guest, the guest memory can be changed to suit the requirements of the package before, during, and after the failover process.

To use dynamic memory, the guest must have the guest management software installed, as described in Section 4.4 (page 78).

8.10.1 Managing Dynamic Memory from the VM Host

On the VM Host, the dynamic memory software is included with Integrity VM. Manage dynamic memory on the VM Host 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 24 provides a complete list of -x keywords used for dynamic memory.

Table 24 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.

8.10.1.1 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 VM Host 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.

8.10.1.2 Viewing Dynamic Memory on the VM Host

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 25 describes the dynamic memory characteristics displayed by the `hpvmstatus` and `hpvmmgmt` commands.

Table 25 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 8.10.3.4 (page 167).
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 VM Host 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               -            -            -            -            -             -            -
```

8.10.1.3 Modifying a Virtual Machine's Memory Size on the VM Host

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
```

8.10.2 Managing Dynamic Memory from the Guest

Dynamic memory management from the guest is disabled by default and must be enabled from the VM Host. 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 VM Host system, enter the following command to enable dynamic memory control on the guest named `host1`:

```
# hpvmmodify -P host1 -x dynamic_memory_control=1
```

8.10.2.1 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 guest management software, as described in Section 4.4 (page 78).

Table 26 describes the options to the `hpvmmgmt` command.

Table 26 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.
`-M`	Displays verbose attribute and resource information in a machine-readable format.

Table 26 Options to the `hpvmmgmt` Command *(continued)*

`-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
```

8.10.2.2 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.
```

8.10.3 Troubleshooting Dynamic Memory Problems

This section describes how to solve problems in the use of dynamic memory.

8.10.3.1 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 VM Host. If the VM Host 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.

8.10.3.2 VM Host 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 VM Host system. If the VM Host 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 VM Host 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 VM Host 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 8.10.3.3 (page 167) and Section 8.10.3.5 (page 168).

8.10.3.3 Guest 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 VM Host 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.

8.10.3.4 Specify Sufficient Guest Memory

If you set the value of `ram_dyn_target_start` too small, the guest operating system might hang or crash while booting. In this case, the guest 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 guest crashes while booting on the VM Host , use the `hpvmmodify` command to increase the value of the `ram_dyn_target_start` parameter. For example, to increase the memory size for the guest named `host1`, enter the following command on the VM Host:

```
# hpvmmodify -P host1 -x ram_dyn_target_start=2GB
```

After you set this parameter, reboot the guest.

If the guest hangs, on the VM Host, use the `hpvmstatus` command to check the memory statistics on the guest. 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 VM Host to increase the memory size of the guest. The guest then boots normally.

8.10.3.5 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.

8.10.3.6 Enable Dynamic Memory on the Guest and on the VM Host

The guest management software must be installed on the guest before you can use dynamic memory parameters on the VM Host system. For example, if the guest management 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
.
.
.
```

If you attempt to modify the guest's dynamic memory from the VM Host, 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 guest, 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 guest management software, see Section 4.4 (page 78).

8.10.3.7 Upgrade the Guest Kit When Upgrading Integrity VM

The dynamic memory software has two components: the VM Host 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 guest 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 VM Host'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 guest management 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
```

For information about installing the HP-UX guest management software, Section 4.4 (page 78).

8.10.4 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 guest must have the guest management software installed, because this is required for dynamic memory. For guest management software installation instructions, see Section 4.4 (page 78).

8.10.4.1 Enabling Automatic Memory Reallocation on the VM Host

On the VM Host, 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.

8.10.4.2 Managing Automatic Memory Reallocation from the VM Host

The following tunables can be set in `/etc/rc.config.d/hpvmconf` and are listed here with the values they take if they are not set:

HPVMAMRCHUNKSIZE=256
HPVMAMRWAITTIME=60

The value of HPVMAMRCHUNKSIZE is the default number of MB by which the guest attempts to grow when it encounters memory pressure. The higher this value is, the more quickly a VM attempts to grow when it encounters memory pressure. HP recommends that chunk-size values be a multiple of the dynamic memory chunk size. For a description of dynamic memory chunk size, see Section 8.10.1 (page 161).

The value of HPVMAMRWAITTIME is the maximum number of seconds that the VM startup process waits for memory to become available before reporting failure due to insufficient memory. This value can be set up to a maximum of 600.

If the values of HPVMAMRENABLE, HPVMAMRCHUNKSIZE, or HPVMAMRWAITTIME in the `hpvmconf` file have been changed, sending SIGNUP to the `hpvmamrd` process causes it to obtain the new values.

8.10.4.2.1 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 VM Host that would cause the VM to shrink below its entitlement.

8.10.4.2.2 Managing Automatic Memory Reallocation on a VM

By default, a VM with automatic memory reallocation enabled has no additional constraints placed on its physical memory. It does not go below its `ram_dyn_min` dynamic memory setting, nor does it go below the "Comfortable minimum" described in Section 8.10.1 (page 161). An additional minimum constraint can be set with the following `-x` option, which is supported on running VMs.:

`-x ram_dyn_entitlement=`*`minimum memory size in MB`*

The following rules apply:

- If it is impossible to grow the running VM to the specified entitlement, this additional minimum constraint specification fails.
- A VM cannot have its physical memory grow beyond its maximum memory.
- A VM cannot have its physical memory modified beyond a lower boundary of `ram_dyn_min` or an upper boundary of `ram_dyn_max`.

By default, a VM uses the system-wide HPVMAMRCHUNKSIZE value as the number of MB by which it attempts to grow when it encounters memory pressure. If a value other than the system-wide default is preferred, a per-VM chunk size can be set with the following `-x` option:

`-x amr_chunk_size=`*`chunk size in MB`*

HP recommends that chunk-size values be a multiple of the dynamic memory chunk size, as described in Section 8.10.1.2 (page 163). This option is supported on running VMs.

8.10.4.3 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
```

8.11 Integrity VM Log Files

Each guest has a log file named `/var/opt/hpvm/guests/`*`guestname`*`/log` on each host.

The VM Host 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.

8.12 Managing the Device Database

Integrity VM cannot detect all potential backing store conflicts, and does not always prevent misconfigured guests from booting. Conflicts can arise from the following:

- Specifying the same backing store for more than one virtual device.

  If you add `disk:scsi::disk:/dev/rdisk/disk2` for guest A, do not add the same device to another guest or to the list of VM Host restricted devices.

- Specifying multiple backing store parameters that lead to the same physical storage.

  If the VM Host has multiple paths to a storage device, like `/dev/rdisk/disk0 and /dev/rdisk/disk4`, only one path should be specified for a `disk:scsi` or `dvd:scsi` in guest A. The other path should not be used as a backing store by guest A or by any other guest or the VM Host.

- 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 VM Host, 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.

8.12.1 The Device Database File

Integrity VM device management stored Integrity 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 a supported Integrity VM commands (such as `hpvmmodify` or `hpvmdevmgmt`) to modify virtual devices.

8.12.2 Using the `hpvmdevmgmt` Command

To list and modify the devices used by the VM Host and the virtual machines, use the `hpvmdevmgmt` command.

Table 27 describes the options to the `hpvmdevmgmt` command.

Table 27 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.

Table 27 Options to the `hpvmdevmgmt` Command *(continued)*

Option	Description
`-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 VM Host, run the `hpvmdevmgmt -I` command after adding the device to the host.

8.12.2.1 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:scsi::null:/var/opt/hpvm/ISO-images/hpux/
# hpvmmodify -P host2 -a dvd:scsi::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 use by one guest only.

8.12.2.2 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.

8.12.2.3 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
```

8.12.2.4 Restricting VM Host Devices

You must set up restricted devices to ensure that no guest uses devices that are reserved for use by the VM Host, including the storage devices that the VM Host 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 6 (page 85).

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.

8.12.3 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.

8.13 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.
- Enumerate all LUNs.

The following example shows how to change the enumeration policy to enumerate boot LUNs only:

```
Shell> drvcfg -s
HP AVIO Stor Driver Configuration
==================================
Enumerate only boot LUNs (Y/N)? [N]:Y

  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
```

9 Migrating Virtual Machines

There are several different forms of Virtual Machine migration. With the `hpvmmigrate` command, you can move either an offline virtual machine or a live, online virtual machine running a guest operating system and applications from a source VM Host system to a target VM Host system. For offline migration, the `hpvmmigrate` command has been available with HP Integrity Virtual Machines Version 1.2 and later. The Online VM Migration feature for online guests is available starting with HP Integrity Virtual Machine Version 4.2.

This chapter includes the following topics:

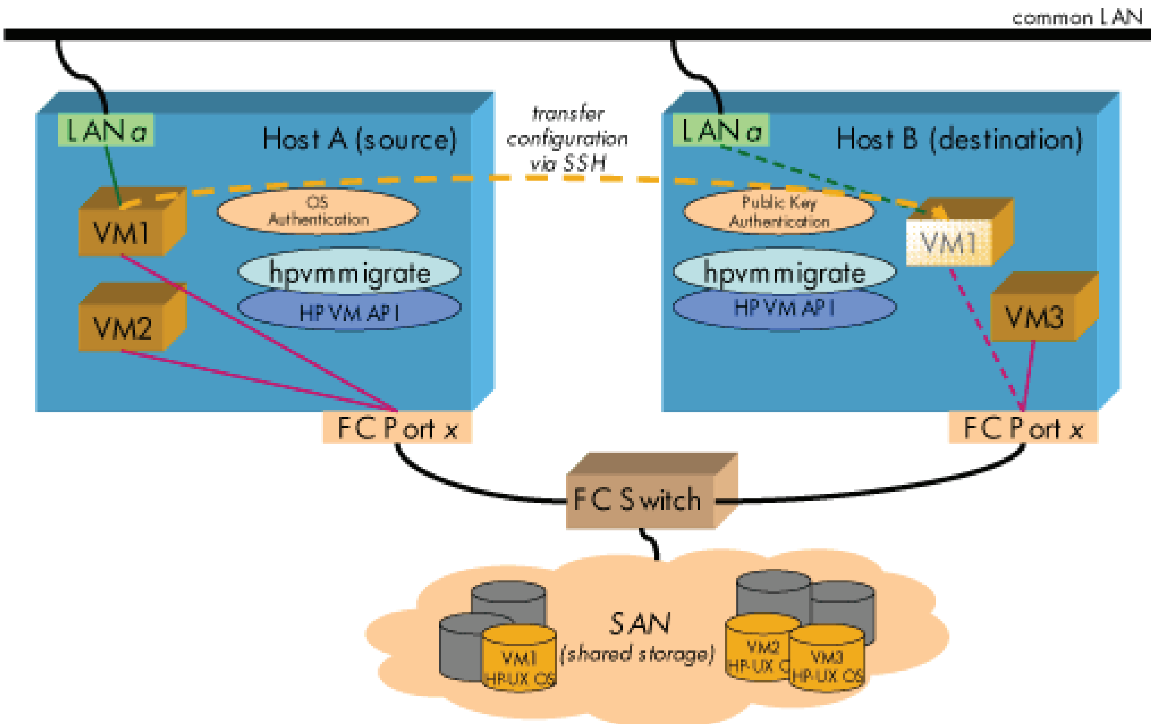
9.1 Introduction to Virtual Machine Migration

Virtual machines can be moved from one VM Host system to another in a variety of ways. Integrity VM provides the following types of virtual machine migration:

- To move the virtual machine from one VM Host system to another, use the `hpvmmigrate` command. The virtual machine can be either a non-running virtual machine configuration (offline migration) or a running guest (online migration). Online migration enables a running guest and its applications to be moved from one VM Host to another without service interruption. All guest I/O connections to storage and networks remain active throughout the online migration, and the guest and all its applications continue operating without a reboot or application restart.
- To migrate a Serviceguard Packaged guest online, use the `hpvmsg_move` command. For more information, see Section 9.4 (page 196) .
- To migrate from a physical machine to a virtual machine, use the `p2vassist` utility. For example, use this method to move virtual machines from a partition or a system to a VM Host system. Guests must be shut down before the procedure and restarted after the migration. For more information, see Section 9.5 (page 196).

Figure 11 illustrates the process of moving a guest from Host A to Host B offline.

Figure 11 Symmetric Hosts Configured for Guest Migration



The basic virtual machine migration environment includes a source machine and a target machine. Both must be running Integrity VM 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:

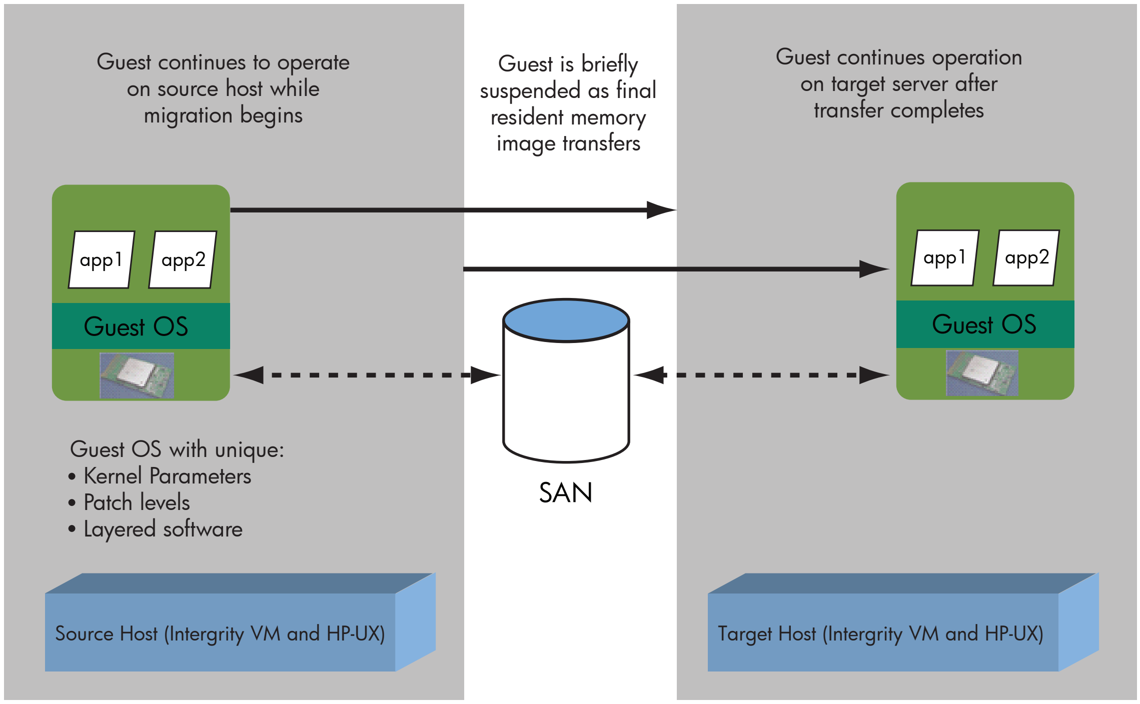
- A common local area network (LAN)
- Identical subnet and vswitch connectivity
- Common access for Storage Area Network (SAN) based storage
- Private, high-speed network connection (for Online VM Migration)

For guidelines about setting up storage for migrating virtual machines, see Section 9.3 (page 187).

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 10–2 illustrates moving a guest online from a source VM Host to a target VM Host.

Figure 12 Online Guest Migration from Source to Target



9.1.1 Reasons for Migrating an Online Guest

There are various reasons why you might want to migrate an online guest. Most can be summarized into three categories:

- Vacating a VM Host system
- Targeting a particular VM Host
- Balancing VM Host workloads
- Optimizing physical resource utilization

With Online VM Migration, you can migrate all your guests off of a VM Host to one or more other VM Hosts without interrupting the workload activity on the guests. A common reason to do this is for maintenance of the VM Host 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 VM Host reboot. A rolling upgrade of VM Host software is possible by moving the running guests to another VM Host, upgrading the VM Host, then migrating the guests back. Being able to move guests 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 guests in response to predictive failure alerts without interrupting your applications.

You might also want to migrate an active guest workload to a particular VM Host to take advantage of a particular resource or feature on that target VM Host without losing application availability. If your current VM Host resources become oversubscribed, you can migrate one or more of the guests to other VM Hosts that have remaining capacity. Perhaps a potential target VM Host has a large quantity of RAM, CPUs or I/O adapters, which might facilitate faster processing or greater I/O bandwidth while on that VM Host. Another possibility is that certain VM Hosts have special devices that are needed only temporarily by guest workloads. Because Online VM Migration enables guests to be migrated without interrupting their workloads, it is convenient and practical to migrate guests temporarily to certain VM Hosts 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 guests to balance the workload on VM Hosts workloads. For example, you might want to separate guests 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 VM Host 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 guests. You can conveniently "park" idle, near-idle, or just currently less-critical guest 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 guests and shrink CPU entitlements to more tightly packed guests on a smaller VM Host.

Table 28 provides the supported online migration paths for HP-UX and OpenVMS guests:

Table 28 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
Integrity VM V4.2.5	Integrity VM V4.2.5 or Integrity VM V4.3
Integrity VM V4.3	Integrity VM V.3

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 VM Host, use the `hpvmmigrate -s` option.

9.1.2 Reasons for Migrating Virtual Machines Offline

This sections lists reasons why you might want to migrate a virtual machine offline. For example:

- The guest might be stopped, so you need to move the configuration information offline.
- Migrating the virtual machine offline does not use the VM Host resources (like memory and CPUs) on the source and target VM Hosts.
- The guest might have local storage, logical volumes or file-backed storage, which must be copied to the target VM Host.
- The source and target VM Hosts might have different processor types that prevent online migration.
- The source VM Host might be running a version of Integrity VM prior to Version 4.3, which does not support Online VM Migration.
- You can offline migrate guests between different processor families.

Table 29 provides the migration path for offline migration:

Table 29 Offline Migration Paths

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

9.2 Command Line Interface for Online and Offline Migration

To migrate a virtual machine to another VM Host, perform the following steps:

1. Set up SSH keys on both the source and target hosts, as described in Section 9.3.3 (page 192).
2. Present all SAN storage assigned to the virtual machine to the target VM Host (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 8 (page 145).
4. On the source host, enter the `hpvmmigrate` command, as described in Section 9.2.1 (page 181). 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 VM Hosts, 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 VM Host, 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 VM Host, you can remove the virtual machine configuration with the `hpvmremove` command. Once the guest is removed from the VM Host, 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.

9.2.1 Using the `hpvmmigrate` Command

Use the `hpvmmigrate` command to move an online guest or an offline virtual machine from a source VM Host to a specified target VM Host. Virtual machines can be migrated while OFF, and online guests can be migrated while ON and running. Use the `-o` option 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 configuration information, and optionally local disk contents, to the target VM Host.

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 VM Host. 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 VM Host.

CAUTION: Use the `-F` option with caution, because some errors can prevent a virtual machine from working properly on the target VM Host.

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 VM Host after it is migrated successfully to the target VM Host. Run the `hpvmstatus` command to see that the state of the virtual machine is now `Off(NR)` on the source VM host and the guest is `On(OS)` on the target VM Host. The guest is running on the target VM Host and is, therefore, considered `Runnable`.

This mechanism allows the same virtual machine to be configured on multiple VM Hosts, while still preventing accidental booting of the same guest on multiple hosts simultaneously. Each virtual machine must be `Runnable` on only one VM Host 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 VM Host, 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 VM Host 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 that is being managed by ID-VSE, use Capacity Advisor to collect utilization data before you migrate the virtual machine. Otherwise, the utilization information for the VM Host prior to the migration is lost.

Table 30 lists the options to the `hpvmmigrate` command.

Table 30 Options to the `hpvmmigrate` Command

Option	Description
`-A`	Attempts to abort an online guest 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.
`-C`	For offline migrations, physically copies the storage device specified with the `-m` option to the target VM Host 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.

Table 30 Options to the `hpvmmigrate` Command *(continued)*

Option	Description
`-d`	For offline migrations, causes `hpvmmigrate` to automatically shut down a running guest before migrating the virtual machine configuration to the target VM Host. Consider migrating the guest online by using the `-o` option instead.
`-D`	Deletes the virtual machine from the source VM Host after migrating the virtual machine to the target VM Host system. If not specified, the virtual machine is marked `Not Runnable` on the source VM Host 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 VM Host. 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 VM Host machine to which the virtual machine is being migrated. The target machine must be a valid VM Host and must be accessible by the source VM Host. 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 VM Host. 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.
`-k`	Creates the virtual machine configuration on the target VM Host and marks it `Not Runnable`, but does not change the virtual machine on the source VM Host. This is used primarily to distribute virtual machine configurations for Serviceguard.

Table 30 Options to the `hpvmmigrate` Command *(continued)*

Option	Description
`-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 VM Host, 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 VM Host 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 VM Host. 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.
`-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.

Table 30 Options to the `hpvmmigrate` Command *(continued)*

Option	Description
`-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 VM Hosts. 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
```

9.2.2 Examples of the `hpvmmigrate` Command

The following example displays the version number of the `hpvmmigrate` command:

```
# hpvmmigrate -v
hpvmmigrate: Version B.04.30
```

Offline Migration Example

The following example shows how to migrate the guest named VM1, residing on the host named HostA, to the target host (HostB). On the system named HostA, enter the following command:

```
# hpvmmigrate -P VM1 -h HostB
```

This example specifies:

- The name of the guest (`-P VM1`)
- 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 VM Host. The guest name is vm3. The target VM Host is called host2, and the target VM Host'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 VM Host `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 VM Host.

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 VM Host simultaneously. If the `hpvmmigrate -D` option is not specified, the guest is marked `Not Runnable (NR)` on the source VM Host after online migration has finished. This prevents the virtual machine from booting on the original source VM Host while it is running on the target VM Host. If the `hpvmmigrate -D` option is used, unpresent the guest's SAN storage from the source VM Host as soon as migration completes, thus avoiding accidental usage of the storage on that VM Host.

9.2.3 Using the `hpvmstatus` Command to See Migration Details

Use the `hpvmstatus` command to see the current state of all virtual machines on this VM Host. 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 VM Host.
- `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 VM Host.

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.

9.2.4 Options to `hpvmmodify` Command for Online Migration

Use the `hpvmmodify -x` option to change the online migration phase timeout values. See Section 9.2.4 (page 186) 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 `nwmgrl`.

If the vswitch connectivity required by the guest on the target VM Host 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 Section 9.4 (page 196).

9.2.5 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. Use the `hpvminfo -V` option to display information about the guest and the current VM Host.

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
```

9.3 VM Host 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 VM Hosts depends on proper configuration of the networks and storage connected to the VM Host 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 VM Host servers, the VM Hosts 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 VM Hosts. There must be equivalent access to guest storage and equivalent network reachability on both the source and the target VM Hosts. The network on the target VM Host must be able to make all the same network connections that can be used by the guest on the source VM Host.

A vswitch of the same name, connected to the same network must be available on the source and target VM Host 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 VM Hosts should be connected with a private 1 GbE (or faster) network for efficient VM Host–to-VM Host communications and for secure guest memory transfer. Using NTP for time synchronization is strongly recommended on all VM Hosts and guests to maintain consistent time accuracy.

9.3.1 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 VM Host

On each VM Host, 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 VM Hosts that you would like to synchronize, you can add "peer" references in the `/etc/ntp.conf` file for each of those associated VM Hosts, 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 VM Host'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 VM Host.

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 VM Host. 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 VM Host 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 VM Host, the VM Host 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 VM Host 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 VM Host to use each other for peer mutual time synchronization.

9.3.2 VM Host Requirements and Setup

The VM Hosts must be configured with Integrity VM Version 4.3 or later. 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 Integrity Virtual Machines Release Notes, available from http://www.hp.com/go/virtualization-manuals, for general Integrity VM installation details, including supported VM Host operating system versions, patches, and other system requirements. Required patches are available on the http://www.itrc.hp.com website.

9.3.2.1 VM Host Processors for Online Migration

VM Hosts 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 VM Hosts. However, for online migration, all the eligible VM Host 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 31 lists the recent Itanium processors showing different values for processor family:

Table 31 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

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
          . . .
```

9.3.2.2 Private Network Setup

Source and target VM Host 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 VM Host 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, VM Host system `host2` is using network interface lan3 as its private network to connect to VM Host `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 VM Host 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` VM Host 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 VM Host 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
```

9.3.2.3 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 VM Host 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 VM Host's *`hostname`* for the `hpvmmigrate -h` option.

9.3.2.4 Using NTP on VM Hosts

Using NTP to synchronize clocks is strongly recommended for Online VM Migration environments. In addition to a typical NTP configuration, all the potential VM Hosts should use each other as mutual peer NTP servers to help maintain time consistency between hosts.

9.3.3 SSH Setup Between the VM Hosts

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 VM Hosts. 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 VM Hosts 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.

9.3.3.1 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,

```
# ll -d /
drwxr-xr-x 20 root root 8192 Apr 29 06:25 /
```

If your VM Host's root directory has different permissions than displayed in the previous example, use the `chmod` command to correct them.

```
# chmod 755 /
```

If a VM Host 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 VM Hosts 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.

9.3.3.2 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.

9.3.4 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.

You can migrate guests that use both the virtual I/O (VIO) and the accelerated virtual I/O (AVIO) drivers for storage and LAN, although only whole disk SAN storage and ejected file-backed DVDs can be migrated while the guest is online. All storage and network supported by Integrity VM are supported by Online VM Migration.

9.3.4.1 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 VM Host. 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.

9.3.4.2 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 VM Hosts
- 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 VM Host. 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.

9.3.4.3 Guest Storage Device Shareable Attribute not Propagated During Online Migration

The guest storage device shareable attribute is not propagated to the target VM Host 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 VM Hosts. 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 VM Host to WWID and then DSF on the target VM Host. Use `ioscan -C disk -P wwid` to see if the virtual machine's disks are presented to both VM Hosts 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 VM Host 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 VM Host). 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.

9.3.4.4 Using NTP on the VM Guests

Using NTP is strongly recommended for Online VM Migration environments. Each guest should include all potential VM Hosts as servers in its ntp.conf file so the current local VM Host 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.

9.3.4.5 Marking a Guest Not Runnable

On all VM Hosts that have a virtual machine configured, the virtual machine should be marked Runnable on only one VM Host 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 VM Host and Not Runnable on all other VM Hosts. If the Runnable state of a virtual machine is not correct on a VM Host, use the hpvmmodify command to correct it.

For information about the hpvmmodifycommand and how to mark a guest Runnable or Not Runnable, see Section 9.2.4 (page 186).

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 VM Host.

9.3.5 Restrictions and Limitations of Online VM Migration

Administrators should carefully configure certain aspects of VM Hosts 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 V4.3. 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 VM Host 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 VM Host, 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 migratingVMs.

Veritas volumes are not supported for Online VM Migration.

9.4 Migrating Serviceguard Packaged Guests and Serviceguard Cluster Nodes

In a Serviceguard cluster, you can use `hpvmsg_move` to migrate a packaged guest online. For example, to migrate the packaged guest `ogmlin` to `node2`, use the `hpvmsg_move` command as follows:

```
# hpvmsg_move -n node2 ogmlin
```

The Integrity VM V4.3 `hpvmsg_move` command disables the package before migrating the guest and re-runs the package after migration is finished. Therefore, the packaged guest is not protected by Serviceguard while migration is in progress.

A guest configured as a Serviceguard package is not protected by Serviceguard while the migration is in progress. The `hpvmsg_move` command temporarily disables package failover prior to the migration and re-enables package failover after migration has completed. If a failure occurs during the migration process, you must re-enable package failover manually and restart the guest by running it on one of the Serviceguard cluster nodes.

For example, to disable the guest `sgnode`, run the `hpvmmodify` command as follows:

```
# hpvmmodify -P sgnode -x online_migration=disabled
```

Disabling online migration for the guest ensures that you do not accidentally attempt to migrate a Serviceguard node.

The `hpvmsg_move` command is used to initiate online migrations of Serviceguard guest packages. This command is located in the `/opt/cmcluster/toolkit/hpvm` directory, which is not added to the users PATH variable during installation. To initiate an online migration, use the full pathname to the `hpvmsg_move` command. The full pathname to the command is: `/opt/cmcluster/toolkit/hpvm/hpvmsg_move`

9.5 Migrating Guests from Physical to Virtual Machines

The P2V assistant is a menu driven utility that helps you consolidate or migrate between systems. The source and target systems can actually be either virtual systems or physical systems, as the feature is implemented as a general purpose utility. To use the P2V assistant, enter the `p2vassist` command on the VM Host.

9.5.1 Requirements for Migrating a Workload

The Integrity VM software must be installed on the source system. The source system and target system must have SSH connectivity. You can connect from the VM Host system to the target server without a password (host-based authentication).

The P2V assistant migrates only applications that have Software Depot (SD) packages. Applications that are not SD compliant cannot be migrated with the P2V assistant.

9.5.2 Using the p2vassist Utility

The `p2vassist` utility is structured as a tree of menus, each of which can execute a function or give access to a deeper-level menu. Use the `p2vassist` command on the VM Host system. For example:

```
# p2vassist
P2V Assistant Manager
======================================
 1 ) General Configurations
 2 ) Manage Applications
 3 ) Manage O.S. Images
 -----
 0 ) Exit


1 - General Configurations
======================================
 1 ) Set Source Depot
 -----
 0 ) Return


2 - Manage Applications
======================================
 1 ) List Applications
 2 ) Select Applications
 3 ) Add Data Directories/Files
 4 ) Consolidate Applications
 -----
 0 ) Return


3 - Manage O.S. Images
======================================
 Function not implemented yet
```

To migrate a workload, follow these steps:

1. Set the location of the depots to use in installing the application. Select menu 1, option 1 and then enter the fully qualified depot path in either the form accepted by the SD products or as alocal directory. The location must be a registered depot source. For example, to specify `depsrv.hp.com:/release/1123.0706/ic054`, enter the location on the screen:

   ```
   P2V Assistant Manager
   ======================================

    1 ) General Configurations
    2 ) Manage Applications
    3 ) Manage O.S. Images
    -----
    0 ) Exit

   Enter Option Number: 1

   General Configurations
   ======================================

    1 ) Set Source Depot
    -----
    0 ) Return

   Enter Option Number: 1

           In this option you can set the location of the depots
           of the application to be installed on the target system.
           It can be a hostname or a local directory.

           Depots source currently configured: NONE
   ```

```
Depots source (type 'q' to return): depsrv.hp.com:/release/1123.0706/ic054
```

2. List the applications available in the source host. Select menu 2, option 1 (Manage Applications - List Applications). Note the names of the products to be selected. For example:

```
P2V Assistant Manager
======================================

 1 ) General Configurations
 2 ) Manage Applications
 3 ) Manage O.S. Images
 -----
 0 ) Exit

Enter Option Number: 2


Manage Applications
======================================

 1 ) List Applications
 2 ) Select Applications
 3 ) Add Data Directories/Files
 4 ) Consolidate Applications
 -----
 0 ) Return

Enter Option Number: 1

  ACXX                  C.06.00        HP aC++
  AudioDevKit           B.11.23.10     HP-UX Audio Desktop Developer Kit
  Auxiliary-Opt         B.11.23.11     Auxiliary Optimizer for HP Languages.
  BullseyeCoverIA       7.5.61         HP-UX IA
  C-ANSI-C              C.06.10        HP C/ANSI C Compiler
  C-Dev-Tools           B.11.23.11     C Language Development Tools
  gcc                   4.3.0          gcc
  gdbm                  1.8.3          gdbm
  .
  .
  .
  hpuxwsAPACHE          B.2.0.55.03    HP-UX Apache-based Web Server
  hpuxwsTOMCAT          B.5.5.9.04     HP-UX Tomcat-based Servlet Engine
  hpuxwsWEBMIN          A.1.070.07     HP-UX Webmin-based Admin
  hpuxwsXML             A.2.00         HP-UX XML Web Server Tools
  .
  .
  .
  vim                   7.0            vim
  xpm                   3.4k           xpm
  zlib                  1.2.3          zlib
```

3. Enter the applications to be migrated. Select menu 2, option 2 (Manage Applications - Select Applications). Enter the application names noted in the previous step, as a space-separated list. For example::
   - hpuxwsAPACHE
   - hpuxwsTOMCAT
   - hpuxwsWEBMIN
   - hpuxwsXML

```
P2V Assistant Manager
======================================

 1 ) General Configurations
 2 ) Manage Applications
 3 ) Manage O.S. Images
 -----
 0 ) Exit

Enter Option Number: 2
```

```
Manage Applications
======================================

 1 ) List Applications
 2 ) Select Applications
 3 ) Add Data Directories/Files
 4 ) Consolidate Applications
 -----
 0 ) Return

Enter Option Number: 2


  Select Applications
==============================

Please enter the list of depots to be installed on the target
system, separated by spaces.
You can list the applications using the first option of this menu.

Depots list (type 'q' to return): hpuxwsAPACHE hpuxwsTOMCAT hpuxwsWEBMIN hpuxwsXML
```

4. Specify application data transfer. To define directories or files, select menu 2, option 3 (Manage Applications - Add Data Directories/Files). Otherwise, proceed to step 5. The `p2vassist` utility suggests directories to set to transfer. For example:

```
P2V Assistant Manager
======================================

 1 ) General Configurations
 2 ) Manage Applications
 3 ) Manage O.S. Images
 -----
 0 ) Exit

Enter Option Number: 2

Manage Applications
======================================

 1 ) List Applications
 2 ) Select Applications
 3 ) Add Data Directories/Files
 4 ) Consolidate Applications
 -----
 0 ) Return

Enter Option Number: 3

Please enter the list of directories/files to be transferred to the
target system, separated by spaces.

The following directories are obtained by examining the depot
configuration, you may copy them to the prompt below in order to be used:

/opt/hpws/apache /opt/hpws/tomcat /opt/hpws/webmin

List (press <Enter> to return): /opt/hpws/apache /opt/hpws/tomcat /opt/hpws/webmin
```

5. Transfer the application and data to the new host. Select menu 2, option 4 (Manage Applications - Consolidate Applications). The target host name or IP address are verified for connectivity. Disk space is verified. When you press **Enter**, the data is transferred to the target host. For example:

```
P2V Assistant Manager
======================================

 1 ) General Configurations
 2 ) Manage Applications
 3 ) Manage O.S. Images
 -----
 0 ) Exit

Enter Option Number: 2
```

```
Manage Applications
======================================

 1 ) List Applications
 2 ) Select Applications
 3 ) Add Data Directories/Files
 4 ) Consolidate Applications
 -----
 0 ) Return

Enter Option Number: 4


  Consolidate Applications
==============================

This option will consolidate application(s) installed
on the current system and then migrate into another a
suitable target system. The target system is the machine
which will host the consolidated application(s).
This wizard requires that non-interactive Secure Shell
(ssh) access be set up between this system and the
target server.


Depots source currently configured: depsrv.hp.com:/release/1123.0706/ic054
Depots to be installed: hpuxwsAPACHE hpuxwsTOMCAT hpuxwsWEBMIN hpuxwsXML

Dirs/Files to be transferred: /opt/hpws/apache /opt/hpws/tomcat /opt/hpws/webmin

Please enter the hostname/IP
of the target system (type 'q' to return): vmp2v
Querying the system vmp2v for current status, this may take a few moments...
Installing depots...

Press ENTER to continue and begin transferring files.

Transferring Configuration files...
Checking application disk space...
Transferring /opt/hpws/apache directory to vmp2v, please wait...
Transferring /opt/hpws/tomcat directory to vmp2v, please wait...
Transferring /opt/hpws/webmin directory to vmp2v, please wait...
Application Sucessfully consolidated on host: vmp2v

Press <Enter> to return to menu:
```

9.5.3 Troubleshooting P2V Problems

If the application was already installed on the target host, the P2V assistant interrupts the installation process with the following message:

```
---------------------- [ Attention ] ----------------------

  One or more of selected application (s) are already
  installed on target system garopaba.
  The applications are listed below:

  Application - Target Version - Repository Version
  FIREFOXsrc - 1.5.0.00.01 - 1.5.0.00.01

  It's strongly recommended to stop that all applications
  listed above (or any other using them) before continuing
  with the installation.

  All selected applications already installed on target
  host will be removed. Notice that any patchs and/or other
  updates may be lost if older versions are installed.
```

```
  Type  below if you would like to continue
-----------------------------------------------------------
```

The P2V assistant prompts you either to continue or to stop the installation. If you choose to continue, the P2V assistant first uninstalls the application on the target host, then installs the selected one.

Do not continue the installation if older versions can break dependencies on another applications. In addition, install older versions of applications might overwrite updated libraries that were installed by patches.

10 Using HP Serviceguard with Integrity VM

After you install Integrity VM and create the guest, you can install HP Serviceguard on either the VM Host system (to provide failover for the guest), or on the guest (to provide failover for applications running on the guest).

This chapter describes how to configure Serviceguard with Integrity VM and assumes you are familiar with HP Serviceguard. The procedures in this chapter use the HP Serviceguard commands to accomplish Serviceguard tasks. You can use Serviceguard Manager instead. For more information, see the *Managing Serviceguard* manual.

10.1 Introduction to HP Serviceguard

HP Serviceguard is used to create high availability clusters using a networked grouping of HP Integrity servers. These servers are typically configured with redundant hardware and software components to eliminate single points of failure (SPOFS). Serviceguard is designed to keep application services running in spite of hardware (for example, system processing unit, disk, LAN, and so forth) or software (for example, operating system, user application, and so forth) failures. In the event of a hardware or software failure, Serviceguard and other high availability subsystems coordinate operational transfer between components.

Using Serviceguard together with Integrity VM provides the ability to:

- Minimize both planned and unplanned downtime of VM guests using Online VM Migration and Serviceguard
- Migrate workloads using the flexibility of Integrity VM and the control of Serviceguard
- Failover Integrity VM environments to other cluster node configurations
- Meet the consolidation and high availability requirements of many business-critical customers

Serviceguard uses packages to group application services and resources that are typically configured to run on several nodes in the cluster, one at a time. In the event of a service, node, network, or other monitored package resource failure on the node where the package is running, Serviceguard can automatically transfer control of the package to another node in the cluster, thus allowing the services to remain available with minimal interruption.

Serviceguard is designed to run on nodes and manage applications encapsulated within packages, with the integration of Serviceguard with Integrity VM, there are several configuration models that you can use to help effectively design Serviceguard clusters when using VMs for consolidation:

- VMs as Serviceguard Nodes

  The virtual machine is a member of a Serviceguard cluster, allowing failover of application packages between other physical or VM nodes in the cluster (Serviceguard runs within the VM guest).

- VMs as Serviceguard Packages

  The virtual machine is encapsulated within a Serviceguard package, allowing failover of the virtual machine between cluster nodes (Serviceguard runs on the VM Host).

- Combining VMs as Serviceguard Package and Node configurations

  Virtual machines as Serviceguard packages and as Serviceguard nodes in separate clusters can coexists on the same VM Host, allowing fail over of the VM guests and application packages within their respective clusters.

10.2 VMs as Serviceguard Nodes

The VMs as Serviceguard Packages configuration provides high availability for virtual machines encapsulated within Serviceguard packages. In a VMs as Serviceguard Nodes configuration (or

VMs as Nodes), HP-UX virtual machines are used as actual Serviceguard cluster nodes to provide the same HA failover capabilities found in traditional Serviceguard cluster configurations. Essentially, you can use Integrity VM to consolidate Serviceguard clusters on to virtual machines. VMs as Serviceguard Node cluster configurations can span across the following:

- VMs on separate VM Hosts
- VMs on the same VM Host
- VMs and separate physical nodes or vPars

With virtual machines and separate physical nodes or vPars, Serviceguard provides high availability in the event of a virtual machine or application failure. A failed application can restarted either within the same virtual machine or failed over to the physical node. With virtual machines on separate VM Hosts, Serviceguard also provides high availability in the event of a virtual machine, physical machine, or application failure. A failed application can be restarted either within the same virtual machine or failed over to another node (either physical or virtual).

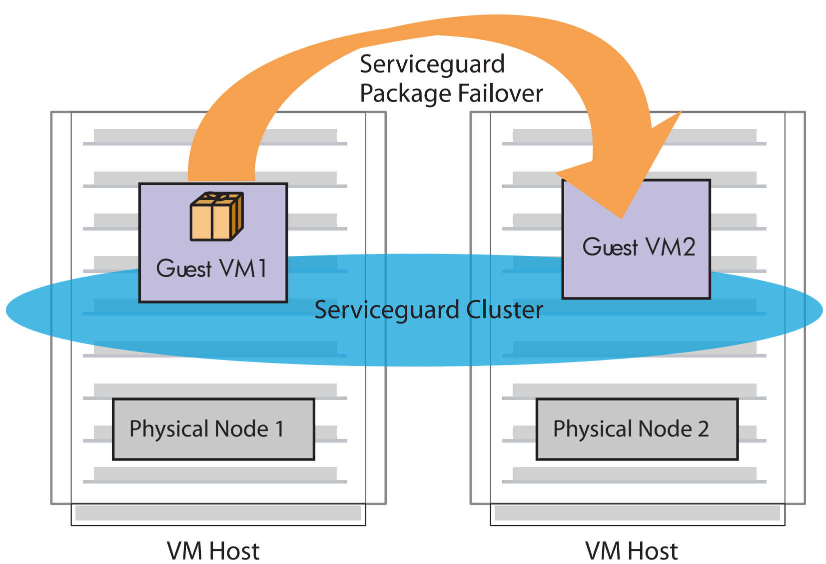
Two or more virtual machines can form a Serviceguard cluster operating within a single physical node. This configuration is similar to using vPars within a single physical system to form a cluster-in-a-box. In this configuration, a failed application can be restarted either within the same virtual machine or failed over to another virtual machine operating on the same physical node.

NOTE: HP recommends that you do not use a cluster-in-a-box configuration for mission-critical applications, because there is no electrical isolation between the virtual machine nodes and the physical node hosting the virtual machine, which creates a single-point-of-failure (SPOF). The VM Host can also be considered an SPOF.

10.2.1 VMs on Separate VM Hosts

Figure 13 shows the configuration of an application package that can fail over to a guest running on a different VM Host system.

Figure 13 Guest Application Failover to a Guest on a Different VM Host

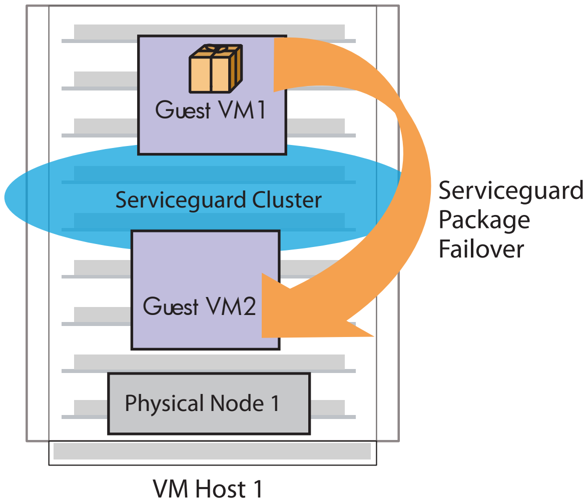


In this configuration, the Serviceguard nodes are guests running on either separate hard partitions (nPars) or HP Integrity servers. Note that Integrity VM does not run on soft partitions (vPars).

10.2.2 VMs on the Same VM Host

Figure 14 shows the configuration of an application package that can fail over to another guest on the same VM Host system.

Figure 14 Guest Application Failover to Another Guest on the Same VM Host



In this configuration, the primary node and the adoptive node are guests running on the same VM Host system. This cluster does not provide protection against single point of failure (SPOF), because both the primary cluster member and the adoptive cluster member are guests on the same physical machine. However, this configuration is useful in testing environments.

If you are running more than one guest on the VM Host system, and you need to share the same storage among the guests, you must change the SHARE attribute of the shared disk to YES using the `hpvmdevmgmt` command. For example:

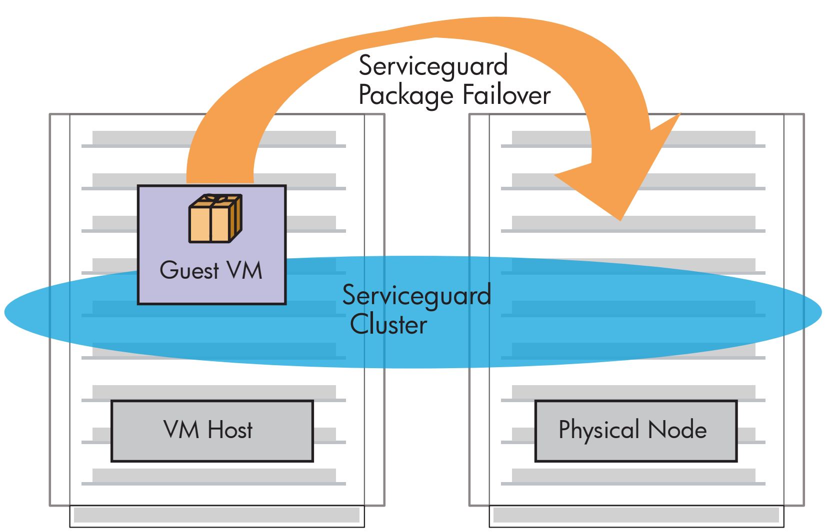
```
# hpvmdevmgmt -m gdev:/dev/rdisk/disk1:attr:SHARE=YES
```

For more information about using the `hpvmdevmgmt` command, see Section 8.12 (page 171).

