HP Integrity Virtual Machines Installation, Configuration, and Administration

HP Integrity Virtual Machines Version 2.0

Printed in the US
HP Part Number: T2767-90024
Published: October 2006, Edition 2

Legal Notices

Acknowledgments

HP-UX Release 10.20 and later and HP-UX Release 11.00 and later (in both 32 and 64-bit configurations) on all HP 9000 computers are Open Group UNIX 95 branded products.

UNIX is a registered trademark of The Open Group.

Microsoft and Windows are U.S. registered trademarks of Microsoft Corporation.

Intel and Itanium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Table of Contents

List of Figures

List of Tables

About This Document

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

Refer to the Release Notes accompanying this documentation for recent updates, known issues, and other information.

NOTE: The terms *Integrity Virtual Machines* and *Integrity VM* are used interchangeably throughout this guide.

1 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.

2 New and Changed Information in This Edition

This manual supersedes the manual of the same title for HP Integrity Virtual Machines Version A.01.00 (T2767-90004). For more information about the new version of the product, see "Upgrading from Earlier Versions of Integrity VM" (page 23). For information about the features and changes in this version of Integrity VM, see the *HP Integrity Virtual Machines Release Notes*.

3 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 Web site 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 can display software versions that differ from the actual released product.

4 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

5 Document Organization

This manual consists of the following chapters:

- "Introduction" (page 15) describes the concept of the virtual machine as it applies to Integrity VM.
- "Installing Integrity VM" (page 21) describes how to install the Integrity VM product.
- "Creating Virtual Machines" (page 27) describes how to create virtual machines.
- "Creating HP-UX Guests" (page 41) describes how to create HP-UX guests
- "Creating Windows Guests" (page 47) describes how to create Windows® guests.
- "Creating Virtual Storage Devices" (page 61) describes how to create virtual storage devices.
- "Creating Virtual Networks" (page 89) describes how to create virtual networks.
- "Managing Guests" (page 101) describes how to start, stop, and manage virtual machines.
- "Migrating Virtual Machines" (page 111) describes how to migrate guests to other VM Host systems.
- "Using HP Serviceguard with Integrity VM" (page 115) describes how to set up Serviceguard to manage your guests.
- "Reporting Problems with Integrity VM" (page 129) describes how to solve virtual machine problems.
- "Integrity VM Manpages" (page 133) lists the HP-UX manpages provided with the HP Integrity VM software.
- The "Glossary" (page 181) defines many of the terms used in the Integrity VM documentation.

6 Related Information

You can download the latest version of this document from docs.hp.com. The following related documents can also be downloaded from the same site:

- *HP Integrity Virtual Machines Release Notes*
- *Ignite-UX Reference*
- *Troubleshooting Ignite-UX Installation Booting* White Paper
- *HP-UX Installation and Update Guide*
- *HP-UX Reference*
- *Managing Serviceguard*
- *Windows on Integrity: Smart Setup Guide*
- *HP Auto Port Aggregation (APA) Support Guide*
- *Using HP-UX VLANS*

The web site docs.hp.com also includes technical papers about using virtual machines.

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

7 Publishing History

Manufacturing Part Number	Supported Operating Systems	Supported Versions	Edition Number	Publication Date
T2767-90004	HP-UX	11i v2	1.0	October 2005
T2767-90024	HP-UX	11i v2	2.0	October 2006

8 HP Encourages Your Comments

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

Your comments and suggestions regarding product features will help us develop future versions of the Virtual Server Environment Management Software. Use the following e-mail address to send feedback directly to the VSE Management Software development team: **vse@hpuxweb.fc.hp.com**.

NOTE: HP cannot provide product support through this e-mail address. To obtain product support, contact your HP Support representative, your HP Services representative, or your authorized HP reseller. For more information about support services, see the support web site at http://www.hp.com/go/support.

For other ways to contact HP, see the Contact HP web site at http://welcome.hp.com/country/us/en/contact_us.html.

1 Introduction

This chapter describes the Integrity Virtual Machines product, including:

1.1 About HP Integrity Virtual Machines

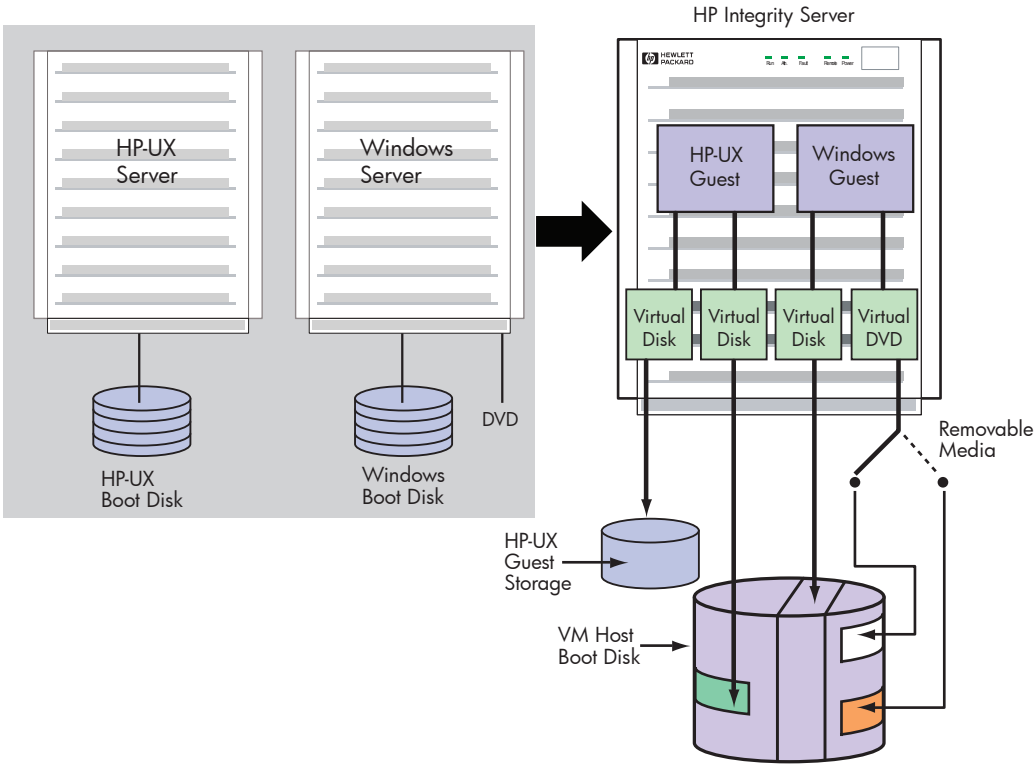
Integrity Virtual Machines is a soft partitioning and virtualization technology that provides operating system isolation, with sub-CPU allocation granularity and shared I/O. Integrity VM can be installed on an Integrity server or hardware partition (nPartition) running HP-UX. The Integrity VM environment 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 1-1 Hardware Consolidation using Integrity VM



Guests are fully loaded, operational systems, complete with operating system, system management utilities, applications, 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 1-1 illustrates how an HP-UX system and a Windows 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 Windows guest storage. The Windows 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 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 includes the following:

- HP-UX Foundation Operating Environment (FOE)

NOTE: The HP-UX FOE and license is included with the Integrity VM media so that you can install and run Integrity VM on the VM Host system. For HP-UX guests, you must purchase FOE licenses.

- 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

Do not run the following types of software on the VM Host system:

- vPars (Virtual Partitions and virtual machines are mutually exclusive.)
- Workload Manager (WLM)

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 Foundation Operating Environment (FOE)
- Windows 2003 for Integrity Servers (Enterprise edition)
- 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
- Hardware diagnostic tools and support tools (should be run on the VM Host)
- 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.

You can install the VM Host on a system that is running HP-UX 11i v2 May 2005 or later. Guests must also be running HP-UX 11i v2 May 2005 or later. 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.

1.3 Related Products

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

- HP Integrity VM Manager — A graphical user interface for creating and managing HP Integrity virtual machines. Runs under either HP System Management Homepage (SMH) or HP Systems Insight Manager as part of the HP Integrity VSE. For more information, see the *Getting Started with Integrity Virtual Machine Manager* guide.
- HP Integrity Virtual Server Environment (VSE) — A graphical user interface for managing HP Integrity central managed systems (CMS). Runs under HP Systems Insight Manager. For more information, see the *HP VSE Management Software Quick Start Guide*.
- HP-UX operating system — Integrity VM runs on HP-UX 11i v2 Integrity systems. For more information, see the *HP-UX 11i v2 Installation and Update Guide*.
- HP Integrity Support Pack and Microsoft® Windows® Server 2003 Service Pack 1 — HP recommends that you install the Support Pack and SP1 on all HP Integrity servers running Windows Server 2003, 64-bit.. For more information, see the *HP Integrity Support Pack and Microsoft Windows Server 2003 Service Pack 1 Release Notes*.
- 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 Integrity Virtual Machines VMMigrate utility — An optional, separately-installed software package that allows you to move virtual machines from one VM Host to another. For more information, see "Migrating Virtual Machines" (page 111) in this manual.

- 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 "Using HP Serviceguard with Integrity VM" (page 115), in this manual.
- HP Integrity Essentials Global Workload Manager (gWLM) — A software product that 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.

1.4 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 1-1 describes each chapter in this manual.

Table 1-1 Chapters of this Manual

Chapter	Read if...
"Introduction" (page 15)	You are new to HP Integrity Virtual Machines.
"Installing Integrity VM" (page 21)	You are installing the HP Integrity Virtual Machines product.
"Creating Virtual Machines" (page 27)	You are setting up new virtual machines on your VM Host system.
"Creating HP-UX Guests" (page 41)	You are creating virtual machines that will run the HP-UX operating system.
"Creating Windows Guests" (page 47)	You are creating virtual machines that will run the HP Integrity Windows 2003 Enterprise operating system.
"Creating Virtual Storage Devices" (page 61)	You need to make changes to the storage devices used by the VM Host or virtual machines.
"Creating Virtual Networks" (page 89)	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.
"Managing Guests" (page 101)	You need to manage existing virtual machines.
"Migrating Virtual Machines" (page 111)	You need to move virtual machines from one VM Host system to another.
"Using HP Serviceguard with Integrity VM" (page 115)	You need to set up Serviceguard to manage your VM Host system or your virtual machines.
"Reporting Problems with Integrity VM" (page 129)	You encounter problems while creating or using virtual machines.
"Integrity VM Manpages" (page 133)	You need to understand how to use an Integrity VM command.
"Glossary" (page 181)	You do not understand the definition of a term used in the Integrity VM product documentation.

1.5 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.

1.5.1 Integrity VM Manpages

For online information about using Integrity VM, refer to the following manpages:

- *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.
- *hpvmdevmgmt*(1M) - describes how to modify the way virtual devices are handled.
- *hpvminfo*(1M) - describes how to get information about the VM Host.
- *hpvmmigrate*(1M) - describes how to migrate virtual machines from one VM Host to another.