10.2.3 VMs and Separate Physical Nodes

Figure 15 shows the configuration of an application package that can fail over to a physical node or partition that is not running Integrity VM software. In this case, the physical node may be a discreet physical system, a hard partition (nPar), or a soft partition (vPar).

Figure 15 Guest Application Failover to an HP Integrity Server



The Serviceguard cluster consists of a VM Host system and a Serviceguard node that is not running Integrity VM. The application configured as a Serviceguard package can fail over to the physical node. Alternatively, you can run the application on the physical node and configure the guest on the VM Host system as the adoptive node.

10.2.4 Usage Considerations

VMs as Serviceguard Nodes configurations should be considered whenever there is a need for consolidating systems in Serviceguard clusters, and the applications require full HA monitoring and failover functionality that is provided by Serviceguard. These configurations allow for a reduction in the total number of physical systems required for clusters by moving cluster nodes from individual physical systems to multiple VMs running on single systems.

As with VMs as Serviceguard Packages configurations, Integrity VM Host nodes should run only virtual machines and not other user applications. Doing so could adversely affect the allocation of system resources to the VM guests running on the VM Hosts.

Monitoring of applications within a VMs as Node configuration is the same as in a traditional Serviceguard cluster, because Serviceguard is running within the virtual machine. No specialized monitoring agents are required as is the case for VMs as Package configurations.

Package failover times for VMs as Serviceguard Node configurations are similar to traditional Serviceguard cluster failovers, because application packages are simply restarted on their pre-configured adoptive nodes. These configurations do not experience the additional virtual machine boot time that is part of VMs as Package configurations, where the failed over virtual machine package must restart the virtual machine on its adoptive node.

Integrity VM has a default 5-second network polling interval that increases Serviceguard failover time by several seconds. To insure the vswitches used by Serviceguard failover is less than 5 seconds, this polling interval can be set when using Serviceguard with Integrity VM to a recommended value of 2-seconds by modifying the `HPVMNETINTVAL=n` parameter in the file `/etc/rc.config.d/hpvmconf`.

NOTE: The value *n* is an integer between 1 and 10 that specifies the number of seconds for the polling interval.

10.2.5 Cluster Reformation

Cluster-reformation time is somewhat longer for VMs as Serviceguard Node configurations (approximately 40-70 seconds longer compared to a Serviceguard cluster consisting only of physical nodes). This allows for all outstanding I/O requests from the VM through the VM Host virtualization layer to complete before cluster activities can resume following a cluster reformation. Serviceguard uses an `io_timeout_extension` parameter that is set at cluster-configuration time to extend the quiescence period of the cluster reformation based on whether a VM node is present in the cluster and the I/O timeout settings on the VM Host. It is important to note the following:

- The `io_timeout_extension` parameter is set internally by Serviceguard and is not configurable by the user; however its value can be viewed using the Serviceguard `cmviewconf`, `cmviewcl v f` commands, or can be found in the system log file.
- HP highly recommends that you install the VM management software, especially on VM functioning as Serviceguard nodes, for Serviceguard to determine an optimal `io_timeout_extension` value (otherwise, Serviceguard assumes the most conservative value of 70 seconds resulting in unnecessarily lengthening the cluster recovery time).
- Be aware that the online addition or removal of VM cluster nodes or changes to cluster membership parameters (available with the Serviceguard A.11.18 and later releases) can affect the cluster quiescence period.

  In a failure scenario where the pending I/Os from a VM are not cleared within its extended quiescence time period, the Integrity VM software performs a TOC (Transfer of Control, or CPU reset) on the VM Host servicing the guest to ensure data integrity by terminating any outstanding I/O requests from the affected VM.

When performing a Serviceguard cluster consolidation, as with any workload consolidation using Integrity VM, careful planning of the VM configuration is required to ensure proper performance of the virtual machines by having a sufficient number of processors and available memory, in addition to storage and network I/O connections, to handle their workloads. Any initial performance problems with a virtual machine can be compounded when application workloads are failed over to it by Serviceguard in response to a failure in one of the other cluster members.

10.2.6 Availability

Integrity VM instances are not highly available in VMs as Nodes configurations. A failure of a VM is similar to a node failure in a Serviceguard cluster. The use of Serviceguard within the VM provides high availability for the applications running in the VM. VMs as Serviceguard Nodes configurations do have a shortcoming in that the adoptive failover VMs must be executing and consuming some degree of VM Host resources, which could potentially be used by other VMs that are not part of the Serviceguard cluster. The Integrity VM dynamic memory allocation feature should be considered to better manage adoptive VM memory usage during application failovers.

NOTE: HP recommends not using VMs as Serviceguard Nodes configuration with virtual machines on the same VM Host (cluster-in-a-box) for mission- or business-critical applications, because the physical VM Host is an SPOF. If the physical system fails, the entire cluster also fails.

10.2.7 Storage Considerations

An important distinction between VMs as Serviceguard Package and Node configurations is that VMs as Serviceguard Node configurations have limitations on the backing store selection depending on the use of the storage on the node. The guest root or systems disks, those not used by application that can fail over, can be of any supported backing store type. Shared storage disks, those used by applications on more than one node that will be accessed by failover applications, must be only whole disk VM backing stores.VMs as Serviceguard Node configurations support only whole disk VM backing stores, because:

- It not possible to set timeouts on logical volumes or file systems presented as backing stores to the VM. Any errors generated from these types of backing stores are not passed through the virtualization layers from the VM Host to VM that would allow Serviceguard running in the VM to react to these conditions.
- Disk I/O performance and the speed at which I/O requests can be completed prior to a VM node failure can affect cluster-reformation time. For more information about handling outstanding I/O requests during a VM node failure, see Usage Considerations.
- Data used by applications protected by Serviceguard packages must reside on shared storage that is physically connected to all nodes in the cluster and can be placed in LVM or VxVM logical volumes or on a cluster file system (CFS) that is accessible by the VM.
- The storage for the application data presented to the VM guest by the VM host must be whole disks so the logical volume and file system structures on this storage can be accessed by the other nodes in the cluster during a Serviceguard package failover.

10.2.8 Limitations Associated with These Configurations

Online migration of VMs as Serviceguard Nodes is not supported at the present time due to the guest freeze time causing a loss of cluster node heartbeats in the migrating VM. This triggers a cluster reformation and the removal of the VM from the cluster. To avoid accidental online migration of VMs as Serviceguard nodes, HP recommended that you use the following command to disable online migration of the VM:

```
# hpvmmodify -P Serviceguard-node vm-name –x online_migration=disabled
```

The following limitations apply:

- Guest OS boot disks can be of any backing-store type; however, shared storage, the storage used by the application on the guest that is accessed from more than one virtual machine, can be only whole disk or SLVM.
- DVD and tapes devices can be defined on one or more VM and are typically ignored by Serviceguard. For additional information about excluding a Device from probing, see the *Managing Serviceguard* manual.

10.2.9 Additional Considerations for VMs as Serviceguard Nodes Configurations

Serviceguard clusters rely on a cluster daemon process called `cmcld` that determines cluster membership by sending heartbeat messages to other `cmcld` daemons on other nodes within the cluster. The `cmcld` daemon runs at a real-time priority and is locked in memory. Along with handling the management of Serviceguard packages, `cmcld` also updates a safety timer within the kernel to detect kernel hangs, checks the health of networks on the system and performs local LAN failovers. Status information from `cmcld` is written to the node's system log file.

In VMs as Serviceguard Nodes configurations, there are some situations where VMs defined with multiple vCPUs, or a single vCPU with insufficient entitlement, can potentially experience `cmcld` run-time delays under heavy processing load conditions. If the run-time delay is longer than the configured cluster NODE_TIMEOUT value (that is, the time after which a node might decide that the other cluster node has become unavailable), `cmcld` triggers a cluster reformation just as if a node had failed. However, because no node has actually failed, the cluster reforms potentially with the same number of nodes it originally had before the `cmcld` run delay was reported depending on the length of the run delay.

Other factors that might contribute to this situation include vCPU processing entitlement percentages and the number of vCPUs assigned per VM as they relate to HP-UX kernel time-slice processing. `Cmcld` run delays can be identified by the following warning reported in the system log file:

```
[date/time VM name] cmcld [PID]: Warning: cmcld process was unable to
run for the last x.yz seconds
```

HP recommends that you install the latest Fair Share Scheduler patches on all VM Hosts to minimize the possibility of encountering this problem and triggering false cluster reformations. Another option is to increase the Serviceguard cluster NODE_TIMEOUT to a value larger than the run delay reported in the syslog file. The default value for this cluster parameter is 2 seconds, and the maximum recommended value is 30 seconds. However, for most installations, a setting between 5–8 seconds is appropriate.

NOTE: Increasing the NODE_TIMEOUT value causes the cluster to take longer to react to an actual node failure.

10.2.10 Creating the VMs as Serviceguard Nodes Configuration

This section assumes that you have installed the appropriate HP-UX operating environment and Integrity VM software on the VM Hosts and that the Serviceguard software has been installed on any physical nodes. The interrelated configuration of the physical hosts, VM Hosts, virtual machine, Serviceguard cluster, and Serviceguard packages dictates that all components be configured and created in a particular sequence.

The following process is one of many ways to create a VMs as Serviceguard Packages configuration. Note, you can use the Virtualization Manager or the Serviceguard Manager as alernatives when appropriate.

On the VM Hosts and the physical Serviceguard Nodes, perform the following steps:

1. Create the network configuration on each of the VM Hosts and physical Serviceguard Nodes.

NOTE: APA is not supported in the VM guest, and primary and secondary LAN failover is not provided in the VM Host.

2. On each of the VM Hosts, use the `hpvmnet` command to create vswitches.
3. Create the storage configuration on each of the VM Hosts and physical Serviceguard Nodes.
   - System disk storage for physical nodes, as well as VM nodes, can be of any type support by Serviceguard and Integrity VM.
   - System disks are required to be presented or available only on the VM Host or physical node where the node runs.
   - Shared package storage is limited to whole disk storage.
   - Shared package storage must be presented to each VM Host and physical node where the package application runs.
4. Ensure that each guest has access to a quorum server or cluster lock disk.
5. Run the `hpvmcreate` command to create the virtual machine.
6. Run the `hpvmstart` command to start the virtual machine.

On the Integrity VM guest, perform the following steps:

1. Install and configure the guest operating system and patches.
2. Install the Integrity VM guest kits.
3. If Serviceguard is not included in the HP-UX OE, install Serviceguard on the virtual machine nodes.
4. Install Serviceguard on the HP-UX guests that will run the application.

On any virtual machine node or physical node, perform the following steps:

1. Use the `cmquerycl` command to specify the nodes to be included in the cluster and to generate a template for the cluster configuration file. For example, to set up a cluster named `gcluster` that includes nodes `host1` and `host2`, enter the following command:

   ```
   # cmquerycl -v -C /etc/cmcluster/gcluster.config -n host1 -n host2 -q quorum-server-host
   ```

   Include the `-q` option if a quorum server is used on the cluster.
2. Edit the `/etc/cmcluster/`*`cluster-name`*`.config` file (where *`cluster-name`* is the name of the cluster specified in the `cmquerycl` command). For details about modifying the information in the cluster configuration file, see the *Managing Serviceguard* manual.
3. Use the following command to verify the contents of the file:

   ```
   # cmcheckconf -k -v -C /etc/cmcluster/gcluster.config
   ```

   This command ensures that the cluster is configured properly.
4. Generate the binary configuration file and distribute it using the following command:

   ```
   # cmapplyconf -k -v -C /etc/cmcluster/gcluster.config
   ```
5. Start the cluster using the following command:

   ```
   # cmruncl
   ```

These procedures provide an example of creating guest application packages. For information about how to set up your Serviceguard configuration, see the *Managing Serviceguard* manual.

10.2.11 Modifying and Managing the Virtual Machine and the Cluster

While the interrelated configuration of the physical hosts, VM Hosts, virtual machine and Serviceguard must always be taken into consideration; the modification and management of the virtual machine and Serviceguard cluster, in a VM as Serviceguard Node configuration, are relatively independent of each other. You modify and manage a Serviceguard cluster by using typical Serviceguard tools and procedures ,while you modify and manage virtual machines by using Integrity VM tools and procedures. The virtual machine is basically unaware that Serviceguard and all of its processes, packages, and applications are running as a cluster in the guest, just as Serviceguard considers the virtual machine environment as an alternative platform type. Even with its independent management focus, you should consider certain interaction between the virtual machine and Serviceguard environment.

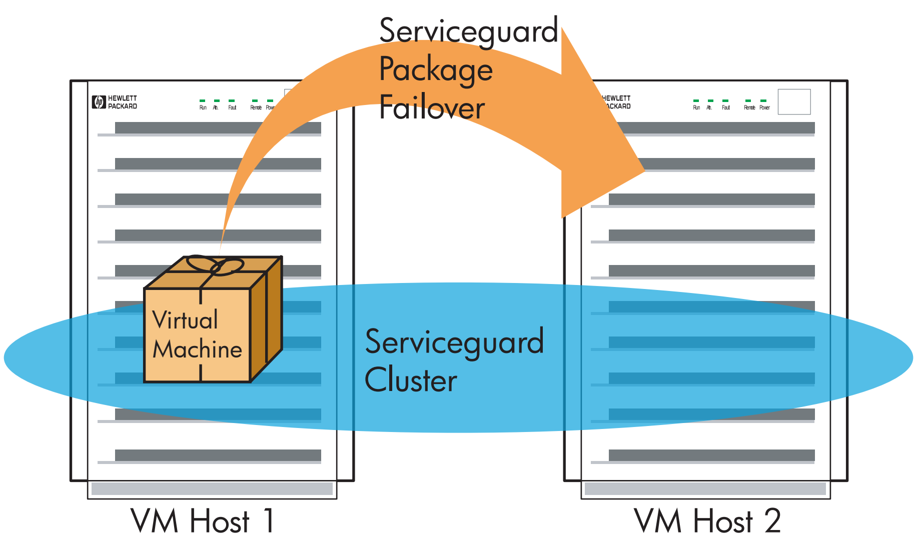
Changes to the VM Host and virtual machine and resources associated with the virtual machine might impact the Serviceguard environment running in the virtual machine, which could impact performance and availability. To minimize this impact, you should failover all applications and packages and optionally halt the node running on the virtual machine prior to making changes on the VM Host.

Unlike a Serviceguard cluster based solely on physical hardware, redundancy and availability is divided between the VM Host and the Serviceguard cluster. You should carefully consider this arrangement when planning hardware and software upgrades.

10.3 VMs as Serviceguard Packages

This section describes how to configure a VM Host as a Serviceguard Package. Figure 16 shows the configuration of an application package that can fail over to another guest on the same VM Host system.

Figure 16 Virtual Machine Failover to Another Cluster Member



10.3.1 Cluster Components

The resources on the VM Host system configured as VMs as Serviceguard Packages are shared by Serviceguard, the VM Host, and all the running guests. Some resources must be made exclusive to Serviceguard, such as the heartbeat network, and some must be shared with the VM Host, such as the VM Host operating system boot disk. Guests running simultaneously share the remaining memory and processing power.

To provide a high level of availability, a typical cluster uses redundant system components, for example two or more SPUs and two or more independent disks. This redundancy eliminates single points of failure. In general, the more redundancy, the greater your access to applications, data, and supportive services in the event of a failure.

In addition to hardware redundancy, you must have the software support that enables and controls the transfer of your applications to another SPU or network after a failure. Servceguard provides this support as follows:

- In the case of LAN failure, Serviceguard switches to a standby LAN or moves affected packages to a standby node.
- In the case of SPU failure, your application is transferred from a failed SPU to a functioning SPU automatically and in a minimal amount of time.
- For failure of other monitored resources, such as disk interfaces, a package can be moved to another node.
- For software failures, an application can be restarted on the same node or another node with minimum disruption.

Serviceguard also gives you the advantage of easily transferring control of your application to another SPU to bring the original SPU down for system administration, maintenance, or version upgrades.

10.3.2 VM Hosts as Serviceguard Nodes

When planning the VM Host and Serviceguard nodes, you need to consider the following:

- Proper memory and physical CPU allocation

  The number of physical CPUs per Serviceguard node limits the number of virtual CPUs (vCPUs) with which a guest can run. If your Serviceguard nodes differ in the number of CPUs on each Serviceguard node, then it might not be possible to failover a VM package running on a Serviceguard node with a large number of physical CPUs (pCPUs) to a Serviceguard node with a smaller number of physical CPUs.

  Do not reduce the physical CPU count below the virtual CPU (vCPU) count of any guest. No running guest should be allocated more vCPUs than the VM Host system has physical processors.

  A set of Integrity VM servers can be grouped into an Integrity VM multiserver environment, MSE. This grouping is required when a set of Integrity VM Hosts is also configured as a Serviceguard cluster. The MSE environment must be configured consistently on each node of the VM Host cluster. To set up an Integrity VM multiserver environment, use the `hpvmdevmgmt` command on each of the servers in the group to establish each server entry and the MSE group entry. For more details about MSE, see the Integrity VM manpages section in this document or on your Integrity VM system.

- Online migrations

  To be compatible with online migration, VM configured as Serviceguard packages must conform with the restriction and limitations outlined in Section 9.3.5 (page 195).

- Virtual machine entitlement

  Virtual machine entitlement is the minimum amount of processing power guaranteed to the virtual machine from each vCPU. When allocating minimum entitlement to a virtual machine, it is important to plan your failover strategy, so that a failover of a set of virtual machine packages to a single Serviceguard node does not exceed the total available CPU processing on that node.

- CPU processing overhead

  Take into consideration the CPU processing overhead that both Serviceguard and the VM Host require.

- The amount of physical memory on each Serviceguard node

  The amount of physical memory on each Serviceguard node must also be considered when planning your VMs as Serviceguard Packages configurations and failover strategy.

- Differing amounts of physical memory of each Serviceguard node

  Consideration should be given to the impact and effects of configuring dynamic memory and the minor memory overhead associated with Serviceguard running on the VM Host. If your Serviceguard nodes differ in the amount of physical memory on each Serviceguard node, it might not be possible to failover a virtual machine package running on a Serviceguard node with a large amount of physical memory to a Serviceguard node with less available 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. In VMs as Serviceguard Package configurations, the virtual machines can be configured with differing dynamic memory configurations on different Serviceguard nodes. If you plan to use the dynamic memory feature, you should understand how it might impact your virtual machine configurations and failover strategy. For details about configuring dynamic memory, see Section 9.7 (page 151).

10.3.3 Network Components

To maximize availability, a minimum of three physical network interface cards (pNIC) are recommended, one each VM Host. One pNIC is configured as a dedicated Serviceguard heartbeat, and therefore, is not available for any use by Integrity Virtual Machines. Two additional pNICs are configured as a primary or standby LAN pair for use by VM guests and are monitored by Serviceguard on the VM Host. The Serviceguard network monitor provides network failure detection for identifying failed network cards based on inbound and outbound message counts and failing over to configured standby LANs.

Auto-Port Aggregation (APA) is supported and can be used to provide network bandwidth scalability, load balancing between the physical links, automatic fault detection and HA recovery.

NOTE: When using APA, make sure you have at least two pNICs configured to avoid a single-point-of-failure for the Serviceguard cluster heartbeat connections.

As with all VM configurations, to allow guests to access network devices, you must create vswitches on the VM Host. When configuring vswitches in a VMs as Serviceguard Packages configuration, you configure vswitches only on the pNICs associated with the primary LANs. You do not configure a vswitches on the pNIC associated with the Serviceguard Heartbeat network or the pNIC associated with the standby LAN.

When the Serviceguard network monitor detects a network failure on the pNIC associated with a primary LAN, it fails the LAN over to the pNIC associated with the standby LAN. When this occurs, the vswitch monitor, which is responsible for monitoring the activities of the Serviceguard network monitor, automatically moves the vswitch configuration from the pNIC associated with the primary LAN to the pNIC associated with the standby LAN.

All VLAN configurations supported by Integrity Virtual Machines are supported in VMs as Serviceguard Packages configurations. All VLAN configurations supported by Serviceguard are supported in VMs as Serviceguard Packages configurations.

10.3.4 Storage Considerations

VMs as Serviceguard Package configurations support all VM guests backing store types, including:

- Whole disks
- LVM logical volumes
- VxVM logical volumes
- Files on any of the storage types listed, including files on a Cluster File System (CFS).

The VM guest backing stores must reside on shared storage, so that it is accessible by all VM Hosts in the Serviceguard cluster to allow failover of the VM guests. Configuring shared storage for VM Hosts is accomplished in a similar manner as a standard Serviceguard cluster configuration, with the only difference being the shared storage must be defined as VM storage devices that are exclusively used by a specific VM guest.

Both standard LVM and VxVm logical volumes are only active and available on one Serviceguard node at a time. Whole disks, Shared LVM logical volumes, Vxvm logical volumes managed by the Cluster Volumes Manager (CVM) as well as files on a Cluster Files System can be configured to be active and available on multiple Serviceguard nodes simultaneously. Additional care and planning should be taken when configuring VMs as Serviceguard Packages to insure that no more then one guest on one node can access these backing store types.

The Integrity VM toolkit script, `hpvmsg_package`, is used to create Serviceguard package configuration file and control script templates for the VMs to be protected by Serviceguard packages. This script is designed to determine the cluster-shared backing store and application data storage used by the VM guest to be packaged and adds the appropriate logical volume and mount point entries into the package control script for guest failover.

When creating a Serviceguard cluster with virtual machines acting as packages, the `hpvmsg_package` script correctly identifies CVM logical volumes and CFS files backing stores used by guests but requires users to verify or provide activation modes and package dependencies for the backing stores.

The `hpvmsg_package` script creates the package configuration in the package directory: `/etc/cmcluster/`*`hpvm-name`*`/`*`hpvm-name`*`.conf`. Inside the configuration file are instructions, examples, and default and assigned named-values pairs describing the resources used by the virtual machines and controlled and monitored by Serviceguard.

For information about selecting the appropriate values for these items, see the templates files in the *Managing Servicguard* manual and the *Veritas Storage Foundation 5.0 Cluster File System Administration Guide Extracts for HP Serviceguard Storage Management Suite*.

10.3.5 Limitations Associated with These Configurations

The following limitations are associated with Serviceguard configurations:

- The only backing-store types permitted with Online VM Migrations are:
  - Whole disk
  - Shared LVM logical volumes

    If you plan to use Online VM Migration with VMs as Serviceguard Packages, you must limit your VM configuration to one of these backing-store types.

- The VM Host presents devices to the virtual machines as virtual devices. Attached I/O devices; such as tape, DVD, burner, and changed, are not presented as virtual devices; but instead, they are presented as direct I/O devices. In a typical Serviceguard configuration, these devices are ignored and are not expected to be present on more than one Serviceguard node. The `hpvmsg_package` script does not insert information about these devices into the package configuration file. For information about excluding a device from probing, see the *Managing Serviceguard* manual.
- When Integrity VM guests are configured as packages in a Serviceguard (SG) cluster, `hpvmstatus` displays which VM Host is running the distributed guests as an SG package. Because this information comes from SG, it can be delayed by as much as 10 seconds. This delay does not cause any risk of starting the same guest on two different Integrity VM Hosts, because SG is controlling the start of these guests and allows a single instance to run at any one time.
- The `hpvmsg_package` does not add appropriate entries to the package configuration and control script files. After running the `/opt/cmcluster/toolkit/hpvmsg_package` script

to package a guest that contains CVM or CFS backing stores, review and modify the package configuration and control scripts for each cluster member. As part of this process, add CVM and/or CFS backing store entries to these files.

- Guests configured as Serviceguard packages should be stopped and started using Serviceguard package control commands. Do not use the Integrity VM commands (`hpvmstart`, `hpvmstop`, and `hpvmconsole`) to start and stop these types of guests.

There is no limitation associated with the use of AVIO functionality with VMs as Serviceguard Packages configurations.

Depending on the system configuration, you might see an error similar to the following when running `cmcheckconf` or `cmapplyconf` to the virtual machine packages:

```
# cmapplyconf -P /etc/cmcluster/hpvm-sg1-g4/hpvm-sg1-g4.conf
Incorrect permissions for /etc/cmcluster/hpvm-sg1-g4 (40777). Directory must be executable for owner, and not
writable by group and others on node hpvm-sg1-h1.

cmapplyconf: Error found in package file: /etc/cmcluster/hpvm-sg1-g4/hpvm-sg1-g4.conf.
```

To correct this issue, use a `chmod` command similar to the following to correct the permission of the package directory on each of the cluster members:

```
# chmod go-w /etc/cmcluster/hpvm-name
```

10.3.6 Operating System and Software Requirements

The following software is assumed to be properly installed and configured:

- Appropriate HP-UX version and the Data Center OE
- HP-UX patches
- Serviceguard Depots, if not included in the OE
- Serviceguard Patches
- Integrity VM Depot, if not included in the OS and OE
- Integrity VM Patches

Additionally, depending on your particular requirements, the following optional products might also need to be installed and configured:

- APA (HP-UX 11i v2 only)
- SecurePath (HP-UX 11i v2 only)
- SMS 5.0 Depots (provides CFS/CVM)
- SGeRAC Depot (provides SLVM) — T1907BA A.11.19.00 Serviceguard Extension for RAC

10.3.7 Creating the Package Configuration

The interrelated configuration of the Serviceguard cluster, VM Host, virtual machine, guest, and Serviceguard package dictates that all components be configured and created in a particular sequence. The following steps provide a sequence of activities, which is one of many ways to create a VMs as Serviceguard Packages configuration. Virtualization and Serviceguard Managers might be used as alternatives when appropriate. Note, some steps are performed on all nodes, the primary node, or the secondary nodes.

1. On all nodes, perform the following steps:
   a. Configure networks (IP address, APA, VLANS).
   b. Run the `hpvmnet` command to configure the vswitch for guest access.
   c. Configure the MSE using the `hpvmdevmgmt` command.
   d. Configure the lock disk or quorum server.
   e. Configure storage: LVM, shared LVM, VxVM, and file systems.
2. On the primary node, perform the following steps:

   a. Run the cmquerycl command to create Serviceguard ASCII configuration file.
   b. Edit configuration file to insure that the cluster name, networks usage, LVM devices, and lockdisk or quorum server are correct.
   c. Run the Serviceguard command cmcheckconf to check the configuration and cmapplconf command to apply the configuration file.
   d. Run the cmruncl command to start the cluster.
3. On all nodes, perform the following step:
   a. Configure CFS and CVM system-wide packages and guest storage devices and packages.
4. On the primary node, perform the following steps:
   a. Run the hpvmcreate command to create the virtual machine.
   b. Run the hpvmstart command to start the virtual machine.
   c. Install and configure the guest operating system.
   d. Install the guests kits and optionally the cmappserver software.
   e. Optionally configure any guest applications and their associated application monitor run commands.
   f. Run the hpvmstop command to stop the virtual machine.
   g. Unmount, export and deactivate storage.
5. On the secondary nodes, perform the following step:
   a. Configure, import, activate, and mount virtual machine storage.
6. On the primary node, perform the following step:
   a. Run the hpvmmigrate command with the -k option to distribute the virtual machine to the next node.
7. On the secondary nodes, perform the following steps:
   a. Run the hpvmstart command to verify the virtual machine operation.
   b. Verify the machine is functioning.
   c. Run the hpvmstop command to stop the virtual machine.
   d. Unmount, export and deactivate storage.
   e. Repeat steps 4 f. through step 7 d., running the hpvmmigrate command from the current Secondary Node for each of the remaining nodes.
8. On the primary node, perform the following steps:
   a. Reconfigure, import, activate, and mount virtual machine storage.
   b. Run the hpvmstart command to verify the virtual machine operation.
   c. Verify the machine is functioning.
   d. Run hpvmsg_package to package the virtual machine as a Serviceguard package.
   e. Run the hpvmstop command using the force option to stop the virtual machine.
   f. Run cmrunpkg to run the Package

The hpvmsg_package command, referenced in the previous steps, assists a user who is developing and managing a set of Serviceguard package configurations and associated toolkit files. The command creates these configurations by performing the steps required to associate a virtual machine to a Serviceguard failover package. For details about this command, ee the hpvmsg_package manpage described in this manual or the see the hpvmsg_package(1) manpage.

10.3.8 Managing the Package with Serviceguard Commands

After the virtual machine has been packaged, it is managed as a Serviceguard package using Serviceguard commands. Many of the Integrity VM commands are aware of this packaging and no longer function in the same manner as they did prior packaging. You can use any of the available Serviceguard tools the Serviceguard Manager or the command-line tools to manage basic package functionality.

The following command are used to perform these functions:

- The `cmrunpkg` command starts a package.
- The `cmhaltpkg` command stops a package.
- To perform a package failover, which is similar to an offline migration, use the `cmhaltpkg` command to halt the package and then the `cmrunpkg` command to start the package.

By default, a package is configured to automatically run on cluster start or in the event of a failure.

NOTE: The following situation might occur when you are using Insight Dynamics — VSE to manage distributed guests (guests that are configured as Serviceguard packages):

- There is a guest configuration file for each guest on each VM Host. Therefore, when you modify a distributed guest you must modify the guest on each VM Host that is a cluster node.

10.3.9 Online VM Migration

Online VM Migration (OVMM) is an optional feature of Integrity VM that enables a running virtual machine and its applications to be moved from one VM Host to another without service interruption. Integrity VM supports OVMM of virtual machines and VMs as Serviceguard packages for HP-UX guests.

Throughout the migration, all virtual machine I/O connections to storage and networks remain active. The primary benefits of online migration include:

- Minimizing downtime for planned maintenance activities on VM Hosts
- Balancing static workload between VM Hosts

You can also migrate virtual machines encapsulated within Serviceguard packages, providing the additional benefits of protection against unplanned VM Host hardware and software failures at any time other than during an online migration. The type of migration also protects planned movement of workloads that cannot be migrated online due to VM Host hardware or software differences.

To perform an online migration of a VM as a Serviceguard Package, execute the `hpvmsg_move` command that performs the following operations:

1. Disables Serviceguard VM package switching (that is, monitoring and failover functions are not available); however, the VM continues to run on the source VM Host.
2. Migrates the VM to the target VM Host online using the `hpvmmigrate` command, automatically invoked by the `hpvmsg_move` command.
3. Now that the migration is complete, re-enables VM packaging switching (that is, Serviceguard monitors and can failover the VM.

The hpvmsg_move command performs online migration of virtual machines protected by Serviceguard with the benefit of minimizing both the virtual machine planned and unplanned downtime. In this case, virtual machines can be migrated online with minimal interruption during planned events while Serviceguard protects the virtual machines from unplanned failures during normal operation.

NOTE: Serviceguard does not monitor or protect the virtual machine during the online migration process; however, the virtual machine and its applications are fully available except during migration process freeze time.

Online VM Migration does not support migrating guests that are Serviceguard cluster nodes. If a Serviceguard node is migrated while online, it might lose connection to other cluster members and be automatically removed from the cluster. The Integrity VM software does not know whether you have installed Serviceguard in a guest, so it cannot automatically restrict online migration of your Serviceguard node. Therefore, you should disable online migration for all Serviceguard nodes.

For example, to disable Online VM Migration for the guest **sgnode**, run the `hpvmmodify` command as follows:

```
# hpvmmodify -P sgnode -x online_migration=disabled
```

Disabling online migration for the guest ensures that you do not accidentally attempt to migrate a Serviceguard node.

10.3.10 Maintenance Mode

Using the Integrity VM commands, you can make changes to the VM configuration, disable maintenance mode and then distribute the VM configuration using the `hpvmmigrate` command. If the changes made to the VM configuration impact the Serviceguard Package configuration, the package should be repacked using the `hpvmsg_package` command, and then re-applied using the `cmapplyconf` files as previously outlined.

Maintenance Mode is enabled only on the local node, and when enabled, will periodically note that it is enabled for each VM in the system `syslog` file. In the event that the Serviceguard node on which the VM is running fails during maintenance mode, the VM package will be failed over to a secondary node. Depending on where in the process the failure occurred, changes made to the VM or package configuration might or might not have been distributed to the secondary node prior to the failover. After recovery, ensure that the VM package is not in maintenance mode and that the configuration is consistent of all nodes.

To enable or disable maintenance mode, use the `hpvmsg_package` command:

```
# hpvmsg_package -m [0|1] -P vm-name
```

Where specifying 1 enables maintenance mode and specifying 0 disables maintenance mode.

10.3.11 Serviceguard Virtual Machine Application Monitoring

Starting with Serviceguard A.11.19 and Integrity VM B.04.10 releases, HP supports the monitoring and control of applications within a virtual machine configured as a Serviceguard package. This functionality is available for any application running in an HP-UX guests using the Java Runtime Environment (JRE) and provides the following benefits:

- Checks the status of applications within virtual machines from the VM Host under control of Serviceguard.
- Provides startup and failure detection capabilities for monitored virtual machine applications using Serviceguard functionality.
- Provides a supported application monitoring framework without having to rely on custom-written software.

The `hpvmsg_package` script can help you develop guest applications monitors.

10.4 Combining the VMs as Serviceguard Package and Node Models

VMs as Serviceguard Packages and Serviceguard nodes in separate clusters can co-exist on the same VM Hosts, allowing failover of the VMs and application packages within their respective clusters. The combined configuration provides the most flexibility in meeting recovery-time objectives (RTOs) for VMs and applications while efficiently consolidating systems on VM Hosts. At the same time, the mission-critical applications with a higher RTO can be protected by Serviceguard packages running within HP-UX VMs, all running together on a minimum number of VM Hosts.

NOTE: Different versions of Serviceguard can be used for either cluster as long as they are listed as supported. For a list of supported combinations of Integrity VM and Serviceguard, see the *HP Integrity Virtual Machines 4.3: Release Notes*.

When implementing this configuration, there is a possibility that a failure of all heartbeat links between both clusters could cause the VM node cluster to fail depending on the timing in which the surviving nodes in each cluster achieves quorum and which cluster reforms first. This failure scenario can be mitigated by:

- Having multiple redundant heartbeat networks for both cluster
- Using Serviceguard A.11.19 for both clusters and setting the *MEMBER_TIMEOUT* parameter on the VMs as Serviceguard Node cluster to at least 1 second greater than the reported *max_reformation_duration* value of the VM Host cluster. To determine this value, use the following command:

  cmviewcl -v -f line | grep max_reformation_duration

NOTE: With Serviceguard and Integrity VM running, you might see the following types of messages in the `syslog` file:

```
Syslog entries - cmcld[XXXX]: Warning: cmcld process was unable to run for the last X.XX seconds
```

These messages can be ignored.

10.5 Migrating VMs That are Using Logical Volume Backing Stores

Integrity VM now supports the migration of virtual machines that are using LVM logical volume (lvol) backing stores configured with `avio_stor`, also known as shared LVM (SLVM). To enable SLVM functionality, Serviceguard A.11.19 and patch PHSS_40152 must be installed on the VM Host systems. In addition, for virtual machines using `avio_stor` adapters, version B.11.31.0910.01 of the HostAVIOStor software bundle must be installed. This software is available for download from http://software.hp.com/.

To configure SLVM, follow these steps:

1. Create the volume group and logical volumes.

   NOTE: If the LVM storage is already comprised of shared storage (LUNs), you do not need to recreate them, and therefore; may skip the following steps.

   If the virtual machine is already using LVM logical volume backing stores, make sure that the volume group(s) are comprised of storage (LUNS) that are exposed to all the VM Hosts (among which you want to migrate the virtual machine.) If the volume group (VG) is not comprised of share storage, create the LVM structures on the shared storage and copy the data from the existing logical volumes. For more information about LVM procedures, see the *HP-UX System Administrator's Guide: Logical Volume Management*.

   a. Select one of the VM Hosts and login to it. (This VM Host will be known as the primary VM Host in the future steps.

   b. On the primary VM Host, identify the shared LUN for the VM Hosts (for example, `/dev/disk/disk15`). Ensure that it is being shared among the other VM Hosts by verifying that storage is exposed to each VM Host. Use the `ioscan -P wwid` command to show the LUN wwid in the last field:

   ```
   # ioscan -P wwid /dev/disk/disk15
   Class     I  H/W Path  wwid
   ===============================
   disk     15  64000/0xfa00/0xb  0x600508b4000157f000020000027c0000
   #
   ```

   c. Create the physical volume:

```
# pvcreate /dev/rdisk/disk15
Physical volume "/dev/rdisk/disk15" has been successfully created.
#
```

d. Create the volume group (VG). With LVM Version 2.1, you can use the `-E` option of the `vgcreate` command to help identify the VG and extent sizes (which are required for LVM V2.1 of `vgcreate`).

```
# diskinfo -b /dev/rdisk/disk15
37748736
# vgcreate -V 2.1 -E -S 37748m
Max_VG_size=36g:extent_size=1m
# vgcreate -V 2.1 -S 36g -s 1m /dev/vgsharedA /dev/disk/disk15
Volume group "/dev/vgsharedA" has been successfully created.
Volume Group configuration for /dev/vgsharedA has been saved in /etc/lvmconf/vgsharedA.conf
#
```

e. Create the logical volumes:

```
# lvcreate -L 18748m -n lvdisk0 /dev/vgsharedA
Logical volume "/dev/vgsharedA/lvdisk0" has been successfully created with
character device "/dev/vgsharedA/rlvdisk0".
Logical volume "/dev/vgsharedA/lvdisk0" has been successfully extended.
Volume Group configuration for /dev/vgsharedA has been saved in /etc/lvmconf/vgsharedA.conf
# ...
```

2. Distribute the Volume Group Definition among the Serviceguard Nodes that are VM Host systems.

   With the volume group defined on the initial VM Host, that VG's definition needs to be distributed among the other VM Hosts:

   a. On the initial VM Host, deactivate the VG:

```
# vgchange -a n /dev/vgsharedA
```

   b. Export the LVM definition for the VG:

```
# vgexport -v -p -s -m /tmp/vgsharedA.map /dev/vgsharedA
Beginning the export process on Volume Group "/dev/vgsharedA".
/dev/disk/disk15
vgexport: Preview of vgexport on volume group "/dev/vgsharedA" succeeded.
```

   c. Copy (for example, with ftp or rcp) the map file to all the other VM Hosts.

   d. On the other VM Host systems, import the VG configuration by scanning the disks (`-s` option) on the VM Host populating persistent DSFs (`-N` option) taking care to use the same VG path as on the source VM Host:

```
# vgimport -N -v -s -m /tmp/vgsharedA.map /dev/vgsharedA
Creating "/etc/lvmtab_p".
Beginning the import process on Volume Group "/dev/vgsharedA".
Logical volume "/dev/vgsharedA/lvdisk0" has been successfully created
with lv number 1.
Logical volume "/dev/vgsharedA/lvdisk1" has been successfully created
with lv number 2.
vgimport: Volume group "/dev/vgsharedA" has been successfully created.
Warning: A backup of this volume group may not exist on this machine.
Please remember to take a backup using the vgcfgbackup command after activating the volume group.
#
```

3. Configure Serviceguard:

   a. On **all** VM Hosts on which this virtual machine needs to run, install Serviceguard A.11.19 (if it is not already installed.):

```
# swinstall -s depot-location T1905CB
```

   b. Download and install the Serviceguard patch that enables SLVM: PHSS_40152 (or later).

   c. Configure your VM Hosts for Serviceguard. For information about configuring your VM Hosts, see the Managing Serviceguard manual.

      For proper configuration and functionality, SLVM configurations require the following tasks be properly executed:

      1) Configure root-level access.
      2) Configure name resolution.

3) Enable network time protocol.
4) Choose cluster lock disks or quorum server.

d. On the primary VM Host, create the Serviceguard cluster. Start by defining the configuration file. In this example, the new configuration file is `/etc/cmcluster/creekQuorumRiver.cfg`. There are two nodes in the cluster, creek1 and creek2. The SG quorum is River. Use the quorum server for your location:

```
# cmquerycl -v -C /etc/cmcluster/creekQuorumRiver.cfg -n node1 -n creek2 -q creek
Looking for other clusters ... Done
Gathering storage information
Found 23 devices on node creek1
Found 23 devices on node creek2
Analysis of 46 devices should take approximately 5 seconds
0%----10%----20%----30%----40%----50%----60%----70%----80%----90%----100%
Found 2 volume groups on node creek1
Found 2 volume groups on node creek2
...

Writing cluster data to /etc/cmcluster/creekQuorumRiver.cfg.
#
```

e. Edit the configuration file and change the entry for each of the SLVM volume groups, so that they are listed as an HPVM_VOLUME_GROUP. If each entry is already listed as aVOLUME_GROUP, replace each line so that each entry is listed as an HPVM_VOLUME_GROUP. When you are finished, you should have this:

```
# grep VOLUME_GROUP /etc/cmcluster/creekQuorumRiver.cfg
VOLUME_GROUP          /dev/vgdatabase
VOLUME_GROUP          /dev/vg02
HPVM_VOLUME_GROUP     /dev/vgmobileG
HPVM_VOLUME_GROUP     /dev/vgsharedA
#
```

NOTE: Each SLVM volume group being used by Integrity VM must have an entry.

---

f. Check the cluster:

```
# cmcheckconf -k -v -C /etc/cmcluster/creekQuorumRiver.cfg
Begin cluster verification...
Checking cluster file: /etc/cmcluster/creekQuorumRiver.cfg
Defaulting MAX_CONFIGURED_PACKAGES to 300.
Checking nodes ...


Creating the cluster configuration for cluster River
Adding node creek1 to cluster River
Adding node creek2 to cluster River
cmcheckconf: Verification completed with no errors found.
Use the cmapplyconf command to apply the configuration.
#
```

Note the next to last message, “Verification completed with no errors found.”

g. Apply the configuration:

```
 # cmapplyconf -v -C /etc/cmcluster/creekQuorumRiver.cfg
Begin cluster verification...
Checking cluster file: /etc/cmcluster/creekQuorumRiver.cfg
Defaulting MAX_CONFIGURED_PACKAGES to 300.
Checking nodes ...


Creating the cluster configuration for cluster River
Adding node creek1 to cluster River
Adding node creek2 to cluster River
Marking/unmarking volume groups for use in the cluster
```

```
Completed the cluster creation
#
```

h. Start the cluster:

```
# cmruncl
cmruncl: Validating network configuration...
cmruncl: Network validation complete
Waiting for cluster to form .... done
Cluster successfully formed.
Check the syslog files on all nodes in the cluster to verify that no warnings occurred during startup.
#
```

4. Activate the volume groups correctly:
   a. On **all** the VM Hosts where this cluster is defined, activate the VG:

      ```
      # vgchange -a s -p /dev/vgsharedA
      Activated volume group in Shared Mode.
      This node is the Server.
      Volume group "/dev/vgsharedA" has been successfully changed.
      #
      ```

   b. Check the VG Status line of the `vgdisplay –v /dev/vgname` output to make sure that it is shared:

      ```
      # vgdisplay -v /dev/vgsharedA | grep "VG Status"
      VG Status                   available, shared, client
      ```

      If the VG is not listed as shared, deactivate it and reactivate it with `vgdisplay –v /dev/vgname`.

5. Install HostAVIOStor bundle.

   If you plan to migrate VMs using logical volumes with `avio_stor` adapter, install a new HostAVIOStor bundle on all VM Hosts. You do not need to install this bundle if your VMs use the legacy VIO `scsi` adapters.

   ```
   # swinstall -s depot-location HostAVIOStor
   ```

6. Troubleshooting

   The following problems might occur:

   - The `hpvmmigrate` fails with a message “Online migration requires only disk or null backing stores.”

     This is most likely caused by not having the latest patch installed on VM Host systems.

   - If you receive the following response from the `hpvmmigrate` command:

     ```
     hpvmmigrate:ERROR vmname: Source AVIO SCSI failure, status 252
     ```

     The system does not have the right HostAVIOStor bundle installed to support migration of VMs with logical volume backing stores. Make sure that the HP-UX 0910 version of the HostAVIOStor bundle is installed on all VM Hosts among which you plan to migrate your VMs.

7. Limitations

   Online migration of VMs with logical volume backing stores is not supported for the following configurations:

   - Logical volume backing stores contained in volume groups defined on disk partitions
   - VxVM logical volume backing stores
   - Logical volumes that are not shared across VM Host systems.
   - Logical volumes with different pathnames. That is, the pathname for the shared LVM logical volume must be identical across all VM Hosts.

10.5.1 Creating and Configuring VMs as Serviceguard Nodes Having SLVM Backing Storage

When creating and configuring virtual machines as Serviceguard nodes having SLVM backing storage, the following limitations apply:

- SLVM backing stores can be used only as non-shared disks (system storage) in VMs as Serviceguard Nodes configurations.
- SLVM backing stores cannot be used as shared disks (package storage) in VMs as Serviceguard Nodes configurations.
- No special configuration other than what is described in Section 10.3, is required for these configurations.
- Online migration of VMs is not supported in VMs as Serviceguard Nodes configurations.

10.5.2 Creating and Configuring VMs as Serviceguard Packages Having SLVM Backing Storage

Deploying virtual machines as Serviceguard packages is a common occurrence. Multiple virtual machines can have virtual disks mapped to logical volumes in a single SLVM volume group. In the event that one of those virtual machines is moved to another node (VM Host) in the Serviceguard cluster, the other virtual machines need to have continued access to logical volumes in that volume group. To facilitate this behavior, configure Serviceguard multi-node packages for the SLVM volume groups by following these steps:

1. Create a Serviceguard multi-node package configuration file for each SLVM volume group:

   ```
   # mkdir /etc/cmcluster/hpvm_vgsharedA
   # cd /etc/cmcluster/hpvm_vgsharedA
   # ls
   # cmmakepkg -m sg/multi_node -m sg/volume_group > hpvm_vgsharedA.conf
   Package template is created.
   This file must be edited before it can be used.

   #
   ```

2. Edit the configuration file, making the following modifications:
   a. Provide the package with a name (for example, hpvm_vgshareA).
   b. Change the `vgname_cmd` to `vgchange -a s`.
   c. Add the vg name for the SLVM volume group. For example, if vgsharedA is the SLVM volume group being used by the virtual machines on this VM Host, add the following line:

      vg vgsharedA

   d. If only some nodes in the cluster have the SLVM volume groups, then identify teach of those nodes with the `node_name` attribute. If all nodes can run the SLVMs, then leave the attribute's value as asterisk (*).
   e. Verify the configuration and, if it is valid, apply the package configuration to the cluster:

      ```
      # cmcheckconf -P /etc/cmcluster/hpvm_vgsharedA/hpvm_vgsharedA.conf
      Attempting to validate hpvm_vgsharedA.
      ...
      cmcheckconf: Verification completed with no errors found.
      Use the cmapplyconf command to apply the configuration.
      # cmapplyconf -P /etc/cmcluster/hpvm_vgsharedA/hpvm_vgsharedA.conf
      Attempting to validate hpvm_vgsharedA.
      ...
      Modify the package configuration ([y]/n)? y
      ```

```
Completed the cluster update
#
```

3. Create the package for the VM using `/opt/cmcluster/toolkits/hpvm/hpvmsg_pkg` and, for this example, assume that the configuration is in /etc/cmcluster/slvmtest/slvmtest.conf. Associate this package with the multi-node package created with a SLVM volume group as follows:
   a. Modify the package to specify its independence on the multi-node package for the SLVM volume group (created for the SLVM volume group that the virtual machine uses for storage):

      ```
      # cd /etc/cmcluster/slvmtest/
      # cmmakepkg -i ./slvmtest.conf -m sg/dependency > slvmtest2.conf
      #
      ```

   b. Modify the dependency characteristics for the VMs package configuration, so that it is dependent on the SLVM VG package being available:

      ```
       dependency_name                        hpvm_vgsharedA
      dependency_condition                   hpvm_vgsharedA = up
      dependency_location                    same_node
      ```

   c. Check the configuration of the package:

      ```
      # cmcheckconf -P /etc/cmcluster/slvmtest/slvmtest2.conf
      Attempting to validate slvmtest.
      ...
      cmcheckconf: Verification completed with no errors found.
      ```

   d. Save the original configuration file and use the modified configuration in its place to apply the package configuration to the cluster:

      ```
      # cp slvmtest.conf slvmtest.conf.orig
      # mv slvmtest2.conf slvmtest.conf
      # cmapplyconf -P /etc/cmcluster/slvmtest/slvmtest.conf
      Attempting to validate slvmtest.
      ...
      Modify the package configuration ([y]/n)? y
      Completed the cluster update
      ```

   e. Run the package associated with the SLVM and virtual machine independent order:

      ```
      # cmrunpkg hpvm_vgsharedA
      Running package hpvm_sharedA on node ghost1
      Successfully started package hpvm_sharedA on node ghost1
      Running package hpvm_sharedA on node ghost2
      Successfully started package hpvm_sharedA on node ghost2
      cmrunpkg: All specified packages are running
      # cmrunpkg slvmtest
      Running package slvmtest on node ghost1
      Successfully started package slvmtest on node ghost1
      cmrunpkg: All specified packages are running
      ```

NOTE: If the package associated with the SLVM is not run prior to starting the package associated with the virtual machine, then a message similar to the following is displayed:

```
# cmrunpkg slvmtest
Unable to execute command. Dependency on the following packages not met:
hpvm_vgsharedA
cmrunpkg: Unable to start some package or package instances
```

10.5.3 Online Migration of VMs with SLVM Virtual Disks in VM as Serviceguard Package Configurations Requires Manual Reconfiguration

Successful online migration of VMs with SLVM virtual disks in VMs as Serviceguard Package configurations requires you to configure each SLVM as an independent multinode package. Additionally, the VM as Serviceguard Package must be modified to indicate a new package dependency that is the package name associated with the SLVM package.

If the SLVM created and associated with VM is created and associated with the package, perform the following steps:

1. Create and apply to the cluster an SLVM package for each using multinode and volume_group modules.
2. Edit the VM package configuration:
   a. Remove all references to the volume data reference for each SLVM.
   b. Add a dependency for each SLVM package associated with the VM.
   c. Check and apply the VM package configuration.
3. Ensure that all SLVM packages are running on all nodes.
4. Ensure that the VM package is running on the desired node.
5. Verify failover and online migration function as expected prior to placing systems into production.

10.6 Troubleshooting Network Problems When Using Serviceguard

If the guest has network problems after failover:

- Make sure the vswitches are configured on the adoptive node. If you are using the VLAN feature of Integrity VM vswitches, make sure that appropriate VLAN IDs are assigned to each port.
- Adjust the values of the following Serviceguard parameters in the cluster configuration file. The correct settings for the `HEARTBEAT_INTERVAL` and the `NODE_TIMEOUT` parameter are system- and load-dependent.
  - The `HEARTBEAT_INTERVAL` parameter specifies the normal interval between the transmission of heartbeat messages from one node to the other in the cluster. The value of the `HEARTBEAT_INTERVAL` parameter is entered in microseconds; the default value is 1,000,000 microseconds. Setting the value of this parameter to less than the default is not recommended. The default should be used where possible. The maximum value recommended is 15 seconds, and the maximum value supported is 30 seconds. This value should be at least half the value of the `NODE_TIMEOUT` parameter.
  - The `NODE_TIMEOUT` parameter specifies the amount of time after which the Serviceguard node can determine that the other node is unavailable and can initiate cluster reformation. This parameter is entered in microseconds; the default value is 2,000,000.The minimum is two times the value of the `HEARTBEAT_INTERVAL` parameter. The maximum recommended value for this parameter is 30,000,000. The default setting yields the fastest cluster reformations. However, using the default value increases the potential for spurious reformations because of momentary system hangs or network-load spikes. For

many installations, a setting of 5,000,000 to 8,000,000 (5 to 8 seconds) is more appropriate. The maximum value recommended is 30 seconds and the maximum value supported is 60 seconds.

11 Reporting Problems with Integrity VM

Report 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 VM Host and Integrity VM to best match the environment when the VM Host 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 VM Host, reboot the guest that caused the VM Host to crash.
3. Run the `hpvmcollect` command on the guest again. Include this information in the `hpvmcollect` archive from the VM Host.
4. Report the information through your support channel.

This chapter describes how to use the `hpvmcollect` command and how to investigate Integrity VM log files for information, including the following topics:

- "Collecting Integrity VM Data"
- "Managing the Size of the VMM Driver Log File"

11.1 Collecting Integrity VM Data

You can use the `hpvmcollect` command on the VM Host or on the guest to collect Integrity VM information that is useful in analyzing system problems. The options available for the `hpvmcollect` command on the VM Host are different from those available on guests. For information about using the `hpvmcollect` command, see one of the following sections:

- Using the `hpvmcollect` command the VM Host: see Section 11.1.1 (page 227).
- Using the `hpvmcollect` command on guests: see Section 11.1.2 (page 230).

11.1.1 Using the hpvmcollect Command on the VM Host

Table 32 describes the options to the `hpvmcollect` command on the VM Host:

Table 32 Options to the `hpvmcollect` Command on the VM Host

Option	Description
`-P` *`vm-name`*	Specifies the virtual machine name, where *`vm-name`* is the name of the virtual machine.
`-p` *`vm-number`*	Specifies the virtual machine number, where *`vm-number`* is the number of the virtual machine.
`-s` *`host`*	Specifies a host name to receive the archive, which is copied using the `scp` command. Verify that you can log in to the host 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 32 Options to the `hpvmcollect` Command on the VM Host *(continued)*

Option	Description
`-c`	Includes the latest crash dump directory in the archive. This option is used if the guest or the VM Host 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 guests on the VM Host for inclusion in the collection. Valid only on the VM Host.
`-r`*`directory`*	Specifies a remote target directory in which to store the collected archive, overriding the default of/crashes.Valid on both the VM Host and the guest. The `-r` option is valid only with the `-s` option.

If the VM Host hangs, generate a crash dump using the `TC` command on the VM Host console. When the VM Host 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 Integrity VM problems.

The `hpvmcollect` command is a shell script that can be run on either the VM Host or the guest to gather system information, log files, Integrity VM logs, and configuration files for later analysis.

Because the `hpvmcollect` command collects generic Integrity VM and HP-UX operating system and system information, it may 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 guest 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 Integrity VM and VM Host information. For example, to gather information for a guest named `host1` on the VM Host, 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 VM Host 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 VM Host 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 VM Host:

```
# hpvmcollect -P host1
```

```
HPVM host crash/log collection tool version 0.8
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_Sep.28.06_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 VM Host) as well as VM Host 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
```

11.1.2 Using the `hpvmcollect` Command on Guests

To use the `hpvmcollect` command on the guest, you must first install the guest management software on the guest as described in Section 8.6 (page 156).

Table 33 describes the options to the `hpvmcollect` command on the guest.

Table 33 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 guest or the VM Host fails or hangs.
`-f`	Forces an archive to be overwritten, if it exists, rather than renamed with an appended time stamp.
`-g`	Deletes old guest 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 host name to receive the archive, which is copied using the `scp` command. Verify that you can log in to the host without a password.

When you use the `hpvmcollect` command on the guest, do not specify the guest name. By default, the guest 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 guest `host1`:

```
host1# hpvmcollect -c
HPVM guest crash/log collection tool version 0.8
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"
```

11.2 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.
```

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 VM Host. Because this is not always possible for all users the following method should work.

The VM Host 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/backups` 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.04.30.
2. Stop Integrity VM (`/sbin/init.d/hpvm stop`)
3. Remove Integrity VM V4.3 software (This causes a system reboot.).

   ```
   # swremove -x autoreboot=true T2767CC T8718AC VMGuestLib VMGuestSW VMKernelSW vmProvider
   ```

4. Move the `/var/opt/hpvm` area aside:

   ```
   # mv /var/opt/hpvm /var/opt/hpvm_4.3
   ```

5. Install the previously installed version of Integrity VM following the directions for installing Integrity VM in the *HP Integrity Virtual Machine Installation, Configuration, and Administration* 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_4.2/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
```

Integrity VM Manpages

hpvm(5)

NAME

hpvm – HP Integrity Virtual Machines (Integrity VM).

SYNOPSIS

`Virtualization technology`

DESCRIPTION

HP Integrity Virtual Machines allows the creation and management of virtual machines, in which unmodified operating systems designed for HP Integrity servers can run. Integrity VM provides a **VM Host**, which manages the physical machine and allocates system resources, such as memory, CPU time, and I/O devices to virtual machines. The VM Host is the HP-UX operating system installed on the physical machine and running the Integrity Virtual Machines product. Virtual machines run on the same physical machine as the VM Host and appear to be ordinary HP-UX processes. Each virtual machine emulates a real Integrity machine, including firmware. A virtual machine is sometimes referred to as a **guest**. The operating system running in a virtual machine is referred to as the **guest operating system**, or **guest OS**.

Following are the Integrity VM commands:

- `hpvmclone`: Creates a cloned copy of a virtual machine. Runs on the VM Host.
- `hpvmcollect`: Collects crash dumps, logs, system status, and configuration on host and guest. Runs on the VM Host and on guests.
- `hpvmconsole`: Connects to the console of a virtual machine. Runs on the VM Host.
- `hpvmcreate`: Creates a new virtual machine. Runs on the VM Host.
- `hpvmdevinfo`: Reports about storage for a virtual machine.
- `hpvmdevmgmt`: Manages the device database. Runs on the VM Host.
- `hpvmdevtranslate`: Translates Integrity VM guests to agile devices.
- `hpvmhostgdev`: Manages Integrity VM Host devices available for virtual machine access.
- `hpvmhostrdev`: Manages virtual machine access to devices used by the Integrity VM Host system.
- `hpvminfo`: Displays information about the Integrity VM environment. Runs on the VM Host and on guests.
- `hpvmmgmt`: Manages the guest memory allocation. Runs on guests only.
- `hpvmmigrate`: Moves a virtual machine from one VM Host to another. Runs on the VM Host.
- `hpvmmodify`: Renames or modifies the attributes of a virtual machine. Runs on the VM Host.
- `hpvmmove_suspend`: Moves suspend files to a different directory.
- `hpvmnet`: Configures virtual network devices. Runs on the VM Host.
- `hpvmnvram`: Displays, creates, edits and removes EFI variables in NVRAM files from a VM Host.
- `hpvmpubapi`: Provides the Integrity VM public application interface descriptions.
- `hpvmremove`: Removes a virtual machine. Runs on the VM Host.
- `hpvmresume`: Manages Integrity VM Host devices available for virtual machine access.
- `hpvmsar`: Displays performance information about one or several guests on the same host.
- `hpvmsg_move`: Initiates an online migration (move) of a virtual machine that has been associated with a Serviceguard package.

- `hpvmsg_package`: Assists the user with developing and managing the Serviceguard package configuration.
- `hpvmstart`: Starts a virtual machine. Runs on the VM Host.
- `hpvmstatus`: Displays status of one or more virtual machines. Runs on the VM Host.
- `hpvmsuspend`: Suspends a virtual machine.
- `hpvmstop`: Stops a virtual machine. Runs on the VM Host.
- `hpvmupgrade`: Assists an Integrity VM upgrade.
- `p2vassist`: Moves a system workload from a discreet server to a virtual machine. Runs on the VM Host.

All commands except `hpvmconsole` require superuser privileges.

To use Integrity VM commands on the guest, install the guests management software as described in *HP Integrity Virtual Machines Installation, Configuration, and Administration*.

AUTHORS

HP Integrity Virtual Machines was developed by HP.

SEE ALSO

On the VM Host:

hpvmclone(1M), *hpvmcollect*(1M), *hpvmconsole*(1M), *hpvmcreate*(1M), *hpvmdevinfo*(1M),*hpvmdevmgmt*(1M), *hpvmdevtranslate*(1M),*hpvmhostgdev*(1M), *hpvmhostrdev*(1M), *hpvminfo*(1M), *hpvmmigrate*(1M), *hpvmmodify*(1M), *hpvmmove_suspend*(1M), *hpvmnet*(1M), *hpvmnvram*(1M), *hpvmpubapi*(3), *hpvmremove*(1M), *hpvmresources*(5), *hpvmresume*(1M), *hpvmsg_move*(1M), *hpvmsg_package*(1M), *hpvmstart*(1M), *hpvmsar*(1M)*hpvmstatus*(1M), *hpvmstop*(1M), *hpvmsuspend*(1M), *hpvmupgrade*(1M)*p2vassist*(1M)

On the Integrity VM guest:

hpvmcollect(1M), *hpvminfo*(1M), *hpvmmgmt*(1M), *hpvmpubapi*(3)

hpvmclone(1M)

NAME

hpvmclone -- Create a new virtual machine that is a copy of an existing virtual machine.

SYNOPSIS

hpvmclone { -P *vm_name* | -p *vm_number* } [-K *console_IP_Addr*] [-L *console_IP_Addr_Netmask*] -N *clone_VM_name* [-F | -s] [-l *vm_label*] [-B *start_attr*] [-O *os-type* [:*version*]] [-c *number_vcpus*] [-e *percent[:max_percent]* | -E *cycles[:max_cycles]*] [-r *amount*] [-S] [-g *user*] [-g [+]*group*[: {*admin* |*oper*}]] [-u *user*] [-u [+]*user*[: {*admin*| *oper*}]] [-a *rsrc*]... [-m *rsrc*]... [-d *rsrc*]... [[-x *name=value* [:*name=value*]]]

DESCRIPTION

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. The `clone_vm_name` must not already exist on this host.

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. *Note that resource conflicts could occur if both the original and clone virtual machines are booted together.*

Resources are checked to determine whether the virtual machine can boot by itself on the server. Any problems are reported as WARNINGS. These warnings do not prevent the new virtual machine from being created.

NOTE: Cloning is not allowed during Online VM Migration.

Only superusers can execute the `hpvmclone` command.

Options

To print the warnings without creating a new virtual machine, use the `-s` option.

Because there is no guarantee that other virtual machines are running at the same time as the new virtual machine is running, use the following command to verify whether a device has dependents:

`hpvmdevmgmt -l` *`gdev:entry_name`*

where *`entry_name`* is the device name in the device-management database.

If you omit an option, the associated attribute remains unchanged.

`-P` *vm_name*	Specifies the name of the existing virtual machine to be cloned. You must specify either the `-P` or the `-p` option.
`-p` *vm_number*	Specifies the number of the existing virtual machine to be cloned. You can obtain the value of vm_number using the `hpvmstatus` command. You must specify either the `-P` 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. If an address is supplied, then the option must also be specified. The virtual iLO Remote Console will be accessible using telnet or ssh using this IP address. Accessing the virtual

iLO Remote Console will automatically run the command `hpvmconsole` for this guest.

To facilitate virtual iLO Remote Console access, an IP alias interface will be created using this IP address and the netmask specified in the option. If the IP address specified is 0, the IP alias interface for this guest will be removed. If this option is not specified, then the new guest being created will have no virtual iLO Remote Console set up. The virtual iLO Remote Console configuration for the guest from which this new guest is cloned is left untouched.

`-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 to be assigned to the new virtual machine. The name can be composed of up to 256 alphanumeric characters, including A-Z, a-z, 0-9, the dash (—), the underscore (_), and the period (.).

The virtual machine name cannot start with a dash (—). You must specify the `-N` option.

`-e` *`percent[:max_percent]`*
: Specifies the percentage of CPU resources to which each of the new guest's virtual CPUs is entitled. If the entitlement is not specified with this option or the `-E` option, the new virtual machine's entitlement is that of the existing virtual machine.

The percentage can be set to an integral value between 0 and 100. If the value specified is less than 5, then the virtual machine is allocated the minimum percentage of 5%. The default is 10%.

The maximum entitlement cannot exceed 100 percent and cannot be less than the minimum. Each group has a default maximum setting of 100 percent.

The entitled CPU resources inherited from the existing virtual machine, specified in cycles or percentages, are replaced in the new virtual machine by this percentage.

The `-e` and the `-E` options are mutually exclusive.

`-E` *`cycles[:max_cycles]`*
: Specifies the virtual machine's CPU entitlement in number of CPU clock cycles. If the cycles are not specified with this option and the `-e` option is not specified, the new virtual machine's entitled CPU resources is that of the existing virtual machine.

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 value of entitlement inherited from the existing virtual machine (specified in either cycles or percentages) is replaced in the new virtual machine by the new value in CPU clock cycles.

The `-e` and the `-E` options are mutually exclusive.

`-l` *vm_label*

Specifies a descriptive text string for the new virtual machine. This option can be useful in identifying a specific virtual machine in the `hpvmstatus -V` display. The label can be 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 (" ").

`-B` *start_attr*

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. (This is the default.)

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

If the attribute is set to `manual`, the virtual machine does not start automatically when Integrity VM is initialized on the VM Host. This is the default. 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.

`-O` *os_type*[*:version*]

Specifies the type and version of the operating system running on the virtual machine. The response affects the default selection of certain virtual machine attributes, such as amount of memory and CPU power. The *os_type* is one of the following: `HPUX` or`OPENVMS`. This parameter is not case sensitive. The *version* is specific to the operating system type.

The version specifies a descriptive text string of the version of the operating system. The version string can consist of up to 256 alphanumeric characters, including A-Z, a-z, 0-9, the dash (—), the underscore r (_), and the period (.). To specify white space, then “version” must be quoted.

NOTE: You can specify the `os_type` here manually, but when the guest boots, the `os_type` is set to whatever operating system was last booted. The operating system *`version`* is only set manually with the `-O` option.

`-a` *`rsrc`*

Adds an I/O resource to the new virtual machine. The resource specification (*`rsrc`*) is described in *hpvmresources*(5).

This option can be specified more than once.

`-d` *`rsrc`*

Deletes an I/O resource from the new virtual machine. The resource specification (*`rsrc`*) is described in *hpvmresources*(5).

This option can be specified more than once.

`-m` *`rsrc`*

Modifies an I/O resource on the cloned virtual machine.

This option can be specified more than once.

Integrity VM recognizes the following types of guest virtual devices:

- Virtual disks, which can be backed by files in a VM Host file system, by logical volumes or by whole disks.
- Virtual DVDs, which can be backed by files in a VM Host file system or by the physical DVD drive.
- Virtual network switches (vswitches), which are created using the `hpmvnet` command and backed by physical LAN cards. For more information about vswitches, see the *hpvmnet*(1M) manpage.

For information about specifying storage and network resources for guests, see *hpvmresources*(5).

`-F`

Ignores all virtual machine configuration warnings, including oversubscribing of resources (Force mode).

This option is primarily intended for use by scripts and other noninteractive applications.

The `-F` option is deprecated in Integrity Virtual Machines commands.

`-c` *`number_vcpus`*

Specifies the number of virtual CPUs visible to the new virtual machine. If unspecified, the number defaults to that of the existing virtual machine. The maximum number of vCPUs supported per guest is 16.

`-r` *`amount`*

Specifies the amount of memory available to the new virtual machine at boot time.

The sizes are expressed as integers, optionally followed by one of these units:

- M (megabytes)
- G (gigabytes)

If the letter is left off, the unit type defaults to megabytes. If the `-r` option is omitted, the amount of memory is that of the existing virtual machine.

`-S`	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.
`[-g [+]`*`group`*` [ : {admin| oper}]]`	Specifies group authorization. Use the `-g [+]`*`group`*`:{admin | oper}` syntax to add a group, where + is optional. When adding a group authorization, the default authorization type is `oper`. To remove a group authorization, specify the `-g -`*`group`* syntax. This option can be specified more than once.
`[-u [+]`*`username`*`[ :{admin | oper}]]`	Specifies user authorization. Remove user authorization by using the `-u `*`user`* syntax. The virtual machine user account specified here can use the `hpvmconsole` command to manage the virtual machine. Add user authorization using the `-u [+]`*`user`*`{:admin |oper }` syntax, where + is optional. When adding a group authorization, the default authorization type is `oper`.
`-s`	Verifies the new virtual machine configuration and returns warnings or errors but does not create the virtual machine.
`-x`	Specifies whether the new virtual machine uses dynamic memory and the values associated with it by including the following keywords: • `-x dynamic_memory_control={0|1}` • `-x ram_dyn_type={none|any|driver}` • `-x ram_dyn_min=`*`amount`* • `-x ram_dyn_max=`*`amount`* • `-x ram_dyn_target_start=`*`amount`* Specifies whether the virtual machine's dynamic memory settings are automatically adjusted. The ram_dyn_entitlement and amr_enable options must be set to enable adjustments. • `-x ram_dyn_entitlement=`*`amount`* Specifies the minimum guaranteed amount of memory. • `-x amr_enable={0|1}` Enables or disables AMR monitoring for a guest, where 1 enables and 0 disables. The monitor (amr daemon) adjusts the guest size and take its entitlement into account. • `- x amr_chunk_size=`*`amount`* Specifies the increment amount of changes in memory size (default is 256 MB). Larger values result in faster memory size growth. • `-x mac_address={`*`new`*`/`*`same`*`}`

- `-x serial_number={`*`new|same`*`}`
- `-x sched_preference={`*`none|cell|ilm`*`}`

  where:

  - *`none`* — The default preference. If your application is predominantly CPU bound, specifying either *`ilm`* or *`cell`* will perform the same.
  - *`cell`* — The cell with the most CPU and memory space is chosen. When that guest is active, the scheduler then optimizes where the guest runs, so that it can be closest to its memory.
  - *`ilm`* — Indicates that guests that are larger than any single cell and contain highly threaded applications.
- `-x graceful_stop_timeout={0|`*`amount`*`}`

Used for Online VM Migration:

- `-x migrate_init_phase_timeout={`*`number of seconds`*`}` — Specifies the maximum number of seconds the online migration spends during the initialize phase of the migration. The default is 10 seconds.
- `–x migrate_copy_phase_timeout= {`*`number of seconds`*`}` — Specifies the maximum number of seconds the online migration spends during the full-copy stage. The default is 0 seconds.
- `-x migrate_io_quiesce_phase_timeout= {`*`number of seconds`*`}` — Specifies the maximum number of seconds the migration spends during the quiesce stage. The default is 15 seconds.
- `-x migrate_frozen_phase_timeout= {`*`number of seconds`*`}` — Specifies the maximum number of seconds the migration spends during the freezing stage. The default is 60 seconds.
- `-x online_migration= {`*`enabled | disabled`*`}`
- `-x tunables= {`*`name=value`*` [, `*`name=value`*`,...]}`

To specify the MAC address of the new virtual machine, enter the following: `-x mac_address={new|same}`

To specify the serial number of the new virtual machine, enter the following: `-x serial_number={new|same}`

RETURN VALUES

The `hpvmclone` command exits with one of the following values:

0: Successful completion.

1: One or more error conditions occurred.

DIAGNOSTICS

The `hpvmclone` command displays error messages on stderr for any of the following conditions:

- An invalid option is specified.
- An invalid value is specified for an option.
- A value was omitted for an argument that requires one, or a value was supplied for an argument that does not take one.
- One or more options other than `-a`, `-m`, `-d`, `-g`, or `-u` have been specified more than once.
- The `clone_vm_name` attribute already exists.
- The `vm_name` or `vm_number` attribute does not exist, cannot be accessed, is not a virtual machine, or is corrupt.
- The `hpvmclone` command and Integrity VM are at different revision levels.
- The same resource was allocated more than once.
- A resource allocated to another virtual machine was specified, and the force flag (`-F`) was not used.

EXAMPLES

Clone the virtual machine named host2, to create a new virtual machine named host5.

```
# hpvmclone -P host2 -N host5
```

Following are sample warning messages returned when `hpvmclone` is executed with various configuration problems on the guest `host5`:

```
HPVM guest host5 configuration problems:
Warning 1: Guest needs more vcpus than server supports.
Warning 2: Insufficient free memory for guest.
Warning 3: Insufficient swap resource for guest.
Warning 4: Insufficient cpu resource for guest.
Warning 5 on item /dev/rdisk/disk0: Device file '/dev/rdsk/disk0' in use by another guest.
Warning 6 on item /dev/vg00/rswap: Device file '/dev/vg00/rswap' in use by server.
Warning 7 on item /dev/rdisk/disk3 backing device does not exist.
Warning 8 on item /dev/rdisk/disk4: Device file '/dev/rdisk/disk4' in use by another guest.
Warning 9 on item hostnet: MAC address in use for switch hostnet.
Warning 10 on item offnet: Vswitch offnet is not active.
Warning 11 on item badnet: 'badnet' backing device does not exist.

These problems will prevent HPVM guest host5 from booting.
```

The following example shows how to use the `hpvmclone` command to create a guest named vmclone1 that uses the same ports as the existing guest (vm1). The `hpvmnet` command shows that two guests are sharing ports 1 and 2 on the virtual switch vmlan4. Only the active virtual machine (vm1) can use the port.

```
# hpvmclone -P vm1 -N vmclone1 -S
  # 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         Port    Untagged   Number of      Active VM
  Number  state        Adaptor  VLANID     Reserved VMs
  ======= ============ ========  ============  ============
  1       Active        lan   none       2             vm1
  2       Active        lan   100        2             vm1
  3       Active        avio_lan  none       1             vm2
  4       Active       lan   100        1             vm2
```

OBSOLESCENCE

The -F option is deprecated in Integrity Virtual Machines commands. This flag might be removed in a future release of Integrity Virtual Machines. HP recommends use of the command with the -s option, then resolving any errors or warnings that result before executing the command without the -s option and without the -F option.

AUTHORS

The hpvmclone command was developed by HP.

SEE ALSO

On the VM Host:

hpvm(5), *hpvmcollect*(1M), *hpvmconsole*(1M), *hpvmcreate*(1M),*hpvmdevinfo*(1M), *hpvmdevmgmt*(1M), *hpvmdevtranslate*(1M), *hpvmhostgdev*(1M), *hpvmhostrdev*(1M), *hpvminfo*(1M), *hpvmmigrate*(1M), *hpvmmodify*(1M), *hpvmmove_suspend*(1M), *hpvmnet*(1M), *hpvmnvram*(1M), *hpvmpubapi*(3), *hpvmremove*(1M), *hpvmresources*(5), *hpvmresume*(1M), *hpvmsar*(1M), *hpvmsg_move*(1M), *hpvmsg_package*(1M), *hpvmstart*(1M), *hpvmstatus*(1M), *hpvmstop*(1M), *hpvmsuspend*(1M), *hpvmupgrade*(1M), *p2vassist*(1M)

On the Integrity VM guest:

hpvmcollect(1M), *hpvminfo*(1M), *hpvmmgmt*(1M), *hpvmpubapi*(3)

hpvmcollect(1M)

NAME

hpvmcollect -- Collects crash dumps, logs, system status, and configuration on the VM Host and guests for post-mortem analysis.

SYNOPSIS

On the VM Host:

`hpvmcollect` { `-P` *vm-name* | `-p` *vm-number* | `-a` } [`-f`] [`-g`] [`-h`] [`-b` *bug-report-number*] [`-d` *directory*] [`-c [-n #]`] [`-l` | `-s` *hostname* [`-r` *directory*]]

On the Guest:

`hpvmcollect` [`-f`] [`-h`] [`-b` *bug-report-number*] [`-d` *directory*] [`-c [-n #]`] [`-l` | `-s` *hostname* [`-r` *directory*]]

DESCRIPTION

The `hpvmcollect` command collects log files, system status, device information, system and Integrity Virtual Machines configuration, guest information, and crash dumps.

When run on a VM Host, this command collects systemwide information as well as information for a specified guest. In this case, you can specify one or more guests using virtual machine names or virtual machine numbers or specify all guests on the VM Host using the `-a` option.

When run in a guest, the `hpvmcollect` command collects information associated only with the guest.

The `hpvmcollect` command creates a directory and produces a `tar` archive or a compressed `tar` archive containing the collected information and places it in your current directory. By default, the archive name is constructed by appending a timestamp to the guest name.

Only superusers can execute the `hpvmcollect` command.

Options

The following options can be specified only once.

The `hpvmcollect` command recognizes the following command-line options and arguments:

`-P` *vm_name*
: Specifies the unique name of the virtual machine to be archived. Valid on the VM Host only.

  The `-P` and `-p` options can be mixed and repeated, but they are mutually exclusive with the `-a` option.

`-p` *vm_number*
: Specifies the unique number of the virtual machine to be archived. The `vm_number` is displayed by the `hpvmstatus` command. Valid on the VM Host only.

  The `-P` and `-p` options can be mixed and repeated, but they are mutually exclusive with the `-a` option.

`-a`
: Selects all guests on the VM Host for inclusion in the collection.

  Valid only on the VM Host.

`-b` *bug-report-number*
: Overrides the default archive name with *bug-report-number* plus the specified label. If an archive with the same name exists, it is renamed by appending a timestamp to the original name before the new archive is created. Valid on both the VM Host and the guest.

`-c`	Includes the latest crash dump directory in the archive. This option is used if the guest or the VM Host crashes or hangs. Valid on both the VM Host and the guest.
`-d` *directory*	Specifies a target directory in which to create the `hpvmcollect_archive` directory. Valid on both the VM Host and the guest.
`-f`	Forces an archive to be overwritten, if it exists, rather than renamed with an appended timestamp. Valid on both the VM Host and the guest.
`-g`	Cleans up a guest debug memory dump directory after a memory dump has been collected into the `hpvmcollect` archive. Valid on the VM Host only.
`-n #`	Specifies the number of crash dumps to copy to the archive. Valid on both the VM Host and the guest. 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.
`-l`	Leaves the collected information in the directory rather than an archiving it. Valid on both the VM Host and the guest. The `-l` option and the `-s` option are mutually exclusive.
`-s` *hostname*	Specifies a host name to receive the archive, which is copied using `scp`. Verify that you can login to the host without a password. Valid on both the VM Host and the guest. Note that the `-l` option and the `-s` option are mutually exclusive.
`-h`	Displays the help message. Valid on both the VM Host and the guest.
`-r` *directory*	Specifies a remote target directory in which to store the collected archive, overriding the default of /crashes. Valid on both the VM Host and the guest. The `-r` option is valid only with the `-s` option.

RETURN VALUES

The `hpvmcollect` command exits with one of the following values:

0: Successful completion.

1: One or more error conditions occurred.

DIAGNOSTICS

The `hpvmcollect` command displays the status of each collection line by line:

- OK: The item collection was successful.

  NO: The option was not used to collect the item.

  N/A: `hpvmcollect` was supposed to collect the item but failed. Possible reasons include:

  The command is not available (for example, it may not be in `$PATH`).

  The command exited with an error; thus, there was no collection.

  The condition that triggers the log file generation did not occur.

EXAMPLES

On a VM Host, collect VM Host and guest `myguest` information:

```
# hpvmcollect -P myguest
HPVM host crash/log collection tool version 3.0
  Gathering info for post-mortem analysis of guest 'myguest' on host
```

```
  Copying host's device database ..................................... N/A
  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 ............................................... OK
  Collecting passthru disk info ...................................... N/A
  Collecting file backing store info ................................. OK
  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 ....................................... N/A
  Copying VMM image .................................................. OK
  Copying hpvmdvr image .............................................. OK
  Copying hpvmntdvr image ............................................ OK
  Copying NVRAM image ................................................ OK
  Collecting IPMI logs ............................................... OK
  Copying guest debug memory file .................................... N/A
  Garbage collect guest debug memory files ........................... N/A
  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
  "/hpvmcollect_archive/myguest_Dec.08.06_1739EST.tar.gz"
```

On the VM Host, include crash dump 23 and write the archive directory in `/tmp`:

```
# hpvmcollect -d /tmp -c -n 23 -P myguest
```

On the VM Host, leave collected information in an archive directory rather than creating the `tar` archive:

```
# hpvmcollect -P myguest -l
```

On the guest, collect guest information along with the latest guest crash dump:

```
# hpvmcollect -c
HPVM guest crash/log collection tool version 3.0
  Gathering info for post-mortem analysis on guest 'vm6'(hostname 'hpux06')

  Copying host's device database ..................................... N/A
  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 ................................................. NO
Running crashinfo ..................................................... NO
Collecting tombstone .................................................. NO
Collecting system message buffer ...................................... OK
Collecting system syslogs ............................................. OK
Collecting measureware logs ........................................... N/A

Finished with the collection

Tar archiving and compressing ......................................... TGZ
Remote copying the archive  ........................................... NO

The collection is
"//hpvmcollect_archive/hpux06_Jan.10.07_182804EST.tar.gz"
```

AUTHORS

The `hpvmcollect` command was developed by HP.

SEE ALSO

On the VM Host:

hpvm(5), *hpvmclone*(1M), *hpvmconsole*(1M), *hpvmcreate*(1M), *hpvmdevinfo*(1M), *hpvmdevmgmt*(1M), *hpvmdevtranslate*(1M), *hpvmhostgdev*(1M), *hpvmhostrdev*(1M), *hpvminfo*(1M), *hpvmmigrate*(1M), *hpvmmodify*(1M), *hpvmmove_suspend*(1M), *hpvmnet*(1M), *hpvmnvram*(1M), *hpvmpubapi*(3), *hpvmremove*(1M), *hpvmresources*(5), *hpvmresume*(1M), *hpvmsar*(1M), *hpvmsg_move*(1M), *hpvmsg_package*(1M), *hpvmstart*(1M), *hpvmstatus*(1M), *hpvmstop*(1M), *hpvmsuspend*(1M), *hpvmupgrade*(1M), *p2vassist*(1M)

On the Integrity VM guest:

hpvmmgmt(1M), *hpvminfo*(1M), *hpvmpubapi*(3)

hpvmconsole(1M)

NAME

hpvmconsole -- Connect to the console of a virtual machine.

SYNOPSIS

`hpvmconsole` { `-P` *vm-name* | `-p` *vm-number* } [`-c` *command*] [`-e` *echar*] [`-f`] [`-i`] [`-q`]

DESCRIPTION

An Integrity VM virtual machine console is similar in appearance to the maintenance processor of an Integrity system. Each virtual machine has its own virtual console from which you can manage the virtual machine and the guest operating system. The `hpvmconsole` command connects to the virtual console of a specified virtual machine.

If you have logged into the physical console of a VM Host and then run `hpvmconsole` interactively.

To return to the physical console, press **Ctrl/B**.

To return to the virtual console main menu, press **Ctrl/X**.

Options

The `hpvmconsole` command recognizes the following standard Integrity VM options and attributes:

`-P` *vm-name* — Specifies the name of the virtual machine to be booted.

You can specify either the `-P` or the `-p` option, but not both.

`-p` *vm-number* — Specifies the number of the virtual machine to be booted. The *vm-number* is displayed by the `hpvmstatus` command.

You can specify either the `-P` or the `-p` option, but not both.

`-c` *command* — Provides a console command to be performed before reading from standard input. The `-c` option is provided for scripting and logging purposes. You can enter multiple `-c` options; they are processed from left to right. In this mode, you cannot use **Ctrl/B** to get back to command mode. This mode is primarily useful in combination with the `-f` option to enter console mode and watch the OS console output. Even so, the console commands so given will assume a trailing `-nc` option, if they support one, to prevent the reading of standard input unless the `-i` option is also specified.

`-e` *echar* — Overrides the standard **Ctrl/B** escape (or attention) character. The character can be given as a literal control character, or as a caret (^) followed by another character.

`-f` — Continues following the console output after reaching EOF on standard input. (This option exists for scripting and logging purposes.)

`-i` — Interacts with the console (reads from standard input), despite the use of the `-c` and `-f` options.

`-q` — Makes scripted operations less verbose.

RETURN VALUES

The `hpvmconsole` command exits with one of the following values:

0: Successful program execution.

1: Invalid option or invalid argument
to an option (usage error).

2: All other program failures
(operational error).

DIAGNOSTICS

The `hpvmconsole` command displays error messages on stderr for any of the following conditions:

- An invalid option is specified.
- The `hpvmconsole` command and Integrity VM are at different revision levels.
- An operational error occurred.

EXAMPLES

To use the console interactively:

```
# hpvmconsole -p guestname
```

To collect the guest console log in the correct order:

```
# hpvmconsole -P "$GUEST" -q -c cl > $GUEST.conslog
```

Similarly, to collect the guest operation log:

```
# hpvmconsole -P "$GUEST" -q -c 'rec -view' > $GUEST.applog
```

To override the default attention character (**Ctrl/B**) and use **Ctrl/t** instead:

```
# hpvmconsole -e ^t -P guestname
```

AUTHORS

The `hpvmconsole` command was developed by HP.

SEE ALSO

On the VM Host:

hpvm(5), *hpvmclone*(1M), *hpvmcollect*(1M), *hpvmcreate*(1M), *hpvmdevinfo*(1M), *hpvmdevmgmt*(1M), *hpvmdevtranslate*(1M), *hpvmhostgdev*(1M), *hpvmhostrdev*(1M), *hpvminfo*(1M), *hpvmmigrate*(1M), *hpvmmodify*(1M), *hpvmmove_suspend*(1M), *hpvmnet*(1M), *hpvmnvram*(1M), *hpvmpubapi*(3), *hpvmremove*(1M), *hpvmresources*(5), *hpvmresume*(1M), *hpvmsar*(1M), *hpvmsg_move*(1M), *hpvmsg_package*(1M), *hpvmstart*(1M), *hpvmstatus*(1M), *hpvmstop*(1M), *hpvmsuspend*(1M), *hpvmupgrade*(1M), *p2vassist*(1M)

On the Integrity VM guest:

hpvmcollect(1M), *hpvminfo*(1M), *hpvmmgmt*(1M), *hpvmpubapi*(3)

hpvmcreate(1M)

NAME

hpvmcreate – Create a new Integrity VM virtual machine.

SYNOPSIS

`hpvmcreate -P` *vm-name* [`-K` *console_IP_Addr*] [`-L` *console_IP_Addr_Netmask*] [`-F | -s`] [`-l` *vm-label*] [`-B` *start-attr*] [`-O` *os-type* [`:`*version*]] [`-c` *number-vcpus*] [`-e` *percent[:max_percent]* | `-E` *cycles[:max_cycles]*] [`-r` *amount*] [`-g [+]` *group* `[: {`*admin* |*oper*`}]`] [`-u [+]` *user*`[: {`*admin* |*oper*`}]`] [`-a` *rsrc*]... [[`-i SG` | `-i` *SG-pkgname* | `-i GWLM` | `-i` *SG-pkgname,* `GWLM` | `-i NONE`]] [`-j { 0 | 1 }`] [[`-x` *name=value* [`:`*name=value*]]]

DESCRIPTION

The `hpvmcreate` command creates a new virtual machine (a guest) and assigns the specified attributes and resources to it. This command creates an association between the virtual devices seen by the guest and the physical devices managed by the VM Host.

Only superusers can execute the `hpvmcreate` command.

Virtual machine creation is designed for flexibility and assumes that not all created virtual machines are necessarily running at the same time or on the current VM Host. Therefore, the `hpvmcreate` command allows the creation of virtual machines that cannot boot on the current system. A guest configuration receives a warning at creation and an error at start time for any issues that might prevent it from starting on the current VM Host. To verify a particular configuration for the current VM Host without actually creating the guest, use the `-s` option.

Options

`-P` *vm-name*

Specifies the name of the virtual machine. This name must be unique on the VM Host. This virtual machine name is used in other Integrity VM commands to specify which virtual machine the command affects. If you plan to allow remote access to the virtual machine's console, the virtual machine name must be a legal UNIX account name.

The name can consist of up to 256 alphanumeric characters including A-Z, a-z, 0-9, the dash (-), the underscore (_), and the period (.). The virtual machine name cannot start with a dash (-). The `-P` option is required.

`-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. The virtual iLO Remote Console will be accessible using telnet or ssh using this IP address. Accessing the virtual iLO Remote Console will automatically run the command `hpvmconsole` for this guest.

To facilitate virtual iLO Remote Console access, an IP alias interface will be created using this IP address and the netmask specified in the `–L` option. During a migration the virtual iLO Remote Console addressing will follow the migration. Use this option when setting up the IP interface to be used for accessing the virtual iLO Remote Console for this guest.

-L *console_IP_Addr_Netmask* — During a migration the virtual iLO Remote Console addressing will follow the migration.

-e *percent[:max_percent]* — Specifies the percentage of CPU resources to which each of the guest's 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, then the virtual machine is allocated the minimum percentage of 5%. The default entitlement is 10%.

The maximum entitlement may not exceed 100 percent, and may not be less than the minimum. Each group has a default maximum setting of 100 percent. You can view the current settings by using the `hpvmstatus -r` command.

In addition to the guest calculation, Integrity VM reserves processing power for essential system functions like logging, networking, and file system daemons.

The `-e` and `-E` options are mutually exclusive.

-E *cycles[:max_cycles]* — 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 `-E` options are mutually exclusive.

-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.

The `-F` and `-s` options are mutually exclusive.

The `-F` option is deprecated in Integrity Virtual Machines commands.

-a — Specifies the mapping of a guest virtual device to a VM Host backing store. A virtual device is instantiated on physical entities that are managed by the VM Host. These physical entities (for example, network cards, files, logical volumes, and so forth) are collectively referred to as "backing stores."