- *hpvmmodify*(1M) - describes how to modify virtual machines.
- *hpvmnet*(1M) - describes how to create and modify virtual networks.
- *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.
- *hpvmremove*(1M) - describes how to remove a virtual machine.
- *hpvmresources*(1M) - describes how to specify the storage and network devices used by virtual machines.

1.5.2 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 "Using the Virtual Console" (page 105).

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 sections:

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. To install Integrity VM, you need a computer that fits the specifications listed in "VM Host System Requirements" (page 21).

NOTE: Before installing this product, read the *HP Integrity Virtual Machine Release Notes*. The most up-to-date release notes are available on http://docs.hp.com.

2.1.1 VM Host System Requirements

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 and storage 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 2-1 describes the minimum configuration requirements for installing Integrity VM on the VM Host system.

Table 2-1 Requirements for Installing Integrity VM

Resource	Description
Computer	An Integrity server
Operating system	HP-UX 11i v2 May 2005 or later, running on an Integrity server, as well as any appropriate software patches (see the *HP Integrity Virtual Machines Release Notes*). The license for Integrity VM includes the license for running the HP-UX Foundation Operating Environment 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).
Disk storage	Sufficient disk space for the following: • The VM Host operating system (refer to the *HP-UX 11i v2 Installation and Upgrade Guide*) • The VM Host software (50 MB) • Swap space size should be at least as large as physical memory plus 4GB (for example, for 16 GB of RAM, swap space should be 20 GB) **NOTE:** HP-UX uses this space to start up guests, but guests are never swapped out. • Disk space for each guest operating system, including swap space • Disk space for the applications running on each guest • 4.7 MB for each running guest as the allowance for backing up configuration files

Table 2-1 Requirements for Installing Integrity VM (*continued*)

Resource	Description
Memory	Sufficient physical memory (RAM), including the following: • 750 MB + 7.5% of memory beyond the first GB (that is, 7.5% of (total physical memory - 1 GB)) • Total aggregate memory required for each guest (operating system and application requirements) 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. • Additional 7% of aggregate guest memory for overhead For example, for a VM Host with 16 GB of memory and two VMs configured with 3GB of memory each, the memory requirements would be calculated as follows: • 1.86 GB for the VM Host (750 MB plus 7.5% of 15 GB) • 6.42 GB total guest requirement (107% of 6 GB) • Total requirements = 8.28 GB of memory This leaves 7.72 GB of memory for additional guests.
Integrity VM software	The software bundle T2767AC. Refer to "Bundle Names" (page 22) for information about the required software for installing Integrity VM.
Network configuration	A configured and operational network, with at least one LAN card if you plan to allow remote access to guest virtual consoles. 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 "Creating Virtual Networks" (page 89).

2.1.2 Bundle Names

Integrity VM software is bundled as T2767AC, which includes VMAGENT, the Integrity VM fair-share scheduler. When you install Integrity VM, the following software bundles are installed:

- T2767AC
- PRM-Sw-Krn (included with T2767AC)
- VMGuestLib

In addition to the T2767AC bundle, you can install the following optional software bundles:

- VMProvider (to use the HP Integrity VM Manager to manage the VM Host)
- VMMigrate (to be able to migrate virtual machines from one VM Host to another). For information about using the `hpvmmigrate` command, see "Migrating Virtual Machines" (page 111).

2.1.3 Using VM Manager Requires the Latest WBEM Services on the VM Host

The version of HP WBEM Services for HP-UX must be A.02.00.10 or later. Integrity VM fails to install if the version of WBEM Services on your VM Host is older than A.02.00.10. The HP WBEM Services for HP-UX software bundle (B8465BA) is available as part of the HP-UX 11i V2 0606 (June 2006) operating system and later. For VM Hosts running earlier versions of HP-UX, download the latest version of WBEM Services from www.hp.com.

2.2 Installation Procedure

Once you have read the product release notes and verified that you have met the proper system requirements as described in "VM Host System Requirements" (page 21), install the Integrity VM software as described in this section.

NOTE: Installing the Integrity VM software may require 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 pathname that correspond to the software distribution depot that contains the T2767AC 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 T2767AC
   ```

   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 (T2767AC) 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/` 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.

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:

- Integrity VM is installed in the `/var/opt/hpvm` directory.
- Integrity VM data files are installed under the `/var/opt/hpvm` directory.
- Integrity VM commands are installed in the `/opt/hpvm/bin` directory.
- Integrity VM installation modifies certain kernel parameters. If you use multiple shells to manage Integrity VM, change the kernel parameters on all your shells. Table 7-1 lists the kernel parameters that are modified when you install Integrity VM.

Table 2-2 Kernel Parameters

Parameter	Default Value	Modified Value
dbc_max_pct	50	1
dbc_min_pct	5	1
maxdsiz_64bit	4294967296	34359738368
swapmem_on	1	0

You can now create guests using the `hpvmcreate` command, as described in Chapter 3 (page 27).

2.3 Upgrading from Earlier Versions of Integrity VM

When you upgrade Integrity VM from an earlier version, you should:

1. Shut down all running guests (using the `hpvmstop` command).
2. Locate and install the new version of Integrity VM.
3. Install new versions of the `vmmigrate` utility and the VMProvider, if they were previously installed.
4. Reboot the VM Host system.

Existing guest configuration information, operating system software, and application data are not affected when you upgrade Integrity VM.

If you have installed an evaluation version of Integrity VM, you should remove the evaluation software before installing the Integrity VM product. For more information, refer to the *Integrity VM Release Notes*.

2.4 Verifying the Installation of Integrity VM

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

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

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

```
# swlist T2767AC
# Initializing...
# Contacting target "gaggle"...
#
# Target:  gaggle:/
#

# T2767AC               A.02.00.02 Integrity VM
  T2767AC.HPVM          A.02.00.02 Integrity VM HPVM
  T2767AC.VMAGENT       A.02.00.02 HP Resource Allocation Agent for Integrity
 VM
```

NOTE: Specific baselevels on your installation might not exactly match the examples in this manual. For example, you may see A.02.00.01 or A.02.00.02.

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

2.5 Removing Integrity VM

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

- VMProvider (if installed)
- T2767AC
- VMGuestLib
- VMMigrate (if installed)
- VMKernelSW (reboots the system)

To remove these bundles, enter the following commands:

```
# swremove VMProvider
# swremove T2767AC
# swremove VMGuestLib
# swremove VMMigrate
# swremove -x autoreboot=true VMKernelSW
# rm -rf /opt/hpvmprovider
# rm -rf /opt/hpvm
```

Guests are not affected by this procedure. To remove guests, see the procedures in "Removing Virtual Machines" (page 38).

2.6 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/rscsi/c2t0d0` device, enter the following command:

```
# hpvmdevmgmt -a rdev:/dev/rscsi/c2t0d0
```

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 "Managing the Device Database" (page 107).

2.7 Troubleshooting Installation Problems

If the installation verification fails, report the problem using the procedures described in "Reporting Problems with Integrity VM" (page 129). Some problems encountered in the process of installing Integrity VM are described in the following sections.

2.7.1 Error messages during installation

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

```
could not write monParams: Device is busy

hpvmnet * already exists

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

You can ignore these messages.

3 Creating Virtual Machines

After you install Integrity VM, you can begin to create guests. This chapter includes the following sections:

3.1 Specifying Virtual Machine Characteristics

When you create a new virtual machine, you specify its characteristics. Later, you can change the virtual machine characteristics. The characteristics of a virtual machine are listed in Table 3–1.

You can create a virtual machine using the following commands:

- `hpvmcreate`
- `hpvmclone`

After you create a virtual machine, you can modify it using the the `hpvmmodify` command. All of these commands accept the same options for specifying virtual machine characteristics. Each option and characteristic is described in more detail later in this chapter.

Table 3-1 Characteristics of an Integrity Virtual Machine

Command Option	Virtual Machine Characteristic	Where Described
`-P` *`vm-name`*	Virtual machine name. You must specify a name when you create or modify the virtual machine. You cannot modify this characteristic.	"Virtual Machine Name" (page 27)
`-O` *`os_type`*	Operating system. If you do not specify the operating system type, it is set to UNKNOWN.	"Guest Operating System" (page 28)
`-c` *`number_vcpus`*	Virtual CPUs (vCPUs). If you omit this option when you create the virtual machine, the default is one vCPU.	"Virtual CPUs" (page 28)
`-e` *`percent`* `-E` *`cycles`*	CPU entitlement. If you omit this option when you create the virtual machine, the default is 10%.	"Entitlement" (page 28)
`-r` *`amount`*	Memory. If you omit this option when you create the virtual machine, the default is 2 GB.	"Guest Memory Allocation" (page 29)
`-a` *`rsrc`*	Virtual devices. If you omit this option when you create the virtual machine, it has access to no network and storage devices.	"Virtual Devices" (page 29)

3.1.1 Virtual Machine Name

Use the `-p` *`vm-name`* option to the `hpvmcreate` command to specify the name of the new virtual machine. This option is required. In the following example, the new virtual machine is named `compass1`:

```
# hpvmcreate -P compass1
```

The virtual machine name can be up to 256 alphanumeric characters. To provide remote console access to the guest, its name must be a legal UNIX account name (no more than eight characters, where the colon (:) and newline (\) characters are not valid). See *password*(1M) for more information about HP-UX account

names. For more information about setting up remote console access to the guest, see “Using the Virtual Console” (page 105).

3.1.2 Guest Operating System

Use the `-o os_type` option to the `hpvmcreate` command 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:

- `hpux`
- `windows`

If you do not supply the operating system type, it defaults to UNKNOWN. When you install the operating system, this value in the guest configuration file is automatically set to the appropriate operating system type.

In the following example, the virtual machine `compass1` is specified as an HP-UX guest:

```
# hpvmcreate -P compass1 -o hpux
```

For more information about creating HP-UX guests, refer to Chapter 4 (page 41).

For more information about creating Windows guests, refer to Chapter 5 (page 47).

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.

3.1.3 Virtual CPUs

Use the `-c number_vcpus` option to the `hpvmcreate` 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 `compass1` to have two vCPUs, enter the following command:

```
# hpvmcreate -P compass1 -c 2
```

Every virtual machine has at least one vCPU. A virtual machine cannot use more than 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.)

Integrity VM allows you to create a virtual machine with more vCPUs than the number of physical CPUs on the VM Host system. Warning messages are dislayed if there are not enough physical CPUs to run the virtual machine. This feature allows you to create virtual machines for future configurations. However, the virtual machine is not allowed to start on a VM Host system that does not have enough physical CPUs.

3.1.4 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 compass1, enter the following command:

```
# hpvmcreate -P compass1 -e 20
```

When the virtual machine is booted, the VM Host ensures that sufficient processing power is available for each running virtual machine to receive its entitlement. For virtual machines with multiple virtual CPUs, the entitlement is guaranteed on all the vCPUs 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 a physical CPU's processing power is available to that virtual machine. As

many physical processors as the virtual machine has vCPUs can contribute to the total processing power of the 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 power is available on thehost system, the virtual machine can receive more than its entitlement. When there is contention for processing power (on a VM Host system with busy virtual machines), each virtual machine is limited to its entitlement.