Integrity VM recognizes the following types of guest virtual devices:

- Virtual DVDs, which can be backed by files in a VM Host file system or by physical DVD drives.
- Virtual disks, which can be backed by files in a VM Host file system, by logical volumes or by whole disks.

- Attached I/O devices (DVD, tape, changer, and other peripheral device types).
- Attached AVIO devices (tape, changer, and burner).
- Virtual network devices, which are created using the `hpvmnet` command and backed by physical LAN cards. See the *hpvmnet* manpage for more information about virtual network devices.

For information about specifying storage and network resources for guests, see *hpvmresources*(5).

`-i` *`package-name`*
: Specifies whether the virtual machine is managed by Serviceguard or gWLM (or both). For the argument, specify the Serviceguard package name, *`GWLM`*, both, or *`NONE`*. This option is used by Integrity VM software; do not use this option without express instruction by HP.

`-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). This option is used by Integrity VM software; do not use this option without express instruction by HP.

`-l` *`vm-label`*
: Specifies a descriptive label for this virtual machine. This 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 ("").

`-B` *`start-attr`*
: 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 occurs when the VM Host system is booted, and when the Integrity VM software is stopped and restarted on a running VM Host. For example, when you upgrade Integrity VM to a new version on a running system, the software is started automatically. The VM Host 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 VM Host. This is the default behavior. 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.

`-O` *`os-type`*`[:`*`version`*`]`

Specifies the type and version of the operating system running on the virtual machine. The response will affect the default selection of certain virtual machine attributes, such as amount of memory and CPU power. The *`os_type`* is one of the following: `HPUX` or `OPENVMS`. This parameter is not case-sensitive.

The *`version`* is specific to the operating system type. The version specifies a descriptive text string of the version of the operating system. The version string can consist of up to 256 alphanumeric characters, including A-Z, a-z, 0-9, the dash (—), the underscore (_), and the period (.). If white space is desired then `version` must be quoted.

NOTE: You can specify the *`os_type`* here manually, but when the guest boots, the *`os_type`* is set to whatever operating system was last booted. The operating system *`version`* is only set manually with the `-O` option.

`-c` *`number-vcpus`*

Specifies the number of virtual CPUs this virtual machine sees at boot time. If unspecified, the number defaults to one.

The maximum number of virtual CPUs that can be allocated to a guest is 16.

`-r` *`amount`*

Specifies the amount of memory available to this virtual machine.

The size is expressed as an integer, optionally followed by one of these units:

- M (megabytes)
- G (gigabytes)

If unspecified, the unit defaults to megabytes. If the `-r` option is omitted, the size defaults to 2 GB.

`-g` *`group`*`[:{`*`admin`*` | `*`oper`*`}]`

Specifies the group authorization. A VM Host user account that is a member of this group can use the `hpvmconsole` command to manage this guest. The *`group`* attribute specifies the name of the group. The argument specifies the privilege level available at the virtual console: either `admin` or `oper` (the default).

This option can be specified more than once.

`-u` *`user`*`[:{`*`admin`*` | `*`oper`*`}]`

Specifies the user authorization. A VM Host user account specified here can use the `hpvmconsole` command to manage this guest. The *`user`* attribute specifies the user name. The argument specifies the privilege level available at the virtual console: either `admin` or `oper` (the default).

This option can be specified more than once.

`-s`

Verifies the virtual machine configuration and returns warnings or errors but does not create the virtual machine.

This option is used to start 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.

The `-F` and `-s` options are mutually exclusive.

`-x`

Specifies whether the new virtual machine uses dynamic memory and the values associated with it by including the following keywords:

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

  Specifies whether the virtual machine's dynamic memory settings are automatically adjusted. The `ram_dyn_entitlement` and `amr_enable` options must be set to enable adjustments.

- `-x ram_dyn_entitlement=`*`amount`*

  Specifies the minimum guaranteed amount of memory.

- `-x amr_enable={0|1}`

  Enables or disables AMR monitoring for a guest, where 1 enables and 0 disables. The monitor (amr daemon) adjusts the guest size and take its entitlement into account.

- `-x 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.

- `-x sched_preference={`*`none|cell|ilm`*`}`

  where:

  - *`none`* — The default preference. If your application is predominantly CPU bound, specifying either *`ilm`* or *`cell`* will perform the same.
  - *`cell`* — The cell with the most CPU and memory space is chosen. When that guest is active, the scheduler then optimizes where the guest runs, so that it can be closest to its memory.
  - *`ilm`* — Indicates that guests that are larger than any single cell and contain highly threaded applications.

- `-x graceful_stop_timeout=`{0 | *`number`*}

  Specifies the amount of time in seconds to allow a graceful stop when `hpvmstop –g` is specified. The default is 30 seconds.

NOTE: If the graceful stop does not finish within the time frame specified, a hard stop is then executed.

Used for Online VM Migration:

- `–x migrate_copy_phase_timeout=` {*number of seconds*} — Specifies the maximum number of seconds the online migration spends during the full-copy stage. The default is 0 seconds.
- `-x migrate_frozen_phase_timeout=` {*number of seconds*} — Specifies the maximum number of seconds the migration spends during the freezing stage. The default is 60 seconds.
- `-x migrate_init_phase_timeout=`{*number of seconds*} — Specifies the maximum number of seconds the online migration spends during the initialize phase of the migration. The default is 10 seconds.
- `-x migrate_io_quiesce_phase_timeout=` {*number of seconds*} — Specifies the maximum number of seconds the migration spends during the quiesce stage. The default is 15 seconds.
- `-x online_migration=` {*enabled* | *disabled*}
- `-x tunables=`{*name*=*value* [, *name*=*value*,...]}

RETURN VALUES

The `hpvmcreate` command exits with one of the following values:

0: Successful completion.

1: One or more error conditions occurred.

DIAGNOSTICS

The `hpvmcreate` displays error messages on stderr for any of the following conditions:

- An invalid option is specified.
- An invalid value is specified for an option or a value is omitted.
- The specified *vm-name* already exists. Use the `hpvmmodify` command to modify an existing guest.
- One or more options other than `-a`, `-g` or `-u` were specified more than once or the same resource was allocated more than once.
- An unavailable resource (allocated to another virtual machine, or exceeding the available resource limit) was specified.
- A value was omitted for an argument that requires one, or a value was supplied for an argument that does not take one.
- The `hpvmcreate` command and the Integrity VM software are at different version levels.

EXAMPLES

The following example creates a virtual machine named myguest1, specifying 4 virtual CPUs, and 2 GB of memory, and `/dev/rdisk/disk0` as a SCSI disk device:

```
# hpvmcreate -P myguest1 -c 4 -r 2G -a disk:scsi::disk:/dev/rdisk/disk0
```

The following example creates a virtual machine named myguest2, specifying 2 virtual CPUs and a virtual switch named vswitch1. Each virtual CPU has a 50% entitlement.

```
# hpvmcreate -P myguest2 -c 2 -e 50 -a disk:scsi::disk:/dev/rdisk/disk0 \
-a network:avio_lan::vswitch:vswitch1
```

The following example creates a virtual machine named cougar with 2 virtual CPUs, 2 GB memory, a virtual disk backed by a whole disk, a virtual disk backed by a partition, a virtual disk backed by an LVM volume, a virtual DVD backed by an ISO file, a virtual network interface backed by virtual switch localnet, and an accelerated virtual network interface backed by virtual switch hostnet:

```
# hpvmcreate -P cougar -c 2 -r 2G \
-a disk:scsi::disk:/dev/rdisk/disk0 \
-a disk:scsi::disk:/dev/rdisk/disk1 \
-a disk:scsi::lv:/dev/vg00/rguestvol1 \
-a dvd:scsi::file:/var/opt/hpvm/ISO-images/hpux/1123505GOLD.ISO \
-a network:lan::vswitch:localnet \
-a network:avio_lan::vswitch:hostnet
```

The following are sample warning messages that are returned when the `hpvmcreate` command is executed with various configuration problems on the guest myguest3:

```
HPVM guest myguest3 configuration problems:
Warning 1: Guest needs more vcpus than server supports.
Warning 2: Insufficient free memory for guest.
Warning 3: Insufficient swap resource for guest.
Warning 4: Insufficient cpu resource for guest.
Warning 5 on item /dev/rdisk/disk0: Device file '/dev/rdisk/disk0' in use by another guest.
Warning 6 on item /dev/vg00/rswap: Device file '/dev/vg00/rswap' in use by server.
Warning 7 on item /dev/rdisk/disk3 backing device does not exist.
Warning 8 on item /dev/rdisk/disk3: Device file '/dev/rdisk/disk3' in use by another guest.
Warning 9 on item hostnet: MAC address in use for switch hostnet.
Warning 10 on item offnet: Vswitch offnet is not active.
Warning 11 on item badnet: 'badnet' backing device does not exist.
These problems will prevent HPVM guest myguest3 from booting.
```

The following example shows how to create the guest `myguest1` and specify dynamic memory control.

```
# hpvmcreate -P myguest1 -c 4 -r 2G -a disk:scsi::disk:/dev/rdisk/disk0 -x dynamic_memory_control=1
```

The following example creates a guest named `testguest` with a 100 Mhz minimum and a cap of 250 Mhz.

```
# hpvmcreate -P testguest -E 100:250
```

OBSOLESCENCE

The `-F` option is deprecated in Integrity Virtual Machines commands. This flag might be removed in a future release of Integrity Virtual Machines. HP recommends use of the command with the `-s` option, then resolving any errors or warnings that result before executing the command without the `-s` option and without the `-F` option.

AUTHORS

The `hpvmcreate` command was developed by HP.

SEE ALSO

On the VM Host:

hpvm(5), *hpvmclone*(1M), *hpvmcollect*(1M), *hpvmconsole*(1M), *hpvmdevinfo*(1M), *hpvmdevmgmt*(1M), *hpvmdevtranslate*(1M), *hpvmhostgdev*(1M), *hpvmhostrdev*(1M), *hpvminfo*(1M), *hpvmmigrate*(1M), *hpvmmodify*(1M), *hpvmmove_suspend*(1M), *hpvmnet*(1M), *hpvmnvram*(1M), *hpvmpubapi*(3), *hpvmremove*(1M), *hpvmresources*(5), *hpvmresume*(1M), *hpvmsar*(1M), *hpvmsg_move*(1M), *hpvmsg_package*(1M), *hpvmstart*(1M), *hpvmstatus*(1M), *hpvmstop*(1M), *hpvmsuspend*(1M), *hpvmupgrade*(1M), *p2vassist*(1M)

On the Integrity VM guest:

hpvmcollect(1M), *hpvminfo*(1M), *hpvmmgmt*(1M), *hpvmpubapi*(3)

hpvmdevinfo(1M)

NAME

hpvmdevinfo – Report about storage for a virtual machine.

SYNOPSIS

On the VM Host:

`hpvmdevinfo [ -P vm-name | -p vm-number ][ -M | -S | -V ]`

On the guest:

`hpvmdevinfo [ -M | -S | -V | -Q ]`

DESCRIPTION

The `hpvmdevinfo` command displays the available information about the storage devices assigned to a virtual machine. When run on the VM Host, `hpvmdevinfo` displays information about selected guests or all guests and their assigned storage. This information always includes the guest name and the VM Host resource information. When available, it includes the guest device name corresponding to the assigned resource.

Depending on the report formatting options (`-M`, `-S`, `-V`), the information might also include the guest number, the local VM Host name (for merging reports), and the WWID of the VM Host resource.

When the `hpvmdevinfo` command is run in the guest, it displays the VM Host resource information and the guest device name, if available. Use the `ioscan` command after dynamic device changes to avoid insufficient reporting of some hardware device information.

Options

The `hpvmdevinfo` command recognizes the following command-line options:

-M
: Select machine-parseable output format. This causes all fields for a given device to be displayed in a one-line format, with major fields separated by a colon (:). Subfields are separated with a semicolon (;). The `-M`, `-Q`, `-S`, and `-V` options are mutually exclusive.

`-P` *vm-name*
: (VM host only) Select reporting for only the specified virtual machine. The `-P` and `-p` options are mutually exclusive.

`-p` *vm-number*
: (VM Host only) Select reporting for only the specified virtual machine. The `-p` and the `-P` options are mutually exclusive.

`-Q`
: (Guest only) Suppress the display of any information, and only send guest device information to the VM Host. The `-M`, `-Q`, `-S`, and `-V` options are mutually exclusive

`-S`
: Select spreadsheet compatible output format. This causes all fields for a given device to be displayed in a one-line format, with major fields separated by a comma (,). Subfields are separated with a semicolon (;). The `-M`, `-Q`, `-S`, and `-V` options are mutually exclusive.

`-V`
: Select verbose output format. This causes all fields for a given device to be displayed in a multi-line format, with user-readable information for each field displayed on a separate line. The `-M`, `-Q`, `-S`, and `-V` options are mutually exclusive.

RETURN VALUES

The command exits with one of the following values:

0: Successful completion.

1: One or more error conditions
occurred.

EXAMPLES

The following example, specified on a VM Host, displays information about all guests, in a format suitable for loading into a spreadsheet.

```
# hpvmdevinfo -S
```

The following example, specified on a guest, displays information matching guest devices with VM Host resources:

```
# hpvmdevinfo
```

AUTHOR

The `hpvmdevinfo` command was developed by HP.

SEE ALSO

On the VM Host:

hpvm(5), *hpvmclone*(1M), *hpvmcollect*(1M), *hpvmconsole*(1M), *hpvmcreate*(1M), *hpvmdevmgmt*(1M), *hpvmdevtranslate*(1M), *hpvmhostgdev*(1M), *hpvmhostrdev*(1M), *hpvminfo*(1M), *hpvmmigrate*(1M), *hpvmmodify*(1M), *hpvmmove_suspend*(1M), *hpvmnet*(1M), *hpvmnvram*(1M), *hpvmpubapi*(3), *hpvmremove*(1M), *hpvmresources*(5), *hpvmresume*(1M), *hpvmsar*(1M), *hpvmsg_move*(1M). *hpvmsg_package*(1M), *hpvmstatus*(1M), *hpvmstop*(1M), *hpvmsuspend*(1M), *hpvmupgrade*(1M)*p2vassist*(1M)

On the Integrity VM guest:

hpvmcollect(1M), *hpvminfo*(1M), *hpvmmgmt*(1M), *hpvmpubapi*(3)

hpvmdevmgmt(1M)

NAME

hpvmdevmgmt – Manage the devices that are associated with the VM Host and the guests.

SYNOPSIS

hpvmdevmgmt -a { server | env |rdev | gdev }*:entry-name* [-V]
hpvmdevmgmt -d { server | env |rdev | gdev }*:entry-name* [-V]
hpvmdevmgmt -d gdev: { all | *entry-name* } :depend*:dependent-name* [-V]
hpvmdevmgmt -d { gdev |rdev }*:/devpath* IGNOREWWID
hpvmdevmgmt -d { server | env |rdev | gdev }*:entry-name*:attr*:attr-name*
[-V]
hpvmdevmgmt -d gdev_alias:/dev/rdisk/disk*nn*
hpvmdevmgmt -m { server | env |rdev | gdev
}*:entry-name*attr*:attr-name=attr-value* [-V]
hpvmdevmgmt -m {[gdev] | [rdev]} *:/devpath*:IGNOREWWID={YES|NO}
hpvmdevmgmt -n {[gdev] | [rdev]}*:oldentry-name:newentry-name0*
[,*newentry-name1*] [-V]
hpvmdevmgmt -l { all |server env | rdev | gdev } [:entry-name] [-V]
hpvmdevmgmt -l { all |server env | rdev | gdev
}:depend*:dependent-name* [-V]
hpvmdevmgmt -l { all |server env | rdev | gdev }*:attr-name=attr-value*
[-V]
hpvmdevmgmt -I [-V]
hpvmdevmgmt -r > repair_script.h
hpvmdevmgmt -v
hpvmdevmgmt [-A] -S *file-size file-name* [-V]

DESCRIPTION

Manages entries in the Integrity VM device-management database, which tracks and validates guest-device usage, ensures that devices are only shared deliberately, and restricts guest access to devices used by the VM Host. Guest devices are added, modified, and removed from this database when you use Integrity VM commands, such as hpvmcreate, hpvmmodify, and hpvmclone. The hpvmdevmgmt command allows you to examine the database entries and alter specific device attributes, such as shared devices. You can also use the hpvmdevmgmt command to create database entries for restricted devices (to which guest access is prohibited) and for pre-extending files used as virtual devices.

The device management database contains four types of entries:

- Restricted devices (rdev)
- Guest devices (gdev)
- Environmental entries (env)
- VM Host devices (server)

A device management database entry contains a name or alias, attributes in the form *attribute-name=value*, a list of guest names and device entities that depend on the entry (called its dependents), and a unique identifier.

The Integrity VM device-management facility allows you to restrict devices and directories for exclusive use by the VM Host in two ways:

- By specifying individual device, file, or directory paths:

```
# hpvmdevmgmt -a rdev:/some-device
# hpvmdevmgmt -a rdev:/some-file
# hpvmdevmgmt -a rdev:/some-directory
```

This method restricts the ability to later designate the specified device, file, or directory for use by a guest.

- By specifying recursive directory paths:

```
Restrict everything in and under a specified directory
that is also in the same file system that the specified
restricted directory is in.

# hpvmdevmgmt -a rdev:/some-directory:attr:RECURSIVE=FS

Restrict everything in and under a specified directory,
even if sub-paths to the specified restricted directory
contain mount points to other file systems.

# hpvmdevmgmt -a rdev:/some-directory:attr:RECURSIVE=DIR
```

This method restricts the ability to later designate anything in or hierarchically below the specified directory for use by a guest. Recursion can be specified as either limited to directories, files, and devices in the same file system as the restricted directory (RECURSIVE=FS), or as unlimited by the file system, applying to everything hierarchically below the restricted directory (RECURSIVE=DIR).

The RECURSIVE attribute can be specified only for restricted directories and can be disabled or removed entirely. Note the following:

```
Disable restricted directory recursion, but keep the
RECURSIVE attribute.

# hpvmdevmgmt -m rdev:/some-directory:attr:RECURSIVE=NONE

Disable restricted directory recursion by removing the
RECURSIVE attribute.

# hpvmdevmgmt -d rdev:/some-directory:attr:RECURSIVE
```

The Integrity VM device-management facility automatically makes two directories, `/etc/` and `/stand`, recursively restricted by file system when you run `hpvmdevmgmt -I`. This command is always run when Integrity VM is started with `/sbin/init.d/hpvm start`. To inhibit the automatic restriction of these directories, add one of the following lines to the `/etc/rc.config.d/hpvmconf` file:

```
HPVMRESDIRELIST=""
HPVMRESDIRLIST=
```

To replace the default list of restricted directories, the *HPVMRESDIRLIST* parameter can be specified as follows:

```
HPVMRESDIRLIST="RECURSIVE={DIR|FS}:/dir1[,...]"
HPVMRESDIRLIST=RECURSIVE={DIR|FS}:/dir1[,...]
```

If the device pathnames have changed for disk backing stores and have been assigned as guest devices, run the `hpvmdevmgmt -r` report and repair function to determine the new pathnames for those disks and have Integrity VM reassign them for you. This command generates a shell script that performs the reassignment using the `hpvmdevmgmt -n` command. See the examples, for syntax and usage.

If the report and repair function determines that more than one device path can be selected to replace an obsolete device path, you must manually edit the script to make the selection. The report and repair script might also suggest that you remove certain device entries with which it has found a problem from the device database. You must manually edit the script to enable all such removals. If the report and repair function finds errors, it might write warning and information messages to stderr. All repair-script output is written to stdout. If no errors are found, you do not need to manually edit the repair script in order for it to run.

The IGNOREWWID attribute allows you to set the WWID. The attribute value can be YES or NO. The following example sets the WWID to WWID_NULL and forces the Integrity VM device management utilities to ignore the WWID field and do no WWID field checking:

```
# hpvmdevmgmt -m gdev:/someguestdevice:attr:IGNOREWWID=YES
```

NOTE: HP recommends that you do not use IGNOREWWID=YES on a multipath device, or use it only on one path of a multipath device, because serious device or guest conflicts could arise.

The UUID and PHYSUUID are set up in the Virtual Connect profile.

Only superusers can execute the hpvmdevmgmt command.

A set of Integrity VM servers can be grouped into an Integrity VM multiserver environment, MSE. This grouping is required when a set of Integrity VM servers is also configured as a Serviceguard cluster. To set up an Integrity VM multiserver environment, use the hpvmdevmgmt command on each of the servers in the group to establish each server entry and the MSE group entry. The same set of hpvmdevmgmt commands can be run on each Integrity VM server in the group with the exception of the local server entry. In most cases, the local server entry is already present. If it is already present, use the hpvmdevmgmt -m command to add the attributes to the existing entry. If you attempt to add a new server entry, an error is issued advising you that an entry with the same name already exists.

NOTE: If the Serviceguard cluster is configured before establishing the MSE group, the Integrity VM servers form an MSE server group name by adding the prefix HPVM-SG- to the Serviceguard cluster name. This MSE group entry is automatically formed on each MSE server once the server entries are defined and guests are configured into Serviceguard packages.

When the HPVM_MSE_GROUP_ENTRY is made automatically, it receives a required UUID attribute. This UUID attribute must be the same on each MSE server. To do this, choose one of the UUIDs and change the other servers to match.

The following commands set up the an Integrity VM MSE group and also change the UUID attribute:

```
# hpvmdevmgmt -a env:HPVM_MSE_GROUP_ENTRY:attr:GROUPNAME=HPVM-SG-clustername
# hpvmdevmgmt -m env:HPVM_MSE_GROUP_ENTRY:attr:UUID=b7a4ec05-c2a2-11db-be42-adc07415534
```

In this command sequence, you supply the IP address (*ip-address*) used by Serviceguard to monitor the cluster. Also, specify a server identifier (*server-id*) from 1 to 255.

```
# hpvmdevmgmt -a server:hostname:attr:SERVERADDR=ip-address,SERVERID=server-id
```

If the entry already exists, use the following command:

```
# hpvmdevmgmt -m server:hostname:attr:SERVERADDR=ip-address,SERVERID=server-id
```

To delete a multiserver entry, enter the following command:

```
# hpvmdevmgmt -d server:hostname
```

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:scsi::null:/var/opt/hpvm/ISO-images/hpux/
# hpvmmodify -P host2 -a dvd:scsi::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 use by one guest only.

The `PRESERVE` attribute has the values {YES|NO}. When set to YES, it preserves the existence of a device database entry if, and when, the last guest listed as a dependency on that device is deleted. If it is set to NO, or not set, when the last guest dependency is deleted for an entry, that entry is deleted from the device database. When you add it to the device entry, it prevents the entry from being automatically device deleted by an `hpvmmodify -d` command that deletes the last guest dependency for a particular device entry. That is, the entry is preserved for future use (whereas, the default behavior is to clean up non-used entries by deleted them.) You can also set `PRESERVE` to NO, which is the same as not having the attribute at all. Note the following example:

```
# hpvmdevmgmt -m gdev:/device:attr:PRESERVE=YES
```

You can give file entries in the device database that are used as backing storage for disks or DVDs a `USAGE` attribute with the value DISK or DVD. This attribute is valid only for files and cannot be set for a real disk, DVD, or burner (when the burner is used as a DVD). Its values are:

`USAGE={DISK|DVD}`

The `USAGE` attribute labels the usage of a file, so that non Integrity VM management software can determine what type of device the file is to be presented as.

The following examples show how to set the `USAGE` attribute:

```
# hpvmdevmgmt -a gdev:/path/file:attr:USAGE=DISK
# hpvmdevmgmt -a gdev:/path/isofile.iso:attr:USAGE=DVD

or

# hpvmdevmgmt -a gdev:/path/file
# hpvmdevmgmt -m gdev:/path/file:attr:USAGE=DISK

# hpvmdevmgmt -a gdev:/path/isofile.iso
# hpvmdevmgmt -m gdev:/path/isofile.iso:attr:USAGE=DVD
```

Although not required, you should give the attributes `SHARE=YES` and `PRESERVE=YES` to the ISO files that are used to back DVD devices.

For attached AVIO devices, you can specify the `SHARE_LUNPATHS` attribute, which allows multiple guests to boot at the same time when they use different lunpaths to the same device. Two different guests can use different lunpaths to the same device at the same time (SHARE_LUNPATHS=YES) of the paths use *different* initiators; however, Integrity VM blocks the use of different guests using different lunpaths at the same time if they use the *same* initiator. SHARE_LUNPATHS=YES can still be set when potential initiator-sharing conflicts exists. The `hpvmstart` command blocks the guests from booting when the same-iinitiator conflict (of using different paths) exists.

Options

The following options can be specified only once.

The `hpvmdevmgmt` command recognizes the following command-line options and arguments:

`-l {server| rdev| gdev|env}`

Lists an entry. This option can perform the following actions:

- List all entries:

  `hpvmdevmgmt -l all`

- List all devices with a specific attribute:

  `hpvmdevmgmt-V -l {all|server|rdev|gdev}:`*`attr-name=attr-value`*

- List all devices with a specific dependency:

  `hpvmdevmgmt -V -l {all|server|rdev|gdev}:depend:`*`dependent-name`*

- List a single entry by name:

  `hpvmdevmgmt -V -l {all|server|rdev|gdev}:`*`entry-name`*

`-v`

Displays the version number of the `hpvmdevmgmt` output format. The version number is displayed first, 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. The *`filename`* is the path name of the file to be created. An error is returned on an attempt to overwrite an existing file.

`—A`

The `-A` option adds a file created by `-S` option to the device database. It adds the device with the following attributes:

- PRESERVE=YES
- USAGE=DISK

For example:

`#` **`hpvmdevmgmt -A -S file_size file_name`**

`-I`

Creates attached (passthrough/sctl) devices. Attached devices include tape devices, media changers, and CD/DVD burners. Also, automatically adds `/etc` and `/stand` to the device database as restricted directories (using RECURSIVE:FS).

NOTE: The `hpvmdevmgmt -I` command works only on device directories that have changed. If you want the command to rigorously search all device directories regardless of whether changes have happened since the last search, remove the `/var/opt/hpvm/common/hpvm_devinit` file before running `hpvmdevmgmt -I`.

`-m {server | rdev |gdev |env}:`*`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 | env}:`*`entry_name`*`[:attr:`*`attr-name=attr-value`*`]`

Adds an entry. This option can be used for:

- Adding a restricted device (rdev)
- Adding a VM Host device (server)
- Adding a guest device (gdev)
- Adding a Serviceguard cluster entry

-d {server | rdev | gdev |env}:*entry-name*[:*param*:*arg*]

Deletes an entry. This option can perform the following deletions:

- Deletion of an entry:

  -d {server | rdev | gdev}:*entry-name*

  An entry cannot be deleted if it has dependents.

- Delete a dependent from one or all entries of a certain type:

  -d gdev:{all | *entry-name*}:depend:*dependent_name*

- Delete an attribute from an entry:

  -d {server | rdev | gdev}:*entry-name*:attr:*attr-name*

- Delete one alias if a device has multiple aliases defined.

  -d gdev_alias:/dev/rdisk/disk*nn*

  The following limitations apply:

  - Valid to delete only guest device aliases.
  - More than one alias must be defined, so that the alias delete does not attempt to delete the only alias. If there is only one alias, use the following command to delete the entire entry:

    # **hpvmdevmgmt -d gdev:/device**

  - The alias must not be in used by any guest. If the alias to be deleted is in use by a guest, you must remove the dependency with the following command:

    # **hpvmmodify -P guest -d <resources>**

    Note that the aliases for the device can be used by any guest.

To delete a cluster entry, specify the server host name for the *entry-name*.

-n gdev|rdev:*oldentry-name*:*newentry_-name0*[,*newentry-name1*]

Replaces a device. Typically used when a device goes bad.

-r

Generates a report script that can be used after inspection to fix various device database problems.

RETURN VALUES

The hpvmdevmgmt command exits with one of the following values:

0: Successful completion.

1: One or more error conditions occurred.

DIAGNOSTICS

The hpvmdevmgmt command displays error messages for any of the following conditions:

- An invalid option is specified.
- An invalid value is specified for an option.
- A value was omitted for an argument that requires one, or a value was supplied for an argument that does not take one.
- The hpvmdevmgmt command and Integrity VM are at different revision levels.

EXAMPLES

Generate a device report and repair script and redirect the output to a file. This example also sets the script to executable and writable, starts an editor to edit the file, and then runs the edited file. If the script is not edited and there are errors to report or repair, you receive a message directing you to edit it. If there are no errors to report or repair, the script runs without editing.

```
# hpvmdevmgmt -r > repair_script.sh
# chmod +wx repair_script.sh
# ./repair_script.sh
Device database status - Total problems found: zz.
Number of restricted devices with problems: xx.
Number of guest devices with problems: yy.

Since errors were found, please edit and inspect this script by hand to enable its repair functions.

# vi repair_script.sh
# ./repair_script.sh
```

List a guest-device entry:

```
# hpvmdevmgmt -l gdev:/dev/rdisk/disk2
```

List all the restricted devices:

```
# hpvmdevmgmt -l rdev
```

List all the guest devices used by the guest `phantom`:

```
# hpvmdevmgmt -l gdev:depend:phantom
```

List all shareable guest devices (those with the attribute SHARE=YES):

```
# hpvmdevmgmt -l gdev,SHARE=YES
```

Allocate a 4 GB file:

```
# hpvmdevmgmt -S 4G /var/opt/hpvm/guests/mirage/disk_4G_file
```

Create attached (passthrough/esctl) devices:

```
# hpvmdevmgmt -I
```

Modify a guest device attribute on an ISO file from not shared to shared:

```
# hpvmdevmgmt -m gdev:/var/opt/hpvm/ISO-images/hpux/kit:attr:SHARE=YES
```

Add a restricted device entry:

```
# hpvmdevmgmt -a rdev:/dev/vg00/lvol8
```

Delete a restricted device:

```
# hpvmdevmgmt -d rdev:/dev/vg00/lvol8
```

Delete the guest `mirage` dependent from all guest devices:

```
# hpvmdevmgmt -d gdev:all:depend:mirage
```

Replace a guest device:

```
# hpvmdevmgmt -n gdev:/dev/vgvm/lvol5:/dev/rdisk/disk4
```

Share 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 Integrity VM- generated alias for the hardware path. The Integrity 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.

Share a tape device using different initiators (different lunpaths):

1. Add different paths to each guest:

   ```
   # 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/).

2. 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
   ```

3. 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
   ```

4. 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.

5. 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
   ```

AUTHORS

The `hpvmdevmgmt` command was developed by HP.

SEE ALSO

On the VM Host:

hpvm(5), *hpvmclone*(1M), *hpvmcollect*(1M), *hpvmconsole*(1M), *hpvmcreate*(1M), *hpvmdevinfo*(1M), *hpvmdevtranslate*(1M), *hpvmhostgdev*(1M), *hpvmhostrdev*(1M), *hpvminfo*(1M), *hpvmmigrate*(1M), *hpvmmodify*(1M), *hpvmmove_suspend*(1M), *hpvmnet*(1M),*hpvmnvram*(1M), *hpvmpubapi*(3), *hpvmremove*(1M), *hpvmresources*(5), *hpvmresume*(1M), *hpvmsar*(1M), *hpvmsg_move*(1M), *hpvmsg_package*(1M), *hpvmstart*(1M), *hpvmstatus*(1M), *hpvmstop*(1M), *hpvmsuspend*(1M), *hpvmupgrade*(1M), *p2vassist*(1M)

On the Integrity VM guest:

hpvmcollect(1M), *hpvminfo*(1M), *hpvmmgmt*(1M), *hpvmpubapi*(3)

hpvmdevtranslate(1M)

NAME

hpvmdevtranslate – Translate Integrity VM guest devices into HP-UX 11i v3 agile devices.

SYNOPSIS

`hpvmdevtranslate` [-v] `-a` *hpvm_mgmtdb*
`hpvmdevtranslate` [-v] `-a`
`hpvmdevtranslate` [-v] `-P` *guestname*
`hpvmdevtranslate` [-v] `-u`
`hpvmdevtranslate` [-v]

DESCRIPTION

The primary purpose of this program is to help Integrity VM administrators upgrade an Integrity VM host from HP-UX 11i v2 to HP-UX 11i v3. This program produces a script, `/var/opt/hpvm/common/hpvm_dev_convert`, which can be executed to convert existing guest devices into 11i v3 agile devices. This program is most useful if the upgrade from 11i v2 to 11i v3 or the upgrade between two 11i v3 versions was accomplished using a cold-install of 11i v3, rather than update-ux. The update-ux procedure preserves all device names.

The cold-install can map storage devices to different device names, which prevents guests from running. The cold-install changes legacy device names to agile device names, which is why you need to run this script if you do a cold-install. With this program, you can also change a single guest or all guests backing storage devices to use the new 11i v3 agile devices.

Use the following procedure if a cold install upgrade is required.

1. Stop Integrity VM by running the following:

   `/sbin/init.d/hpvm stop`
2. Do a backup of `/var/opt/hpvm`.
3. Do the cold-install of HP-UX.
4. Install Integrity VM V4.*
5. Stop Integrity VM by running the following:

   `/sbin/init.d/hpvm stop`
6. Tar in or recover the backup of `/var/opt/hpvm` to `/var/opt/hpvm`.
7. Move `/var/opt/hpvm/common/hpvm_mgmtdb` to `/var/opt/hpvm/common/hpvm_mgmtdb.orig`.
8. Start Integrity VM with the following command:

   `/sbin/init.d/hpvm start`
9. Run the following command:

   `#` **`hpvmdevtranslate -a /var/opt/hpvm/common/hpvm_mgmtdb.orig`**
10. Carefully edit the created script, `/var/opt/hpvm/common/hpvm_dev_convert` , making sure that you comment out the early exit line and taking note of any ERROR lines. The ERROR lines flag translations that could not be made. These ERRORs are from devices that could not be uniquely identified, and therefore require the administrator to resolve after running the script.
11. Run the edited script `/var/opt/hpvm/common/hpvm_dev_convert`.
12. Continue with the remaining 11i v3 Integrity VM Host configuration until the host is functionally equivalent to the former 11i version Host configuration.

You can use the `hpvmdevtranslate` command to convert all existing guests or a single guest to use the 11i v3 agile devices. For example, to convert all existing guests, use the following:

`#` **`/opt/hpvm/bin/hpvmdevtranslate -a`**

To convert a single guest, use the following command:

```
# hpvmdevtranslate -P guestname
```

You can use the `hpvmdevtranslate` command to undo the previous translation if the translation is not more than five days old. The undo operation is run with the following command:

```
# hpvmdevtranslate -u
```

Options

The following options can be specified only once.

The `hpvmdevtranslate` command recognizes the following command-line options and arguments:

`-v`	Displays the version number of the `hpvmdevtranslate` command. The version number is displayed first, followed by the information specified by the other options.
`-a`	Translates all existing guests. If an hpvm_mgmtdb file is specified, the translation uses this file.
`-P` *guestname*	Translates the specific guest.
`-u`	If the translation is not more than five days old, the `-u` option, performs and undo operation.

RETURN VALUES

The `hpvmdevtranslate` command exits with one of the following values:

0: Successful completion.

1: One or more error conditions occurred.

NOTE: An error exit occurs only if the program was unable to complete its processing. However, if the program finds guest devices that cannot be translated, the output script will have comment lines indicating the error with “ERROR:”

DIAGNOSTICS

`The hpvmdevtranslate` command displays error messages for any of the following conditions:

A legacy device has no agile device translation.

EXAMPLES

The following example lists the version of the `hpvmdevtranslate` command:

```
# hpvmdevtranslate -v
```

The following example translates the devices using an 11i v2 `hpvm_mgmtdb` file:

```
# hpvmdevtranslate [-v] -a /var/opt/hpvm/common/hpvm_mgmtdb_pre1131
```

The following example translates all existing guests:

```
# hpvmdevtranslate [-v] -a
```

The following example translates a specific guest:

```
# hpvmdevtranslate [-v] -P guestname
```

The following examples performs an undo operation if the translation is not older than 5 days:

```
# hpvmdevtranslate [-v] -u
```

AUTHORS

The `hpvmdevtranslate` command was developed by HP.

SEE ALSO

On the VM Host:

hpvm(5), *hpvmcollect*(1M), *hpvmconsole*(1M), *hpvmcreate*(1M), *hpvmdevinfo*(1M), *hpvmdevmgmt*(1M), *hpvmhostgdev*(1M), *hpvmhostrdev*(1M), *hpvminfo*(1M), *hpvmmigrate*(1M), *hpvmmodify*(1M), *hpvmmove_suspend*(1M), *hpvmnet*(1M), *hpvmnvram*(1M), *hpvmpubapi*(3), *hpvmremove*(1M), *hpvmresources*(5), *hpvmresume*(1M), *hpvmsar*(1M), *hpvmsg_move*(1M), *hpvmsg_package*(1M), *hpvmstart*(1M), *hpvmstatus*(1M), *hpvmstop*(1M), *hpvmsuspend*(1M), *hpvmupgrade*(1M), *p2vassist*(1M)

On the Integrity VM guest:

hpvmcollect(1M), *hpvminfo*(1M), *hpvmmgmt*(1M), *hpvmpubapi*(3)

hpvmhostgdev(1M)

NAME

hpvmhostgdev – Manages Integrity VM Host devices available for virtual machine access.

SYNOPSIS

hpvmhostgdev [`-H` | `-u` | { `-a` | `-c` {*disklist*|*lvlist*} | `-l` [>*devlistfile*] } | `-d` | `-f` *devlistfile*| `-r` | `-v` { `all`|`assigned`|`unassigned` }]

DESCRIPTION

The `hpvmhostgdev` command analyzes disklist and lvlist output searching for unused disks and logical volumes (LVs). It uses its results to add unused devices to the Integrity VM device database as guest devices (gdevs).

When executed without any options or with the `-H` option, the `hpvmhostgdev` command displays a description of the command.

NOTE: Do not use a disk to create volume group after it has been made a gdev (either by the script or by hand). Once a disk has been made a gdev, it is free to be used as a disk only by a guest.

To use a disk as part of a volume group of which you want to create logical volumes, perform the following steps:

1. If the disk is a gdev, unmark it as one, if it is not in use by a guest:

   `#` **`hpvmdevmgmt -d gdev:/dev/rdisk/disk#`**

2. Follow one of the following scenarios:
   - Scenario 1:
     a. Make the disk restricted:

        `#` **`hpvmdevmgmt -a rdev:/dev/rdisk/disk#`**

     b. Create the volume group with the disk (or add it to an existing volume group), followed by the LV creation.
   - Scenario 2:
     a. Create the volume group with the disk (or add it to an existing volume group), followed by the LV creation.
     b. Make the disk restricted, either by:
        - `#` **`hpvmdevmgmt -a rdev:/dev/rdisk/disk#`**
        - `#` **`hpvmhostrdev -u`**

          If many disks are being used for volume groups, then running `hpvmhostrdev` –`u` when you are done is more efficient than running the individual adds with `hpvmdevmgmt -a rdev`.

NOTE: Only a superuser may execute the `hpvmhostrdev` command.

Options

The `hpvmhostgdev` command recognizes the following command-line options:

`-H` Displays a description of the `hpvmhostgdev` command.

`-a` Adds unused logical volumes and disks for guest use.

`-c` Displays the `disklist` or `lvlist` command output.

-d Deletes guest devices added by this script.

-f Uses specified device-list file to add devices for guest use.

-l Lists unused devices that can be added for guest use. (For example, pipe –l output to a file for use with –f.)

-r Recovers the Integrity VM device management database.

-u Uses disklist device cache (Do not force hardware scan.).

NOTE: This option is used to speed up the *disklist* command to obtain unused disks.

Use the -u option before using the following options:

- -a
- -c *disklist*
- -l

-v Verifies add and delete operations by this script (with the AUTOGDEV=YES attribute).

- all — lists all guest devices.
- assigned — lists all guest devices configured by any guests.
- unassigned — lists all guest devices not configured by any guests.

RETURN VALUES

The hpvmhostgdev command exits with one of the following values:

0: Successful completion.

1: One or more error conditions occurred.

DIAGNOSTICS

The hpvmhostgdev command displays error messages for any of the following conditions:

An invalid option is specified.

An empty or invalid file is specified with the -f option.

EXAMPLES

The following example adds unused disks and logical volumes as gdevs:

```
# hpvmhostgdev -a
```

The following example illustrates how to make a file of unused disk and logical volumes, and then how to use the list to add them to as gdevs:

```
# hpvmhostgdev -l > devicelist
# hpvmhostgdev -f devicelist
```

NOTE: You can edit the devicelist file to remove disks and logical volumes that you do not want added as gdevs.

The following example verifies guest devices added to the device database by the hpvmhostgdev script:

```
# hpvmhostgdev -v all
```

The following example deletes guest devices added by the hpmvhostgdev script:

```
# hpvmhostgdev -d
```

AUTHORS

The hpvmhostgdev command was developed by HP.

SEE ALSO

On the VM Host:

hpvm(5), *hpvmcollect*(1M), *hpvmconsole*(1M), *hpvmcreate*(1M), *hpvmdevinfo*(1M), *hpvmdevmgmt*(1M), *hpvmdevtranslate*(1M), *hpvmhostrdev*(1M), *hpvminfo*(1M), *hpvmmigrate*(1M), *hpvmmodify*(1M), *hpvmmove_suspend*(1M), *hpvmnet*(1M), *hpvmnvram*(1M), *hpvmpubapi*(3), *hpvmremove*(1M), *hpvmresources*(5), *hpvmresume*(1M), *hpvmsar*(1M), *hpvmsg_move*(1M), *hpvmsg_package*(1M), *hpvmstart*(1M), *hpvmstatus*(1M), *hpvmstop*(1M), *hpvmsuspend*(1M), *hpvmupgrade*(1M), *p2vassist*(1M)

On the Integrity VM guest:

hpvmcollect(1M), *hpvmmgmt*(1M), *hpvmpubapi*(3)

hpvmhostrdev(1M)

NAME

hpvmhostrdev – Manages virtual machine access to devices used by the Integrity VM Host system.

SYNOPSIS

`hpvmhostrdev` [`-D | -E | -d | -h | -r | -u`]

DESCRIPTION

The `hpvmhostrdev` command manages virtual machine access to devices — disks and volumes — used by the VM host system. The Integrity VM device-management database (`hpvmdevmgmt` (1M) provides the capability to restrict virtual machine access to devices used by the VM Host. The `hpvmhostrdev` command can automatically identify storage used by the VM Host physical system and restrict that storage from being used by virtual machines. This identification is consistent with that of system management tools such as the HP System Management Homepage (HP SMH) and the HP System Administration Manager (HP SAM).

When executed without any options or with the `-u` option, the `hpvmhostrdev` command operates as follows:

- Any devices previously identified and restricted by the `hpvmhostrdev` command are removed from the Integrity VM device database if they are no longer used by the VM Host system. Subsequently, such devices become available for use by virtual machines. All device currently used by the VM Host system are identified and restricted by the `hpvmhostrdev` command.
- To restrict a device from use by virtual machines, `hpvmhostrdev` inserts an entry for that device into the Integrity VM device database as a restricted device with the attribute AUTODEV set to YES (see the `hpvmdevmgmt` (1M) command).
- By default, `hpvmhostrdev -u` is executed during startup of the Integrity VM software. This startup execution can be enabled or disabled with the `-E` or `-D` options, respectively.
- Execution of the `hpvmhostrdev` command without any option causes the VM Host system to rescan its storage devices and might take some time to execute.

  NOTE: Do not use the `-u` option immediately after hardware has been added online (that is, without rebooting the VM Host system to add the hardware).

- The `hpvmhostrdev` command restricts only the user of disks and logical volumes used by the VM Host system for its swap space, file systems, or (in the case of disks), logical volume groups. Any unused devices are ignored (that is, not restricted) by `hpvmhostrdev`.
- When `hpvmhostrdev` is executed with the `-u`, the `-d`, or without any options, it creaets a backup copy of the Integrity VM Host device database. This copy can be recovered using the `-r` option.

NOTE: Only a superuser may execute the `hpvmhostrdev` command.

Options

The `hpvmhostrdev` command recognizes the following command-line options:

`-D` Disables execution of `hpvmhostrdev` during startup of the Integrity VM software. Subsequent startup of the Integrity VM system (that is, **`/sbin/init.d/hpvm start`**), does not automatically identify, update, and restrict use of the VM Host system's storage devices. Note that this option does not remove restricted device entries form the Integrity VM device database.

`-E` Enables automatic execution of `hpvmhostrdev` during startup of the Integrity VM software. Subsequent startup of the Integrity VM system (that is, **`/sbin/init.d/hpvm start`**), automatically identifies, updates, and restricts use of the VM Host system's storage devices.

Note that execution of `hpvmhostrdev -E` does not — by itself — add restricted device entries to the Integrity VM device database.

`-d` Removes restricted device entries from the Integrity VM device database that were previously added by execution of the `hpvmhostrdev` command. Such devices are no longer restricted from use by virtual machines.

NOTE: Any devices explicitly restricted through use of the `hpvmdevmgmt -a` *`rdev`* command are not removed by execution of the `hpvmhostrdev` command.

`-h` Displays a brief overview of the `hpvmhostrdev` command.

`-r` Restores the Integrity VM device database to the state previous to the last execution of the `hpvmhostrdev` command without any options or executed with the `-u` or `-d` options.

`-u` Prevents the rescan and update of the VM Host system's I/O system. Do not use this option immediately after I/O devices are added to the VM Host system online without a reboot. This option provides the best performance.

RETURN VALUES

The `hpvmhostrdev` command exits with one of the following values:

0: Successful completion.

1: One or more error conditions occurred.

DIAGNOSTICS

The `hpvmhostrdev` command displays error messages for any of the following conditions:

An invalid option is specified.

Multiple options are specified simultaneously.

EXTERNAL INFLUENCES

`hpvmdevmgmt` (1M)

The `hpvmhostrdev` command attaches a special attribute, AUTORDEV, to the entries it makes in the Integrity VM device database. Do not use the `hpvmdevmgmt` command to add or modify the AUTORDEV attribute of a restricted device entry in the device database. This might cause unexpected results, such as the entry being inadvertently removed or undisturbed by `hpvmhostrdev`.

EXAMPLES

The following example disables execution of `hpvmhostrdev` (and restriction of VM Host storage devices from use by virtual machines) at startup of the Integrity VM system.

```
# hpvmhostrdev -D
```

The following example removes all restricted devices added by previous execution of the `hpvmhostrdev` command:

```
# hpvmdevrdev -d
```

AUTHORS

The `hpvmhostrdev` command was developed by HP.

SEE ALSO

On the VM Host:

hpvm(5), *hpvmcollect*(1M), *hpvmconsole*(1M), *hpvmcreate*(1M), *hpvmdevinfo*(1M), *hpvmdevmgmt*(1M), *hpvmdevtranslate*(1M), *hpvmhostgdev*(1M), *hpvminfo*(1M), *hpvmmigrate*(1M), *hpvmmodify*(1M), *hpvmmove_suspend*(1M), *hpvmnet*(1M), *hpvmnvram*(1M), *hpvmpubapi*(3), *hpvmremove*(1M), *hpvmresources*(5), *hpvmresume*(1M), *hpvmsar*(1M), *hpvmsg_move*(1M), *hpvmsg_package*(1M), *hpvmstart*(1M), *hpvmstatus*(1M), *hpvmstop*(1M), *hpvmsuspend*(1M), *hpvmupgrade*(1M), *p2vassist*(1M)

On the Integrity VM guest:

hpvmcollect(1M), *hpvmmgmt*(1M), *hpvmpubapi*(3)

hpvminfo(1M)

NAME

hpvminfo -- Display information about the Integrity VM environment.

SYNOPSIS

`hpvminfo [ -V | -M | -X ][ -v | -S ]`

DESCRIPTION

Allows you to determine whether you are running in a guest or on the VM Host. When run in a guest, this command returns information to identify the VM Host as well as the guest.

Information can be presented in several formats. The `-M` option displays in a machine-readable format, while the `-X` option displays in the XML format.

Only superusers can execute the `hpvminfo` command.

Options

The following options can be specified only once.

The `hpvminfo` command recognizes the following command-line options and arguments:

`-v` Displays the version number of the `hpvminfo` command. The version number is displayed first, followed by the information specified by other options.

`-V` Displays detailed information about the VM Host and guests (verbose mode). For whole disks used by guests, the SCSI timeout information is displayed.

The `-V`, `-M`, and `-X` options are mutually exclusive.

`-M` Displays verbose information in a machine-readable format.

Individual fields are separated by one of three delimiters:

- The colon (:) separates each field and resource type.
- The semicolon (;) separates subfields of a resource type.
- The comma (,) separates individual items in a list of similar items.

The `-V`, `-M`, and `-X` options are mutually exclusive.

`-X` Displays verbose information in the XML format.

The `-V`, `-M`, and `-X` options are mutually exclusive.

`-S` Displays all information accessible through the supported interfaces defined in `hpvm_api_public.h`.

RETURN VALUES

The `hpvminfo` command exits with one of the following values:

0: Successful completion.

1: One or more error conditions occurred.

DIAGNOSTICS

The `hpvminfo` command displays error messages on stderr for any of the following conditions:

An invalid option is specified.

The `hpvminfo` command and Integrity VM are at different revision levels.

EXAMPLES

The following example demonstrates the command running on the VM Host.

```
# hpvminfo
hpvminfo: Running on an HPVM host.
```

The following example demonstrates the command running inside a guest.

```
# hpvminfo
hpvminfo: Running inside an HPVM guest.
```

The following example shows detailed information about the VM Host from within a guest.

```
# hpvminfo -V
hpvminfo: Running inside an HPVM guest.
Configured guest name: vm0512
Host chassis information
  Host model string                : ia64 hp server rx5670
  Host serial number               : USR4319L4J
  Host partition ident             : a7d6d186-9f74-11d7-867a-636e2282571a
  Host machine ident               : a7d6d186-9f74-11d7-867a-636e2282571a
  Host physical serial number      : USE4346Y5T
  Host physical ident              : a7d6d186-9f74-11d7-867a-636e2282571a
Host Inet information
  Hostname                         : rake
  Number of host IPv6 Addresses    : 0
  Number of host IPv4 Addresses    : 1
    IP Address                     : 1.2.3.4
Host SCSI information
  Timeout                          : 40000
```

AUTHORS

The `hpvminfo` command was developed by HP.

SEE ALSO

On the VM Host:

hpvm(5), *hpvmclone*(1M), *hpvmcollect*(1M), *hpvmconsole*(1M), *hpvmcreate*(1M), *hpvmdevinfo*(1M), *hpvmdevmgmt*(1M), *hpvmdevtranslate*(1M), *hpvmhostgdev*(1M), *hpvmhostrdev*(1M)*hpvmmigrate*(1M), *hpvmmodify*(1M), *hpvmmove_suspend*(1M), *hpvmnet*(1M), *hpvmnvram*(1M), *hpvmpubapi*(3), *hpvmremove*(1M), *hpvmresources*(5), *hpvmresume*(1M), *hpvmsar*(1M), *hpvmsg_move*(1M), *hpvmsg_package*(1M), *hpvmstart*(1M), *hpvmstatus*(1M), *hpvmstop*(1M), *hpvmsuspend*(1M), *hpvmupgrade*(1M)*p2vassist*(1M)

On the Integrity VM guest:

hpvmcollect(1M), *hpvmmgmt*(1M), *hpvmpubapi*(3)

hpvmmgmt(1M)

NAME

hpvmmgmt -- Manage dynamic data within the Integrity VM guest environment.

SYNOPSIS

`hpvmmgmt` [`-l` *type*] [`-V` | `-M` | `-X`] [`-v`]
`hpvmmgmt` [`-l` *type*] [`-t` *interval*] [`-V`] [`-v`]
`hpvmmgmt` [`-c` *num*] [`-v`]
`hpvmmgmt` [`-v`] [`-x` *keyword=value*] | [`-x` *keyword=value* [:*keyword=value*]]

DESCRIPTION

The `hpvmmgmt` command allows you to view and potentially manage dynamic guest data. To manage the data, the VM Host must have configured the guest to do so.

To obtain a display in machine-readable or XML format, use the `-M` or `-X` option.

The `hpvmmgmt` command without options displays a summary of types of Integrity VM management data that the guest can view and, if appropriate, change.

Only superusers can execute the `hpvmmgmt` command.

Options

Only the `-x` option can be specified more than once.

The `hpvmmgmt` command recognizes the following options and attributes:

`-l` *type* — Specifies the type of data about which to list more information. For the *type* parameter, enter *ram*, which lists dynamic memory values, or *cpu*, which lists dynamic cpu values.

`-t` *interval* — Specifies the interval in seconds to redisplay values. Used with *type* ram in conjunction with the `-l` option, this option allows the `hpvmmgmt` command to continuously refetch the requested type of data using the value specified for the *interval* attribute (specified in seconds) between updates. If the `-t` option is not specified, the data is only fetched once. The `hpvmmgmt` command forces the value to a maximum interval.

`-c` *num* — Specifies the number of virtual CPUs to enable on the guest. The parameter *num* must be greater than 0 and cannot exceed the number of vCPUs attached to the guest when it was started. To view the number of vCPUs available on the guest, use the `hpvmmgmt -l cpu` command.

`-v` — Displays the version number of the `hpvmmgmt` command. The version number is displayed first, followed by information specified by other options.

`-V` — Displays detailed information (verbose mode) about the virtual machines. When the `hpvmmgmt` command is run with only this option, the output displays the valid name value pairs for each type of modifiable dynamic data type known.

The `-V`, `-M`, and `-X` options are mutually exclusive.

`-M` Displays verbose attribute and resource information in machine-readable format.

Individual fields are separated by one of three delimiters:

- The colon (:) separates each field and resource type.
- The semicolon (;) separates subfields of a resource type.
- The comma (,) separates individual items in a list of similar items.

The `-V`, `-M`, and `-X` options are mutually exclusive.

`-X` Displays verbose attribute and resource information in the XML format.

The `-V`, `-M`, and `-X` options are mutually exclusive.

`-x` *`keyword=value`* Specifies virtual machine characteristics. Use the following keywords:

- `-x ram_target={0 | start |` *`amount`*`}`

  Specifies the guest RAM target, where the values are as follows:

  - 0 indicates the dynamic memory driver reduces the memory on the guest to what it determines is a comfortable minimum that does not result in guest memory being paged out.
  - `start` indicates that the guest dynamic memory size is set back to the boot time value.
  - *`amount`* is a specific target memory size for the guest.

    Each amount is expressed as an integers, optionally followed by one of these units:

    – M (megabytes)

    – G (gigabytes)

    If the letter is omitted, the default unit is megabytes.

RETURN VALUES

The `hpvmmgmt` command exits with one of the following values:

0: Successful completion.

1: One or more error conditions occurred.

DIAGNOSTICS

The `hpvmmgmt` command displays error messages on stderr for any of the following conditions:

- An invalid option is specified.
- The `hpvmmgmt` command and Integrity VM are at different revision levels.

EXAMPLES

The following example lists all the types of data that can be managed, as well as the name value pairs:

```
# hpvmmgmt
Type                Guest Control
 ======              =============
 ram                 yes
 cpus                yes
```

The following example show the information about the number of vCPUs available to the guest:

```
# hpvmmgmt -l cpu
Virtual CPUs: 2 activated, 0 deactivated, 2 total
```

The following example specifies the number of vCPUs that should be enabled on the guest:

```
# hpvmmgmt -c 1
hpvmmgmt: 1 virtual CPUs enabled, 1 disabled
```

The following example shows the detailed information about the manageable data:

```
# hpvmmgmt -l ram -V
[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
```

The following example sets the dynamic RAM target to the original boot amount:

```
# hpvmmgmt -x ram_target=0
Attempting to decrease memory from 2039 MB to 745 MB.
 Successfully began to change ram_target to 745 MB.
# hpvmmgmt -x ram_target=start
Attempting to increase memory from 929 MB to 2039 MB.
 Successfully began to change ram_target to 2039 MB.
```

AUTHORS

The `hpvmmgmt` command was developed by HP.

SEE ALSO

On the VM Host:

hpvm(5), *hpvmclone*(1M), *hpvmcollect*(1M), *hpvmconsole*(1M), *hpvmcreate*(1M), *hpvmdevinfo*(1M), *hpvmdevmgmt*(1M), *hpvmdevtranslate*(1M), *hpvmhostrdev*(1M), *hpvmhostrdev*(1M), *hpvminfo*(1M), *hpvmmigrate*(1M), *hpvmmodify*(1M), *hpvmmove_suspend*(1M), *hpvmnet*(1M), *hpvmnvram*(1M), *hpvmpubapi*(3), *hpvmremove*(1M), *hpvmresources*(5), *hpvmresume*(1M), *hpvmsar*(1M), *hpvmsg_move*(1M), *hpvmsg_package*(1M), *hpvmstart*(1M), *hpvmstatus*(1M), *hpvmstop*(1M), *hpvmsuspend*(1M), *hpvmupgrade*(1M), *p2vassist*(1M)

On the Integrity VM guest:

hpvmcollect(1M), *hpvminfo*(1M), *hpvmpubapi*(3)

hpvmmigrate(1M)

NAME

hpvmmigrate – Migrate an active guest or an offline virtual machine to a specified VM Host.

SYNOPSIS

hpvmmigrate { -P *vm-name* | -p *vm-number* } -h *target-host-alias-or-IP-addr* [-l *new-vm-label*] [-N *new-vm-name*] [-D] [-n] [-s] [-F] [-q] [-e *percent* [*:max_percent*] | -E *cycles*[*:max_cycles*]] [-c *number-vcpus*] [-r *amount*] [-m *rsrc-with-absolute-path* [-C] [-t]]... [-T] [-b] [-d]
hpvmmigrate -o { -P *vm-name* | -p *vm-number* } -h *target-host-alias-or-IP-addr* [-l *new-vm-label*] [-N *new-vm-name*] [-D] [-n] [-s] [-F] [-q] [-Q] [-w] [-Y | -y]
hpvmmigrate -o -s -h *target-host-alias-or-IP-addr*
hpvmmigrate -k { -P *vm-name* | -p *vm-number* } -h *target-host-alias-or-IP-addr*
hpvmmigrate -A { -P *vm-name* | -p *vm-number* }
hpvmmigrate -v
hpvmmigrate -H

DESCRIPTION

The `hpvmmigrate` command moves an online guest or an offline virtual machine from a source VM Host to a specified target VM Host. Virtual machine can be migrated while OFF, and active guests can be migrated while ON and running. Use the `-o` option to migrate running guests online, which involves moving the virtual machine, memory and active state for a running guest.

To move a guest from a source VM Host to a target VM Host, both hosts must be configured to allow common access to all of the required resources of the migrating guest. The guest can be either a non-running virtual machine, offline migration, or a running guest, online migration. Online migration enables a running guest and its applications to be moved from one VM Host to another without service interruption. SAN storage is assumed for Online VM Migration.

The resources that are defined in the virtual machine's configuration are checked to determine whether the migrated guest can boot on the target VM Host. If there is a problem, it is reported, and the guest is not migrated. You can specify the `-F` option (force) to suppress the errors and force the guest migration to the target VM Host.

△ **CAUTION:** Use the `-F` option with caution, because some errors can prevent a virtual machine from working properly on the target VM Host.

Only superusers can execute the `hpvmmigrate` command. In addition, the migration of a guest is controlled by a set of secure remote operations that must be enabled on both systems. SSH security must be set up so that root can use the equivalent of the `ssh` and `scp` commands between the source and the target VM Hosts without requiring passwords. The easiest way to do this is to use the `secsetup` script provided by Integrity VM. Otherwise, see the HP-UX `ssh-keygen` command.

If you specify the `-D` option, the virtual machine is deleted from the source VM Host system after it is successfully migrated to the target VM Host. By default, however, Integrity VM retains the virtual machine and marks it `Not Runnable (NR)` on the source VM Host after it is migrated successfully to the target VM Host. Specifying a subsequent `hpvmstatus` command on the source VM Host after you have migrated the virtual machine will show the state of the virtual machine as `Off (NR)`, that is Not Runnable on the source VM Host, because the virtual machine is now runnable on the target VM Host.

This mechanism allows the same virtual machine to exist on multiple VM Hosts, while still preventing accidental booting of the same virtual machine on multiple hosts simultaneously. Each virtual machine must be runnable on only one VM Host at any given time to prevent the possibility of two virtual machines using the same resources at the same time. Use the `hpvmmodify` command, if necessary, to mark the guest `Runnable` on only one single VM Host, and `Not Runnable` on all other VM Hosts that have the virtual machine configured.

When `hpvmmigrate` sets the guest to `Not Runnable` state (`runnable_status=disabled`) , it also sets `modify_status=disabled` and `visible_status=disabled`. Likewise, when `hpvmmigrate` sets the guest to a `Runnable` state, it also sets `modify_status=enabled` and `visible_status=enabled`.

NOTE: Mark a migrated guest as `Runnable` only in rare circumstances and with great care. Inappropriate use can cause disk corruption if a guest is booted on more than one host at the same time.

If you migrate a virtual machine that is being managed by ID-VSE, use Capacity Advisor to collect utilization data before you migrate the virtual machine. Otherwise, the utilization information for the VM Host prior to the migration is lost.

Cloning or modifying a guest during online migration is not allowed.

Online migration of a guest from a V4.3 VM Host back to a V4.2 VM Host is not supported with one exception. If a guest was originally booted on a V4.2 VM Host and then migrated online to a V4.3 VM Host, you can perform an online migrate on that guest back to a V4.2 VM Host.

Options

The `hpvmmigrate` command recognizes the following command-line options and arguments:

`-A` — Aborts an active guest migration.

`-b` — For offline migrations, causes the `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 exist or must be copied.

`-c` *`number-vcpus`* — For offline migrations, specifies the number of virtual CPUs this virtual machine sees at boot time.

`-C` — For offline migrations, physically copies the single storage device specified with the `-m` option to the target VM Host 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 guest to the target VM Host. Consider migrating the guest online by using the `-o` option.

`-D` — Deletes the virtual machine from the source VM host after migrating the virtual machine to the target VM Host. If not specified, the virtual machine is marked `Not Runnable` on the source VM Host after migration.

`-e` *`percent[:max-percent]`*

For offline migrations, specifies the percentage of CPU resources to which each of the virtual machine's 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` *`cycles[:max-cycles]`*

For offline migration, 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 guest, whether or not there are resource validation errors (such as resource conflict, resource nonexistence, and so forth). Use the `-F` option only rarely and with **caution**. This option ignores all resource validation errors, including oversubscribing of resources. It is important to note that these errors can prevent the virtual machine from booting on the target VM Host. Any validation errors are logged in the Integrity VM command log.

The `-F` option is deprecated in Integrity Virtual Machines commands.

`-h` *`target-host-alisas-or-IP-addr`*

Specifies the host alias or IP address of a network connection to the target VM Host machine to which the virtual machine is being migrated. The target machine must be a valid VM Host and must be accessible by the source VM Host. 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 VM Host.

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 VM Migration does not check to ensure that 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 the usage of the `hpvmmigrate` command.
`-k`	Creates the virtual machine on the target VM Host and marks it `Not Runnable`, but does not change the virtual machine on the source VM Host.
`-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` 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 migration, specifies a resource of a virtual machine for copying, translation, and so on. For a network specification, use the following syntax: **`-m network:lan::vswitch:OLD:network:lan::vswitch:NEW`** This option can be specified more than once. For information about specifying virtual machine storage and network resources for guests, 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 name can consist of up to 256 alphanumeric characters, including A-Z, a-z, 0-9, the dash (—), the underscore (_), and period (.). The virtual machine name cannot start with a dash (—). If the virtual machine name exists on the target VM Host, 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 VM Migration of an active guest. To be compatible for online migrations, both the source and the target host must have the same processor family (as reported by the `machinfo` command). To maintain active guest network connectivity, a vswitch with the same name and connected to the same subnet must be configured on the target VM Host. Also, only whole disk backing storage consisting of SAN LUNs and null backing store DVD devices are supported for online guest migration storage.
`-p` *`source-vm-number`*	Specifies the unique number of the virtual machine to be migrated. The *`vm-number`* is reported with 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 integers, optionally followed by one of these letter: • M (megabytes) • G (gigabytes) If the letter is omitted, the default unit is megabytes.
`-s`	Indicates that the migration should not occur, but the `hpvmmigrate` command should check whether or not the migration is possible. Because guests and their VM 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 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 VM Hosts. 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 number of the `hpvmmigrate` command.
`-w`	For online migrations, bypasses all vswitch connectivity checks. Use the `-w` option only of you are certain that the source and target vswitches are connected to the same subnet; otherwise, your active guest will lose network connectivity after migrating.
`-Y`	Suppresses encryption negotiation and sends guest memory data in the clear. You can modify the default behavior by setting variables in the `/etc/rc.config.d/hpvmconf` file using the `ch_rc` command. The variable `HPVM_OVMM_ENCRYPT_BY_DEFAULT` controls whether any attempt at encryption negotiation is done. The default setting of 1 attempts an encryption negotiation. The variable `HPVM_OVMM_ENCRYPT_ALGORITHM` can be changed from its default value of *`aes-128-cbc`* to *`aes-256-cbc`* for sites with stronger security concerns. If you do not specify either the `-Y` or the `-y` option, and you set the `HPVM_OVMM_ENCRYPT_BY DEFAULT` variable to its default setting of 1, the *`target-host-alisas-or-IP-addr`* name given to the

-h option for an online migration is examined to see whether it matches the private network convention for guest migration. If the name given ends in –hpvm-migr, or if such a name was chosen because of finding a match in the /etc/hosts file, encryption is suppressed for the private network. The default is to encrypt guest data for all other connections. Explicit use of the -Y or -y option, as well as changing the HPVM_OVMM_ENCRYPT_BY_DEFAULT variable to 0, changes this default behavior.

-y

Requires encryption negotiation and sends guest memory data with protection.

You can modify the default behavior by setting variables in the /etc/rc.config.d/hpvmconf file using the ch_rc command. The variable HPVM_OVMM_ENCRYPT_BY_DEFAULT controls whether any attempt at encryption negotiation is done. The default setting of 1 attempts an encryption negotiation. The variable HPVM_OVMM_ENCRYPT_ALGORITHM can be changed from its default value of *aes-128-cbc* to *aes-256-cbc* for sites with stronger security concerns.

If you do not specify either the -Y or the -y option, and you set the HPVM_OVMM_ENCRYPT_BY DEFAULT variable to its default setting of 1, the *target-host-alisas-or-IP-addr* name given to the -h option for an online migration is examined to see whether it matches the private network convention for guest migration. If the name given ends in –hpvm-migr, or if such a name was chosen because of finding a match in the /etc/hosts file, encryption is suppressed for the private network. The default is to encrypt guest data for all other connections. Explicit use of the -Y or -y option, as well as changing the HPVM_OVMM_ENCRYPT_BY_DEFAULT variable to 0, changes this default behavior.

RETURN VALUES

The hpvmmigrate command exits with one of the following values:

0: Successful completion.

1: One or more error conditions occurred.

2: One or more operational failures occurred.

DIAGNOSTICS

The hpvmmigrate command displays error messages on stderr for any of the following conditions:

- An invalid option is specified.
- An invalid value is specified for an option.
- A value is omitted for an argument that requires one, or a value is supplied for an argument that does not take one.

- The *source-vm-name* or *source-vm-number* attribute does not exist, cannot be accessed, is not a virtual machine, or is corrupt.
- SSH does not seem to be properly configured between the source and the target VM Hosts.
- The `hpvmmigrate` command and Integrity Virtual Machines are at different revision levels.
- The virtual machine already exists on the target VM Host and is `Runnable`, or the virtual machine name exists on the target but the configuration has a different UUID.
- The guest is running, and online migration is not being used. Or, the virtual machine is not running, and online migration is attempted.
- Invalid virtual machine configuration.
- Remote execution error.
- Virtual machine resource validation error.
- A remote vswitch is not present or not active.
- The version of the `hpvmmigrate` command is incompatible with the version on the target VM Host.

EXAMPLES

Display the version number of the `hpvmmigrate` command.

```
# hpvmmigrate -v
hpvmmigrate: Version B.04.30
```

Migrate the offline virtual machine named host1, to the host abc.def.com.

```
# hpvmmigrate -P host1 -h abc.def.com
```

Migrate the active guest named `drop` to the VM Host named `bucket` through the private network alias `bucket-hpvm-migr`.

```
# hpvmmigrate -o -P drop -h bucket-hpvm-migr
```

OBSOLESCENCE

The `-F` option is deprecated in Integrity Virtual Machines commands. This flag might be removed in a future release of Integrity Virtual Machines. HP recommends use of the command with the `-s` option, then resolving any errors or warnings that result before executing the command without the `-s` option and without the `-F` option.

AUTHORS

The `hpvmmigrate` command was developed by HP.

SEE ALSO

On the VM Host:

hpvm(5), *hpvmclone*(1M), *hpvmcollect*(1M), *hpvmconsole*(1M), *hpvmcreate*(1M), *hpvmdevinfo*(1M), *hpvmdevmgmt*(1M), *hpvmdevtranslate*(1M), *hpvmhostgdev*(1M), *hpvmhostrdev*(1M), *hpvminfo*(1M), *hpvmmodify*(1M), *hpvmmove_suspend*(1M), *hpvmnet*(1M), *hpvmnvram*(1M), *hpvmpubapi*(3), *hpvmremove*(1M), *hpvmresources*(5), *hpvmresume*(1M), *hpvmsar*(1M), *hpvmsg_move*(1M), *hpvmsg_package*(1M), *hpvmstart*(1M), *hpvmstatus*(1M), *hpvmstop*(1M), *hpvmsuspend*(1M), *hpvmupgrade*(1M), *p2vassist*(1M)

On the Integrity VM guest:

hpvmcollect(1M), *hpvminfo*(1M), *hpvmmgmt*(1M), *hpvmpubapi*(3)

hpvmmodify(1M)

NAME

`hpvmmodify` -- Rename a virtual machine or modify the attributes of a virtual machine.

SYNOPSIS

hpvmmodify { -P *vm-name* | -p *vm-number* } [-K *console_IP_Addr*] [-L *console_IP_Addr_Netmask*] [-F | -s] [-N *new-vm-name*]
[-l *vm-label*] [-B *start-attr*] [-O *os-type* [:*version*]]
[-c *number-vcpus*] [-r *amount*]
[-e *percent[:max_percent]* | -E *cycles[:max_cycles]*]
[-g -*group*]... [-g [+]*group*[: {*admin* | *oper*}]]...
[-u -*user*]... [-u [+]*user:*[{*admin* | *oper*}]]...
[-a *rsrc*] [-m *rsrc*] [-d *rsrc*] [-i SG | -i *SG-pkgname* | -i GWLM | -i *SG-pkgname* , GWLM | -i NONE] [{-j *0* | *1* }]
[[-x *name=value* [:*name=value*]]]
hpvmmodify -A { -P *vm-name* | -p *vm-number* } [-F] [-l *vm-label*] [-B *start-attr*] [-e *percent[:max_percent]* | -E *cycles[:max_cycles]*] [
[-a *rsrc*] ... [-m *rsrc*] ... [-d *rsrc*] ...]
[[-x *name=value* [:*name=value*]]

DESCRIPTION

The `hpvmmodify` command modifies the attributes and resources of the specified virtual machine.

All attributes and resources can be changed statically, so that changes take effect when the virtual machine is next restarted.

Some attributes and resources can also be changed dynamically. Dynamic changes take effect immediately and remain in effect when the virtual machine is next started, unless you explicitly specify otherwise with the `-A` option.

Only specified attributes or resources are changed. All others retain their original values.

When changing from -E cycles to -e percent on a guest that previously had a maximum configured, if you do not specify a new maximum, then the default maximum value is chosen. This same action occurs when changing from -e percent to -E cycles.

Virtual machine modification is designed for flexibility and assumes that all existing virtual machines are not necessarily running at the same time or on the current VM Host. Therefore, the `hpvmmodify` command allows virtual machines to be modified in such a way that they cannot boot on the current system. A guest configuration receives a warning at modification and an error at start time, for any issues that might prevent it from starting on the current VM Host. To verify a particular configuration for the current VM Host without actually modifying the guest, use the `-s` option.

NOTE: Modifications are not allowed during online VM migration.

When a new device is added to a guest without specifying the *`bus/target/lun`* in the resource parameter, a default order of *`bus/target/lun`* allocation is followed. This ordering fills 0-14 targets on the lowest numbered bus before proceeding to the next higher-numbered bus. This default ordering is the same for both AVIO and VIO guest adapters.

NOTE: Targets higher than 14 are supported only on AVIO adapters.

Only a superuser can execute the `hpvmmodify` command.

Options

The `hpvmmodify` command recognizes the following command-line options and arguments.

`-P` *vm-name*

Specifies the name of the virtual machine to be modified.

You must specify either the `-P` or the `-p` option.

`-p` *vm-number*

Specifies the number of the virtual machine to be modified. The `vm-number` is displayed by the `hpvmstatus` command.

You must specify either the `-P` 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. If an address is supplied, then the option must also be specified. The virtual iLO Remote Console will be accessible with telnet or ssh using this IP address. Accessing the virtual iLO Remote Console will automatically run the command `hpvmconsole` for this guest.

To facilitate virtual iLO Remote Console access, an IP alias interface will be created using this IP address and the netmask specified in the option. If the IP address specified is 0, the IP alias interface for this guest will be removed.

`-L` *console_IP_Addr_Netmask*

This option 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.

`-A`

Specifies that the addition, modification, or deletion of resources is done to an active virtual machine's configuration file. These modifications will be effective until the virtual machine is rebooted. Not all modifications can be done to an active virtual machine; in this case, an error message indicates the changes that require the virtual machine to be rebooted.

NOTE: Changing the VM name of an active guest fails and requires a reboot.

`-e` *percent[:maximum_percent]*

Specifies the percentage of CPU resources to which each of the guest's virtual CPUs is entitled and optionally specific the maximum percent or entitlement cap. The default cap is 100%.

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 will be allocated the minimum percentage of 5%. The default is 10%.

The maximum entitlement may not exceed 100 percent, and may not be less than the minimum. Each group has a default maximum setting of 100 percent.

In addition to the guest calculation, 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.

NOTE: If you change from `-e` *`percent`* to `-E` *`cycles`* on a guest that previously had a maximum configured and do not specify a maximum, then the default maximum value is chosen.

`-E` *`cycles[:maximum_cycles]`*

Specifies the virtual machine's CPU entitlement in CPU cycles and optionally specifies the maximum cycles or entitlement cap. The default entitlement cap is the total maximum cycles supported by the physical CPU.

The cycles are expressed as an integer, followed by one of the following letters to specify units:

- M (megahertz)
- G (gigahertz)

  If no letter is specified, the default unit is megahertz.

The `-e` and the `-E` options are mutually exclusive.

NOTE: If you change from `-E` *`cycles`* to `-e` *`percent`* on a guest that previously had a maximum configured and do not specify a maximum, then the default maximum value is chosen.

`-F`

Suppresses all resource conflict checks and associated warning messages (force mode). Force mode is provided for scripts and other noninteractive applications. Note that you will receive no notification of potential resource problems for a virtual machine modified with the `-F` option.

The `-F` and `-s` options are mutually exclusive.

The `-F` option is deprecated in Integrity Virtual Machines commands.

`-i` *`package-name`*

Specifies whether the virtual machine is managed by Serviceguard or gWLM (or both). For the argument, specify the Serviceguard package name or `gWLM`, both, or NONE. This option is used by Integrity VM software; do not use this option without explicit instruction from HP.

`-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). This option is used by Integrity VM software; do not use this option without explicit instruction from HP.

`-l` *`vm-label`*

Specifies a descriptive label for the virtual machine, which can be useful in identifying a specific virtual machine in the `hpvmstatus` 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 specifiy white space, the label must be quoted (" ").

-B *start-attr*

Specifies the startup behavior of the virtual machine. The `start_attr` argument 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. This is the default.

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

If the attribute is set to `manual`, the virtual machine does not start automatically when Integrity VM is initialized on the VM Host. 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.

[-O *os-type*[*:version*]]

Specifies the type and version of the operating system running on the virtual machine. The response affects the default selection of certain virtual machine attributes, such as amount of memory and CPU power. The *os_type* is one of the following: `HPUX` or `OPENVMS`. This parameter is not case sensitive. The *version* is specific to the operating system type.

The version specifies a descriptive text string of the version of the operating system. The version string can consist of up to 256 alphanumeric characters, including A-Z, a-z, 0-9, the dash (—), the underscore (_), and the period (.). To specify white space, *version* must be quoted (" ").

NOTE: You can specify the *os_type* here manually, but when the guest boots, the *os_type* is set to whatever operating system was last booted. The operating system *version* is only set manually with the `-O` option.

-a *rsrc*

Adds an I/O resource to a virtual machine.

NOTE: When assigning a null device to a resource on a virtual machine, the file name serves as a placeholder. Therefore, if the file does not exist, you do not receive an error. For example, in the following command, if the file `XXX.iso` does not exist, no error is given.

```
# hpvmmodify -P vm1 -a
disk:scsi::null:/opt/XXX.iso
```

The resource is specified as described in *hpvmresources*(5).

This option can be specified more than once.

`-d` *`rsrc`*

Deletes an I/O resource from a virtual machine. The resource is specified as described in *hpvmresources*(5). The physical device portion of the `rsrc` is optional.

This option can be specified more than once.

`-m` *`rsrc`*

Modifies an existing I/O resource for a virtual machine. The resource is specified as described *in hpvmresources*(5). You must specify the hardware address of the device to modify. The physical device portion of the `rsrc` specifies a new physical device to replace the one in use. To move from VIO to AVIO, use the following command:

```
# hpvmmodify -P guest1 -m hba:avio_stor:0,5
```

This option can be specified more than once.

The `rsrc` specifies the mapping of a guest virtual device to a VM Host backing store. Integrity VM guests access virtual devices that are instantiated on physical entities managed by the VM Host. These physical entities (for example, network cards, files, logical volumes, and so forth) are collectively referred to as backing stores.

Integrity VM recognizes the following types of guest virtual devices:

- Virtual disks, which can be backed by files in a VM Host file system, by logical volumes or by whole disks.
- Virtual DVDs, which can be backed by files in a VM Host file system or by the physical DVD drive.
- Virtual network devices, which are created through the `hpvmnet` command and backed by physical LAN cards. For more information about virtual network devices, see *hpvmnet*(1M).

For information about specifying storage and network resources for guests, see the *hpvmresources* manpage.

Integrity VM Version 4.3 supports the usage of both legacy and agile devices in guest configurations. Over time, you should migrate from using legacy device names to agile device names to simplify the management of devices. You can use the `hpvmmodify -P` *`guest`* `-m` *`rsrc`* command to change the existing legacy device to its agile device equivalent. The `ioscan -m dsk` command displays the mapping of the agile devices to legacy devices for the specific device special file, `dsf`.

NOTE: The `rmsf -L` command should not be executed on an Integrity VM server until all guest devices names have been changed to agile device names.

The following commands are helpful for reconfiguring guests to use agile device names:

- `insf`

  By default, creates both persistent and legacy DSFs for new devices.

- `insf -L`

  Restores legacy DSFs and legacy configuration information.

- `rmsf -L`

  Aids in migration by removing all legacy DSFs and legacy configuration information.

- `ioscan —m dsf`

  Maps persistent DSFs to their equivalent legacy DSFs and vice versa.

- `ioscan -N`

  Prints persistent DSFs when used with the `-n` option to list DSFs.

- `io_redirect_dsf`

  Associates a new disk with an existing set of DSFs. Used when replacing an internal disk or a disk in a JBOD.

- `hpvmmodify -P guest —m rsrc`

  Used to modify backing devices for guest devices.

`-N` *`new-vm-name`*

Specifies the new name for the virtual machine being modified, assuming no virtual machine with that name already exists. The name can consist of up to 256 alphanumeric characters, including A-Z, a-z, 0-9, the dash (—), the underscore (_), and period (.). The virtual machine name cannot start with a dash (-).

The virtual machine name can be changed only by using the `-N` option.

The name change takes effect immediately.

NOTE: Do not change the VM name while the virtual machine is running.

`-c` *`number-vcpus`*

Specifies the number of virtual CPUs and vCPUs his virtual machine sees at boot time along with the range that the vCPUs must be within. The specification of the range is optional. When the range is not specified, a default range is used consisting of a minimum of one vCPU and a maximum equal to the maximum vCPUs supported for the specific guest operating system type.

Integrity VM Version 4.3 supports up to 16 vCPUS for all guest types. For example, to set a guest with four vCPUs with no range, use the following command:

`# hpvmmodify -P guestname -c 4`

To set a quest with four vCPUs with a range of two to six vCPUs, use the following command:

`# hpvmmodify -P guestname -c 4:2:6`

`-r` *amount*

Specifies the amount of memory available to this virtual machine.

The size is expressed as an integer, optionally followed by one of these units:

- M (megabytes)
- G (gigabytes)

  If the letter is omitted, the default unit is megabytes.

`-g [+]group[:{admin | oper}]`

Adds (+ or unspecified) or removes (-) a group authorization. A VM Host user account that is a member of an authorized group can use the `hpvmconsole` command to manage this guest. The `{admin|oper}` argument specifies the privilege level available at the `hpvmconsole` — either `admin` or `oper` (the default). Do not specify the privilege level when you are removing a group.

This option can be specified more than once.

`-u [+]`*user*`[:{admin | oper}]`

Adds (+ or unspecified) or removes (-) a user authorization. An authorized VM Host user account can use the `hpvmconsole` command to manage this guest. The `{admin|oper}` argument specifies the privilege level available at the `hpvmconsole`, either `admin` or `oper` (the default). Do not specify the privilege level when you are removing a user.

This option can be specified more than once.

`-s`

Verifies the virtual machine configuration and returns warnings or errors, but suppresses the action that the command would normally perform. This option starts resource checking for the specified virtual machine configuration without actually modifying the virtual machine. In typical cases, where `-s` is not specified, the virtual machine is modified even if resource warnings occur.

The `-F` and `-s` options are mutually exclusive.

`-x` *name=value*

Specifies whether the virtual machine uses dynamic memory and the values associated with it by including the following keywords:

- `-x dynamic_memory_control={0|1}`
- `-x ram_dyn_type={none|any|driver}`
- `-x ram_dyn_min=`*amount*

- `-x ram_dyn_max=`*`amount`*
- `-x ram_dyn_target_start=`*`amount`*

  Specifies whether the virtual machine's dynamic memory settings are automatically adjusted. The `ram_dyn_entitlement` and `amr_enable` options must be set to enable adjustments.

- `-x ram_dyn_entitlement=`*`amount`*

  Specifies the minimum guaranteed amount of memory.

- `-x amr_enable={0|1}`

  Enables or disables AMR monitoring for a guest, where 1 enables and 0 disables. The monitor (amr daemon) adjusts the guest size and take its entitlement into account.

- `- x amr_chunk_size=`*`amount`*

  Specifies the increment amount of changes in memory size (default is 256 MB). Larger values result in faster memory size growth.

- `-x ram_target={0 |start | `*`amount`*`}`

  Specifies the guest RAM target, where the values are:

  - 0 indicates the dynamic memory driver reduces the memory on the guest to what it determines is a comfortable minimum that does not result in guest memory being paged out.
  - `start` indicates the guest dynamic memory size set back to the boot time value.
  - *`amount`* is a specific target memory size for the guest.

    Each amount value is expressed as an integer, optionally followed by one of these units:

    - M (megabytes)
    - G (gigabytes)

    If the letter is omitted, the default unit is megabytes.

- `-x suspend={enable | disable}`

  Enables or disables the ability of the guest to be suspended. If the disable is specified, the guest cannot be suspended.

- `-x suspend_file=delete`

  Deletes the suspend file.

- `-x guest_uuid=uuid_string`

  To be used only if directed by HP Customer Support.

- `-x tunables={name=value[,name=value, ...]}]`

  To be used only if directed by HP Customer Support.

- `-x runnable_status={disabled|enabled}`

  Specifying `runnable_status=disabled` prevents the guest from being started. It also prevents other guests from booting if they share devices with a disabled guest whose devices are marked SHARE=NO. Specifying enabled allows guest to be started again.

  Set `-x runnable_status=disabled` only with the `hpvmmodify` command. An error message is printed if it is specified with the `hpvmcreate` or the `hpvmclone` commands. In addition, specify `-x runnable_status=disabled` only if the guest is not being managed by Serviceguard. If the `hpvmcreate`, `hpvmmodify`, or the `hpvmclone` command specifies -i SG, -i GWLM,SG, or `-i` SG_packagename, an error message is printed if an attempt is made to disable the guest.

- `-x not_runnable_reason="string"`

  Displays the specified string in a message printed by the `hpvmstatus -V` command when the guest is disabled, or when `-F` is specified. If no reason is given for disabling the guest, a default string is used. This parameter is ignored when `-x runnable=enabled` is specified on the command line. An error message is displayed if this parameter is specified by itself and the guest is enabled.

- `-x graceful_stop_timeout=`{0 | *`number`*}

  Specifies the amount of time in seconds to allow a graceful stop when `hpvmstop –g` is specified. The default is 30 seconds.

  NOTE: If the graceful stop does not finish within the time frame specified, a hard stop is then executed.

- `-x modify_status=` [*`enabled|disabled`*]

  Allows you to enable or disable the ability to modify the status of a virtual machine.

- `-x visible_status=` [*`enabled|disabled`*]

  Allows you to set whether the virtual machine is visible to graphical tools like Logical Server Manager.

- `-x register_status=` [*`enabled|disabled`*]

  Allows the alias `register_status` to equate to `modify_status`, `visible_status`, and `runnable_status`. When a guest is registered with VMM, the following attributes are set:

  - `-x runnable_status=enabled`
  - `-x modify_status=enabled`
  - `-x visible_status=enabled`

VMM and LSM ensure that a virtual machine is not registered (and therefore runnable) on more than one VM Host at a time.

NOTE: HP does not recommend using `-x runnable_status`, `-x modify_status`, or `-x visible_status` except with extreme caution. Integrity VM ensures that the VM is runnable only on one VM Host at a time. Marking a VM runnable on more than one VM Host can lead to accidentally booting the VM on more than one VM Host.

If `modify_status` is disabled, the VM is most likely running on another VM Host. Any modification made to this VM's configuration will be lost when it is migrated back to this VM Host.

You can enable or disable `visible_status`; however, when a VM has the `visible_status` set to disabled, the graphical tools will not display the VM.

HP does not recommend using `-x register_status`. Integrity VM commands ensure that the VM is registered only on one VM Host at a time. Registering a VM on more than one VM Host can lead to accidentally booting the VM on more than one VM Host, which could cause inconsistencies with the display of graphical tools.

- `-x sched_preference=`{*`none`*|*`cell`*|*`ilm`*}

  where:

  - *`none`* — The default preference. If your application is predominantly CPU bound, specifying either *`ilm`* or *`cell`* will perform the same.
  - *`cell`* — The cell with the most CPU and memory space is chosen. When that guest is active, the scheduler then optimizes where the guest runs, so that it can be closest to its memory.
  - *`ilm`* — Indicates that guests that are larger than any single cell and contain highly threaded applications.
- `-x migrate_copy_phase_timeout=` {*`number of seconds`*} — Specifies the maximum number of seconds the online migration spends during the full-copy stage. The default is 0 seconds.
- `-x migrate_frozen_phase_timeout=` {*`number of seconds`*} — Specifies the maximum number of seconds the migration spends during the freezing stage. The default is 60 seconds.
- `-x migrate_init_phase_timeout=`{*`number of seconds`*} — Specifies the maximum number of seconds the online migration spends during the initialize phase of the migration. The default is 10 seconds.

- `-x migrate_io_quiesce_phase_timeout=` {*number of seconds*} — Specifies the maximum number of seconds the migration spends during the quiesce stage. The default is 15 seconds.
- `-x online_migration=` {*enabled* | *disabled*}

RETURN VALUES

The `hpvmmodify` command exits with one of the following values:

0: Successful completion.

1: One or more error conditions occurred.

DIAGNOSTICS

The `hpvmmodify` command displays error messages on stderr for any of the following conditions:

- An invalid option is specified.
- An invalid value is specified for an option, or a value is omitted.
- The *vm_name* or *vm_number* does not exist, cannot be accessed, is not a virtual machine, or is corrupt.
- The *new_vm_name* already exists.
- One or more options other than `-a`, `-m`, `-d`, `-g` or `-u` have been specified more than once.
- The same resource was allocated more than once.
- A resource allocated to another virtual machine was specified, and the force flag (`-F`) was not used.
- A resource exceeded an available resource limit, and the force flag (`-F`) was not used.
- A value was omitted for an argument that requires one, or a value was supplied for an argument that does not take one.
- For the modified (`-m`) or delete (`-d`) options, the specified resource is not presently assigned to the *vm_name*.
- The `hpvmmodify` command and Integrity VM are at different revision levels.

Using a colon (:), semicolon (;), or comma (,) when entering device names causes the machine-readable format of `hpvmstatus` to be misaligned.