3.1.5 Guest Memory Allocation

Use the `-r amount` option to the `hpvmcreate` command to specify the amount of virtual memory (in either gigabytes or megabytes) 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 compass1, enter the following command:

```
# hpvmcreate -P compass1 -r 3G
```

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

- The amount of memory required by the guest operating system. For example, the HP-UX 11i v2 operating system requires 1 GB of memory.
- 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 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 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 “Installation Requirements” (page 21).

3.1.6 Virtual Devices

Use the `-a` option to the `hpvmcreate` command to allocate network and storage devices to the virtual machine. The VM Host presents the devices to the virtual machine as “virtual 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.

3.1.7 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. A vswitch functions like a physical network interface card (pNIC), accepting network traffic from one or more virtual machines and directing network traffic to an associated port. A vswitch can be associated with a VM Host pNIC, or it can be local to the virtual machines on the VM Host and provide a dedicated network among guests.

Integrity VM always creates a vswitch named `localnet`. This network is not associated with a pNIC. It is used only for communication between the guests running on the same VM Host. The `localnet` vswitch does not use a name server or router, and the VM host does not access the `localnet` vswitch. For more information, see “Local Networks” (page 91).

You can create vswitches any time, 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. For example:

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

where:

- *`vswitch-name`* is the name you assign to 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
```

To allocate the vswitch to the virtual machine named `compass2`, use the –`a` option to the `hpvmcreate` command. For example:

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

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

```
# hpvmcreate -P  -a network:lan::vswitch:vswitch-name
```

For more information about using the `hpvmnet` command, see "Creating Vswitches" (page 90).

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.

With Integrity VM A.02.00 and later, you can create virtual LANs (VLANs), which 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 VLANs, see the manual *Using HP-UX VLANs*.

For more information about creating and managing vswitches, see Chapter 7 (page 89).

3.1.8 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. The VM Host system must have sufficient physical storage for the VM Host and for all of the virtual machines.

When you create a virtual machine with the `hpvmcreate` command, you can specify both the virtual devices that the virtual machine recognizes and the physical backing stores on the VM Host system. Use the `-a` option to create and allocate the virtual device to the virtual machine. For example:

```
# hpvmcreate -a device-type:adapter-type:[hardware-address]:storage-type:device
```

where:

- *device-type* is the type of virtual device that the virtual machine will use. This can be one of the following:
  - `disk`
  - `dvd`
  - `tape`

  - changer
  - burner
- *adapter-type* is always `scsi`.
- *hardware-address* (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:

  ```
  # hpvmcreate -a device-type:adapter-type::storage-type:device
  ```

- *storage-type* indicates the type of physical backing store:
  - disk
  - lv
  - file
  - null
  - attach
- *device* is the specific physical device ID (for example, `/dev/rdsk/c4t3d2`). To display the device IDs on your VM Host system, enter the `ioscan` command.

The physical backing store that you associate with a virtual device 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/dsk/c8t2d0` as a backing store, the raw device (`/dev/rdsk/c8t2d0`) cannot also be used as a backing store. For more information about specifying virtual devices, see "Creating Virtual Storage Devices" (page 61).

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 exising 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 `compass1`, 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 compass1 -a dvd:scsi::disk::/c1t1d2
```

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 "Restricting VM Host Devices" (page 109).

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 3-2.

3.2 Using the hpvmcreate Command

To create a virtual machine, enter the `hpvmcreate` command in the following format:

```
hpvmcreate -P vm-name [-F | -s] [-l vm_label] [-B start_attr]
[-O os_type[:version]] [-c number_vcpus] [-e percent | -E cycles]
```

```
[-g group[:{admin|oper}]] [-u user[:{admin|oper}]]
[-a rsrc] [-r amount]
[-i {SG | -i SG_pkgname | -i GWLM | -i SG_pkgname,GWLM | -i NONE}]
[-j {0|1}]
```

Table 3-2 describes the options you can use with the `hpvmcreate` command.

Table 3-2 Options to the hpvmcreate Command

Option	Description
`-P vm-name`	Specifies the name of the virtual machine. The virtual machine name can be up to eight alphanumeric characters. To provide remote console access to the guest, its name must be a legal UNIX account name (no more than eight characters, where the colon (:) and newline (\) characters are not valid). The `-P` option is required.
`-F`	Suppresses all resource-conflict checks and associated warning messages (force mode). Use force mode for troubleshooting purposes only.
`-s`	Sanity checks the virtual machine configuration and returns warnings or errors, but does not create the virtual machine.
`-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 character (_), 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. `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. The *os_type* parameter can have the following (case-insensitive) values:
	`HPUX`
	`Windows`

`-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.

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

`-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.

`-a rsrc`	Creates a virtual device for the virtual machine. To create a virtual storage device, enter the *rsrc* as:
	`virtual_devicetype:scsi:[bus,device,target]:phstorage_type:physical_device`
	For information about forming a virtual storage device specification, see Chapter 6.
	To create a virtual network device for a virtual machine, enter the *rsrc* as:
	`network:adaptertype:[bus,device,mac-addr]:vswitch:vswitch-name:portid:portnumber`
	For information about forming a virtual network device specification, see Chapter 7 (page 89).

Table 3-2 Options to the hpvmcreate Command (*continued*)

Option	Description
`-r amount`	Specifies the amount of memory available to this virtual machine. Specify the amount as either *amount*`M` (for megabytes) or *amount*`G` (for gigabytes).
`-i package-name`	Specifies whether the virtual machine is managed by Serviceguard or gWLM (or both). The argument is one of the following: • `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. 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). Do not specify this option. This option is used internally by Integrity VM.

3.2.1 Example of Virtual Machine Creation

To create a virtual machine named `compass1`, enter the following command:

```
# hpvmcreate -P compass1
```

This command creates a virtual machine named `compass1` 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
winguest1                5 WINDOWS On (OS)         1     5     1    1 GB        0
winguest2                9 WINDOWS Off             1     3     1    2 GB        0
compass1                12 UNKNOWN Off             1     0     0    2 GB        0
```

The `compass1` 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 "Using HP Serviceguard with Integrity VM" (page 115).

3.3 Starting Virtual Machines

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

The `hpvmstart` command syntax is:

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

Table 1–3 describes the options to the `hpvmstart` command.

Table 3-3 Options to the hpvmstart 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.

Table 3-3 Options to the hpvmstart Command (*continued*)

Option	Description
-F	Forces the command to act without requiring confirmation.
-s	Checks the VM Host system resources without starting the guest.

For example, to start the new virtual machine compass1, enter the following command:

```
# hpvmstart -P compass1
(C) Copyright 2000 - 2006 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 86016 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 'compass1'
```

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
winguest1                5 WINDOWS On (OS)        1     5     1    1 GB        0
winguest2                9 WINDOWS Off            1     3     1    2 GB        0
compass1                13 UNKNOWN On (EFI)       1     0     0    2 GB        0
```

For more information about using the hpvmstatus command, see "Managing Guests" (page 101).

3.4 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 compass1 with 3 vCPUs and 4 MB of allocated memory:

```
# hpvmcreate -P compass1 -c 3 -r 4GB
HPVM guest compass1 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 compass1 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 compass1
HPVM guest compass1 configuration problems:
Warning 1: Insufficient free memory for guest.
Warning 2: Insufficient cpu resource for guest.
   These problems may prevent HPVM guest compass1 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. Table 3-4 describes the options you can use on the `hpvmmodify` command.

Table 3-4 Options to the hpvmmodify Command

Option	Description
`-P` *vm-name*	Specifies the name of 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 (-). 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.
`-s`	Sanity-checks the virtual machine configuration and returns warnings or errors, but does not create the virtual machine.
`-N` *new-vm-name*	Specifies a new name for the virtual machine.
`-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 character (_), 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. The *os-type* can have the following (case-insensitive) values:
	`HPUX`
	`Windows`

Table 3-4 Options to the hpvmmodify Command (*continued*)

Option	Description
`-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 | -E cycles`	Modifies the virtual machine's CPU entitlement in CPU cycles. To specify the percentage of CPU power, enter the following option:
	`-e percent`
	To specify the clock cycles, enter one of the following options:
	`-E cyclesM (for megahertz)`
	`-E cyclesG (for gigahertz)`

`-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.
`-a rsrc`	Adds a virtual storage or network device to the virtual machine. For more information, see *hpvmresources*(1M).
`-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 will replace the one in use.
`-d rsrc`	Deletes a virtual resource.
`-r amount`	Modifies the amount of memory available to this virtual machine. Specify the amount as either *amount*`M` (for megabtyes) or *amount*`G` (for gigabytes).
`-i package-name`	Specifies whether the virtual machine is managed by Serviceguard or gWLM (or both). For the argument, specify one or more of the following parameters: • `SG` indicates that the 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_compass1,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.

For example, to modify the characteristics of the problematic virtual machine `compass1` to remove vCPUs and memory, enter the following command:

```
# hpvmmodify -P compass1 -c 1 -r 2 GB
```

This command changes the following characteristics of the virtual machine named `compass1`:

- `-c 1` specifies one vCPU
- `–r 2 GB` specifies one 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 compass1
(C) Copyright 2000 - 2006 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 86016 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 'compass1'
```

The virtual machine `compass1` is started. Now the guest operating system must be installed. For information about creating HP-UX guests, see Chapter 4 (page 41). For information about creating Windows guests, see Chapter 5 (page 47).

3.5 Cloning Virtual Machines

Once you have created a guest, you can easily create an identical guest by using the `hpvmclone` command. Like the `hpvmcreate` and `hpvmmodify` commands, the `hpvmclone` command accepts the `-a` option for specifying virtual devices and network interfaces. This allows you to create new guests with similar characteristics but different virtual resources. For more information, see *hpvmclone*(1M).

3.6 Stopping Virtual Machines

To stop a running virtual machine, use the `hpvmstop` command. You must confirm this command. Table 3-5 describes the options to the `hpvmstop` command:

Table 3-5 Options to the hpvmstop 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.
`-h`	Performs a hard stop on the virtual machine, similar to a power failure.
`-g`	Performs a graceful shutdown on the virtual machine. This is the default.

Table 3-5 Options to the hpvmstop Command (*continued*)

Option	Description
-F	Forces the command to act without requiring confirmation.
-q	Quiet mode. Used for scripting purposes.

For example, the following command stops the virtual machine named compass1. The hpvmstatus command shows that the virtual machine is Off.

```
# hpvmstop -P compass1
hpvmstop: Stop the virtual machine 'compass1'? [n]: 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
winguest1                5 WINDOWS On (OS)        1     5     1    1 GB         0
winguest2                9 WINDOWS Off            1     3     1    2 GB         0
compass1                12 UNKNOWN Off            1     0     0    2 GB         0
```

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.