EXAMPLES

Change the name of the virtual machine called myguest1 to myguest2:

```
# hpvmmodify -P myguest1 -N myguest2
```

Set the autoboot attribute for the virtual machine myguest1:

```
# hpvmmodify -P myguest1 -B auto
```

Add a new virtual DVD backed by a file to virtual machine myguest2:

```
# hpvmmodify -P myguest2 -a dvd:scsi::file:/var/opt/myguest.file
```

Change the virtual disk with hardware address 0,0,4 to a different physical device, `/dev/rdisk/disk1`:

```
# hpvmmodify -P myguest2x -m disk:scsi:0,0,4:disk:/dev/rdisk/disk1
```

Change the network device at hardware address 0,2 to a different vswitch, (called myswitch), thereby preserving its original virtual MAC address:

```
# hpvmmodify -P myguest2 -m network:avio_lan:0,2,1a-01-5a-8e-99-fa:vswitch:myswitch
```

Delete the virtual disk at hardware address 0,0,2 from the virtual machine myguest2:

```
# hpvmmodify -P myguest2 -d disk:scsi:0,0,2
```

Delete the network device at hardware address 0,1 from the virtual machine myguest2:

```
# hpvmmodify -P myguest2 -d network:avio_lan:0,1
```

Cap myguest2 with a the CPU entitlement of 20% with a minimum of 10%:

```
# hpvmmodify -P myguest2 -e 10:20
```

Temporarily change the CPU entitlement to 50% until virtual machine myguest2 is rebooted:

```
# hpvmmodify -A -P myguest2 -e 50
```

The following are sample warning messages returned when the `hpvmmodify` command is executed with various configuration problems on the guest myguest1:

```
HPVM guest myguest1 configuration problems:
Warning 1: Guest needs more vcpus than server supports.
Warning 2: Insufficient free memory for guest.
Warning 3: Insufficient swap resource for guest.
Warning 4: Insufficient cpu resource for guest.
Warning 5 on item /dev/rdisk/disk0: Device file '/dev/rdisk/disk0' in use by another guest.
Warning 6 on item /dev/vg00/rswap: Device file '/dev/vg00/rswap' in use by server.
Warning 7 on item /dev/rdisk/disk3 backing device does not exist.
Warning 8 on item /dev/rdisk/disk4: Device file '/dev/rdisk/disk4' in use by another guest.
Warning 9 on item hostnet: MAC address in use for switch hostnet.
Warning 10 on item offnet: Vswitch offnet is not active.
Warning 11 on item badnet: 'badnet' backing device does not exist.

These problems will prevent HPVM guest myguest1 from booting.
```

The following example shows how to enable dynamic memory control for the guest myguest1:

```
# hpvmmodify -P myguest1 -x dynamic_memory_control=1
```

OBSOLESCENCE

The `-F` option is deprecated in Integrity Virtual Machines commands. This flag might be removed in a future release of Integrity Virtual Machines. HP recommends use of the command with the `-s` option, then resolving any errors or warnings that result before executing the command without the `-s` option and without the `-F` option.

AUTHORS

The `hpvmmodify` command was developed by HP.

SEE ALSO

On the VM Host:

hpvm(5), *hpvmclone*(1M), *hpvmcollect*(1M), *hpvmconsole*(1M), *hpvmcreate*(1M), *hpvmdevinfo*(1M), *hpvmdevmgmt*(1M), *hpvmdevtranslate*(1M), *hpvmhostgdev*(1M), *hpvmhostrdev*(1M), *hpvminfo*(1M), *hpvmmigrate*(1M), *hpvmmove_suspend*(1M), *hpvmnet*(1M), *hpvmnvram*(1M), *hpvmpubapi*(3), *hpvmremove*(1M), *hpvmresources*(5), *hpvmresume*(1M), *hpvmsar*(1M), *hpvmsg_move*(1M), *hpvmsg_package*(1M), *hpvmstart*(1M), *hpvmstatus*(1M), *hpvmstop*(1M), *hpvmsuspend*(1M), *hpvmupgrade*(1M)*p2vassist*(1M)

On the Integrity VM guest:

hpvmcollect(1M), *hpvminfo*(1M), *hpvmmgmt*(1M), *hpvmpubapi*(3)

hpvmmove_suspend(1M)

NAME

hpvmmove_suspend -- Move suspend files to a different directory.

SYNOPSIS

`hpvmmove_suspend` { `-d` | `-D` *`directory`* } [`-q`]

DESCRIPTION

The `hpvmmove_suspend` command changes the directory where suspend files are stored and moves all existing suspend files to the new directory. Suspend files can be very large, up to the size of guest memory for each suspended guest. If there is insufficient space in `/var` to accommodate them, this command can be used to move them to a larger disk.

Only superusers can execute the `hpvmmove_suspend` command.

Options

The `hpvmmove_suspend` command recognizes the following command-line options and arguments:

`-d`
: Changes the directory to the default, `which is /var/opt/hpvm/guests/`.

  You must specify either the `-d` option or the -`D` option.

`-D` *`directory`*
: Changes the directory to the specified directory.

  You must specify either the `-d` option or the `-D` option.

`-q`
: Displays fewer informative messages. Some potential error conditions are still reported.

RETURN VALUES

The `hpvmmove_suspend` command exits with one of the following values:

0: Successful completion.

1: One or more error conditions occurred when parsing the command.

2: One or more error conditions occurred during the suspension.

DIAGNOSTICS

The `hpvmmove_suspend` command displays error messages on stderr for any of the following conditions:

- An invalid option is specified.
- An invalid value is specified for an option
- A value is omitted for an argument that requires one, or a value is supplied for an argument that does not take one.
- The new directory is not an absolute path.
- The new directory does not exist and cannot be created.
- There is not enough disk space available in the new directory.

EXAMPLES

Display the version number of the `hpvmmove_suspend` command.

```
# hpvmmove_suspend -D /large-disk/hpvm-suspend/
```

Use the new directory for all suspend files.

AUTHOR

The `hpvmmove_suspend` command was developed by HP.

SEE ALSO

On the VM Host:

hpvm(5), *hpvmclone*(1M), *hpvmcollect*(1M), *hpvmconsole*(1M), *hpvmcreate*(1M), *hpvmdevinfo*(1M),*hpvmdevmgmt*(1M), *hpvmdevtranslate*(1M), *hpvmhostgdev*(1M), *hpvmhostrdev*(1M)*hpvminfo*(1M), *hpvmmigrate*(1M), *hpvmmodify*(1M), *hpvmmove_suspend*(1M), *hpvmnet*(1M), *hpvmnvram*(1M), *hpvmpubapi*(3), *hpvmremove*(1M), *hpvmresources*(5), *hpvmresume*(1M), *hpvmsar*(1M), *hpvmsg_move*(1M), *hpvmsg_package*(1M), *hpvmstart*(1M), *hpvmstatus*(1M), *hpvmstop*(1M), *hpvmsuspend*(1M), *hpvmupgrade*(1M)*p2vassist*(1M)

On the Integrity VM guest:

hpvmcollect(1M), *hpvminfo*(1M), *hpvmmgmt*(1M), *hpvmpubapi*(3)

hpvmnet(1M)

NAME

hpvmnet – Create and control an Integrity Virtual Machines virtual network switch (vswitch).

SYNOPSIS

hpvmnet [-S *vswitch-name* | -s *vswitch-number*] [-X | -M | -V] [-v]
hpvmnet -c -S *vswitch-name* [-n *nic-id*]
hpvmnet -C [-S *vswitch-name* | -s *vswitch-number*] -n *nic-id* [-N *new-vswitch-name*]
hpvmnet -d [-S *vswitch-name* | -s *vswitch-number*] [-F] [-Q]
hpvmnet -b [-S *vswitch-name* | -s *vswitch_number*]
hpvmnet -h [-S *vswitch-name* | -s *vswitch_number*] [-F] [-Q]
hpvmnet -r [-S *vswitch-name* | -s *vswitch-number*] [-F] [-Q]
hpvmnet { -S *vswitch_name* | -s *vswitch-number* } -u portid:*portnum[,...]*:vlanid: { *vlanid* ... | none }
hpvmnet { -S *vswitch-name* | -s *vswitch-number* } [-p {all | portid}] [-A] [-M | -X] [-Z] [-v]
hpvmnet { -S *vswitch_name* | -s *vswitch-number* } [-i | -o] portid:*portnum[,...]*:vlanid: {[*vlanid*] ... | [all]}

DESCRIPTION

A virtual machine accesses its network through a virtual network interface (vNIC) connected to a virtual network switch (vswitch). The virtual network switch is connected in turn to a single physical network interface (pNIC) on the VM Host. Use the `hpvmnet` command to create and manage vswitches.

A vswitch works like an actual network switch. It accepts outbound network traffic from all guests configured to use it and transmits the traffic over the physical interface. It accepts inbound network traffic for all guests configured to use it and directs the traffic to the appropriate guest.

A virtual switch can be associated with at most one physical network interface. The VM Host's physical network interface must be attached to a network with connectivity to the desired subnets. The network interface can be configured on the VM Host with an IP address or multiple IP alias addresses, but this is only necessary if the VM Host shares the interface with the vswitch and directs its own network traffic over the card. If you alter any characteristics of a network interface associated with a running vswitch, for instance, through the `ifconfig` commands on the VM Host, you must stop and restart the vswitch. Otherwise, any guests using that vswitch experience intermittent network failures. Stopping and restarting a vswitch can occur while its guests are running; guest shutdown is not required.

Integrity VM supports guest-based VLANs, (GBVs). With the `hpvmnet` command, you can manage the tagged VLANIDs on vswitch ports. The `-i` option enables a set of tagged VLANs and the `-o` option disables them. The *-p* option displays the list of enabled VLANIDs from the vswitch configuration file. Specifying *all* allows you to enable or disable all VLANs at once.

You must reboot the vswitch (using the `-r` option) when:

- You replace the physical network card associated with the vswitch.
- You change a VM Host IP address associated with the vswitch's network interface card.
- You change network interface characteristics, for example, by using the `lanadmin` command to change checksum offloading (CKO).
- You notice that there is no communication from an `avio_lan` interface to a `lan` interface after booting the guests while the vswitch is down.

You do not need to restart the guests that are using the vswitch. After you restart the vswitch, restart communication from the guest side. For example, on the guest, ping the VM Host.

By default, Integrity VM creates a vswitch named `localnet` that is not associated with a physical interface. It is used only for communication between the guests running on the same VM Host; the VM Host itself does not participate in a `localnet`. There is no nameserver or router configured on a `localnet`, unless one of the guests performs this function.

Only superusers can execute the `hpvmnet` command.

Two classes of virtual network interface are supported in Integrity VM: `lan` and `avio_lan`. The `lan` interface is supported on all underlying HP-UX network interface cards. The `avio_lan` interface is supported on selected HP-UX network interface cards and provides accelerated performance. For examples of using the `lan` and `avio_lan` keywords, see *hpvmcreate* and *hpvmmodify*.

Options

The following options can be specified only once.

The `hpvmnet` command without options displays summary information about all vswitches configured on the VM host.

The `hpvmnet` command recognizes the following command-line options and arguments:

`-S` *vswitch-name*
: Specifies the unique name of the virtual switch. The name of the vswitch is limited to eight characters.

  The `-S` and `-s` options are mutually exclusive.

`-s` *vswitch-number*
: Specifies the unique number of the virtual switch. The vswitch number is reported using the `hpvmnet` command.

  The `-S` and `-s` options are mutually exclusive.

`-v`
: Displays the version number of the `hpvmnet` output format. The version number is displayed first, followed by the display specified by the other options. In addition, it also displays VLAN information.

`-V`
: 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. Network counters are cleared each time statistics are reported; the display reports the counts since the previous display. Use the `-S` or `-s` option to specify the vswitch for which to display network counters.

  The `-V`, `-M` , and `-X` options are mutually exclusive.

`-A`
: Displays statistics for the `avio_lan` virtual network interfaces in the following format:

```
Vswitch Name           :  hostnet
Max Number of Ports    :  512
Port Number            :  3
  Port State           :  Reserved
  Active VM            :
  Untagged VlanId      :  none
  Reserved VMs         :  aviotest
  Adaptor              :  avio_lan
    Inbound Octets                          :
9698882
    Inbound Unicast Pkts (wire)          : 20000

    Inbound Unicast Pkts (local)         : 40000

    Inbound Non-Unicast Pkts (wire)      : 10000
```

```
    Inbound Non-Unicast Pkts (local)    : 30000

    Inbound Discards                     : 36
    Outbound Octets                      :
3557451342b
    Outbound Unicast Pkts (wire)        : 20000

    Outbound Unicast Pkts (local)      : 300000

    Outbound Non-Unicast Pkts          : 400000

    Outbound Discards                    : 0
```

`-M` Displays verbose resource information in a machine-readable format.

Individual fields are separated by one of the following delimiters:

- The colon (`:`) separates each field and resource type.
- The semicolon (`;`) separates subfields of a resource type.
- The comma (`,`) separates individual items in a list of similar items.

The `-V`, `-M` and `-X` options are mutually exclusive.

`-X` Displays verbose resource information in the XML format.

The `-V`, `-M`, and `-X` options are mutually exclusive.

`-Z` Used with to the `-A` option, clears statistics after retrieving them.

`-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).

The `-F` option is deprecated in Integrity Virtual Machines commands.

`-Q` Quietly performs the command. The default is to prompt for confirmation of the command before performing it.

`-c` Creates a new vswitch.

`-d` Deletes an existing vswitch.

`-b` Starts an existing vswitch. Vswitches must be started before they accept guest traffic. Note that Integrity VM attempts to automatically start all existing vswitches when Integrity VM is started.

`-h` Stops a vswitch.

`-i` Enables the list of VLAN ids on the list of ports. Specifying *`all`* allows you to enable all VLANs at once.

`-n` *`nic-id`* Designates the network interface on the VM Host that this vswitch uses. Network interfaces are displayed by the `lanscan` command. If `-n` is not specified when creating a vswitch, a local vswitch is created.

`-o` Disables the list of VLAN ids on the list of ports. Specifying *`all`* disables all VLANs at once.

`-r`	Stops and restarts a vswitch.
`-u portid:`*`portnum`*`:vlanid:{`*`vlanid`*` | none}`	Specifies the VLAN identifier for the specific vswitch and port. Configure VLANs by specifying the number of the port on the vswitch (*`portnum`*) to use for VLAN communication, and the VLAN identifier (`vlanid`). Virtual machines that are configured to use the VLAN can communicate with one another. To disable a VLAN, specify `none` for the VLAN identifier. Only virtual machines that are configured with the same VLAN identifier on their ports can communicate with one another. To disable VLANs on a port, specify `none` for the VLAN identifier. Virtual machines that do not have any VLANs configured on their vswitch port (which is the default) cannot communicate over that port with virtual machines that have a VLAN configured on their vswitch port. The port number is used to reserve a specific port for a particular virtual machine's network resource. This port number can be used later in the `hpvmnet` command to configure VLAN rules on the port. You can also first set up VLAN rules on the virtual switch and later create virtual machines referencing the specific port numbers. **NOTE:** The maximum number of ports supported per vswitch is 512; however, port 0 is reserved for the VM Host, and ports 1 through 511 are available for general use.
`-p {all | `*`portid`*`}`	Specifies the port number on the vswitch. To specify all the ports on the vswitch, enter `all`. This command displays information about VLAN ports configured for the specified vswitch, including a list of enabled VLAN IDs. Do not use the `-p` option with the `-V` option.
`-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 the `-s` option. Only those ports with tagged VLAN are cloned.
`-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 is limited to eight characters. You must include either the `-S` or the `-s` option.

RETURN VALUES

The `hpvmnet` command exits with one of the following values:

0: Successful completion.

1: One or more error conditions occurred.

DIAGNOSTICS

The `hpvmnet` command displays error messages for any of the following conditions:

- An invalid option is specified.
- An invalid value is specified for an option.

- The *vswitch-name* or *vswitch-number* argument does not exist, cannot be accessed, is not a vswitch, or is corrupt.
- A value was omitted for an argument that requires one, or a value was supplied for an argument that does not take one.
- The `hpvmnet` command and Integrity Virtual Machines are at different revision levels.

EXAMPLES

Create the vswitch `switch2` on lan0:

```
# hpvmnet -c -S switch2 -n 0
```

Display all the vswitches on this VM Host:

```
# hpvmnet
Name     Number State   Mode      NamePPA  MAC Address    IPv4 Address
======== ====== ======= ========= ======== ============== ===============
localnet      1 Up      Shared             N/A            N/A
hostnet     296 Up      Shared    lan1     0x00306e4a92ef .1.8.99
vmvlan       13 Up      Shared    lan900   0x00306e4bc7bf
myswitch    241 Up      Shared    lan0     0x00306e4a93e6
```

Display the verbose attributes of vswitch hostnet:

```
# hpvmnet -S hostnet -V
Name                     : hostnet
number                   : 296
PID                      : 72
State                    : Up
Mode                     : Shared
NamePPA                  : lan1
Host MAC Address         : 0x00306e4a92ef
Host IPv4 Address        : .1.8.99
Host IPv6 Address        : fe80::230:6eff:fe4a:92ef
Number of guests         : 5
Guest's Name             : armyvm1
  MAC Address            : 0xEE8D40584D28
  IPv4 Address           : Guest halted
  Vswitch Port           : 4
Guest's Name             : armyvm1
  MAC Address            : 0x8A0F12DD5B48
  IPv4 Address           : Guest halted
  Vswitch Port           : 3
Guest's Name             : armyvm2
  MAC Address            : 0x42F86E004C5D
  IPv4 Address           : Guest halted
  Vswitch Port           : 5
Guest's Name             : armyvm4
  MAC Address            : 0x762704ACBC4D
  IPv4 Address           : 106..5.24
  IPv6 Address           : fe80::ee8d:40ff:fe58:4d28
  Vswitch Port           : 1
Guest's Name             : armyvm3
  MAC Address            : 0xD2E856A02638
  IPv4 Address           : armyvm3.mycom.com
  Vswitch Port           : 2
Packets in               : 37934
Packets out to stream    : 967
Packets out to guest     : 31381
Packets dropped          : 4686
Broadcasts               : 31029
Counter 0                : 0
Counter 1                : 0
Counter 2                : 0
Counter 3                : 0
Cycles in                : 7728
```

```
Cycles out to stream    : 27387
Cycles out to guest     : 5509
Cycles switch packet    : 2345
```

Delete the vswitch named switch2:

```
# hpvmnet -d -S switch2
hpvmnet: Remove the vswitch 'switch2'? [n/y]: y
```

Delete the vswitch with the vswitch ID of 6, and omit the confirmation dialog:

```
# hpvmnet -d -s 6 -F
```

Start the vswitch named switch1:

```
# hpvmnet -b -S switch1
```

Stop the vswitch named switch1:

```
# hpvmnet -h -S switch1
hpvmnet: Halt the vswitch 'switch1'? [n/y]: y
```

Configure port 2 on vswitch switch1 with VLAN identifier 100:

```
# hpvmnet -S switch1 -u portid:2:vlanid:100
```

Display information about VLAN ports configured for the vswitch named myswitch:

```
# hpvmnet -S myswitch
Name     Number State   Mode      NumberPPA  MAC Address   IPv4 Address
======== ====== ======= ========= ====== ==============  ===============
myswitch      2 Up      Shared    lan1   0x00306ef3120c  1.2.3.4

[Port Configuration Details]
Port    Port         Port   Untagged  Number of    Active VM     Tagged
Number  state        Adaptor  VLANID    Reserved VMs      VLANIDs
======= ============ ========  ========  ============ ============  ============
1       Reserved     lan   none      1
2       Active       lan   none      1            config2
3       Reserved     avio_lan none      1
4       Active       lan   none      1            winguest2

[Configured IP Address(es)]
        106..4.12
        fe80::230:6eff:fe4a:92ef

# hpvmnet -S myswitch2 -p 2
Vswitch Name             :  vswitch2
Max Number of Ports      :  512
Port Number              :  2
  Port State             :  Active
  Active VM              :  guest31
  Untagged VlanId        :  none
  Reserved VMs           :  guest31
  Adaptor                :  avio_lan
  Tagged VLANs           :  20, 30, 40
```

To switch over the lan1 from lan0:

```
# hpvmnet -C -S vswitch2 -n 0
# hpvmnet -S vswitch2
 Name     Number State   Mode      NamePPA  MAC Address    IPv4 Address
======== ====== ======= ========= ======== ============== ===============
vswitch2      4 Up      Shared    lan2     0x001a4bf3021e

[Port Configuration Details]
Port    Port         Port     Untagged Number of    Active VM    Tagged
Number  State        Adaptor  VLANID   Reserved VMs              VLANIDs
======= ============ ======== ======== ============ ============ =============
1       Active       avio_lan none     1            vm206_5_31   1
3       Reserved     avio_lan none     1                         1,10,30,55
4       Available    unknown  30       0                         none
10      Available    unknown  none     0                         11,21,31,41...
```

To clone the vswitch, enter the following command:

```
# hpvmnet -C -n 1 -S vmvlan -N clone
# hpvmnet -S clone
```

```
Name     Number State   Mode      NamePPA  MAC Address    IPv4 Address
======== ====== ======= ========= ======== ============== ===============
clone       297 Down    Shared    lan1

[Port Configuration Details]
Port    Port         Port     Untagged  Number of     Active VM    Tagged
Number  state        Adaptor   VLANID    Reserved VMs               VLANIDs
======= ============ ========  ========  ============  ============ ============
2       Available    avio_lan  20        0
```

AUTHORS

The `hpvmnet` command was developed by HP.

OBSOLESCENCE

The `-F` option is deprecated in Integrity Virtual Machines commands. This flag might be removed in a future release of Integrity Virtual Machines. HP recommends use of the command with the `-s` option, then resolving any errors or warnings that result before executing the command without the `-s` option and without the `-F` option.

SEE ALSO

On the VM Host:

hpvm(5), *hpvmclone*(1M), *hpvmcollect*(1M), *hpvmconsole*(1M), *hpvmcreate*(1M), *hpvmdevinfo*(1M), *hpvmdevmgmt*(1M), *hpvmdevtranslate*(1M), *hpvmhostgdev*(1M), *hpvmhostrdev*(1M)*hpvminfo*(1M), *hpvmmigrate*(1M), *hpvmmodify*(1M), *hpvmmove_suspend*(1M), *hpvmnvram*(1M), *hpvmpubapi*(3), *hpvmremove*(1M), *hpvmresources*(5), *hpvmresume*(1M), *hpvmsar*(1M), *hpvmsg_move*(1M), *hpvmsg_package*(1M), *hpvmstart*(1M), *hpvmstatus*(1M), *hpvmstop*(1M), *hpvmsuspend*(1M), *hpvmupgrade*(1M)*p2vassist*(1M)

On the Integrity VM guest:

hpvmcollect(1M), *hpvminfo*(1M), *hpvmmgmt*(1M), *hpvmpubapi*(3)

hpvmnvram(1M)

NAME

hpvmnvram -- Display, create, edit and remove EFI variables in NVRAM files from a VM Host.

SYNOPSIS

hpvmnvram { -P *vm-name* | -p *vm-number* | -f *filename* } -l
hpvmnvram { -P *vm-name* | -p *vm-number* | -f *filename* } -a *devpath*[:[boot_order]] [:[efi_file_path]] [-Q]
hpvmnvram { -P *vm-name* | -p *vm-number* | -f *filename* } -d *boot_order* [-Q]
hpvmnvram { -P *vm-name* | -p *vm-number* | -f *filename* } -m *from_boot_order:to_boot_order* [-Q]
hpvmnvram { -P *vm-name* | -p *vm-number* | -f *filename* } -L
hpvmnvram { -P *vm-name* | -p *vm-number* | -f *filename* } -G [*guid:*]*name* -b *binary_filename*
hpvmnvram { -P *vm-name* | -p *vm-number* | -f *filename* } -S [*guid:*]*name:binary_filename* [-Q]
hpvmnvram { -P *vm-name* | -p *vm-number* | -f *filename* } -R [*guid:*]*name* [-Q]

DESCRIPTION

A virtual machine's NVRAM file contains the EFI variables that are used for communication between the virtual machine's virtualized firmware and the operating system. The `hpvmnvram` command is used from the VM Host's command line to display, create, edit and remove EFI variables in an inactive virtual machine's NVRAM file.

Variables that are in active use during a virtual machine's boot sequence are displayed. The variables in the file are of the type NV, or non-volatile. The specified NVRAM file must already have the proper EFI-readable format complete with storage bank headers.

Only superusers can execute the `hpvmnvram` command.

Options

The following options can be specified only once.

The `hpvmnvram` command recognizes the following options and arguments:

–l
: Lists all the boot options in the specified virtual machine or NVRAM file.

-a *dev_path*[:[*boot_order*]] [:*efi_file_path*]]
: Adds a new boot option to the specified virtual machine or NVRAM file. It first creates a new EFI boot variable, `Boot####`, using *dev_path* and the optional *efi_file_path* and then adds the `Boot####` variable to the *boot_order* entry of the EFI BootOrder ordered list.

- *dev_path*

  Specifies the path of a guest virtual disk device that is to be used to create the new boot option. The guest virtual disk device must be currently assigned to the specified virtual machine and is backed by a file in the VM Host file system by a logical volume or by a whole disk. The `hpvmstatus` command can be used to obtain a list of virtual disk devices currently assigned to the virtual machine.

- *boot_order*

  Specifies the entry where the new boot option is to be placed in the EFI BootOrder ordered list. If *boot_order* is not specified, a default value of "1" is assigned as the boot order.

- *efi_file_path*

  Specifies the path of an EFI application, such as an EFI OS loader, to be used with the new boot option. If the *efi_file_path* is not specified and the virtual machine's OS type is HP-UX, a default file path, "`\EFI\HPUX\HPUX.EFI`", is used.

`-d` *boot_order*	Deletes a boot option from the specified virtual machine or NVRAM file. It first deletes the specified *boot_order* entry from the EFI BootOrder ordered list and then deletes the `Boot####` variable found at the *boot_order* entry.
`-m` *from_boot_order*`:`*to_boot_order*	Changes boot order of the specified virtual machine or NVRAM file. It moves a boot option from the *from_boot_order* entry to the *to_boot_order* entry in the EFI BootOrder ordered list.
`—L`	Lists all the variables and their data that are currently set to active in the specified virtual machine or NVRAM file.
`-G [`*guid*`:]`*name*	Gets a specific variable and its data from the specified virtual machine or NVRAM file by matching the specified guid and variable name input. If guid is not specified, a default GUID, EFI_GLOBAL_VARIABLE (61DFE48B-CA93-D211-AA0D00e098032b8c), is used.
`-S [`*guid*`:]`*name*`:`*binary_filename*	Sets a new variable in the specified virtual machine or NVRAM file. If the variable already exists, this existing one is marked as obsolete, and a new variable is created. The variable created consists of the specified *guid* and *name*. The new variable contains the machine-readable, formatted data in the specified *binary_filename* file. If *guid* is not specified, a default GUID, EFI_GLOBAL_VARIABLE, is used.
`-R [`*guid*`:]`*name*	Removes a variable from the specified virtual machine or NVRAM file. If the variable exists and is not used by the EFI BootOrder variable, the variable is marked as obsolete. If it does not exist or is already marked as obsolete, nothing is done to the NVRAM file contents. If *guid* is not specified, a default GUID, EFI_GLOBAL_VARIABLE, is used.

NOTE: The `-l`, `-a`, `-d`, `-m`, `-L`, `-G`, `-S`, and `-R` options are mutually exclusive.

`-b` *binary_filename*	When used with the `-G` options, it creates a machine-readable format of the retrieved variable and its data and stores them in the specified *binary_filename* file.

–Q	Quietly performs the command. When used with the -a, -d, -m, -S, or -R options, the reminder to back up the NVRAM file is suppressed.
-P *vm-name*	Specifies the name of the virtual machine whose current NVRAM file is used.
-p *vm-number*	Specifies the number of the virtual machine whose current NVRAM file is used. The vm_number is assigned when a virtual machine is created and is displayed by the *hpvmstatus* command.
-f *filename*	Specifies the name of an NVRAM file that is not a virtual machine's current NVRAM file. **NOTE:** The -P, -p and -f options are mutually exclusive.

RETURN VALUES

The hpvmnvram command exits with one of the following values:

0: Successful completion.

1: One or more error conditions occurred.

DIAGNOSTICS

The hpvmnvram command displays error messages on stderr for any of the following conditions:

- An invalid option is specified.
- An invalid value is specified for an option.
- The *vm-name* or *vm-number* does not exist, cannot be accessed, is not a virtual machine, or is corrupt.
- The NVRAM file does not exist, cannot be accessed, or is corrupt.
- A value was omitted for an argument that requires one, or a value was supplied for an argument that does not take one.

EXAMPLES

List all boot options for the virtual machine named guest3:

```
# hpvmnvram -P guest3 -l
Boot   EFI Boot
Order  Variable  Description
=====  ========  ==================================================
1      Boot0000  EFI Shell [Built-in]
```

Add /dev/rdisk/disk7 as first boot option for the virtual machine named guest3:

```
# hpvmnvram -P guest3 -a /dev/rdisk/disk7
You should make a backup copy of this nvram file before proceeding with changes.
Continue? Enter Y or N:y
hpvmnvram: Adding boot option 'Guest Device: Bus 0, Dev 1, Ftn 0, Tgt 0, Lun 0' (/dev/rdisk/disk7) ...
Boot   EFI Boot
Order  Variable  Description
=====  ========  ==================================================
1      Boot0001  Guest Device: Bus 0, Dev 1, Ftn 0, Tgt 0, Lun 0
2      Boot0000  EFI Shell [Built-in]
```

Add /hpvm/disk1 as second boot option for the virtual machine named guest3 and specify the EFI file path to \efi\newpath\hpux:

```
# hpvmnvram -P guest3 -a /hpvm/disk1:2:"\efi\newpath\hpux"
You should make a backup copy of this nvram file before proceeding with changes.
Continue? Enter Y or N:y
hpvmnvram: Adding boot option 'Guest Device: Bus 0, Dev 1, Ftn 0, Tgt 2, Lun 0' (/hpvm/disk1) ...
Boot   EFI Boot
Order  Variable  Description
=====  ========  ==================================================
1      Boot0001  Guest Device: Bus 0, Dev 1, Ftn 0, Tgt 0, Lun 0
```

```
2      Boot0002  Guest Device: Bus 0, Dev 1, Ftn 0, Tgt 2, Lun 0
3      Boot0000  EFI Shell [Built-in]
```

Move the second boot option to be the first boot option in the virtual machine named guest3:

```
# hpvmnvram -P guest3 -m 2:1
You should make a backup copy of this nvram file before proceeding with changes.
Continue? Enter Y or N:y
Boot   EFI Boot
Order  Variable  Description
=====  ========  ===================================================
1      Boot0002  Guest Device: Bus 0, Dev 1, Ftn 0, Tgt 2, Lun 0
2      Boot0001  Guest Device: Bus 0, Dev 1, Ftn 0, Tgt 0, Lun 0
3      Boot0000  EFI Shell [Built-in]
```

Delete the second boot option from the virtual machine named guest3:

```
# hpvmnvram -P guest3 -d 2
You should make a backup copy of this nvram file before proceeding with changes.
Continue? Enter Y or N:y
Boot   EFI Boot
Order  Variable  Description
=====  ========  ===================================================
1      Boot0002  Guest Device: Bus 0, Dev 1, Ftn 0, Tgt 2, Lun 0
2      Boot0000  EFI Shell [Built-in]
```

List all active variables for the virtual machine named guest3:

```
# hpvmnvram -P guest3 -L
Variable NV+RT+BS '61DFE48B-CA93-D211-AA0D00E098032B8C:Lang' DataSize = 3
        0000 | 0x 65 6e 67
Variable NV+RT+BS '61DFE48B-CA93-D211-AA0D00E098032B8C:ConOut' DataSize = 4e
           0000 | 0x 02 01 0c 00 d0 41 03 0a 00 00 00 00 01 01 06 00
       0x10 | 0x 00 03 02 01 0c 00 d0 41 01 05 00 00 00 00 03 0e
       0x20 | 0x 18 00 00 00 00 00 00 c2 01 00 00 00 00 00 08 01
       0x30 | 0x 01 00 00 00 00 00 03 0a 14 00 65 60 a6 df 19 b4
       0x40 | 0x d3 11 9a 2d 00 90 27 3f c1 4d 7f ff 04 00
Variable NV+RT+BS '61DFE48B-CA93-D211-AA0D00E098032B8C:ConIn' DataSize = 4e
       0000 | 0x 02 01 0c 00 d0 41 03 0a 00 00 00 00 01 01 06 00
       0x10 | 0x 00 03 02 01 0c 00 d0 41 01 05 00 00 00 00 03 0e
       0x20 | 0x 18 00 00 00 00 00 00 c2 01 00 00 00 00 00 08 01
       0x30 | 0x 01 00 00 00 00 00 03 0a 14 00 65 60 a6 df 19 b4
       0x40 | 0x d3 11 9a 2d 00 90 27 3f c1 4d 7f ff 04 00
 ...
```

Get the variable `BootOrder` for EFI_GLOBAL_VARIABLE GUID (61DFE48B-CA93-D211-AA0D00e098032b8c) in the NVRAM file, `my_nvram`, which is located in the current directory:

```
# hpvmnvram -G 61DFE48B-CA93-D211-AA0D00e098032b8c:BootOrder -f my_nvram
Variable NV+RT+BS '61DFE48B-CA93-D211-AA0D00E098032B8C:BootOrder' DataSize = 4
0000 | 0x 02 00 00 00
```

Get the same `BootOrder` variable from the previous example without specifying the GUID:

```
# hpvmnvram -G BootOrder -f my_nvram
Variable NV+RT+BS '61DFE48B-CA93-D211-AA0D00E098032B8C:BootOrder' DataSize = 4
0000 | 0x 02 00 00 00
```

Get the variable `Timeout` for EFI_GLOBAL_VARIABLE GUID of the virtual machine named guest3:

```
# hpvmnvram -G Timeout -P guest3
Variable NV+RT+BS '61DFE48B-CA93-D211-AA0D00E098032B8C:Timeout' DataSize = 2
0000 | 0x 0a 00
```

Update the same `Timeout` variable from the previous example with the new data content contained in the binary file, `time.0`, which is located in the current directory. Use the`-Q` option to suppress the reminder to back up the NVRAM file:

```
# hpvmnvram -S Timeout:time.0 -P guest3 -Q
```

Verify the new `Timeout` value from the previous example with the `-G` option:

```
# hpvmnvram -G Timeout -P guest3
Variable NV+RT+BS '61DFE48B-CA93-D211-AA0D00E098032B8C:Timeout' DataSize = 2
0000 | 0x 00 00
```

Remove the variable `panicinfo` for 3CC469BA-5744-11D4-868E7F7F00000000 GUID of the virtual machine named guest3:

```
# hpvmnvram -R 3CC469BA-5744-11D4-868E7F7F00000000:panicinfo -P guest3
```

AUTHOR

The `hpvmnvram` command was developed by the HP.

SEE ALSO

On the VM Host:

hpvm(5), *hpvmclone*(1M), *hpvmcollect*(1M), *hpvmconsole*(1M), *hpvmcreate*(1M), *hpvmdevinfo*(1M), *hpvmdevmgmt*(1M), *hpvmdevtranslate*(1M), *hpvmhostgdev*(1M)*hpvmhostrdev*(1M)*hpvminfo*(1M), *hpvmmigrate*(1M), *hpvmmodify*(1M), *hpvmmove_suspend*(1M), *hpvmnet*(1M), *hpvmpubapi*(3), *hpvmremove*(1M), *hpvmresources*(5), *hpvmresume*(1M), *hpvmsar*(1M), *hpvmsg_move*(1M), *hpvmsg_package*(1M), *hpvmstart*(1M), *hpvmstatus*(1M), *hpvmstop*(1M),*hpvmsuspend*(1M),*hpvmupgrade*(1M)*p2vassist*(1M)

On the Integrity VM guest:

hpvmcollect(1M), *hpvminfo*(1M), *hpvmmgmt*(1M), *hpvmpubapi*(3)

hpvmpubapi(3)

NAME

hpvmpubapi – Integrity VM public application interface descriptions.

SYNOPSIS

```
#include <hpvm_api_public.h>
HPVM_boolean hpvm_api_server_check()
HPVM_boolean hpvm_api_virtmach_check()
HPVM_int hpvm_api_version_get(HPVM_str version)
HPVM_int hpvm_api_my_uuid_get(HPVM_str uuid)
HPVM_int hpvm_api_server_uuid_get(HPVM_str uuid)
HPVM_int hpvm_api_server_hostname_get(HPVM_str hostname)
```

DESCRIPTION

The `hpvmpubapi` command contains the following public APIs:

- `hpvm_api_server_check` — returns the HPVM_boolean type. If this API is run on an operational Integrity VM server, it returns HPVM_TRUE; otherwise it returns HPVM_FALSE as defined in `hpvm_api_public.h`.
- `hpvm_api_virtmach` — returns the HPVM_boolean type. If this API is run on a virtual machine that has the appropriate guest kit installed, it returns HPVM_TRUE; otherwise it returns HPVM_FALSE as defined in `hpvm_api_public.h`.
- `hpvm_api_version_get` — returns the HPVM_int type as an error code and, if successful, the Integrity VM version string. This API copies the Integrity VM version string into the buffer referenced by the *`version`* parameter. The *`version`* pointer must be a preallocated buffer that is at least HPVMAPIDEFMEMSIZ bytes long. If an error occurs, a nonzero error code is returned.
- `hpvm_api_my_uuid_get` — returns the HPVM_int type as an error code and, if successful, the Integrity VM version string. This API copies the *`uuid`* string into the buffer referenced by the *`uuid`* parameter. The *`uuid`* pointer must be a preallocated buffer that is at least HPVMAPIDEFMEMSIZ bytes long. If an error code occurs, a nonzero error code is returned.
- `hpvm_api_server_uuid_get` — returns the HPVM_int type as an error code and, if successful, the Integrity VM version string. This API copies the *`uuid`* string of the Integrity VM server into the buffer referenced by the *`uuid`* parameter. The *`uuid`* pointer must be a preallocated buffer that is at least HPVMAPIDEFMEMSIZ bytes long. If an error occurs, a nonzero error code is returned. This routine returns an error if run on an Integrity VM server.
- `hpvm_api_server_hostname_get` — returns the HPVM_int type as an error code and, if successful, the Integrity VM version string. This API copies the *`uuid`* string of the HPVM server into the buffer referenced by the *`hostname`* parameter. The *`hostname`* pointer must be a preallocated buffer that is at least HPVMAPIDEFMEMSIZ bytes long. If an error occurs, a nonzero error code is returned. This routine returns an error if run on an Integrity VM server.

RETURN VALUES

If run on an operational Integrity VM server, the `hpvm_api_server_check` routine returns HPVM_TRUE; otherwise returns HPVM_FALSE.

If run on an Integrity VM virtual machine, the `hpvm_api_virtmach_check` routine returns HPVM_TRUE; otherwise returns HPVM_FALSE.

If successful, the `hpvm_api_version_get`, `hpvm_api_my_uuid_get`, `hpvm_api_server_uuid_get`, and `hpvm_api_server_hostname_get` routines return 0.

DIAGNOSTICS

There are no error returns for the `hpvm_api_server_check` and `hpvm_api_virtmach_check` routines. If an error occurs on the `hpvm_api_version_get`, `hpvm_api_my_uuid_get`, `hpvm_api_server_uuid_get`, or `hpvm_api_server_hostname_get` routine, one of the following error codes defined in `hpvm_api_public.h` is returned:

- HPVMAPINOMEMORY — Inadequate memory resources.
- HPVMAPIINVALARG — Invalid argument.
- HPVMAPIBADFORMAT — Retrieved uuid was formatted badly.
- HPVMNOTVM — Not being run on an HPVM virtual machine.
- HPVMAPINOTSUP — Not supported.

AUTHORS

The `hpvmpubapi` public APIs were developed by HP.

SEE ALSO

On the VM Host:

hpvm(5), *hpvmclone*(1M), *hpvmcollect*(1M), *hpvmconsole*(1M), *hpvmcreate*(1M), *hpvmdevinfo*(1M), *hpvmdevmgmt*(1M), *hpvmdevtranslate*(1M), *hpvmhostgdev*(1M), *hpvmhostrdev*(1M), *hpvminfo*(1M), *hpvmmigrate*(1M), *hpvmmodify*(1M),*hpvmmove_suspend*(1M), *hpvmnet*(1M), *hpvmnvram*(1M), *hpvmremove*(1M), *hpvmresources*(5), *hpvmsar*(1M), *hpvmsg_move*(1M), *hpvmsg_package*(1M), *hpvmstart*(1M), *hpvmstatus*(1M), *hpvmstop*(1M), *hpvmsuspend*(1M),*hpmvupgrade*(1M), *p2vassist*(1M)

On the Integrity VM guest:

hpvmcollect(1M), *hpvminfo*(1M), *hpvmmgmt*(1M), *hpvmpubapi*(3)

hpvmremove(1M)

NAME

hpvmremove – Remove an Integrity VM virtual machine.

SYNOPSIS

hpvmremove { -P *vm_name* | -p *vm_number* } [-F] [-Q]

DESCRIPTION

The hpvmremove command deletes a virtual machine's configuration information and frees any resources associated with it. Once the virtual machine has been removed all resources associated with the virtual machine become available for allocation to other virtual machines.

Unintentional use of this command has serious consequences; therefore, the user is prompted to confirm this operation unless the -F (force) option is specified.

The virtual machine must be in the Off state to be removed: It is an error to remove a running virtual machine.

Only superusers can execute the hpvmremove command.

Options

The following options can be specified only once.

The hpvmremove command recognizes the following command-line options and arguments:

-P *vm_name* Specifies the unique name of the virtual machine to be removed.

You must specify either the -P or the -p option.

-p *vm_number* Specifies the unique number of the virtual machine to be removed. The *vm_number* is reported by the hpvmstatus command.

You must specify either the -P or the -p option.

-F Omits the confirmation dialog before removing the virtual machine. This option is intended for use by scripts and other noninteractive applications (Force mode).

The -F option is deprecated in Integrity Virtual Machines commands.

-Q Quietly performs the command. The default is to prompt for confirmation of the command before performing it.

RETURN VALUES

The hpvmremove command exits with one of the following values:

0: Successful completion.

1: One or more error conditions occurred.

DIAGNOSTICS

The hpvmremove command displays error messages on stderr for any of the following conditions:

- An invalid option is specified.
- An invalid value is specified for an option or value is omitted.
- A value was omitted for an argument that requires one, or a value was supplied for an argument that does not take one.
- The *vm_name* or *vm_number* argument does not exist, cannot be accessed, is not a virtual machine, or is corrupt.

- The `vm_name` argument is in some state other than Off.
- The `hpvmremove` command and Integrity VM are at different revision levels.

EXAMPLES

Delete the virtual machine myguest:

```
# hpvmremove -P myguest
hpvmremove: Remove the virtual machine myguest? [n/y]: y
```

Delete a virtual machine using its unique identifier using the force option:

```
# hpvmremove -F -p 333
```

Remove a running guest:

```
# hpvmremove -P hpux1
hpvmremove: The guest is currently running, not able to remove.
hpvmremove: Unable to continue.
```

OBSOLESCENCE

The `-F` option is deprecated in Integrity Virtual Machines commands. This flag might be removed in a future release of Integrity Virtual Machines. HP recommends use of the command with the `-s` option, then resolving any errors or warnings that result before executing the command without the `-s` option and without the `-F` option.

AUTHORS

The `hpvmremove` command was developed by HP.

SEE ALSO

On the VM Host:

hpvm(5), *hpvmclone*(1M), *hpvmcollect*(1M), *hpvmconsole*(1M), *hpvmcreate*(1M), *hpvmdevinfo*(1M), *hpvmdevmgmt*(1M), *hpvmdevtranslate*(1M), *hpvmhostgdev*(1M), *hpvmhostrdev*(1M), *hpvminfo*(1M), *hpvmmigrate*(1M), *hpvmmodify*(1M), *hpvmmove_suspend*(1M), *hpvmnet*(1M), *hpvmnvram*(1M), *hpvmpubapi*(3), *hpvmresources*(5), *hpvmresume*(1M), *hpvmsar*(1M), *hpvmsg_move*(1M), *hpvmsg_package*(1M), *hpvmstart*(1M), *hpvmstatus*(1M), *hpvmstop*(1M), *hpvmsuspend*(1M), *hpvmupgrade*(1M), *p2vassist*(1M)

On the Integrity VM guest:

hpvmcollect(1M), *hpvminfo*(1M), *hpvmmgmt*(1M), *hpvmpubapi*(3)

hpvmresources(5)

NAME

hpvmresources -- Specifying virtual storage and virtual network devices.

SYNOPSIS

`Virtual resource specification`

DESCRIPTION

Use the `hpvmcreate`, `hpvmclone`, and `hpvmmodify` commands to specify storage devices and vswitches for guests. To specify the name of the storage device or vswitch, use the syntax described here.

The resource specification contains the virtual device information and the backing store information, separated by a colon (:). The resource specification can be used to define a virtual storage device or a virtual network device.

Specifying Storage Devices

For storage devices, enter the resource specification in the following format:

`devicetype``:`*`adaptertype`*`:`*`bus`*`,`*`device`*`,`*`target`*`:`*`storage`*`:`*`device`*

If you omit any part of the resource specification, you must include the colon (:) character as the field delimiter.

The guest virtual device information consists of the following fields, separated by colons:

- *`devicetype`* (virtual device type):
  - `disk`
  - `dvd`
  - `tape`
  - `changer`
  - `burner`
  - `hba`
- *`adaptertype`* (virtual device adapter type): `scsi` or `avio_stor`.
- *`bus, device, target`* (virtual device hardware address) (optional):

  The virtual device hardware address consists of three fields, separated by commas:
  - *`bus`* (the virtual device PCI bus number)
  - *`device`* (the virtual device PCI slot number)
  - *`target`* (the virtual device SCSI target number)

  If you do not specify the virtual device hardware address, it is generated automatically. If you specify a portion of the virtual device hardware address (for example, just the target), you must include the commas (for example, to specify just target 2, enter `,,2`).

The physical device information consists of two fields, separated by a colon:

- *`storage`* (physical storage type)
- *`device`* (physical device)

The physical storage type and device specification can be one of the following:

- `disk`. For the physical device, specify a disk device file (for example, `/dev/rdisk/disk1`).
- `lv`. For the physical device, specify the LVM or VxVM character logical device file (for example, `/dev/vg01/rlv012`).
- `file`. For the physical device, specify a locally mounted, non-NFS VxFS file (for example, `/guestfiles/diskfile1`).
- `null`. Specify an empty storage unit. Do not specify a world-writable directory such as `tmp`. (This is useful for removable media, such as DVDs.)

  NOTE: When assigning a null device to a resource on a virtual machine, the file name serves as a placeholder. Therefore, you do not receive an error if the file does not exist. For example, in the following command, if the file `XXX.iso` does not exist, no error is given.

  ```
  # hpvmmodify -P vm1 -a disk:scsi::null:/opt/XXX.iso
  ```

- `attach`. For VIO, specify an attached device, such as a tape device, media changer, or CD/DVD burner.
- `attach_path`. For AVIO, specify an attached device, such as a tape device, media changer, or CD/DVD burner.

The physical device names must not contain the following characters: colon (:), semicolon (;), and comma (,).

The following example associates a guest virtual disk device with a physical disk device, using VIO:

```
# disk:scsi::0,1,0:disk:/dev/rdisk/disk1
```

- The virtual device type is `disk`.
- The virtual device adapter type is `scsi`.
- The virtual device PCI bus number is `0`.
- The virtual device PCI slot number is `1`.
- The virtual device SCSI target number is `0`.
- The physical storage type is `disk`.
- The physical device is `/dev/rdisk/disk1`.

The following example associates an empty guest virtual DVD with multiple future choices of ISO files:

```
dvd:scsi::null:/docs
```

- The virtual device type is `dvd`.
- The virtual device adapter type is `scsi`.
- The virtual device hardware address is automatically generated.
- The physical storage type is `null`.
- The physical device is `/docs`.

The following example specifies a tape device using VIO:

```
# tape:scsi::attach:/dev/rscsi/c6t5d0
```

The following example associates a guest tape device with a physical device, using AVIO:

```
# tape:avio_stor::0,4,0:attach_path:0/7/1/1.0x500104f00048b29e.0x0
```

- The virtual device type is `tape`.
- The virtual device adapter type is `avio_stor`.

- The virtual device PCI bus number is `0`.
- The virtual device PCI slot number is `4`.
- The virtual device `avio_stor` target number is `0`.
- The lunpath hardware path of the device is `0/7/1/1.0x500104f00048b29e.0x0`.

Specifying Network Devices

To associate a guest virtual network device with a virtual network switch (vswitch), use the following syntax. (Before you can associate the virtual network device to a virtual switch, you must create the vswitch using the `hpvmnet` command.)

The format of the *`rsrc`* parameter for network devices is:

`network:`*`adaptertype`*`:`*`bus`*`,`*`device`*`,`*`mac-addr`*`:vswitch:`*`vswitch-name`*`:portid:`*`portnumber`*

If you omit any portion of the network resource specification, use the colon character (:) as a field delimiter.

The guest virtual network information consists of the following fields, separated by colons:

- `network`
- `adaptertype` (virtual device adapter type): `lan` , `avio_lan`
- *`:bus,device,mac-addr`* (virtual network device hardware address) (optional)

  The virtual network device hardware address consists of three fields, separated by commas:

  - *`bus`* (the virtual network device PCI bus number)
  - *`device`* (the 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 verify that it is unique, because the address cannot conflict with any of the VM Host's physical network adapter MAC addresses, and to verify that the proper bits are set. You can specify either the bus and device with the MAC address or the MAC address without the bus and device, or you can omit the entire hardware address. If you do not specify the virtual network device hardware address, or a portion of it, the missing information is generated automatically. If you specify only a portion of the virtual network device hardware address (for example, just the MAC address), include the commas (for example `,,aa-bb-cc-00-11-22`).

    When you add a virtual NIC to your guest, Integrity VM checks to make sure the MAC address is unique.

    By default, Integrity VM makes three attempts (each with a one-second timeout) to determine the validity of the MAC address for the virtual NIC. This process can result in up to ten seconds of delay for each defined virtual NIC. To speed up this processing, add the following tunable to the `/etc/rc.config.d/hpvmconf` configuration file:

    `HPVMMACADDRFRAMES=`*`n`*

    Where *n* is the number of attempts (1 to 30). The default is 3. A value of 1 or 2 increases performance at the risk of missing a response from a slow NIC.

    You can set the `HPVMMACADDRFRAMES` tunable to zero (0), which completely eliminates the MAC address verification. However, HP recommends that you do so after you configure all of your guests and confirm that there are no conflicts with MAC addresses in your network environment.

    To boost virtual network performance, create additional vswitches and allocate them across guests.

The virtual switch information consists of the following fields, separated by a colon:

- `vswitch`
- *`vswitch-name`* (the name assigned to the virtual network switch)
- *`portnumber`* (the number of the port on the vswitch)

The following example associates a guest virtual network device with a vswitch:

`network:avio_lan:0,1,02-02-03-04-05-06:vswitch:net1`

- The guest virtual network device type is `network`.
- The virtual adapter type is `avio_lan`.
- The virtual PCI bus number is `0`.
- The virtual PCI slot number is `1`.
- The virtual MAC address is `02-02-03-04-05-06`.
- The physical network device type is `vswitch`.
- The vswitch name is `net1`.

AUTHORS

The `hpvmresources` command was developed by HP.

SEE ALSO

On the VM Host:

hpvm(5), *hpvmclone*(1M), *hpvmcollect*(1M), *hpvmconsole*(1M), *hpvmcreate*(1M), *hpvmdevinfo*(1M), *hpvmdevmgmt*(1M), *hpvmdevtranslate*(1M), *hpvmhostgdev*(1M), *hpvmhostrdev*(1M), *hpvminfo*(1M), *hpvmmigrate*(1M), *hpvmmodify*(1M), *hpvmmove_suspend*(1M), , *hpvmnet*(1M), *hpvmnvram*(1M), *hpvmpubapi*(3), *hpvmremove*(1M), *hpvmresume*(1M), *hpvmsar*(1M), *hpvmsg_move*(1M), *hpvmsg_package*(1M), *hpvmstart*(1M), *hpvmstatus*(1M), *hpvmstop*(1M), *hpvmsuspend*(1M)*hpvmupgrade*(1M), *p2vassist*(1M)

On the Integrity VM guest:

hpvmcollect(1M), *hpvminfo*(1M), *hpvmmgmt*(1M), *hpvmpubapi*(3)

hpvmresume(1M)

NAME

hpvmresume – Resume a virtual machine.

SYNOPSIS

`hpvmresume` { `-P` *vm-name* | `-p` *vm-number* } [`-F`] [`-s`] [`-Q`]

DESCRIPTION

The `hpvmresume` command causes the specified virtual machine to resume. The virtual machine must exist and be in the suspended state. The `hpvmresume` command verifies that the virtual machine can be allocated all of the required resources defined by its configuration file. If this is not possible, the virtual machine is not resumed.

Any of the following conditions can prevent the virtual machine from being resumed:

- The virtual machine is not suspended.
- The server has fewer CPUs than the virtual machine requires.
- The server has insufficient free memory.
- The server has insufficient CPU resources.
- The server has insufficient swap resources.
- Another virtual machine is using a specified nonshared backing device.
- The server is using a specified backing device.
- A specified backing device does not exist.
- A specified vswitch is not available.
- The vswitch must be created using the `hpvmnet` command before the guests using it can be resumed.
- The specified MAC address is in use.
- The specified guest is a distributed guest.

Only superusers can execute the `hpvmresume` command.

Options

The following options can be specified only once.

The `hpvmresume` command recognizes the following command-line options:

`-P` *vm-name*
: Specifies the name of the virtual machine to be resumed.

  You must specify either the `-P` or the `-p` option.

`-p` *vm-number*
: Specifies the number of the virtual machine to be booted. The *vm-number* is displayed by the `hpvmstatus` command.

  You must specify either the `-P` or the `-p` option.

`-F`
: Forces the virtual machine to skip all the resource checks. No warnings are issued.

  HP does not recommend using the `-F` option, because it can result in poor virtual machine performance, oversubscription, or data corruption, or it can cause the virtual machine to hang.

  The `-F` option is deprecated in Integrity Virtual Machines commands.

-s Examines the specified guest configuration and reports any errors or warnings that can prevent it from resuming. The guest is not resumed.

-Q Quietly performs the command. The default is to prompt for confirmation of the command before executing it.

RETURN VALUES

The command exits with one of the following values:

0: Successful completion.

1: One or more error conditions occurred.

DIAGNOSTICS

The `hpvmresume` command displays error messages on stderr for any of the following conditions:

- An invalid option is specified.
- The *vm-name* or *vm-number* does not exist, cannot be accessed, is not a virtual machine, or is corrupt.
- The virtual machine is in a state other than suspended and cannot be resumed.
- The virtual machine cannot boot at this time, because of detected resource complaints.
- The `hpvmresume` command and the Integrity VM software are at different version levels.

EXAMPLES

Start the virtual machine called myguest:

```
# hpvmresume -P myguest
```

The following are sample warning messages returned when `hpvmresume` is executed with various configuration problems on guest `myguest`:

```
Warning 1: Guest needs more vcpus than server supports.
Warning 2: Insufficient free memory for guest.
Warning 3: Insufficient swap resource for guest.
Warning 4: Insufficient cpu resource for guest.
These problems will prevent HPVM guest myguest from booting.
```

OBSOLESCENCE

The `-F` option is deprecated in Integrity Virtual Machines commands. This flag might be removed in a future release of Integrity Virtual Machines. HP recommends use of the command with the `-s` option, then resolving any errors or warnings that result before executing the command without the `-s` option and without the `-F` option.

AUTHOR

The `hpvmresume` command was developed by HP.

SEE ALSO

On the VM Host:

hpvm(5), *hpvmclone*(1M), *hpvmcollect*(1M), *hpvmconsole*(1M), *hpvmcreate*(1M), *hpvmdevinfo*(1M), *hpvmdevmgmt*(1M), *hpvmdevtranslate*(1M), *hpvmhostgdev*(1M), *hpvmhostrdev*(1M)*hpvminfo*(1M), *hpvmmigrate*(1M), *hpvmmodify*(1M), *hpvmmove_suspend*(1M), *hpvmnet*(1M), *hpvmpubapi*(3), *hpvmremove*(1M), *hpvmresources*(5), *hpvmresume*(1M), *hpvmsar*(1M), *hpvmsg_move*(1M), *hpvmsg_package*(1M), *hpvmstatus*(1M), *hpvmstop*(1M), *hpvmsuspend*(1M), *hpvmupgrade*(1M),*p2vassist*(1M)

On the Integrity VM guest:

hpvmcollect(1M), *hpvminfo*(1M), *hpvmmgmt*(1M), *hpvmpubapi*(3)

hpvmsar(1M)

NAME

hpvmsar -- Integrity VM system activity reporter.

SYNOPSIS

`hpvmsar` { `-a` | `-A` | *vm-name*[...] } [`-M`] [`-s` *rate*] [`-n` *count*] [`-h` *rate*]
`hpvmsar -I` { `-a` | `-A` | *vm-name*[...] } [`-M`] [`-m` *mode*] [`-s` *rate*] [`-n` *count*] [`-h` *rate*]
`hpvmsar -H` [`-s` *rate*] [`-n` *count*] [`-h` *rate*]
`hpvmsar -F` { `-a` | `-A` | *vm-name*[...] } [`-M`] [`-s` *rate*] [`-n` *count*] [`-h` *rate*]
`hpvmsar -G` { `-a` | `-A` | *vm-name*[...] } [`-s` *rate*] [`-n` *count*] [`-h` *rate*]
`hpvmsar -N` { `-a` | `-A` | *vm-name*[...] } [`-s` *rate*] [`-n` *count*] [`-h` *rate*]
`hpvmsar -S` *vswitch-name*[...] [`-s` *rate*] [`-n` *count*] [`-h` *rate*]
`hpvmsar -D` { `-a` | `-A` | *vm-name*[...] } [`-s` *rate*] [`-n` *count*] [`-h` *rate*]

DESCRIPTION

The `hpvmsar` command displays system performance statistics for the running local host and running local virtual machines. The statistics displayed come from the running guests, the VM host and Integrity VM itself.

With the `-a` option, the `hpvmsar` command displays information about all running guests. The display is reformatted as guests are started or stopped.

With the `-A` option, the `hpvmsar` command displays information about all guests, whether they are running or not. Guests that are not running display as blank entries. Without these options, the `hpvmsar` command displays information about the specified virtual machines.

The options change the number, frequency and rate at which the display headers are printed in the text displays.

The maximum number of guests that can be displayed at one time is 512. The maximum number of vswitches that can be displayed on a single display is 512.

Options

The `hpvmsar` command recognizes the following command-line options and arguments:

`-s` *rate*
: Collects and displays data every *rate* seconds. The default is 1 second.

`-n` *count*
: Counts how many samples are taken, with zero (0) indicating to sample forever. The default is 5 samples in text mode and 0 (infinity) in graphical mode.

`-h` *rate*
: In text mode, sets how frequently the header is displayed. Header information indicates what each column of text represents. The default is to display the header every 8th sample

`-M`
: Displays separate information for each virtual CPU (vCPU) individually. By default, it is displayed per guest, as an average of all vCPUs.

`-m` *mode*
: When displaying information graphically, `hpvmsar` offers multiple viewing modes. The initial mode is selected using the `-m` option, with the default being mode 0, stacked up. This option has no effect if DISPLAY is not set. The display mode can be changed by pressing any key and cycles through all available modes. The following modes are provided:

- 0: Stacked up values — Stacks up busy time, idle time, and wait time vertically. Busy time is on the bottom and wait time is on the top of the column.
- 1: Cumulative view — Stacks the busy time of multiple guests from the left of the window and the wait time of multiple guests from the right of the window. This mode makes it possible to see how multiple guest share the total available host time.
- 2: History — Displays a graph of historical values for busy and wait time, showing the evolution of these values over time.
- 3: Raw data — Displays one bar for each individual value.

`[-a | -A | vm-name [...]]`

The `-a` option displays only running guests. The `-A` option displays all guests, including those that are not running. If you specify one or more virtual machine names, `hpvmsar` displays information about the specified virtual machines.

DISPLAY OPTIONS

Subsets of data to be printed are specified by options:

Default Guest & Host Cpu usage Display.

The command displays the vCpu and host Cpu utilization statistics graphically using X11 only when the DISPLAY environment is defined; otherwise it displays them as text. The –`M` option displays all counts by vCPU.

In graphical mode, the `hpvmsar` command displays information using the following conventions:

- Blue: Busy time - which counts whenever the guest runs. It includes the time necessary to run the guest operating system, guest applications, and virtual machine monitor on behalf of the guest.
- Green: Idle time - which counts when the guest has no activity and the CPU was returned to the host. The same time may be counted as idle in multiple guest simultaneously, therefore; the sum of idle times may exceed 100%.
- Orange: Wait time - which counts when the guest was preempted but had some activity pending. It indicates that the guest could have used the time, but was prevented, because it was preempted.
- Grey: Host time - which counts when the host is running, as seen from the guest's point of view. Under normal conditions, host time is approximately the sum of idle time and wait time. Differences might arise when context switch-time or interrupt time becomes significant.

`-D`

Integrity VM Storage Metrics activity by guest

Display for the sample interval. The VM Host and guest storage disk block devices are combined and presented on a single line of output, along with performance and utilization metrics.

NOTE: This option is for HP-UX guests only.

- GDISK — Name of the guest disk in persistent dsf format.
- G%BUSY – As seen from the guest, this is the percentage of time, over the interval, that the disk device was busy servicing an I/O request.
- GAVQUE —Average number of I/O requests outstanding for this guest disk as seen from the guests point of view.
- GR+W/s — This is the number of read and write I/O transfers performed per second as seen from the guest.
- GBLKS/s — As seen from the guest, this is the number of 512K blocks per second of storage data that has been transferred to and from the guest disk device.
- GAVWAIT — As seen from the guest, this is the guest's average time (in milliseconds) that transfer requests waited idly on queue for the device.
- GAVSERV — As seen from the guest, average time (in milliseconds) to service each transfer request (includes seek, rotational latency, and data transfer times) for the device.
- HDISK — Name of the guest disk in persistent dsf format.
- H%BUSY — As seen from the guest, this is the percentage of time, over the interval, that the disk device was busy servicing an I/O request.
- HAVQUE — Average number of I/O requests outstanding for this guest disk as seen from the guests point of view.
- HR+W/s — This is the number of read and write I/O transfers performed per second as seen from the VM Host.
- HBLKS/s — As seen from the VM Host, this is the number of 512K blocks per second of storage data that has been transferred to and from the guest disk device.
- HAVWAIT — As seen from the VM Host, this is the VM Host's average time (in milliseconds) that transfer requests waited idly on queue for the device.
- HAVSERV — As seen from the VM Host, average time (in milliseconds) to service each transfer request (includes seek, rotational latency, and data transfer times) for the device.

`-F`

Integrity VM core Memory Metrics activity by Guest Display for the sample interval. The `-M` option displays all counts by vCpu.

- DTLB — Number of data TLB entries needed by data access that were missing and had to be inserted.
- RFI — Number of Return From Interrupt instructions.
- SET_PSR — Number of time the PSR was reset during guest execution.
- SET_RR — Number of time the RR was reset during guest execution.
- TPA — Number of TPAs completed for the guest.
- ITLB — Number of instruction TLB entries needed by guest access were missing and had to be inserted.
- ITC_D — Number of va/pa address inserts into translation cache.
- PTC_GA — Number of PTC in guest vCpu. PTC_GA is similar to PTC_G.
- DKEYMISS — Number of Data Key Misses for the guest.

`-G`

Display the Guest Dynamic Memory, Swapping, and Paging Activity for the sample interval.

NOTE: This option is for HP-UX guests only.

- GTOTMEM(MB) — Number of megabytes with which the guest was booted.
- HDYNRCLM(MB) — Number of megabytes reclaimed by the VM Host from this guest by dynamic memory. This memory is able to be distributed to guests and host by dynamic memory based upon memory utilization of VM Host and guests and dynamic memory settings. If AMR is enabled, this distribution is done automatically.
- GCURMEM(MB) — Actual amount of memory that is claimed and allocated to the guest.
- GCURFREE(MB) — Actual amount of free memory on the guest.
- GSWAP — Number of processes being swapped in and swapped out per second on the guest.
- GPAGE — Number of pages paged in and out per second on the guest.
- GADDRTMISS/s — Number of address translation faults per second occurring on the guest. This is the sum of vhptSubPageMiss, trptMiss, spptMiss and lpptMiss counts.

`-H`

Report the Memory usage for various Integrity VM components, including dynamic memory.

NOTE: This option is for HP-UX guests only.

- HCURMEM(MB) — Current megabytes of memory booted on the VM Host.
- HFREE(MB) — Megabytes of memory that is currently free and available for use by the VM Host or guests using dynamic memory distribution.
- HSWAP — Number of process swapins and swapouts per second on the VM Host.
- HPAGE — Number of pages paged in and out per second on the VM Host.
- HADDRTMISS/s — Number of address translation faults per second on the VM Host.
- VADDRTMISS — Number of address translation faults per second for Integrity VM kernel module (vmm).
- HDYNRCLM(MB) — Unused memory reclaimed from all booted guests for use by dynamic memory. This memory is able to be distributed to guests and VM Host by dynamic memory based upon memory utilization of host and guests and dynamic memory settings. If AMR is enabled, this distribution is done automatically.
- GTOTFREE(MB) — Current number of megabytes free for use within all booted guests.

`-I` Report Guest Interrupt Activity. The `-M` option displays all counts by vCPU.

INTS/VCPU — Number of guest interrupts for all vCPUs. With the `-M` option, a guest's interrupt activity is displayed for each vCPU.
EINTS/VCPU — Number of guest external interrupts for all vCPUs. With the `-M` option, a guest's external interrupt activity is displayed for each vCPU.
XMOGS/s — Number of VM Host context switches, per second, between the Integrity VM kernel module (vmm) and the hpvmdvr driver.

`-N` Guest Network traffic by vswitch Activity Display.

- GUEST — Guest these vswitch statistics are for.
- VSWITCH — Vswitch name.
- VSTATE — Current state of vswitch. Up/Down.
- pNIC — Vswitch's pNIC.
- SPEED — Vswitch's operating speed in megabits/second.
- MTU — Vswitch's maximum transfer unit.
- TX — Transmit rate in megabytes/second for the named guest/vswitch over interval specified.
- RX Mb/s — Receive rate in megabytes/second for the named guest/vswitch over interval specified.

- TXDROPS — Number network transmit packets dropped by this vswitch for the given guest over interval specified.
- RXDROPS — Number network receive packets dropped by this vswitch for the given guest over interval specified.

`-S`
Vswitch Network traffic Activity by Port Display

- VSWITCH — Vswitch name
- VSTATE — Vswitch state. Up or Down.
- PORT — Vswitch port number to which the following statistics pertain.
- UVLANID — Untagged VLAN ID
- TVLAN — Tagged VLAN ID
- TX — Transmit rate in megabytes/second for named guest/vswitch/port over interval specified.
- RX Mb/s — Receive rate in megabytes/second for named guest/vswitch/port over interval specified.
- TXDROPS — Number network transmit packet dropped by the port over interval specified.
- RXDROPS — Number network receive packet dropped by the port over interval specified.

NOTE: The Integrity VM guest kit for HP-UX guests must be installed to receive output with the `-G` or `-D` options.

HP OpenVMS guests do not display data with the `-G` or `-D` options.

Only AVIO statistics are displayed for guests with the `-N` and `-S` options. VIO statistics are not shown.

RETURN VALUES

The `hpvmsar` command exits with one of the following values:

0: Successful completion.

1: One or more error conditions occurred.

DIAGNOSTICS

The `hpvmsar` command displays the following messages:

- Invalid display mode x (max y): The display mode is not supported by this version of `hpvmsar`.
- Missing argument to option (followed by usage): A required argument was not provided.
- Missing option (followed by usage): An option was not recognized.
- Nothing to display, exiting: No `-a` specified and no guest either.
- Unable to find guest X: The guest name is not spelled correctly.
- Unable to allocate memory for history: Really low memory on the VM Host.

EXAMPLES

The following displays the vCPU data for all guests, running and idle. The dashes indicate a guest not booted or at the EFI prompt.

```
# hpvmsar -A
HP-UX star B.11.31 U ia64    10/22/10

10:31:27   NAME          %GUEST   %IDLE   %WAIT   %HOST
10:31:29   ux2             6.52   96.76    0.04   96.92
           ux1             3.10   96.86    0.00   96.87
10:31:30   ux2            12.03   87.01    0.09   88.09
           ux1             3.28   96.72    0.00   96.73
10:31:31   ux2            13.58   87.04    0.13   87.39
           ux1             3.68   96.28    0.01   96.29
10:31:32   ux2             3.04   93.09    0.00   93.12
           ux1            11.99   87.08    0.47   87.56
10:31:33   ux2             3.04   97.41    0.00   97.42
           ux1            79.05   14.80    3.62   21.85
```

The following example shows that guest `ux1` has two disks assigned to it. `disk1` is in light use and `disk8` is not in use at all.

```
# hpvmsar -D ux1
HP-UX new4 B.11.31 U ia64    10/21/10

17:43:42   GUEST      GDISK  G%BUSY  GAVQUE  GR+W/s GBLKS/s GAVWAIT GAVSERV   HDISK  H%BUSY  HAVQUE  HR+W/s
HBLKS/s HAVWAIT HAVSERV
17:43:43   ux1        disk1    2.00    0.50      13     134    0.00    1.11  disk12    1.00    0.50      13
  134    0.00    0.98
17:43:43   ux1        disk8    0.00    0.00       0       0    0.00    0.00  disk14    0.00    0.00       0
    0    0.00    0.00
17:43:44   ux1        disk1    0.00    0.50       2           0.00    0.28  disk12    0.00    0.50       2
     0.00    0.12
17:43:44   ux1        disk8    0.00    0.00       0       0    0.00    0.00  disk14    0.00    0.00       0
    0    0.00    0.00
17:43:45   ux1        disk1    0.00    0.00       0       0    0.00    0.00  disk12    0.00    0.00       0
    0    0.00    0.00
```

The following example displays vCPU data about all running guests at the rate of every 5 seconds:

```
# hpvmsar -s 5 -a
```

The following example displays data for all guests, even those not running, in the cumulative view:

```
# hpvmsar -m 1 -A
```

AUTHORS

The `hpvmsar` command was developed by HP.

SEE ALSO

On the VM Host:

hpvm(5), *hpvmclone*(1M), *hpvmcollect*(1M), *hpvmconsole*(1M), *hpvmcreate*(1M), *hpvmdevinfo*(1M), *hpvmdevmgmt*(1M), *hpvmdevtranslate*(1M), *hpvmhostgdev*(1M), *hpvmhostrdev*(1M)*hpvminfo*(1M), *hpvmmigrate*(1M), *hpvmmodify*(1M), *hpvmmove_suspend*(1M), *hpvmnet*(1M), *hpvmnvram*(1M),*hpvmpubapi*(3), *hpvmremove*(1M), *hpvmresources*(5),*hpvmresume*(1M), *hpvmsg_move*(1M), *hpvmsg_package*(1M), *hpvmstatus*(1M), *hpvmstop*(1M), *hpvmsuspend*(1M), *hpvmupgrade*(1M),*p2vassist*(1M)

On the Integrity VM guest:

hpvmcollect(1M), *hpvminfo*(1M), *hpvmmgmt*(1M), *hpvmpubapi*(3)

hpvmsg_move(1M)

NAME

hpvmsg_move – Initiate an online migration of a virtual machine that has previously been associated with a Serviceguard package.

SYNOPSIS

`hpvmsg_move` [`-v`] [`-h` *`target-node`* `|` `-n` *`target-node`*] [`-P` *`vm-name`*]

DESCRIPTION

The `hpvmsg_move` command initiates an online migration (move) of a virtual machine that has previously been associated with a Serviceguard package. The command moves the virtual machine and package named in the *`vm-name`* argument from the current running node to the *`target-node`* specified with the `-h` option or the `-n` option.

The `hpvmsg_move` command disables package switching during the online migration of the virtual machine, but does not halt the package. It keeps the package from being switched to a new node in the event the current node fails. Following a successful online migration, the package switching is re-enabled.

The `hpvmsg_move` command is limited to the same restrictions as specified by the `hpvmmigrate` command. To be compatible for online migration, both the source VM Host and the target VM Host must have the same processor family (as reported by the `machinfo` command). To maintain active guest network connectivity, configure a vswitch with the same name and connected to the same subnet on the target VM Host. Also, only whole disk backing storage consisting of SAN LUNs and null backing store DVD devices are supported for online guest migration storage.

Options

The `hpvmsg_move` command recognizes the following command-line options and arguments:

`-v`	Displays verbose information.
`-h` *`target-node`*	Specifies the node to which the virtual machine and package online migrate. The `-h` option and the `-n` option are mutually exclusive.
`-n` *`target-node`*	Specifies the node to which the virtual machine and package online migrate. This option provides the same functionality as the `-h` option. The `-h` option and the `-n` option are mutually exclusive.
`-P` *`vm-name`*	Specifies the virtual machine and package to be online migrated.

RETURN VALUES

The command exits with one of the following values:

0: Successful completion.

1: One or more error conditions occurred.

2: One or more operational failures occurred.

3: The package is not a virtual machine package.

4: The package is not up and running at the operating system.

5: The target node is not a Serviceguard node.

6: The target node is not up and running.

7: The backing storage is not compatible with online migration.

8: The package is not running on the local node.

9: Attempt to re-enable package on local node failed.

DIAGNOSTICS

The `hpvmsg_move` command displays error messages on stderr for errors and warnings encountered during execution.

EXAMPLE

The following example moves a Serviceguard package online from one node to another.

```
# hpvmsg_move -n hpvm-sg1-h1 -P hpvm-sg1-g4
Moving (online) Serviceguard package hpvm-sg1-g4 from node hpvm-sg1-h2 to node hpvm-sg1-h1
cmmodpkg: Completed successfully on all packages specified
hpvmmigrate: Connected to target host using 'hpvm-sg1-h1'
hpvmmigrate: Starting guest 'hpvm-sg1-g4' on target host 'hpvm-sg1-h1'
    (C) Copyright 2000 - 2009 Hewlett-Packard Development Company, L.P.
    Opening minor device and creating guest machine container
    Creation of VM, minor device 2
    Initialize guest memory mapping tables
    Starting event polling thread

    Online migration initiated by source 'hpvm-sg1-h2.hpvm.usa.hp.com' (.118.88.228)

hpvmmigrate: Init phase completed successfully.
hpvmmigrate: Copy phase completed successfully.
hpvmmigrate: I/O quiesce phase completed successfully.
hpvmmigrate: Frozen phase completed successfully.
hpvmmigrate: Guest migrated successfully.
Waiting for Serviceguard to detect node change....
Package hpvm-sg1-g4 is already enabled on node hpvm-sg1-h1
cmmodpkg: Completed successfully on all packages specified
Running package hpvm-sg1-g4 on node hpvm-sg1-h1
Successfully started package hpvm-sg1-g4 on node hpvm-sg1-h1
cmrunpkg: All specified packages are running
cmmodpkg: Completed successfully on all packages specified
Package hpvm-sg1-g4 is already enabled on node hpvm-sg1-h1
cmmodpkg: Completed successfully on all packages specified
```

AUTHORS

The `hpvmsg_move` command was developed by HP.

SEE ALSO

On the VM Host:

hpvm(5), *hpvmclone*(1M), *hpvmcollect*(1M), *hpvmconsole*(1M), *hpvmcreate*(1M), *hpvmdevinfo*(1M), *hpvmdevmgmt*(1M), *hpvmdevtranslate*(1M), *hpvmhostgdev*(1M), *hpvmhostrdev*(1M), *hpvminfo*(1M), *hpvmmigrate*(1M), *hpvmmodify*(1M), *hpvmmove_suspend*(1M), *hpvmnet*(1M), *hpvmnvram*(1M), *hpvmpubapi*(3), *hpvmremove*(1M), *hpvmresources*(5), *hpvmresume*(1M), *hpvmsar*(1M), *hpvmsg_package*(1M), *hpvmstatus*(1M), *hpvmstop*(1M), *hpvmsuspend*(1M), *hpvmupgrade*(1M)*p2vassist*(1M)

On the Integrity VM guest:

hpvmcollect(1M), *hpvminfo*(1M), *hpvmmgmt*(1M), *hpvmpubapi*(3)

hpvmsg_package(1M)

NAME

hpvmsg_package -- Assist the user who is developing and managing a set of Serviceguard package configurations.

SYNOPSIS

`hpvmsg_package` [`-V` | `-Q` | `-L` | `-s` | `-U`] [`-m` { `0` | `1` }]
[`-x` {*file* | *name*= [*value*] }]... [`-n` *node* [, *node*] . . .]... [`-P` *vm-name*]

DESCRIPTION

The `hpvmsg_package` command assists the user who is developing and managing a set of Serviceguard package configurations and associated toolkit files. The command creates these configurations by performing the steps required to associate a virtual machine to a Serviceguard failover package. These steps include:

- Verifying the Serviceguard cluster environment
- Verifying the virtual machine environment
- Identifying the Serviceguard modules and attribute values
- Creating a Serviceguard package configuration file
- Distributing the configuration to cluster nodes
- Checking, manual customization, and application of the configuration to the cluster
- Configuring Integrity VM cluster and system resources
- Optionally configure guest application monitoring

Options

The `hpvmsg_package` command recognizes the following options and arguments:

`-P` *vm-name*	Specifies the name of the virtual machine.
`-U`	Unpackage the target virtual machine.
`-L`	Create legacy packages. The default is modular packages.
`-V`	Displays detailed information.
`-Q`	Quietly performs the command using the default actions without additional prompts.
`-s`	Sanity checks the specific command, but does perform the requested action.
`-n` *node*, [*node*]	The node names, in priority order, where the package is allowed to run. The default is all nodes in the order listed, using the HP-UX `cmviewcl` command.
`-m` {0|1}	Enable or disable maintenance mode. Specify 1 to enable or 0 to disable.
`-x` *name*=*value*	Applies an extended option expression consisting of a target (name) and one of the following expressions:

- merge — Makes modification to the current configuration file.
- appmon[=[*name*]: [*run-command*]: [*monitor-timeout*]: [*service-restart*]: [*service-halt-timeout*]] — Configures application monitors where:

◦ *name* — the name for the application monitor (service_name).
◦ *run-command* — A full *pathname* to the application run command. See the HP-UX `cmappmgr` command.
◦ *monitor-timeout* — An application timeout value in seconds. See the HP-UX `cmappmgr` command.
◦ *service-restart* — An application monitor restart value (service_restart).
◦ *service-halt-timeout* — An application monitor halt timeout value in seconds (service_halt).
◦ *module-attribute-custom-value* — A Serviceguard module attributes and values. For example, logging_level=5
◦ *file* — A file containing any of the above attributes, one per line.

RETURN VALUES

The `hpvmsg_package` command returns the following values:

0: Successful completion.

1: Inappropriate command line or configuration error.

2: Extended option configuration error.

3: Extended option input data error.

5: Error during merge or creating configuration file.

6: Error configuring extended option.

7: Error setting distributed state or apply to cluster.

8: Error deleting package from cluster.

9: Error removing virtual machine distributed state.

DIAGNOSTICS

The `hpvmsg_package` command displays error messages on `stderr` for errors and warning encountered during execution.

EXAMPLE

The following example shows the HP Integrity Virtual Machines Serviceguard Toolkit script.

```
# hpvmsg_package -P hpvm-sg1-g1

This is the HP Virtual Machines Serviceguard Toolkit package creation
script.

This script will assist the user to develop and manage a set of
Serviceguard package configuration and associated toolkit template
files.

The templates generated by these scripts will handle many guest
configurations, but they are only templates and may not be
appropriate for your particular configuration needs. You are
encouraged to review and modify these template files, as needed,
for your particular environment.


Do you wish to continue? (y/n):y
```

```
[Virtual Machine Details]
Virtual Machine Name VM #  OS Type State
==================== ===== ======= ========
hpvm-sg1-g1              7 HPUX    On (OS)
[Storage Interface Details]
Guest                                Physical
Device  Adaptor    Bus Dev Ftn Tgt Lun Storage   Device
======= ========== === === === === === ========= =========================
disk    scsi         0   0   0   0   0 lv        /dev/vx/rdsk/g1-N-C-SCRATCH/g1-N-C-SCRATCH_lv
disk    scsi         0   0   0   1   0 file      /guest/g1-N-CF-SCRATCH/file
disk    scsi         0   0   0   2   0 disk      /dev/rdisk/disk978
disk    scsi         0   0   0   3   0 file      /guest/g1-N-DF-SCRATCH/file
disk    scsi         0   0   0   4   0 lv        /dev/g1-N-L-SCRATCH/rlvol1
disk    scsi         0   0   0   5   0 file      /guest/g1-N-LF-SCRATCH/file
disk    scsi         0   0   0   6   0 lv        /dev/vx/rdsk/g1-N-V-SCRATCH/g1-N-V-SCRATCH_lv
disk    scsi         0   0   0   7   0 file      /guest/g1-N-VF-SCRATCH/file
[Network Interface Details]
Interface Adaptor    Name/Num   PortNum Bus Dev Ftn Mac Address
========= ========== ========== ======= === === === =================
vswitch   lan        intl100    1         0   1   0 8a-56-ea-60-fb-6e
vswitch   lan        igelan     1         0   2   0 5e-04-af-bf-b3-25

Modify the packaged Virtual Machine Summarized above? (y/n):y

Checking the virtual machine and cluster configuration

The virtual machine hpvm-sg1-g1 is currently configured as a
Serviceguard package.  It is recommended that all existing packages be
re-packaged whenever any of the following changed:

   HP Integrity Virtual Machine and related virtualization products
   Serviceguard and related high availability products

   HP Integrity Virtual Machine configuration or environment
   Serviceguard cluster configuration or environment

Changes to any of these items may alter the functionality and behavior
of your package.

Continue modifying the package for this virtual machine? (y/n):y

Continuing with virtual machine and cluster analysis...

Determining package attributes and modules...

Creating modular style package files for virtual machine : hpvm-sg1-g1

Review and/or modify the package configuration file (optional)? (y/n):y
Invoking editor : emacs on file /etc/cmcluster/hpvm-sg1-g1/hpvm-sg1-g1.conf

Copy the package files to each cluster member? (y/n):y

Configure this virtual machine as a distributed guest? (y/n):y

The virtual machine has been successfully configured as a Serviceguard package.

Use cmcheckconf check the package configuration file (optional)? (y/n):y

Attempting to validate hpvm-sg1-g1.
The specified run script timeout is NO_TIMEOUT; using 1200 seconds as timeout value.
cmapplyconf will wait for this amount of time for the
script to complete before giving up.
Validation for package hpvm-sg1-g1 succeeded via /etc/cmcluster/scripts/mscripts/master_control_script.sh.
cmcheckconf: Verification completed with no errors found.
Use the cmapplyconf command to apply the configuration.

Apply the package configuration file to the cluster (optional)? (y/n):y

Attempting to validate hpvm-sg1-g1.
The specified run script timeout is NO_TIMEOUT; using 1200 seconds as timeout value.
cmapplyconf will wait for this amount of time for the
script to complete before giving up.
Validation for package hpvm-sg1-g1 succeeded via /etc/cmcluster/scripts/mscripts/master_control_script.sh.

Modify the package configuration ([y]/n)? y
Completed the cluster update

Please see the HP Integrity Virtual Machines documentation for additional
instructions on configuring Virtual Machines as Serviceguard packages.

Before running this package the following steps may need to be performed:

1. Review the files located in /etc/cmcluster/hpvm-sg1-g1/.
2. Add new LVM Volume Groups to the cluster configuration file.
3. Check the cluster and/or package configuration using the cmcheckconf command.
4. Apply the cluster and/or package configuration using the cmapplyconf command.
5. If it is running, stop the virtual machine using: hpvmstop -F -P hpvm-sg1-g1
6. Umount filesystems and deactivate volume groups used by the virtual machine
7. Start the package using: cmrunpkg hpvm-sg1-g1
```

AUTHORS

The `hpvmsg_package` command was developed by HP.

SEE ALSO

On the VM Host:

hpvm(5), *hpvmclone*(1M), *hpvmcollect*(1M), *hpvmconsole*(1M), *hpvmcreate*(1M), *hpvmdevinfo*(1M), *hpvmdevmgmt*(1M), *hpvmdevtranslate*(1M), *hpvmhostgdev*(1M), *hpvmhostrdev*(1M), *hpvminfo*(1M), *hpvmmigrate*(1M), *hpvmmodify*(1M), *hpvmmove_suspend*(1M), *hpvmnet*(1M), *hpvmnvram*(1M), *hpvmpubapi*(3), *hpvmresources*(5), *hpvmresume*(1M), *hpvmsar*(1M), *hpvmsg_move*(1M) , *hpvmstop*(1M), *hpvmsuspend*(1M),*hpvmupgrade*(1M)*p2vassist*(1M)

On the Integrity VM guest:

hpvmcollect(1M), *hpvminfo*(1M), *hpvmmgmt*(1M), *hpvmpubapi*(3)

hpvmstart(1M)

NAME

hpvmstart – Start a virtual machine.

SYNOPSIS

`hpvmstart` { `-P` *vm-name* | `-p` *vm-number* }[`-F` | `-s`][`-Q`]

DESCRIPTION

The `hpvmstart` command causes the specified virtual machine to start. The virtual machine must exist and be in the off state. The `hpvmstart` command verifies that the starting virtual machine can be allocated all of the required resources defined by its configuration file. If not, the virtual machine is not started.

Any of the following conditions can prevent the virtual machine from starting:

- The server has fewer CPUs than the virtual machine requires.
- The server has insufficient free memory.
- The server has insufficient CPU resources.
- The server has insufficient swap resources.
- Another virtual machine is using a specified nonshared backing device.
- The server is using a specified backing device.
- A specified backing device does not exist.
- A specified vswitch is not available. The vswitch must be created using the `hpvmnet` command before the guests using it can be started.
- The specified MAC address is in use.
- The specified guest is a distributed guest.

NOTE: Only superusers can execute the `hpvmstart` command.

Options

The following options can be specified only once.

The `hpvmstart` command recognizes the following command-line options and arguments:

`-P` *vm-name* Specifies the name of the virtual machine to be started.

You must specify either the `-P` or the `-p` option.

`-p` *vm-number* Specifies the number of the virtual machine to be booted. The *vm-number* is displayed by the `hpvmstatus` command.

You must specify either the `-P` or the `-p` option.

`-F` Forces the virtual machine to skip all the resource checks. No warnings are issued.

CAUTION: HP does not recommend using the `-F` option because it can result in poor virtual machine performance, oversubscription, or data corruption, or it can cause the virtual machine to hang.

The `-F` option is deprecated in Integrity Virtual Machines commands.

`-s` Examines the specified guest configuration and reports any errors or warnings that can prevent it from starting. The guest is not started.

`-Q` Quietly performs the command. The default is to prompt for confirmation of the command before executing it.

RETURN VALUES

The command exits with one of the following values:

0: Successful completion.

1: One or more error conditions occurred.

DIAGNOSTICS

The `hpvmstart` command displays error messages on stderr for any of the following conditions:

- An invalid option is specified.
- The `vm-name` or `vm-number` does not exist, cannot be accessed, is not a virtual machine, or is corrupt.
- The virtual machine is in a state other than off and cannot be started. Use the `hpvmstop` command to stop the virtual machine.
- The virtual machine cannot boot at this time because of detected resource complaints.
- The `hpvmstart` command and the Integrity VM software are at different version levels.

EXAMPLES

Start the virtual machine called myguest:

```
# hpvmstart -P myguest
```

The following are sample warning messages returned when `hpvmstart` is executed with various configuration problems on guest `myguest`:

```
Warning 1: Guest needs more vcpus than server supports.
Warning 2: Insufficient free memory for guest.
Warning 3: Insufficient swap resource for guest.
Warning 4: Insufficient cpu resource for guest.
Warning 5 on item /dev/rdisk/disk0: Device file '/dev/rdisk/disk0' in use by another guest.
Warning 6 on item /dev/vg00/rswap: Device file '/dev/vg00/rswap' in use by server.
Warning 7 on item /dev/rdisk/disk3 backing device does not exist.
Warning 8 on item /dev/rdisk/disk4: Device file '/dev/rdisk/disk4:' in use by another guest.
Warning 9 on item hostnet: MAC address in use for switch hostnet.
Warning 10 on item offnet: Vswitch offnet is not active.
Warning 11 on item badnet: 'badnet' backing device does not exist.
These problems will prevent HPVM guest myguest from booting.
```

OBSOLESCENCE

The `-F` option is deprecated in Integrity Virtual Machines commands. This flag might be removed in a future release of Integrity Virtual Machines. HP recommends use of the command with the `-s` option, then resolving any errors or warnings that result before executing the command without the `-s` option and without the `-F` option.

AUTHORS

The `hpvmstart` command was developed by HP.