3.7 Removing Virtual Machines

To remove a virtual machine from the VM Host, use the hpvmremove command. You must confirm this action. Table 3-6 describes the options to the hpvmremove command.

Table 3-6 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 without requiring confirmation.

For example, the following command removes the virtual machine named compass1. The subsequent hpvmstatus command shows that compass1 is gone:

```
# hpvmremove -P compass1
hpvmremove: Remove the virtual machine 'compass1'? [n]: 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
winguest1                5 WINDOWS On (OS)        1     5     1    1 GB         0
winguest2                9 WINDOWS Off            1     3     1    2 GB         0
```

This command removes compass1 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.)

3.8 Troubleshooting Virtual Machine Creation Problems

If you encounter problems with creating virtual machines, report them through your support channel. For information about collecting information to report the problem, see "Reporting Problems with Integrity VM" (page 129).

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

3.8.1 Configuration Error on Starting the Virtual Machine

When you start the virtual machine, the following message is displayed:

```
Configuration error: Device does not show up in guest
```

If you encounter this type of problem:

1. Verify that the path name to the file-backing store is correct and that the physical storage device is mounted.
2. Verify that the size of the physical storage device is divisible by 512 bytes (for a disk device) or 2048 (for a DVD device).
3. Modify the virtual machine using the `hpvmmodify` command.

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" (page 41)
- "Installing HP-UX Guest Management Software" (page 44)
- "Troubleshooting HP-UX Guest Creation" (page 44)

4.1 Installing the HP-UX Guest 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 `compass1`, enter the following command. The `hpvmstatus` command shows that the vritual machine is started.

```
# hpvmstart -P compass
(C) Copyright 2000 - 2006 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 86016 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 'compass1'

# 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
winguest1                5 WINDOWS On (OS)         1     5     1    1 GB         0
winguest2                9 WINDOWS Off             1     3     1    2 GB         0
compass1                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 compass1

 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

[compass1] 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:

```
[compass1] vMP> co


EFI Boot Manager ver 1.10 [14.62] [Build: Fri Aug  4 11:37:36 2006]

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 Integrity server.

When the basic installation process is complete, the software is copied from the distribution media to the guest's disk. Then the operating sytem 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:

```
[compass1] 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 compass1
```

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 compass1 -d dvd:scsi:0,0,0
```

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

4.2 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`
- The VM Provider, which allows you to use the VM Manager to manage the guest.

The guest management software repository is installed on the VM Host system automatically. The following directory contains the HP-UX guest management software:`/opt/hpvm/guest-images/hpux/`

The HP-UX guest management software is included in this directory. A README file contains instructions for installing the guest management software. Install the guest management software on every HP-UX guest.

4.3 Troubleshooting HP-UX Guest Creation

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

4.3.1 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 Windows Guests

You can install HP Integrity Windows 2003 on your virtual machines. These Windows guests can be managed like a Windows server running on an independent Integrity server or nPartition, by either the VM Host administrator or the Windows system administrator. This chapter describes:

5.1 Windows Guest Requirements

To run Windows on a virtual machine, you must install the following on the virtual machine:

- HP Integrity Windows 2003 software media with Service Pack 1. Make sure you have the product key, which appears on the lower section of your Certificate of Authenticity.
- Integrity VM Windows guest management software.
- Optional: For management from HP Integrity Virtual Machines Manager (VM Manager) and the HP Virtual Server Environment (VSE), the provider utilities are available on the Windows Smart Setup Media (SSM).

The following procedure assumes that HP Integrity Virtual Machines A.02.00 or later is installed on the VM Host system.

To install the Windows operating system on the virtual machine, select a physical backing storage unit to be used as the guest's boot disk. For applications requiring optimal I/O performance, HP recommends using a whole disk or disk partition of at least 34 GB. Logical volumes and file backing stores do not perform as well as whole disks and partitions. To determine the device file name, enter the following command:

```
# ioscan -funC disk
```

To prepare for the installation:

- If you are installing from the host's physical CD/DVD, find the physical CD/DVD in the `ioscan` output.
- If you are installing from an ISO file on the host, determine the full path name to that file.

If you are using a disk, partition, or logical volume for the backing storage unit, it should be cleaned to reduce the chance of existing installations causing errors during this installation. For a whole disk, enter the following command:

```
# dd if=/dev/zero of=/dev/rdsk/c0t0d0 bs=1024k
```

where `/c0t0d0` is the disk device name.

For a disk partition, enter the following command:

```
# dd if=/dev/zero of=/dev/rdsk/c0t0d0s0 bs=1024k
```

where `/c0t0d0s0` is the disk partition name.

If you are using a file for the backing storage, use the `hpvmdevmgmt` utilitity with the `-S` option to create the file. The utility automatically cleans the file when it is created

The installation procedure automatically repartitions the virtual disk.

5.2 Installing Windows Guests

To install the Windows guest operating system, follow these steps:

1. Create a Windows guest. In the following example, the virtual machine name (guest name) is `win1`:

```
# hpvmcreate -P win1 -O windows -c 1 -r 2G \
-a disk:scsi::disk:/dev/rdsk/c3t2d0 \
```

```
-a dvd:scsi::disk:/dev/rdsk/c0t0d0 \
-a network:lan::vswitch:switch1
```

The hpvmcreate command creates the /var/opt/hpvm/guests/win1/ directory and creates the guest configuration in that directory.

2. Boot the virtual machine to the EFI menu and take control of the virtual console by entering the following command:

```
# hpvmconsole -P win1 -fi -c "pc -on"

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> pc -on

System will be powered on.

   -> System is being powered on.

Please wait for the guest start sequence to complete.  Use of the attention
character can prevent the guest from running.
(C) Copyright 2000 - 2006 Hewlett-Packard Development Company, L.P.
Opening minor device and creating guest machine container
Creation of VM, minor device 1
Allocating guest memory: 2048MB
  allocating low RAM (0-80000000, 2048MB)
/opt/hpvm/lbin/hpvmapp (/var/opt/hpvm/uuids/682da886-06b2-11db-a3aa-00306e4a931c
/vmm_config.current):
 Allocated 2147483648 bytes at 0x6000000100000000
  allocating firmware RAM (ffaa0000-ffab5000, 84KB)
/opt/hpvm/lbin/hpvmapp (/var/opt/hpvm/uuids/682da886-06b2-11db-a3aa-00306e4a931c
/vmm_config.current):
 Allocated 86016 bytes at 0x6000000180000000
Loading boot image
Image initial IP=102000 GP=62A000
Initialize guest memory mapping tables
Starting event polling thread
Starting thread initialization
Daemonizing....
hpvmstart: Successful start initiation of guest 'win1'

-> Command successful.
[guest1] vMP> CO



        (Use Ctrl-B to return to vMP main menu.)



Loading device drivers
```

```
EFI Boot Manager ver 1.10 [14.62] [Build: Thu Jun  8 12:30:44 2006]

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
Boot option maintenance menu
```

3. Typically, the EFI shell is automatically be selected upon startup. If not, choose **EFI Shell [Built-in]** from the menu.

```
Loading.: EFI Shell [Built-in]
EFI Shell version 1.10 [14.62]
Device mapping table
fs0 : Acpi(PNP0A03,0)/Pci(0|0)/Scsi(Pun1,Lun0)/CDROM(Entry0)
blk0 : Acpi(PNP0A03,0)/Pci(0|0)/Scsi(Pun0,Lun0)
blk1 : Acpi(PNP0A03,0)/Pci(0|0)/Scsi(Pun1,Lun0)
blk2 : Acpi(PNP0A03,0)/Pci(0|0)/Scsi(Pun1,Lun0)/CDROM(Entry0)

Shell>
```

4. Connect to the CDROM file system (listed in the EFI output) by entering the name of the file system. For example:

```
EFI Shell> fs0:
```

5. Enter the `ls` command to make sure you are connected to the right device (look for the `setupldr.efi` file):

```
Fs0: ls
```

Now that the EFI is ready, you can install the Windows® operating system. The procedure for installing the operating system depends on the type of media you are installing from:

- To install the Windows operating system from the reinstall kit (OPK), follow the instructions in "Installing from HP Reinstall (OPK) Media" (page 49).
- To install from Microsoft media, follow the instructions in "Installing from Windows Media" (page 53).

5.2.1 Installing from HP Reinstall (OPK) Media

1. Start the operating system loader by entering the following command:

```
fs0:\> setupldr
```

The installation script runs. The `SAC>` prompt is displayed. When the EVENT message is displayed indicating that the `cmd` command is available, type `cmd` and press **Enter**.

```
Starting BUILD ENT20K.US HP Re-install Environment. Please wait...
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
Press F6 if you need to install a third party SCSI or RAID driver...
.
.
.
SAC>
EVENT: The CMD command is now available.
SAC>cmd
```

```
The Command Prompt session was successfully launched.
SAC>
```

2. In response to the following prompt, press the **ESC** key quickly followed by the **TAB** key. Then press **Enter** to change to the new command channel:

```
Press <esc> <tab> for next channel.
Press 0 to return to the SAC channel.
Use any other key to view this channel.

X:\ia64\system32>
```

3. At the command prompt, type `txtrestore` and press **Enter**:

```
X:\ia64\system32> txtrestore
This is txtrestore.cmd Batch file

Microsoft Windows [Version 5.2.3790] (C) Copyright 1985-2003 Microsoft Corp.
You are about to install Windows Server on the following drive:
Drive 0: HP Virtual Disk SCSI Disk Device (36GB) Bus Number 0, Target ID 0, LUN 0 H
Attached to SCSI Controller In Embedded Slot (PCI bus 0, device 0, function 0)
Drive Layout: Partition table style is MBR
Signature = 0C9C0C9C
WARNING:  If you continue with the installation, all data, including
partition table information, on the above drive will be erased
and permanently lost.

Continue with installation (Y/N)? y
```

Enter `y` to continue using the specified device. To use a different device, enter `n`.

4. If you continue, the following information is displayed:

```
Target drives set to default.

TARGETHDD=0
TARGETPQI=1
ImageDrive=X:
ImagePath=\Images
ScriptDrive=X:
ScriptPath=\IA64\ADDINS
Support=X:\Supp
IVPQET2373ALABA.PQI

================================================================
Note:  <Ctrl/C> Will always exit.
========================== TASK MENU ===========================

A - 16GB Drive Partition Size.
B - 33GB Drive Partition size.
C - Full Drive size.
Q - Exit to console.

================================================================
Select a task by typing a letter A, B, C or Q and then press enter:
WARNING, SELECTING A, B OR C WILL ERASE THE DATA ON BOOT DRIVE.
Input : c
```

Type your selection from the following list, and press **Enter**. HP recommends that you specify `C`.

In response to the following prompt, enter `y`:

```
WARNING, This could potentially wipe out important data!
Recommended: Only the boot controller and hard drive be installed.
Target Drive selected: Drive0
Target Partition Size Selected: Full Drive size
Input [Y/N] : y
```

5. Do not power off the system while the files are being copied to the hard drive. The restore process displays 99% complete for a long time. At the end of the successful process, the following message is displayed. Type `Exit` to reboot the system.

```
Re-install Total Time: 23 minutes
Re-Install Finished [OK].

To restart the system...
Type [exit] to exit from command console.
At the SAC console, type [restart].

X:\IMAGES> exit
The Command Console session is exiting.
```

If an error message is displayed, solve the problem that causes the error and restart the process.

At the following prompt, press **ESC Tab** to get the SAC prompt.

Restart the system by pressing **Enter** in response to the SAC prompt:

```
Press <esc> <tab> for next channel.
Press 0 to return to the SAC channel.
Use any other key to view this channel.


SAC> Enter
```

6. The virtual machine is restarted. When you are prompted for the boot option, press **Enter** to accept the **Windows Server 2003** option.

```
The SAC will become unavailable soon.  The computer is shutting down.

SAC>

*** VM restarting ***

Scsi(Pun1,Lun0) HP      Virtual FileDVD 0.04
Scsi(Pun1,Lun0) HP      Virtual FileDVD 0.04 ( 80 MBytes/sec)

EFI version 1.10 [14.62] Build flags: Running on Intel(R) Itanium Processor EFI_DEBUG
EFI IA-64 SDV/FDK (BIOS CallBacks) [Thu Jun  8 12:30:44 2006] - HP
Cache Enabled. This image MainEntry is at address 0000000000102000
FPSWA.EFI preload successful.
FPSWA.EFI start successful.

Loading device drivers

EFI Boot Manager ver 1.10 [14.62] [Build: Thu Jun  8 12:30:44 2006]

Please select a boot option
```

```
Windows Server 2003, Enterprise
EFI Shell [Built-in]
Boot option maintenance menu
Use ^ and v to change option(s). Use Enter to select an option
Default boot selection will be booted in 30 seconds


Starting: Windows Server 2003, Enterprise
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

Starting Windows...
```

7. When the Windows SAC> prompt says EVENT:, the CMD command is available. Press the **Escape** key and quickly press the **TAB** key, then press **Enter** to go to the following command prompt:

```
Computer is booting, SAC started and initialized.
Use the "ch -?" command for information about using channels.
Use the "?" command for general help.
SAC>...
EVENT:   A new channel has been created.  Use "ch -?" for channel help.
Channel: Unattended Setup Channel
SAC>
EVENT: The CMD command is now available.
SAC>
SAC>
  END-USER LICENSE AGREEMENT FOR MICROSOFT SOFTWARE
  MICROSOFT WINDOWS SERVER 2003, STANDARD EDITION WITH SERVICE PACK 1
  MICROSOFT WINDOWS SERVER 2003, ENTERPRISE EDITION WITH SERVICE PACK 1

  IMPORTANT-READ CAREFULLY: This end user license agreement ('EULA') is a legal
  agreement between you (either an individual or a single legal entity) and the

  manufacturer ('Manufacturer') of the computer system ('Server') with which you
  acquired the Press F8 to accept or ESC to decline the EULA.
  Press PAGE DOWN for next page.
```

To accept the EULA, press the **Escape** key, then **8**, which emulates the F8 function key.

8. In response to the following prompt, enter the product key, including the dashes:

```
Type the Product Key below in the form
XXXXX-XXXXX-XXXXX-XXXXX-XXXXX

Product ID: XXXXX-XXXXX-XXXXX-XXXXX-XXXXX
```

(The 25–character product key appears on the lower section of your Certificate of Authenticity.)

9. At the following prompt, enter the administrator password and press **Enter**:

```
Enter the password that will be used for the Administrator
Account on this machine.  This password must not be blank.
Administrator Password:*******

Please re-enter the Administrator password.
Password Confirmation:*******

Setup will now proceed in an automated fashion.
```

10. Use **ESC-TAB** to cycle through screens until you see the one running the installation script. The system boots to the Windows operating system.

    Use **ESC-TAB** to go back to the `SAC>` prompt.

11. Obtain the guest's IP address by entering the `i` command to the `SAC>` prompt.

    Use the Remote Desktop Connection from a PC to connect to that IP address and log in as Administrator. The Support Pack completes installation:

```
Computer is booting, SAC started and initialized.
Use the "ch -?" command for information about using channels.
Use the "?" command for general help.
SAC>...
EVENT: The CMD command is now available.
SAC>i
Net: 2, Ip=1.2.3.4  Subnet=255.255.248.0  Gateway=16.116.0.1
SAC>
```

NOTE: If the system displays `System model could not be verified`, simply acknowledge the dialog. When asked to confirm, select **No** and then reboot the virtual machine.

12. From the Windows control panel, follow the initial system setup instructions from the platform user's guide. For example:
    a. **Control-Panel**->**System (Set Computer name)**
    b. **Control-Panel**->**System**->**Advanced**->**Virtual Memory**
13. Reboot the virtual machine.

5.2.2 Installing from Windows Media

To install the Windows Enterprise operating system from Windows media, follow these steps:

1. Allocate a virtual DVD to the virtual machine, as described in Chapter 3 (page 27).
2. Boot the virtual machine to the EFI menu and take control of the virtual console by entering the following command:

```
# hpvmconsole -P win1 -i -c "pc -on"

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

[compass1] vMP> pc -on
Please wait for the guest start sequence to complete.  Use of the attention
character can prevent the guest from running.
(C) Copyright 2000 - 2006 Hewlett-Packard Development Company, L.P.
Opening minor device and creating guest machine container
Creation of VM, minor device 1
Allocating guest memory: 2048MB
  allocating low RAM (0-80000000, 2048MB)
/opt/hpvm/lbin/hpvmapp (/var/opt/hpvm/uuids/682da886-06b2-11db-a3aa-00306e4a931c
/vmm_config.current):
 Allocated 2147483648 bytes at 0x6000000100000000
  allocating firmware RAM (ffaa0000-ffab5000, 84KB)
/opt/hpvm/lbin/hpvmapp (/var/opt/hpvm/uuids/682da886-06b2-11db-a3aa-00306e4a931c
```

```
/vmm_config.current):
 Allocated 86016 bytes at 0x6000000180000000
Loading boot image
Image initial IP=102000 GP=62A000
Initialize guest memory mapping tables
Starting event polling thread
Starting thread initialization
Daemonizing....
hpvmstart: Successful start initiation of guest 'win1'

-> Command successful.
[guest1] vMP>
```

3. From the EFI shell, enter the `map` command:

```
Shell> map

Device mapping table
fs0 : Acpi(PNP0A03,0)/Pci(0|0)/Scsi(Pun1,Lun0)/CDROM(Entry0)
blk0 : Acpi(PNP0A03,0)/Pci(0|0)/Scsi(Pun0,Lun0)
blk1 : Acpi(PNP0A03,0)/Pci(0|0)/Scsi(Pun1,Lun0)
blk2 : Acpi(PNP0A03,0)/Pci(0|0)/Scsi(Pun1,Lun0)/CDROM(Entry0)

Shell>
```

4. Enter the file system entry for the DVD. For example:

```
Shell> fs0:

Fs0:
```

5. Enter the `setupldr` command:

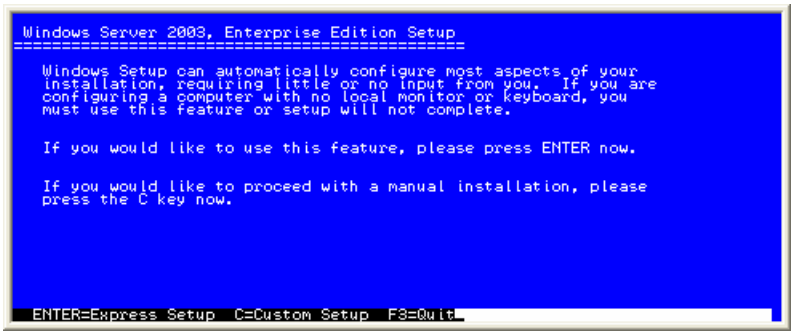
```
Fs0:> setupldr
```

The Windows Setup Begins... screen is displayed. The status bar shows the following:

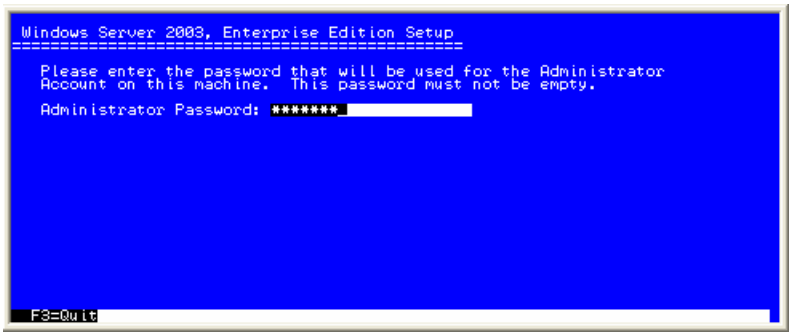
```
Setup is loading files (Windows Executive)...
```

6. On following page, continue by pressing **Enter**:

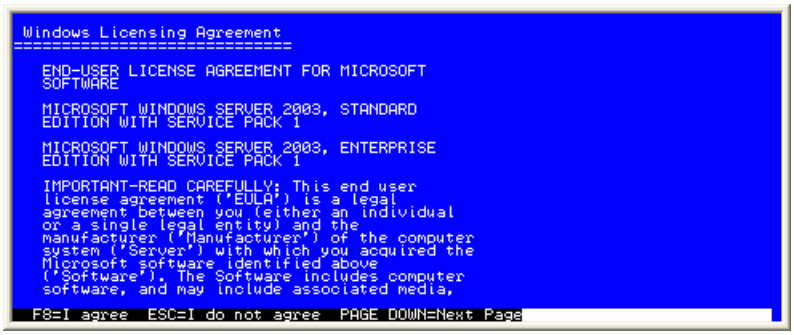
7. When prompted whether to use Express Setup or Custom Setup, select Express Setup. Do not select Custom Setup as its functionality is not yet supported on virtual machines.



8. Enter and confirm an Administrator password:



9. Accept the Windows license by pressing **ESC 8**, which emulates F8.



10. Select the partition in which to install Windows:

If the disk you are installing to has been cleaned as described in "Installing Windows Guests" (page 47), two partitions (EFI and Reserved) are automatically created. The remainder of the disk is unpartioned space. Select the unpartitioned space using the down arrow. Create a new partitiion by entering C.

If the disk was not cleaned in preparation for this procedure, a list of the current partitions is displayed. using the down arrow, select the existing NTFS partion. Enter `D` to delete it, and then enter `L` to confirm, and press **Enter**. Create a new partition by entering `C`.

11. Select the new (raw) partition to which to install Windows:

Choose an NTFS partition. If there is no NTFS partition, format one. The partition is formatted, the files are copied, and the system reboots.

12. Start the Windows installation. In response to the SAC> prompt, press **Escape-Tab**

```
SAC>...Escape-Tab

Setup is being restarted.................
EVENT: A new channel has been created.
 Use "ch -?" for channel help.
 Channel: Unattended Setup Channel
SAC> EVENT: The CMD command is now available
```

The following message is displayed:

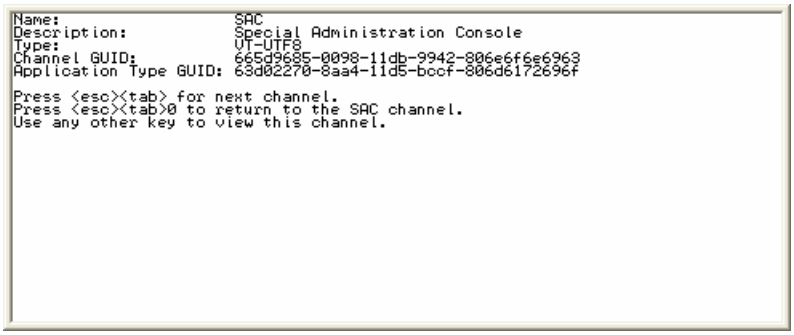
```
Name: Unattended Setup Channel Description:
.
.
.
Press Escape-Tab for next channel.
Press 0 to return to the SAC channel.
Use any other key to view this channel.
```

Press **Enter** to accept the Unattended Setup Channel.

13. In response to the following prompt, enter the 25-character product key.