SEE ALSO

On the VM Host:

hpvm(5), *hpvmclone*(1M), *hpvmcollect*(1M), *hpvmconsole*(1M), *hpvmcreate*(1M), *hpvmdevinfo*(1M), *hpvmdevmgmt*(1M), *hpvmdevtranslate*(1M), *hpvmhostgdev*(1M), *hpvmhostrdev*(1M), *hpvminfo*(1M), *hpvmmigrate*(1M), *hpvmmodify*(1M), *hpvmmove_suspend*(1M), *hpvmnet*(1M), *hpvmnvram*(1M), *hpvmpubapi*(3), *hpvmremove*(1M), *hpvmresources*(5), *hpvmresume*(1M), *hpvmsar*(1M), *hpvmsg_move*(1M), *hpvmsg_package*(1M), *hpvmstatus*(1M), *hpvmstop*(1M), *hpvmsuspend*(1M), *hpvmupgrade*(1M)*p2vassist*(1M)

On the Integrity VM guest:

hpvmcollect(1M), *hpvminfo*(1M), *hpvmmgmt*(1M), *hpvmpubapi*(3)

hpvmstatus(1M)

NAME

hpvmstatus – Display status information about one or more virtual machines.

SYNOPSIS

hpvmstatus [-V | -X | -M] [-v]
hpvmstatus { -P *vm-name* | -p *vm-number* } -D -L [-V | -X | -M]
[-v]
hpvmstatus -e { -P *vm-name* | -p *vm-number* } [-V | -X | -M] [-v]
hpvmstatus -i { -P *vm-name* | -p *vm-number* } [-V | -X | -M] [-v]
hpvmstatus -r { -P *vm-name* | -p *vm-number* } [-V | -X | -M] [-v]
hpvmstatus -d { -P *vm-name* | -p *vm-number* } [-V | -X | -M] [-v]
hpvmstatus -m [-M | -X]
hpvmstatus -S [-M | -X]
hpvmstatus -s [-M | -X]
hpvmstatus -C { -P *vm-name* | -p *vm-number* } [-V | -X | -M] [-v]
[-D]
hpvmstatus -A { -P *vm-name* | -p *vm-number* }

DESCRIPTION

The `hpvmstatus` command displays information about the operational state and virtual hardware configuration of the virtual machines on the VM Host. Information displayed by the `hpvmstatus` command includes the following:

- Version of the command (if you specify the `-v` option).
- Name of the virtual machine (limited to 20 characters in summary format).
- State of the virtual machine. The machine will be in one of the following states:
  - `On`: The virtual machine is "powered on." It may be at its console prompt, or it may have booted its operating system and be fully functional. This is the normal state of a running virtual machine.
  - `Off`: The virtual machine is fully halted.
  - `On (Rmt)`: The virtual machine is a Serviceguard-packaged VM that is running on another member of the cluster. The *`runsysid`* parameter specifies which member. Use the `hpvmstatus –m` to get a list of systems in the multi-server environment, MSE.
  - `Off (NR)`: This designates that the guest is not running and has been set to prevent starting using the not-runnable, (NR), attribute.
  - `Invalid`: The virtual machine configuration file is corrupted or invalid. The configuration file must be corrected before this virtual machine can be started.
- Running condition of the guest. The machine can be in one of the following conditions:
  - `On (EFI)`: The virtual machines is running normally in Extensible Firmware Interface (EFI).
  - `On (OS)`: The virtual machine is powered on and is running normally in the operating system.
  - `On (ATTN!)`: The virtual machine may need attention because it is not responding to interrupts.

- ◦ `On (MGT)`: This designates that the guest is the target of a migration between two Integrity VM servers.
  - ◦ `On (MGS)`: This designates that the guest is the source of a migration between two Integrity VM servers.
- Resources allocated to this virtual machine.
- Attributes assigned to this virtual machine.
- Dynamic memory information, if dynamic memory is enabled for this virtual machine.

The `hpvmstatus` command displays the active configuration for guests that are on, including the resource assignments that are currently in effect. For guests with a status of off, the command displays the configuration to be used when the guest in next booted.

The `hpvmstatus` command displays variety of information:

- To list all the virtual machines that are on the VM Host, enter the `hpvmstatus` command without the `-P`, `-p`, `-e`, or `-r` option.
- To display detailed information about a virtual machine, use the `-P` or `-p` option (without the `-e`, `-r`, or `-d` option) to specify the virtual machine.
- To display devices in the same format used on the command line, use the `-d` option.
- To display a virtual machine's log file, for either the VM Host or the specified virtual machine, include the `-e` option.
- To display the virtual machine's memory and CPU resource allocation and entitlement information, use the `-r` option.
- To display the mode the scheduler is in, use the `-S` option.

To obtain a display in machine-readable format, use the `-M` or `-X` option.

Only superusers can execute the `hpvmstatus` command.

Integrity VM allows the guest to have two configurations, one for the last started configuration and one for the next start configuration or deferred configuration. If the guest is running, the last started configuration is the one that the guest is currently using and the next start configuration is the one that will be used the next time the guest is started. Having two configurations allows the administrator to set up a different configuration for the next start while the guest is running.

To view the next start configuration, the use the `-D` option specifying a deferred start configuration. To view the last start configuration, use the `-L` option specifying the last start configuration.

NOTE: When guests are controlled by Serviceguard in an Integrity VM multi-server environment (MSE), the run status of the SG-controlled guest is distributed to all VM Hosts in the cluster. The guest is marked as “On (Rmt)” on all VM Hosts in the cluster and the MSE serverid of the VM Host running the guest is displayed by `hpvmstatus` as the `Rmt Host`. This allows administrator to be aware that the guest is running on a different server in the cluster. The following `hpvmstatus` command checks if a guest is running locally:

```
hpvmstatus –P guestname -M | awk -F: '$11 !~ /Off/ && $27 != "1" {print
"guest running locally " }'
```

This command displays the serverid of the Host running a guest remotely:

```
hpvmstatus -P guestname -M | awk -F: '$11 !~ /Off/ && $27 == "1" {print
"guest running on serverid " $26}'
```

The default, no options, `hpvmstatus` command output appears as follows:

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

```
config1                      1 HPUX    Off (NR)        1      5      1   512 MB         0
config2                      2 HPUX    Off             1      7      1     1 GB         0
guest1                       5 OPENVMS On (OS)         1      5      1     1 GB         2
```

The following fields are presented in this `hpvmstatus` command output:

- Virtual Machine Name: The unique guest name
- VM #: The unique guest number assigned by the system.
- OS Type: HPUX, OPENVMS
- State:
  - Off => guest is not started
  - Off (NR) => guest is not started and is not runnable
  - On => guest is started
  - On (EFI) => guest is started and running in EFI
  - On (OS) => guest is started and running in the operating system
  - On (RMT) => guest is started and running as a Serviceguard-packaged VM on another member of the cluster. See *`Runsysid`*.
  - On (ATTN)=> guest is started but needs attention
  - On (MGT) => guest is the target of a migration
  - On (MGS) => guest is the source of a migration
  - On (RMT) => guest is a Serviceguard guest running on a remote host.
- #VCPUs: Number of vcpus assigned to the guest
- #Devs: Number of devices assigned to the guest
- #Nets: Number of networks assigned to the guest
- Memory: Amount of memory assigned to the guest
- Rmt Host: The MSE *`serverid`* that is running the guest. If a guest is controlled by Serviceguard, the *`runsysid`* is set to the MSE *`serverid`* of the Integrity VM Host running the guest. If the guest is not running or no MSE is configured, this value is set to zero.

The `hpvmstatus -V` option displays the new attributes, which will be displayed after `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 newhost6.
Modify status            : Disabled
Not modify setby         : Migrate
Not modify reason        : Guest has been migrated to host newhost6.
Visible status           : Disabled
Not visible setby        : Migrate
Not visible reason       : Guest has been migrated to host newhost6.
```

Options

The following options can be specified only once.

The `hpvmstatus` command recognizes the following options and arguments:

`-v` Displays the version number of the `hpvmstatus` command. The version number is displayed first, followed by information specified by other options.

`-V` Displays detailed information (verbose mode) about the virtual machines, including the status of migrating virtual machines. For migrating virtual machines, the verbose status displays in which stage the migration is and percent completed for that stage. The verbose status also displays the source and target VM Host aliases and IP addresses.

The `-V`, `-M`, and `-X` options are mutually exclusive.

`-M` Displays verbose attribute and resource information in machine- readable format, including information on migrating virtual machines.

Individual fields are separated by one of three delimiters:

- The colon (:) separates each field and resource type.
- The semicolon (;) separates subfields of a resource type.
- The comma (,) separates individual items in a list of similar items.

The `-V`, `-M`, and `-X` options are mutually exclusive.

`-X` Displays verbose information about attribute and resource in XML format, including information on migrating virtual machines.

The `-V`, `-M`, and `-X` options are mutually exclusive.

`-P` *vm-name* Specifies the name of the virtual machine for which information is to be displayed.

The `-P` and `-p` options are mutually exclusive.

`-p` *vm-number* Specifies the number of the virtual machine for which information is to be displayed. The `vm_number` is assigned when a virtual machine is created and is displayed by the `hpvmstatus` command.

The `-P` and `-p` options are mutually exclusive.

`-D` Displays resource assignments that takes effect the next time the virtual machine is started (deferred mode).

`-L` Displays the changes from the current configuration.

`-e` Displays the event log for the VM Host or the specified virtual machine. The event log records all changes to virtual machine configurations.

`-i` This option, when used with the `-P` option, prints statistics collected by the monitor. Currently, these include vCPU percentage and durations over the lifetime of the guest.

`-r` Displays the resource entitlement information for the virtual machine or machines. There are two sections, one for virtual CPU entitlement information and one for virtual machine memory entitlement. The virtual CPU section includes the following information:

- #vCPUs: The number of virtual CPUs in this virtual machine.
- Entitlement: The amount of CPU entitlement this virtual machine can use per virtual CPU. Note that the displayed value might be slightly different than what was specified. For example, the value can be rounded down to the nearest whole percentage of CPU entitlement.
- Maximum: The maximum amount of CPU entitlement this virtual machine can use. Note that the displayed value might be slightly different than what was specified. For example, the value might be rounded up to the nearest whole percentage of CPU entitlement. If no maximum is set, the default is 100% or all the CPU cycles. For example, a 1500Mhz CPU displays 1500Mhz.

- Percent Usage: The percentage of the VM Host physical CPUs this virtual machine has used during the last interval period.
- Cumulative Usage: The number of VM Host CPU ticks this virtual machine has consumed since the virtual machine was booted.

When you specify a virtual machine, the `hpvmstatus` command displays the following information for each virtual CPU:

- Cumulative Usage: The number of ticks this virtual CPU has consumed since the virtual machine was booted.
- Guest percent: The CPU percentage that the guest has consumed.
- Host percent: The CPU percentage that the VM Host uses on behalf of the guest.
- Cycles achieved (in MHz).
- Sampling Interval: The time between samples.

The virtual machine memory section includes the following information:

- DynMem Min: The minimum memory that can be dynamically allocated to this virtual machine with the dynamic memory allocation capability or automatic memory reallocation (AMR).
- Memory Entitle: The value of the desired memory allocation for the virtual machine. It may be set manually or automatically (by AMR). : The amount of memory entitlement this virtual machine is guaranteed to have allocated to it, provided it has memory demand. This value is meaningful only if AMR is enabled for the virtual machine. Otherwise, the value is ignored
- DynMem Max: The maximum memory that can be dynamically allocated to this virtual machine with the dynamic memory allocation capability or AMR.
- DynMem Target: The value of the desired memory allocation for the virtual machine. It may be set manually or automatically (by AMR).
- DynMem Current: The actual, current memory allocated to the virtual machine.
- Comfort Min: The memory allocation required to relieve memory "pressure" in the virtual machine.
- Total Memory: The absolute maximum amount of memory this virtual machine may be allocated.
- Free Memory: Amount of free memory in the virtual machine (according to the operating system running there).
- Available Memory: Amount of memory allocated to the virtual machine's user processes but not locked. This memory is available for paging by the virtual machine's operating system.
- Memory Pressure: 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.
- AMR Chunk: The granularity of memory allocation used by AMR to increase or decrease that allocated to a virtual machine.

- AMR State: Indicator of the AMR state of a given VM. The following indicators are displayed:
- ◦ If the amr_enable attribute is not set, DISABLED is displayed.
  ◦ If the VM is not running on the VM Host, but the amr_enable attribute is set, ENABLED is displayed.
  ◦ If the VM is running and the amr_enable attribute is set, then one of two states is displayed: ACTIVE if the VM's memory entitlement is set or PENDING if the VM's memory entitlement is not set.

`-d` Displays the devices on the specified virtual machine in the same format used on the command line.

`-S` Reports the VM Host scheduler mode (CAPPED or NORMAL). If CAPPED, displays information about the controller process.

`-s` Displays the current VM Host resources.

`-m` Displays information about the multiserver environment, including the Serviceguard identifier, state, IP address, and host name. If the VM Host is not a Serviceguard server, the following message is displayed:

`No HPVM multi-server environment configured.`

`-C` Displays whether the guests prefer cell local memory (clm), interleaved memory (ilm) or none. Note that `none`, `ilm` or `cell` is displayed in the `Cell Prefer` column for guests that are not started. Once started, the cell number, or ilm is displayed.

`-A` Displays the guest configuration differences between the next start and the last start guest configurations. If there are no differences, the following messages is printed and the next start configuration replaces the current configuration:

`No differences were found. Copy has been removed.`

If only the last configuration is present, the following message is returned:

`No next start configuration found.`

RETURN VALUES

The `hpvmstatus` command exits with one of the following values:

0: Successful completion.

1: One or more error conditions occurred.

DIAGNOSTICS

The `hpvmstatus` command displays error messages on stderr for any of the following conditions:

- An invalid option is specified.
- An invalid value is specified for an option.
- The *`vm-name`* or *`vm-number`* does not exist, cannot be accessed, is not a virtual machine, or is corrupt.
- A value was omitted for an argument that requires one, or a value was supplied for an argument that does not take one.
- The `hpvmstatus` command and the Integrity VM software are not at the same version levels.

EXAMPLES

Summarize information about all the virtual machines on the VM Host:

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

Display detailed information about a virtual machine named guest1.

```
# hpvmstatus -P guest1 -V
[Virtual Machine Details]
Virtual Machine Name    : guest1
Virtual Machine UUID    : bf4c3714-df65-11da-bd4e-00306e4a93e6
Virtual Machine ID      : 5
Virtual Machine Label   :
VM's Model Name         : server Integrity Virtual Machine
VM's Serial Number      : VM00619004
VM's Config Version     : 4.3.0
VM's Config Label       : HPVM B.04.30 13h00m11s EST
Operating system        : HPUX
OS Version Number       :
State                   : On (OS)
Start type              : Manual
Console type            : vt100-plus
Guest's hostname       :
Guest's IPv4 address    : 106..5.24
Guest's IPv6 address    : fe80::ee8d:40ff:fe58:4d28
EFI Location            : /opt/hpvm/guest-images/common/efi
Pattern File location   : /opt/hpvm/guest-images/common/patterns.vmmpat
...
```

Display the attributes and resources attached to the virtual machine config2:

```
# hpvmstatus -P config2

[Virtual Machine Details]
Virtual Machine Name VM #  OS Type State
==================== ===== ======= ========
config2                  2 HPUX    On (OS)

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

[Virtual CPU Details]
#vCPUs Entitlement Maximum
====== =========== =======
     1       10.0%  100.0%

[Memory Details]
Total    Reserved
Memory   Memory
=======  ========
    5 GB     64 MB

[Dynamic Memory Information]
Minimum     Target      Memory      Maximum
Memory      Memory      Entitlement Memory
=========== =========== =========== ===========
    512 MB     2106 MB        2 GB     5120 MB
```

```
[Storage Interface Details]
Guest                                   Physical
Device  Adaptor    Bus Dev Ftn Tgt Lun Storage   Device
======= ========== === === === === === ========= =========================
disk    scsi         0   1   0   0   0 disk      /dev/rdisk/disk00

[Network Interface Details]
Interface Adaptor    Name/Num   PortNum Bus Dev Ftn Mac Address
========= ========== ========== ======= === === === =================
vswitch   avio_lan   192net     1         0   1   0 de-ad-be-ef-17-67
vswitch  lan        15net      1         0   2   0 66-c7-a6-2d-d6-01
vswitch   lan        192net     3         0   3   0 be-ef-de-ad-17-67
vswitch   lan        localnet   1         0   4   0 be-ef-17-67-de-ad
[Misc Interface Details]
Guest                                   Physical
Device  Adaptor    Bus Dev Ftn Tgt Lun Storage   Device
======= ========== === === === === === ========= =========================
serial  com1                           tty       console
```

Display information about a gWLM managed VM Host:

```
# hpvmstatus -S
HPVM scheduler is running in CAPPED mode.

 Controller process information:
PID=13456,NAME=gWLM,HOST=acms.work.com,URL=http://acms.work.com:280


NONVM group entitlement information:
                           Percent       Cumulative
#VCPUs Entitlement Maximum   Usage            Usage
====== =========== ======= ======= ================
     0      108MHz 1300MHz    1.9%           469302

Startable virtual machines:
vm1,vm2,vm3,vm4,vm5,vm6
```

Display the system resources on the VM Host system:

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

        Processor speed = 900 Mhz
        Total physical memory = 6132 Mbytes
        Total number of processors = 2
        Available memory = 2557 Mbytes
        Available swap space = 1541 Mbytes
        Maximum vcpus for an HP-UX virtual machine = 2
        Available entitlement for a 1 way virtual machine = 900 Mhz
        Available entitlement for a 2 way virtual machine = 810 Mhz
```

Display the Serviceguard server information on the VM Host system:

```
# hpvmstatus -m
HPVM Multi-server environment
        Server group name       : HPVM-SG-hpvm_sg_1131
        Server group uuid       : 45e2a680-af14-11dc-9e00-00306e4a831c
        This server's identifier: 2

        Server_id               : 2
        Server_state            : 1
        Server_ipaddr           : 10.0.128.1
        Server_hostname         : troop
        Server_uuid             : 12345678-abcd-1234-abcd-123456789012
        Server_physical_uuid    : 58d03f5d-79ed-11d9-b720-17c097e9e0d0

        Server_id               : 1
        Server_state            : 1
```

```
        Server_ipaddr            : 10.0.128.159
        Server_hostname          : company
        Server_uuid              : 23456789-abcd-5678-efgh-123456789012
        Server_physical_uuid     : 560ef4fb-7e-11d7-86df-c7749ef083ae
```

Display the devices on the specified virtual machine in the same format used on the command line:

```
# hpvmstatus -P hpvm0014 -d
 [Virtual Machine Devices]

 [Storage Interface Details]
 disk:scsi::0,0,1:lv:/dev/vg01/rlv2

 [Network Interface Details]
 network:lan:0,1,0x56A3E9D74099:vswitch:myswitch

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

Display whether the guests prefer cell local memory (clm), interleaved memory (ilm) or none. Note that `none`, `ilm` or `cell` are displayed in the `Cell Prefer` column for guests that are not started. Once started, the cell number, or ilm is displayed.

Memory listed as `Other Cell` represents memory allocated outside of the preferred (home) cell. If there is insufficient clm, memory can be allocated across all cells. In this case, `ilm` is the `Cell Prefer` entry and the amount of ilm and clm allocated is displayed in the `Interleave` and `Other Cell` columns accordingly.

```
#hpvmstatus -C
# hpvmstatus -c
 [Guest Cell Local Memory Usage]
                               Cell         MB         MB           MB
 Virtual Machine Name VM #  Prefer Interleave  Home Cell Other Cell
 ==================== ===== ====== ========== ========== ==========
lp0                       1   none          0          0          0
lp1                       2      1         15      12371          2
lp2                       3   cell          0          0          0
lp3                       4    ilm         15          0      32783
```

AUTHORS

The `hpvmstatus` command was developed by the HP.

SEE ALSO

On the VM Host:

hpvm(5), *hpvmclone*(1M), *hpvmcollect*(1M), *hpvmconsole*(1M), *hpvmcreate*(1M), *hpvmdevinfo*(1M), *hpvmdevmgmt*(1M), *hpvmdevtranslate*(1M), *hpvmhostgdev*(1M), *hpvmhostrdev*(1M), *hpvminfo*(1M), *hpvmmigrate*(1M), *hpvmmodify*(1M), *hpvmmove_suspend*(1M), *hpvmnet*(1M), *hpvmnvram*(1M), *hpvmpubapi*(3), *hpvmremove*(1M), *hpvmresources*(5), *hpvmresume*(1M), *hpvmsar*(1M), *hpvmsg_move*(1M), *hpvmsg_package*(1M), *hpvmstart*(1M), *hpvmstop*(1M), *hpvmsuspend*(1M), *hpvmupgrade*(1M), *p2vassist*(1M)

On the Integrity VM guest:

hpvmcollect(1M), *hpvminfo*(1M), *hpvmmgmt*(1M), *hpvmpubapi*(3)

hpvmstop(1M)

NAME

hpvmstop -- Stop a virtual machine.

SYNOPSIS

`hpvmstop` { `-P` *vm-name* | `-p` *vm-number* | `-a` } [`-h` | `-g`] [`-F`] [`-q`] [`-Q`]

DESCRIPTION

The `hpvmstop` command stops a running virtual machine by simulating the operations performed at the system console on a physical system. The command can perform a hard stop, which functions like a power failure, or a graceful stop, in which the guest operating system receives notification and time to perform cleanup operations before the stop.

If the `hpvmstop` command returns an error message, the specified virtual machine is shut down.

The `hpvmstop` command does not create a crash dump, and automatic restart is not performed.

Unintentional use of the `hpvmstop` command has serious consequences; therefore, the user is prompted to confirm the operation unless the `-F` (force) option is specified.

Only superusers can execute the `hpvmstop` command.

Options

The following options can be specified only once.

The `hpvmstop` command recognizes the following command-line option and argument:

`-P` *vm-name*
Specifies the unique name of the virtual machine to be stopped.

You must specify the `-P` , `-p`, or `-a` option.

`-p` *vm-number*
Specifies the unique number of the virtual machine to be stopped. The *vm_number* is displayed by the `hpvmstatus` command.

You must specify the `-P` , `-p`, or `-a` option.

`-a`
Stops all active guests.

You must specify the `-P` , `-p`, or `-a` option. When you specify the `-a` option, you must also specify the `-F` option.

`-g`
Specifies that a graceful shutdown be performed within the specified or default *graceful_stop_timeout* period. If the timeout period expires before the graceful shutdown is complete, a hard stop is performed. The guest operating system is notified of an imminent power failure, which gives it time to perform cleanup operations. HP recommends stopping virtual machines using their native operating system commands.

NOTE: To set a graceful shutdown, use the `hpvmmodify -x graceful_stop_timeout={0|`*number*`}` command. See `hpvmmodify(1)` for more information.

The `-h` and `-g` options are mutually exclusive.

`-h`
Performs a hard stop, which is equivalent to a power failure. This is the default action. The guest operating system receives no notice and thus no opportunity to clean up. In these circumstances, the guest operating system does not create a crash dump, and automatic restart is not performed. HP recommends that you stop virtual machines by using their native operating system commands.

NOTE: If neither `-g` nor `-h` are specified, a hard stop is performed.

The `-h` and `-g` options are mutually exclusive.

`-F` Specifies the force option. Omits the confirmation dialog before resetting the virtual machine. This option is intended for use by scripts and other noninteractive applications.

The `-F` option is deprecated in Integrity Virtual Machines commands.

`-q` Makes certain scripted operations less verbose (quiet mode).

`-Q` Quietly performs the command. The default is to prompt for confirmation of the command before performing it.

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.

RETURN VALUES

The `hpvmstop` command exits with one of the following values:

0: Successful completion.

1: One or more error conditions occurred.

DIAGNOSTICS

The `hpvmstop` command displays error messages on stderr for any of the following conditions:

- An invalid option is specified.
- An invalid value is specified for an option.
- The *vm-name* or *vm-number* does not exist, cannot be accessed, is not a virtual machine, or is corrupt.
- A value was omitted for an argument that requires one, or a value was supplied for an argument that does not take one.
- The `hpvmstop` command and Integrity VM software are at different revision levels.
- The specified guest is a distributed guest.

EXAMPLES

Perform a graceful shutdown of the virtual machine called host1:

```
# hpvmstop -P host1 -g
```

OBSOLESCENCE

The `-F` option is deprecated in Integrity Virtual Machines commands. This flag might be removed in a future release of Integrity Virtual Machines. HP recommends use of the command with the `-s` option, then resolving any errors or warnings that result before executing the command without the `-s` option and without the `-F` option.

AUTHORS

The `hpvmstop` command was developed by the HP.

SEE ALSO

On the VM Host:

hpvm(5), *hpvmclone*(1M), *hpvmcollect*(1M), *hpvmconsole*(1M), *hpvmcreate*(1M), *hpvmdevinfo*(1M), *hpvmdevmgmt*(1M), *hpvmhostgdev*(1M), *hpvmhostrdev*(1M), *hpvminfo*(5), *hpvmmigrate*(1M), *hpvmmodify*(1M), *hpvmmove_suspend*(1M), *hpvmnet*(1M), *hpvmnvram*(1M), *hpvmpubapi*(3), *hpvmremove*(1M), *hpvmresources*(5), *hpvmresume*(1M), *hpvmsg_move*(1M), *hpvmsg_package*(1M), *hpvmstart*(1M), *hpvmstatus*(1M), *hpvmsuspend*(1M), *p2vassist*(1M)

On the Integrity VM guest:

hpvmcollect(1M), *hpvminfo*(1M), *hpvmmgmt*(1M), *hpvmpubapi*(3)

hpvmsuspend(1M)

NAME

hpvmsuspend -- Suspend a virtual machine.

SYNOPSIS

hpvmsuspend { -P *vm-name* | -p *vm-number* } [-F] [-n] [-q] [-Q] [-s]
hpvmsuspend -A { -P *vm-name* | -p *vm-number* }
hpvmsuspend -v
hpvmsuspend -H

DESCRIPTION

The hpvmsuspend command suspends a virtual machine. A virtual machine that is suspended does not use CPU or memory resources.

The resources that are defined in the virtual machine's configuration file are checked to determine whether the migrated virtual machine can be suspended. You can specify the -F option (force) to suppress the errors and force the virtual machine to be suspended.

NOTE: Use the -F option with caution, because the suspended virtual machine might not be able to resume safely.

Only superusers can execute the hpvmsuspend command.

Options

The hpvmsuspend command recognizes the following command-line options and arguments:

-P *source-vm-name* Specifies the unique name of the virtual machine to be suspended.

You must specify either the -P option or the -p option.

-p *source-vm-name* Specifies the unique number of the virtual machine to be suspended. The *vm_number* is reported by the hpvmstatus command.

You must specify either the -P option or the -p option.

-A Aborts the suspension in progress.

-s Indicates that the guest should not be suspended but that the hpvmsuspend command should check whether or not the suspend is possible.

-F Forces the suspension of a virtual machine, whether or not there are resources that prevent suspension.

Use the -F option rarely and with caution.

The -F option is deprecated in Integrity Virtual Machines commands.

-q Sets the quiet flag, which slightly reduces the messages.

-Q Displays fewer informative messages. Some potential error conditions are still reported.

-n Quits after starting the suspension. If not specified, the hpvmsuspend command continues to run and reports the suspension status.

-v Displays the version number of the hpvmsuspend command.

-H Displays the usage of the hpvmsuspend command.

RETURN VALUES

The hpvmsuspend command exits with one of the following values:

0: Successful completion.

1: One or more error conditions occurred when parsing the command.

2: One or more error conditions occurred during the suspension.

DIAGNOSTICS

The `hpvmsuspend` command displays error messages on stderr for any of the following conditions:

- An invalid option is specified.
- An invalid value is specified for an option.
- A value is omitted for an argument that requires one, or a value is supplied for an argument that does not take one.
- The source-vm-name or source-vm-number attribute does not exist, cannot be accessed, is not a virtual machine, or is corrupt.
- The `hpvmsuspend` command virtual machines are at different revision levels.
- The guest is not running.
- The guest configuration is invalid.
- The guest has resources that do not allow suspension.
- The guest is distributed.
- The version of the `hpvmsuspend` command is incompatible with the version on the destination VM Host.

EXAMPLES

Display the version number of the `hpvmsuspend` command.

```
# hpvmsuspend -v
hpvmsuspend: Version B.04.00
```

Suspend the virtual machine named `compass1`, to the VM Host `abc.def.com`.

```
# hpvmsuspend -P compass1
```

OBSOLESCENCE

The `-F` option is deprecated in Integrity Virtual Machines commands. This flag might be removed in a future release of Integrity Virtual Machines. HP recommends use of the command with the `-s` option, then resolving any errors or warnings that result before executing the command without the `-s` option and without the `-F` option.

AUTHORS

The `hpvmsuspend` command was developed by HP.

SEE ALSO

On the VM Host:

hpvm(5) ,*hpvmclone*(1M), *hpvmcollect*(1M), *hpvmconsole*(1M), *hpvmcreate*(1M), *hpvmdevinfo*(1M), *hpvmdevtranslate*(1M), *hpvmhostgdev*(1M),*hpvmhostrdev*(1M), *hpvminfo*(1M), *hpvmmigrate*(1M), *hpvmmodify*(1M), *hpvmmove_suspend*(1M), *hpvmnet*(1M), *hpvmnvram*(), *hpvmpubapi*(1M), *hpvmremove*(1M), *hpvmresources*(5), *hpvmresume*(1M), *hpvmsg_move*(1M), *hpvmsg_package*(1M), *hpvmstart*(1M)*hpvmstatus*(1M), *hpvmstop*(1M), *hpvmsuspend*(1M), *p2vassist*(1M)

On the Integrity VM guest:

hpvmcollect(1M), *hpvminfo*(1M), *hpvmmgmt*(1M), *hpvmpubapi*(3)

hpvmupgrade(1M)

NAME

hpvmupgrade – Assist an Integrity VM upgrade.

SYNOPSIS

`hpvmupgrade -e [-v]`

DESCRIPTION

The `hpvmupgrade` command performs operations specific to an Integrity VM version upgrade. The `-e` option examines the current Integrity VM server system to determine whether any virtual machines will have difficulty booting after the upgrade to the next Integrity VM version. The `hpvmupgrade` command produces the following log file:

`/var/opt/hpvm/common/hpvmupgrade.`*`current_date_and_time`*.

This utility was created for the HP-UX 11i v2 to HP-UX 11i v3 Integrity VM upgrade and is designed to run on Integrity VM servers that are at Version 3.0 or later. To determine the version of the utility, use the `-v` option. If the utility is at Version 4.3, it can assist server upgrades up to Integrity VM version 4.3. New versions of `hpvmupgrade` will be available in the future.

Options

The following options can be specified only once.

The `hpvmupgrade` command recognizes the following command-line options:

`-e` Examines the system for an upgrade.

`-v` Displays the version of the utility.

RETURN VALUES

The `hpvmupgrade` command exits with one of the following values:

0: Successful completion.

1: An error occurred during the processing of the command.

DIAGNOSTICS

`hpvmupgrade` outputs all error and warning messages to the log file `/var/opt/hpvm/common/hpvmupgrade.`*`current_date_and_time`*.

AUTHORS

The `hpvmupgrade` command was developed by the HP.

SEE ALSO

On the VM Host:

hpvm(5), *hpvmclone*(1M), *hpvmcollect*(1M), *hpvmconsole*(1M), *hpvmcreate*(1M), *hpvmdevinfo*(1M), *hpvmdevtranslate*(1M), *hpvmhostrdev*(1M), *hpvmhostgdev*(1M), *hpvminfo*(1M), *hpvmmigrate*(1M), *hpvmmodify*(1M),*hpvmmove_suspend*(1M), *hpvmnet*(1M), *hpvmnvram*(1M), *hpvmremove*(1M), *hpvmresources*(5), *hpvmresume*(1M), *hpvmsar*(1M), *hpvmsg_move*(1M), *hpvmsg_package*(1M), *hpvmstart*(1M)*hpvmstatus*(1M), *hpvmstop*(1M), *hpvmsuspend*(1M),*p2vassist*(1M)

p2vassist(1M)

NAME

p2vassist – Physical to virtual migration assistant.

SYNOPSIS

`p2vassist` *`function-number`*

DESCRIPTION

The `p2vassist` command is a menu-driven application that helps migrate applications from a running physical or virtual machine to another. To start the script, enter the `p2vassist` command:

```
# p2vassist
P2V Assistant Manager
======================================

 1 ) General Configurations
 2 ) Manage Applications
 3 ) Manage O.S. Images
 -----
 0 ) Exit

Enter Option Number:
```

The choose from the following menu options:

1 General Configurations

This function sets the server from where the installation depot will be obtained. The format accepted is the same as option `-s` to the `swinstall` command. The following example configures a server called `server.abc.com` and a depot location on `/release/1123.0706/ic054` as the depot server:

```
General Configurations
======================================

 1 ) Set Source Depot
 -----
 0 ) Return

Enter Option Number: 1

In this option you can set the location of the depots
of the application to be installed on the target system.
It can be a hostname or a local directory.

Depots source currently configured: NONE
Depots source (type 'q' to return)
```

Enter the full path for the depot location. For example:

`server.abc.com:/release/1123.0706/ic054`

2 Manage Applications

Use this menu to list the depots installed on the server, and to select the depots to consolidate and the directories with the application data. Entering this option displays the following menu:

```
Manage Applications
======================================

 1 ) List Applications
 2 ) Select Applications
 3 ) Add Data Directories/Files
 4 ) Consolidate Applications
```

```
-----
0 ) Return
```

- List Applications

  This option lists the products installed on the server. You can obtain the names of the depots to consolidate from this listing. The list returned does not contain Operational Environment related packages. The list of packages to exclude is stored in a configuration file (see the Add Data Directories/Files item in this list). For example:

```
Manage Applications
======================================

 1 ) List Applications
 2 ) Select Applications
 3 ) Add Data Directories/Files
 4 ) Consolidate Applications
 -----
 0 ) Return

Enter Option Number: 1

  ACXX                  C.06.00        HP aC++
  BullseyeCoverIA       7.5.61         HP-UX IA
  C-ANSI-C              C.06.10        HP C/ANSI C Compiler
  C-Dev-Tools           B.11.23.11     C Language Development Tools
  gcc                   4.1.0          gcc
  gdbm                  1.8.3          gdbm
  .
  .
  vim                   7.0            vim
  xpm                   3.4k           xpm
  zlib                  1.2.3          zlib
```

  The listing shown on your system might differ from this example.

- Select Applications

  A list of depots (space separated) is the input to this option. The script verifies whether the depots displayed belong to the list of installed software. The following example shows the addition of four depots (belonging to the web server suite):

```
Manage Applications
======================================

 1 ) List Applications
 2 ) Select Applications
 3 ) Add Data Directories/Files
 4 ) Consolidate Applications
 -----
 0 ) Return

Enter Option Number: 2


  Select Applications
==============================

Please enter the list of depots to be installed on the target
system, separated by spaces.
You can list the applications using the first option of this menu.
Depots list (type 'q' to return):
```

  Enter the names of the depots. For example:

```
hpuxwsAPACHE hpuxwsTOMCAT hpuxwsWEBMIN hpuxwsXML
```

- Add Data Directories/Files

  Specifies the directories or files that contain the application configuration and data. Enter absolute paths that refer to directories and files that already exist. Some known, system-level directories are rejected by default (such as `./`, `/var`, and `/usr`). If you specify a software depot, `p2vassist` reads the data directories from the depot definition. These directories appear as a suggestion to the user.

```
Manage Applications
======================================

 1 ) List Applications
 2 ) Select Applications
 3 ) Add Data Directories/Files
 4 ) Consolidate Applications
 -----
 0 ) Return

Enter Option Number: 3

Please enter the list of directories/files to be transferred to the
target system, separated by spaces.

The following directories are obtained by examining the depot configuration, you may copy them to the
 prompt
below in order to be used:

 /opt/hpws/apache /opt/hpws/tomcat /opt/hpws/webmin

List (type ENTER to return):
```

For example:

```
List (type <Enter> to return): /opt/hpws/apache /opt/hpws/tomcat /opt/hpws/webmin
```

- Consolidate Applications

  Consolidates the selected applications, along with their directories and files, to the specified server. For example:

```
Manage Applications
======================================

 1 ) List Applications
 2 ) Select Applications
 3 ) Add Data Directories/Files
 4 ) Consolidate Applications
 -----
 0 ) Return

Enter Option Number: 4

Consolidate Applications
==============================

This option will consolidate application(s) installed
on the current system and then migrate into another a
suitable target system. The target system is the machine
which will host the consolidated application(s).
This wizard requires that non-interactive Secure Shell
(ssh) access be set up between this system and the
target server.


Depots source currently configured: depsrv.hp.com:/release/1123.0706/ic054
Depots to be installed: hpuxwsAPACHE hpuxwsTOMCAT hpuxwsWEBMIN hpuxwsXML

Dirs/Files to be transferred: /opt/hpws/apache /opt/hpws/tomcat /opt/hpws/webmin

Please enter the hostname/IP of the target system (type 'q' to return): vmp2V
```

After you press **Enter**, the process continues:

```
Querying the system vmp2v for current status, this may take a few moments...
Installing depots...

Press <Enter> to continue and begin transferring files.

Transferring Configuration files...
Checking application disk space...
Application disk space Ok.
Transferring /opt/hpws/apache directory to vmp2v, please wait...
Transferring /opt/hpws/tomcat directory to vmp2v, please wait...
Transferring /opt/hpws/webmin directory to vmp2v, please wait...
Application Sucessfully consolidated on host: vmp2v
Press <Enter> to return to menu:
```

3 Manage Operating System Images

This option is not supported in the current version of Integrity VM.

RETURN VALUES

Upon completion, the `p2vassist` command returns with one of the following values:

0: Successful completion. The application was successfully consolidated.

1: An error occurred.

EXAMPLES

The following example shows how to consolidate applications:

```
# p2vassist
P2V Assistant Manager
======================================

 1 ) General Configurations
 2 ) Manage Applications
 3 ) Manage O.S. Images
 -----
 0 ) Exit

Enter Option Number: _1_

General Configurations
======================================

 1 ) Set Source Depot
 -----
 0 ) Return

Enter Option Number: 1

In this option you can set the location of the depots
of the application to be installed on the target system.
It can be a hostname or a local directory.

Depots source currently configured: NONE

Depots source (type 'q' to return): server.abc.com:/release/1123/ic054
```

After the depot source is defined, select the application depots to be installed. To obtain the correct name of the depots, list the installed applications:

```
Manage Applications
======================================

 1 ) List Applications
 2 ) Select Applications
 3 ) Add Data Directories/Files
 4 ) Consolidate Applications
 -----
 0 ) Return

Enter Option Number: 1

  ACXX                     C.06.00        HP aC++
  AudioDevKit              B.11.23.10     HP-UX Audio Desktop Developer Kit
  C-ANSI-C                 C.06.10        HP C/ANSI C Compiler
  C-Dev-Tools              B.11.23.11     C Language Development Tools
  gcc                      4.1.0          gcc
```

```
gdbm                   1.8.3          gdbm
.
.
hpuxwsAPACHE           B.2.0.55.03    HP-UX Apache-based Web Server
hpuxwsTOMCAT           B.5.5.9.04     HP-UX Tomcat-based Servlet Engine
hpuxwsWEBMIN           A.1.070.07     HP-UX Webmin-based Admin
hpuxwsXML              A.2.00         HP-UX XML Web Server Tools
.
.
vim                    7.0            vim
xpm                    3.4k           xpm
zlib                   1.2.3          zlib
```

This example shows the correct names of the depots that can be obtained. In this example, consolidates the web server suite (hpuxwsAPACHE, hpuxwsTOMCAT, hpuxwsWEBMIN and hpuxwsXML).

To enter these names, choose Option 2 from the Manage Applications menu. Then type the depot names at the prompt. For example:

```
Manage Applications
======================================

 1 ) List Applications
 2 ) Select Applications
 3 ) Add Data Directories/Files
 4 ) Consolidate Applications
 -----
 0 ) Return

Enter Option Number: 2


  Select Applications
==============================

Please enter the list of depots to be installed on the target
system, separated by spaces.
You can list the applications using the first option of this menu.

Depots list (type 'q' to return): hpuxwsAPACHE hpuxwsTOMCAT hpuxwsWEBMIN hpuxwsXML
```

After setting the depots, choose the directories to be transferred, along with the installed product. The `p2vassist` utility scans the configuration of the depots to make a list of the directories. To specify a directory, copy the string to the prompt. For example:

```
Manage Applications
======================================

 1 ) List Applications
 2 ) Select Applications
 3 ) Add Data Directories/Files
 4 ) Consolidate Applications
 -----
 0 ) Return

Enter Option Number: 3

Please enter the list of directories/files to be transferred to the
target system, separated by spaces.

The following directories are obtained by examining the depot configuration, you may copy them to the prompt
below in order to be used:

 /opt/hpws/apache /opt/hpws/tomcat /opt/hpws/webmin

Please enter directories or files: /opt/hpws/apache /opt/hpws/tomcat /opt/hpws/webmin
```

Finally, consolidate the application on the new host:

```
Manage Applications
======================================

 1 ) List Applications
 2 ) Select Applications
 3 ) Add Data Directories/Files
```

```
 4 ) Consolidate Applications
 -----
 0 ) Return

Enter Option Number: 4

          Consolidate Applications
         ==============================

         This option will consolidate application(s) installed
         on the current system and then migrate into another a
         suitable target system. The target system is the machine
         which will host the consolidated application(s).
         This wizard requires that non-interactive Secure Shell
         (ssh) access be set up between this system and the
         target server.

         Depots to be installed: Firefox

         Dirs/Files list not defined.
         Please enter the hostname/IP of the target system: vmp2v

         Querying the system vmp2v for current status,
         this may take a few moments...
         Installing depots...

         Transferring Configuration files...
         Checking application disk space...
         Application successfully consolidated on host: vmp2v

         Press <Enter> to return to menu:
```

AUTHORS

The `p2vassist` command was developed by the HP.

SEE ALSO

On the VM Host:

hpvm(5), *hpvmclone*(1M), *hpvmcollect*(1M), *hpvmconsole*(1M), *hpvmcreate*(1M), *hpvmdevinfo*(1M), *hpvmdevmgmt*(1M), *hpvmdevtranslate*(1M), *hpvmhostgdev*(1M), *hpvmhostrdev*(1M), *hpvminfo*(1M), *hpvmmigrate*(1M), *hpvmmodify*(1M), *hpvmmove_suspend*(1M,)*hpvmnet*(1M), *hpvmnvram*(1M), *hpvmpubapi*(3), *hpvmremove*(1M), *hpvmresources*(5), *hpvmresume*(1M), *hpvmsar*(1M), *hpvmsg_move*(1M), *hpvmsg_package*(1M), *hpvmstart*(1M), *hpvmstatus*(1M), *hpvmstop*(1M), *hpvmsuspend*(1M), *hpvmupgrade*(1M)

On the Integrity VM guest:

hpvmcollect(1M), *hpvminfo*(1M), *hpvmmgmt*(1M), *hpvmpubapi*(3)

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.

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 VM host and guests. Participating guests must include a virtual I/O device configured to use the AVIO protocol.

backing store
The physical device on the VM Host that is allocated to guests, such as a network adapter, disk, or file.

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.

captive virtual console account
A special-purpose user account created on the VM Host 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 cell 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 (VM Host or guest) configured to be a part of a Serviceguard cluster.

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.

distributed guests
Guests that has been configured as a Serviceguard package.

EFI
Extensible Firmware Interface. The boot firmware for all HP Integrity systems.

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 Machines 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 VM Host, 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 VM Host for one or more virtual machines.

host administrator The system administrator. This level of privilege provides control of the VM Host system and its resources, as well as creating and managing guests.

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.

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 VM Host. The local network created by this vswitch can be used for communications among guests but not for communication between the VM Host and any guest or between any external system and a VM guest.

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 VM Host systems.

NIC Network Interface Card. Also called “network adapter.”

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 VM Host 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.

PMAN Platform Manager. See *VM Host*.

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 VM Host system. For example, the VM Host 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 VM Host. See *VM Host*.

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 VM Host that is used by the virtual machines running on the VM Host.

symmetric Serviceguard configuration A cluster configuration in which the nodes share access to the same storage and network devices.

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, s a DVD) on the VM Host.

virtual machine Virtual hardware system. Also called VM.

virtual machine application The executable program on the VM Host 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* VM Host.

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 VM Host 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 VM Host or in the same Serviceguard cluster.

virtual switch *See* vswitch.

VM See *Virtual machine*.

VM Host The virtual machine host system.

vNIC Virtual network interface card (NIC). The network interface that is accessed by guest applications.

vswitch Virtual switch. A component in the guest virtual network. By associating the vswitch with a physical working LAN on the VM Host, 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