```
Product ID:
XXXXX-XXXXX-XXXXX-XXXXX-XXXXX
```

Setup proceeds automatically. To monitor the installation, enter **Escape-Tab**.

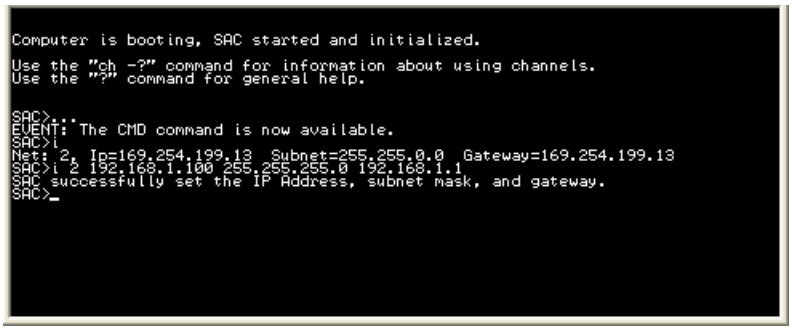


The setup process can take several minutes to complete.

14. When the Windows operating system is installed, Windows reboots and displays the SAC> prompt:

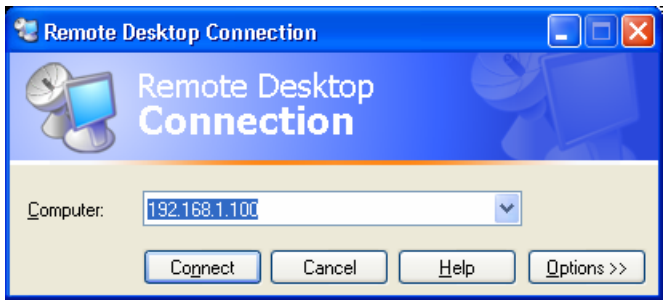
```
SAC> EVENT: The CMD command is now available.
```

15. By default, the machine name is set to a temporary name, and the network defaults to DHCP. To check this information, enter the `id` and `i` commands in response to the SAC> prompt:



5.3 Managing Windows Guests

Use the Windows Remote Desktop to manage the Windows guest:



Log in as Administrator and configure Windows TCP/IP using the same networking information supplied earlier at the SAC> prompt. Install SNMP from the Windows Management and Monitoring Tools and configure it. SNMP is required for the HP Systems Insight Manager (SIM), which provides the Virtual Server Environment (VSE).

The HP Integrity Support Pack for Windows provides the components required for managing the Windows guest using VSE, including:

- Agents for HP SIM
- Providers for VSE
- Basic Windows updates for Integrity VM
- System Management Homepage

To install the SSM media, you must first insert it into the virtual DVD. If the virtual DVD is mapped to the physical DVD, then you will need to physically insert the SSM media disk into the physical DVD drive on the VM Host.

If your virtual DVD is mapped to an ISO file (for example, the Windows installation media), you must virtually eject the current media and virtually insert the SSM. To do this, modify the virtual DVD so that it now maps to the ISO file containing the SSM.

For example, the following `hpvmstatus` command displays the virtual DVD on the Windows guest `vmwin0`:

```
# hpvmstatus -P vmwin0 |grep scsi

disk    scsi     0   0   0   0   0 file    /hpvm/VHD/vmwin0/vhd0
dvd     scsi     0   0   0   1   0 file    /ISO/MS_Ent_Ed_wSP1.iso
```

To change the virtual DVD, you must use the virtual bus, device, and target values for the existing DVD with the `hpvmmodify` command. For example, in the `hpvmstatus` example, the virtual bus, device, and target are 0, 0, and 1, respectively. The corresponding `hpvmmodify` command is:

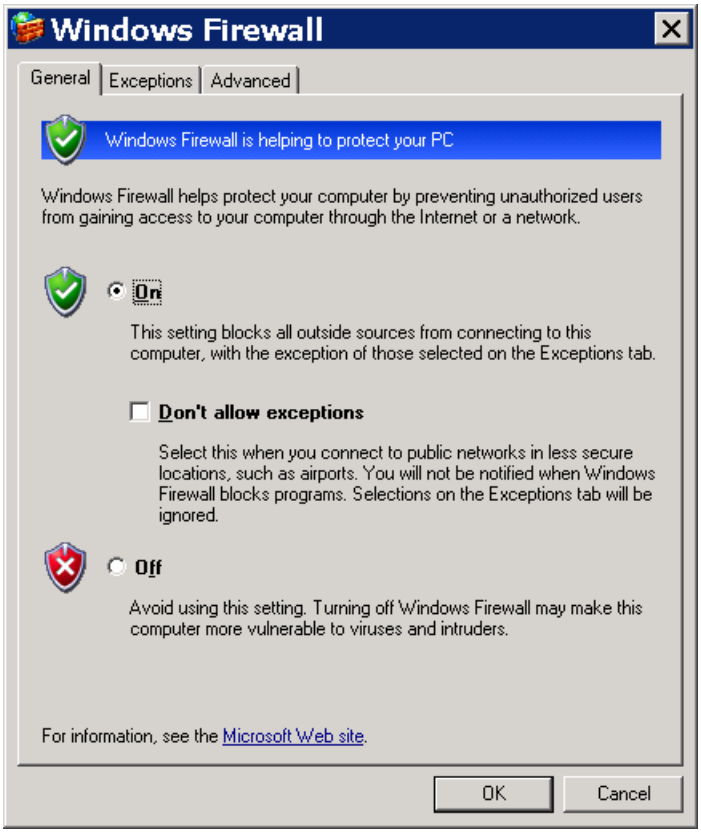
```
# hpvmmodify -P vmwin0 -m dvd:scsi:0,0,1:file:/ISO/SmartSetup.iso
```

Now that the SSM is in the virtual DVD, it will appear in the Windows Explorer display that corresponds to **My Computer**. From the Explorer window, open the DVD and then open the `start.html` file there. This file has infomration to guide you through installation of SSM. Consult the *Smart Setup Guide* for more details.

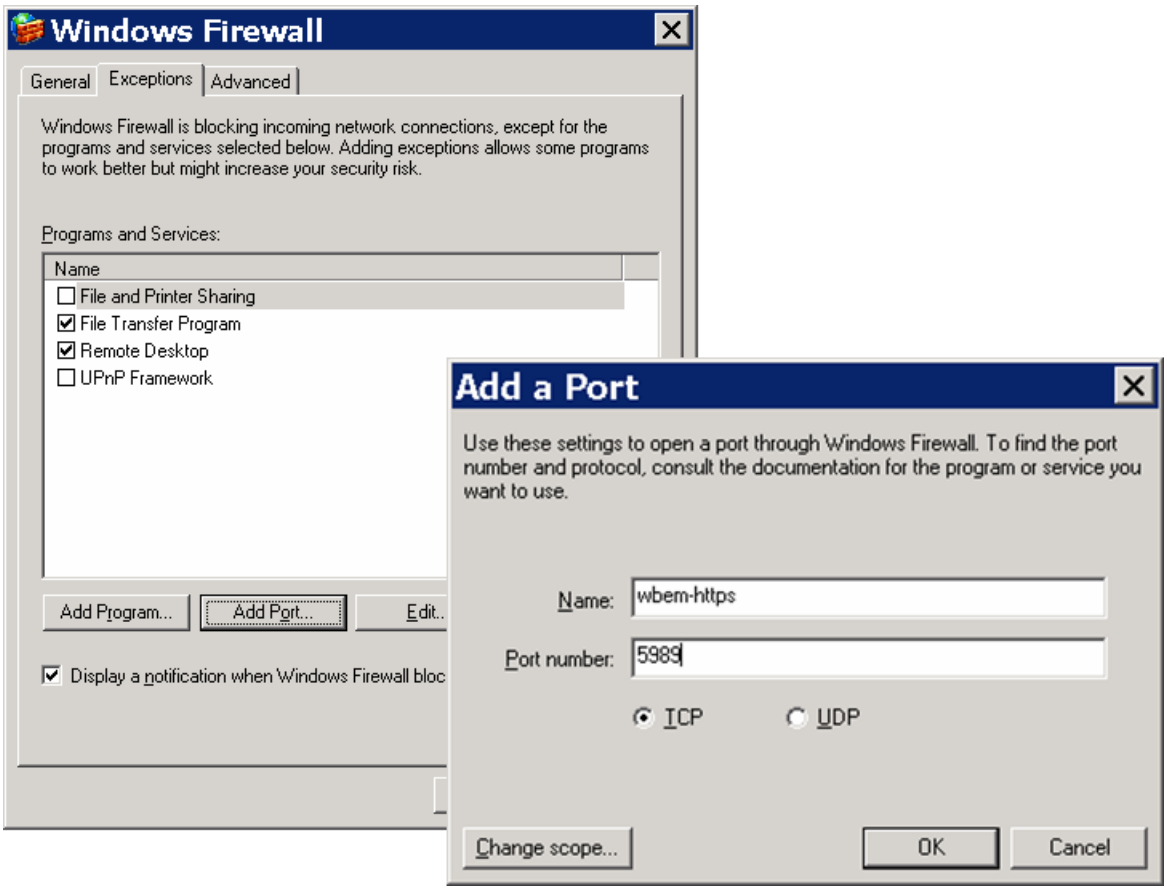
After install the Smart Setup, install the components of HP's Smart Update media. This media is available from HP. Insert the Smart Update media into the virtual DVD as described for SSM. You initiate the Smart Update installation the same way you initiated SSM. Open the virtual DVD, then open the `start.html` file. Install OS and Security components and other components of Smart Update that apply to your Windows system deployment

To enable the Virtual Server Environment (VSE), modify the Windows Firewall settings, as follows:

1. As the adminstrator of the Windows guest, open the Windows Firewall utility from the Control Panel and select the Exceptions tab:



2. Create an opening in the firewall for the VSE providers. Select **Add Port...** Set the name to `wbem-https`. Set the port number to `5989`, and select TCP. For example:



You can now monitor and manage the Windows guest using VM Manager and SMH. For more information, see the *Getting Started with VM Manager* manual.

To shut down a Windows guest, use the Windows system management procedures. The Integrity VM commands for stopping guests do not shut down the Windows software properly and can lead to problems when the Windows guest is rebooted.

5.4 Installing Integrity VM Windows Guest Management Software

After you install the Windows guest operating system, you must install the Integrity VM Windows guest management kit from the VM Host system. When you install Integrity VM, the Windows guest management kit is loaded into the following directory: `/opt/hpvm/guest-images/windows`

This directory contains a README.TXT file that contains instructions for copying the Windows guest management kit to the Windows guest.

5.5 Troubleshooting Windows Guest Installation

You can use the following commands at the SAC prompt:

- `ID` displays the host name and the system up time.
- `I` displays network information.
- `S` shows the system time and date.
- `CMD` creates a command-shell channel. Commands you can use in the command shell include:
  - `TASKLIST`
  - `NET START "TERMINAL SERVICES"`
  - `IPCONFIG`
  - `SAFEBOOT`

5.5.1 Remote desktop unable to connect

If the Windows guest is installed and on, but the Remote Desktop displays the following error, you might have to set the required Windows Registry parameters:

```
Unable to connect
```

Set the following registry parameters:

- `fDenyTSConnections` should be set to 0.
- `TSEnabled` should be set to 1.

6 Creating Virtual Storage Devices

This chapter describes what Integrity VM storage is, how to configure it, and how to use it, including:

- "Introduction to Integrity VM Storage" (page 61)
- "Configuring Integrity VM Storage" (page 64)
- "Using Integrity VM Storage" (page 82)

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" (page 61)
- "Integrity VM Storage Architectures" (page 62)
- "Integrity VM Storage Implementations" (page 63)

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" (page 61)
- "Storage Availability" (page 61)
- "Storage Performance" (page 61)
- "Storage Security" (page 61)
- "Storage Configurability" (page 62)

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 wayt 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 doesn't support on an HP 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 insuring 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" (page 62).

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 hard disk partitions, 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 IO 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 Integrity VM Storage Implementations

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

6.1.3.1 Integrity VM Storage Adapters

Integrity VM provides a virtual PCI parallel SCSI MPT adapter to process virtual storage I/O requests. 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.

6.1.3.2 Integrity VM Storage Devices

Integrity VM supports a variety of virtual and attachable devices. Disk and DVD-ROM devices have been virtualized to support several virtual media types. Physical tapes, media changers, and CD/DVD burners are attachable; they can be used to perform data backups directly from a virtual machine (see "Attached Devices" (page 63)).

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 71)
Virtual PartDisk	VM Host disk partition	"Virtual PartDisks" (page 72)
Virtual LvDisk	VM Host LVM or VxVM logical volume	"Virtual LvDisks" (page 73)
Virtual FileDisk	VM Host VxFS file	"Virtual FileDisks" (page 76)

The following virtual DVD-ROM types are supported:

Virtual DVD Type	Backing Storage Device	For more information, see...
Virtual DVD	Disc in a VM Host physical DVD drive	"Virtual DVDs" (page 76)
Virtual FileDVD	ISO file on a VM Host VxFS file system	"Virtual NullDVDs" (page 78)
Virtual NullDVD (empty)	VM Host physical DVD drive with media inserted or an ISO file in a VxFS directory	"Virtual FileDisks" (page 76)

6.1.3.2.1 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.

Command interpretation for attached devices is based on the following SCSI standards:

Attached Device	SCSI Standard
CD/DVD Burners	MMC-4
Media Changers	SMC-2
Tape Devices	SSC-2

Vendor-specific commands are not interpreted by Integrity VM; therefore, all such commands are rejected.

An attachable device can be attached to only one virtual machine at a time. No reservation commands, persistent or otherwise, are sent to any attached devices. These commands are rejected.

A maximum I/O size of 512 KB is enforced by the Integrity VM pass-through driver. This allows Integrity VM to support many types of VM Host storage adapters.

6.2 Configuring Integrity VM Storage

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

- "Integrity VM Storage Considerations" (page 64)
- "Setting up Virtual Storage" (page 69)

6.2.1 Integrity VM Storage Considerations

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

- "VM Storage Supportability" (page 64)
- "Storage Performance" (page 61)
- "VM Storage Multipath Solutions" (page 66)
- "VM Storage Management" (page 67)
- "VM Storage Changes" (page 68)
- "Virtual Storage Setup Time " (page 69)

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 one of the following adapter and driver types:
  - Fibre Channel adapters supported by the TD driver
  - Fibre Channel adapters supported by the FCD driver
  - SCSI adapters supported by the C8*xx* driver
  - SCSI adapters supported by the MPT driver
  - SCSI adapters supported by the CISS driver
  - IDE adapters supported by the SIDE driver
  - USB adapters supported by the UsbScsiAdaptor driver

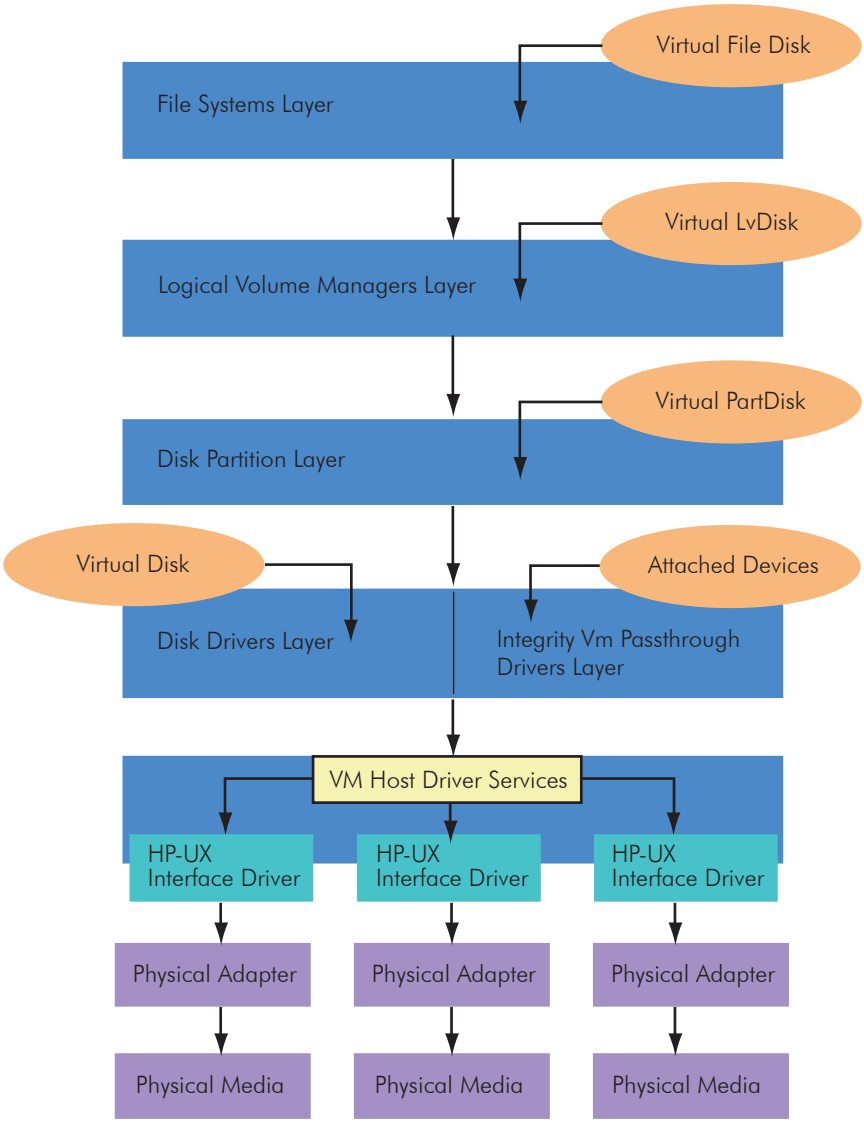
  If the physical storage is not connected with one of above adapter and driver types, it cannot be used by a virtual machine. Use the `ioscan` command to display the VM Host storage that is connected to adapters and drivers.
- Any VM Host attachable devices available for use by a virtual machine must be supported by the guest OS it is attached to. If the physical device is not supported by the guest OS, the device cannot be attached to the virtual machine.

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 6-1.

Figure 6-1 Integrity VM Storage IO 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 6-1, 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, any disk partitions the logical volumes are on, and the disk drivers that control the whole disk where the disk partitions reside. 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 6-1 does not completely illustrate all the choices that can affect the performance:

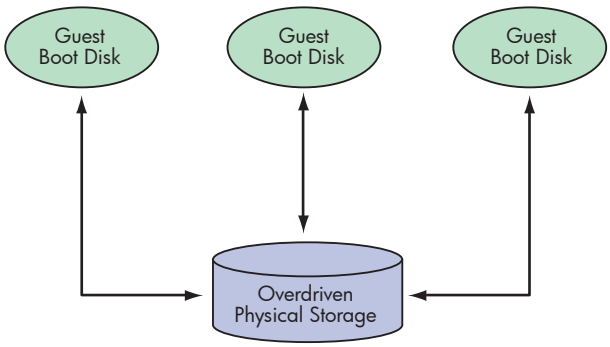
- The software layers are not equal. The difference in performance between disk partitions and whole disks is much smaller than the difference between disk partitions and logical volumes.
- The interfaces to each software layer are different, allowing Integrity VM different ways to send I/O through the layers. For example, disk partitions and 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 http://docs.hp.com.

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 partitions, logical volumes or files. Slicing up physical disks increases utilization, but it can affect the performance of the physical device. The guest OS treats the virtual disk as a whole disk, not as a part of a physical one. Over-slicing physical storage can overload a physical device's ability to handle virtual I/O that is meant for whole disks. Figure 6-2 shows a common mistake of overdriving physical storage with multiple guest OS boot disks, which are often I/O intensive.

Figure 6-2 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 6-1, 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 installing a multipath solution on the VM Host. See "VM Storage Multipath Solutions" (page 66) for help with selecting a multipath solution for a virtual media type.

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

For load balancing and higher availability for virtual machines, consider using a multipath solution on the VM Host. Currently there are no multipath solutions for the attachable device types of tapes, media changers, and CD/DVD burners. However, there are several VM Host multipath options for virtual devices.

Multipath solutions are supported on the VM Host only, not on virtual machines, 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 6-1 (page 65)).
- 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 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 on to the virtual machine.

Each multipath software solution for HP-UX 11.23 interacts at different layers on the I/O stack. Since Integrity VM also interacts with different layers in the I/O stack, only certain options apply to each virtual media type.

Table 6-1 lists the multipath solutions to use on a VM Host for each type of virtual storage media:

Table 6-1 Multipath Solutions

Virtual Media Type	Multipath Options
Whole Disk Disk Partition	EMC PowerPath HP Autopath/SecurePath
LVM Logical Volume	PVLinks EMC PowerPath HP Autopath/SecurePath
VxVM Logical Volume	Veritas DMP EMC PowerPath HP Autopath/SecurePath
VxFS File System	PVLinks Veritas DMP EMC PowerPath HP Autopath/SecurePath

Although Table 6-1 lists the possible solutions for each virtual media type, it cannot determine what is supported on your specific VM Host configuration. Each multipath solution is only supported for specific hardware and software. The solution vendors provide this information for their multipath products. Review the installation and release notes of these products carefully to form a valid VM Host configuration before using it for any virtual machine. Some multipath options do not work together and they all have different load balancing features.

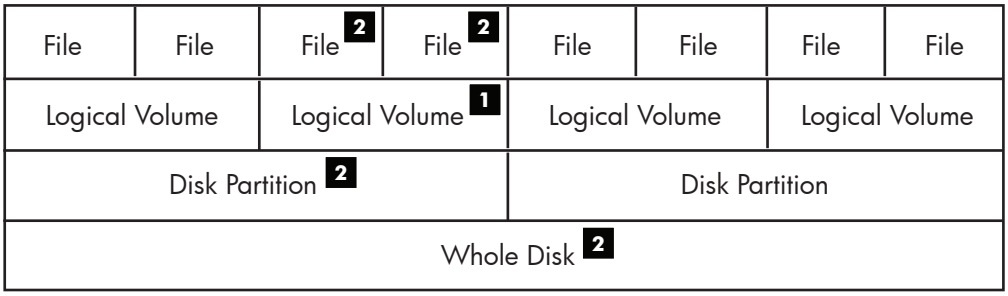
6.2.1.4 VM Storage Management

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 where the virtual media is being allocated from.

Figure 6-3 shows an example of a VM Host I/O stack as it applies to a single LUN:

Figure 6-3 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 marked **1**.
- The parts of the disk that cannot be allocated are marked **2**.

The remaining parts of the disk can be allocated to a virtual machine.

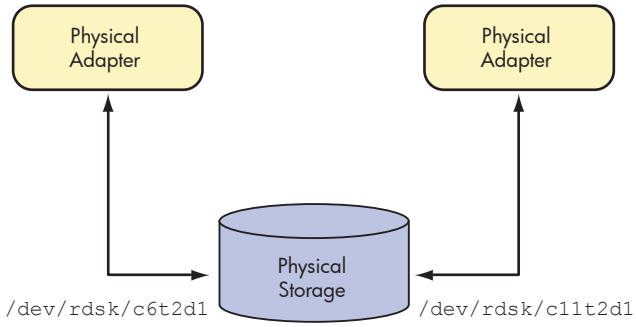
Those parts that are no longer available include the files that were on the logical volume, the disk partition that makes up part of volume group, and the whole disk that makes up part of the volume group. If any of these parts are allocated for other virtual devices, data corruption can occur on the Virtual LvDisk.

Those parts that are still available for reallocation include other logical volumes that are on the disk, files that are on those other logical volumes on the disk, and the other disk partition that is not part of the volume group that the Virtual LvDisk is on. 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. Under HP-UX 11.23, most storage device files are defined per path. Be careful not to specify a given device twice. Figure 6-4 shows an example of two device files, `/dev/rdsk/c6t2d0` and `/dev/rdsk/c11t2d0` pointing

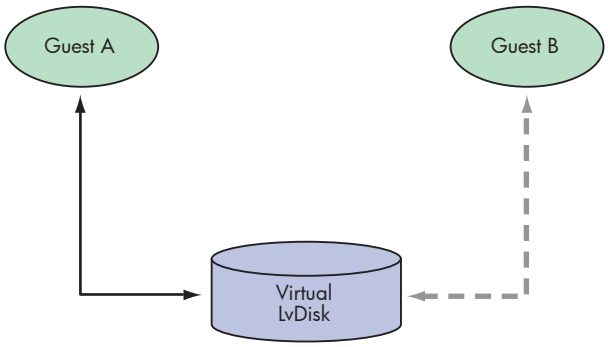
to the same physical disk. Once the `/dev/rdsk/c6t2d0` device file is specified for a Virtual Disk, the `/dev/rdsk/c11t2d0` device file is no longer available.

Figure 6-4 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 6-5 shows a Virtual LvDisk being shared across virtual machines, which is not supported.

Figure 6-5 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. Use the HP System Administration Manager or the *sam(1M)* utility. The `sam` utility provides the ability to track disk devices, volume groups, logical volumes, and file systems. It attempts to consolidate multipaths to not show disks more than once. Additionally, it provides the ability to annotate devices so that VM Host administrators can see exactly which virtual machines are using what VM Host storage devices. The `sam` utility does not track individual disk partitions. However, you should use all of the parts of a single disk on a single virtual machine when you are dividing up the disk. Allocating different parts of the same disk to different virtual machines makes it difficult to manage and to isolate problems.

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 may permit the adjustment of a LUN without losing that LUN's data. Disk partitions are not adjustable without losing the disk's data. However, soft partitions, such as 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 may 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 may 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.

Disk partitions require the creation of the hard disk partitions and their corresponding device files. Though not difficult, this can be time consuming.

Logical volume creation is relatively simple. Logical volumes are used widely on HP-UX systems. The `sam` utility or 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. Use `sam` to accomplish this.

To create empty files for virtual disks, use the `hpvmdevmgmt` command (see "Managing the Device Database" (page 107)).

To create ISO files from physical CD/DVD media for use in virtual DVDs, use the *mkisofs(1M)* or the *dd(1M)* 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(1M)* for information about creating pass-through device files using *mknod(1M)*.

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 "Creating Virtual Networks" (page 89)), or virtual storage devices.

This section describes how to enter resource statements for use with the `hpvmcreate` command (described in Chapter 3 (page 27)) and the `hpvmmodify` command (described in Chapter 8 (page 101)). 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 outline of a complete resource statement for specifying a virtual storage device is the following:

`VM guest storage specification``:`*`VM Host storage specification`*

where:

- *`VM guest storage specification`* defines where and what storage is seen in the virtual machine.
- *`VM Host storage specification`* defines where and how the virtual machine storage is supplied on the VM Host.

6.2.2.1 VM Guest Storage Specification

All virtual storage is addressed from virtual PCI buses. There are 8 PCI buses on the Integrity VM virtual platform. 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 virtual machine.

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.

  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.

- *pcislot* is an integer from 0-7.

  The virtual MPT adapters are only supported on PCI buses 0-6. PCI bus 7 is reserved for other use.

- *scsitgt* is an integer from 0-14 (15 is reserved for the virtual SCSI adapter).

  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.

A PCI function number is not specified. It is implicitly zero because the virtual MPT storage adapter supports only a single channel.

A virtual SCSI MPT adapter can only be added 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 will use. For example, the selection of lv for a virtual disk, makes it a Virtual LvDisk to the VM.

  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.

  There may be more than one VM Host system file that points to the same VM Host storage entity. For example, if there are multiple paths to storage present on the VM Host, there can be more than one disk system file that points 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 "VM Storage Management" (page 67)).

Not all virtual device types support all VM Host storage types (see "Integrity VM Storage Implementations"). Complete VM storage resource statements are discussed in the next section.

6.2.2.3 VM Storage Resource Statements

This subsection provides information on 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"VM Guest Storage Specification" (page 69)) and the VM Host Storage Specification (described in "VM Host Storage Specification" (page 70)). This section provides examples of complete resource statements for each of the following types of virtual storage devices:

- "Virtual Disks" (page 71)
- "Virtual PartDisks" (page 72)
- "Virtual LvDisks" (page 73)
- "Virtual FileDisks" (page 76)
- "Virtual DVDs" (page 76)
- "Virtual FileDVDs" (page 77)
- "Virtual NullDVDs" (page 78)
- "Attachable Devices" (page 79)

A virtual machine can have up to 30 devices total (number of virtual and attached devices).

The maximum size of a virtual storage resource is 2 TB. 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 "VM Storage Management" (page 67)). 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 "VM Storage Multipath Solutions" (page 66)).

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 must be owned by the HP-UX sdisk driver.

Virtual Disk resources cannot be shared simultaneously across active virtual machines. 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 "VM Storage Management" (page 67).

To provide a multipath solution for a Virtual Disk, see "VM Storage Multipath Solutions" (page 66).

The Virtual Disk resource statement takes the form of:

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

Where *`/dev/rdsk/cXtYdZ`* is an HP-UX character `sdisk` device file.

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 sees during boot. New devices connected or created after boot time, require the use of `ioscan` and `insf` to create the new `sdisk` device files. Old device files for storage not longer present can be removed with `rmsf`. For example:

```
# ioscan

# ioscan -funC disk

disk 110 0/5/1/0.11.16.0.0.0.2 sdisk CLAIMED DEVICE HP      A6188A
disk 116 0/5/1/0.11.16.0.0.0.3 sdisk CLAIMED DEVICE HP      A6188A
/dev/dsk/c19t0d3   /dev/rdsk/c19t0d3

# insf -H 0/5/1/0.11.16.0.0.0.2

# ioscan -funC disk

disk 110 0/5/1/0.11.16.0.0.0.2 sdisk CLAIMED DEVICE HP      A6188A
/dev/dsk/c19t0d2   /dev/rdsk/c19t0d2
disk 116 0/5/1/0.11.16.0.0.0.3 sdisk CLAIMED DEVICE HP      A6188A
/dev/dsk/c19t0d3   /dev/rdsk/c19t0d3
```

In this example, the Virtual Disk Resource Statement is `disk:scsi::disk:/dev/rdsk/c19t0d2`.

If you are using EMC PowerPath or HP SecurePath/Autopath for a Virtual Disk, use their respective commands to ensure the `sdisk` device files chosen are enabled for use by the multipath product. Consult the multipath vendor's documentation for more information.

6.2.2.3.2 Virtual PartDisks

A Virtual PartDisk is an emulated SCSI disk whose virtual media comes from a VM Host disk partition. The VM Host disk partition is specified using a character device file. The character device file is owned by the HP-UX `sdisk` driver.

Virtual PartDisks cannot be shared simultaneously across active virtual machines. Only one active virtual machine at time can be given a particular Virtual PartDisk resource. Virtual PartDisk resources can be changed dynamically between active virtual machines (see "Using Integrity VM Storage" (page 82)).

VM Host disk partitions must be managed to prevent virtual media conflicts that can result in data corruption. To help with the accounting, HP recommends that all disk partitions on a single VM Host disk LUN be used with a single virtual machine. See "VM Storage Management" (page 67) for more information on tracking virtual media allocation

To provide a multipath solution for a Virtual PartDisk, see "VM Storage Multipath Solutions" (page 66).

To create a disk partition on a VM Host disk LUN, use the `idisk` utility. This utility was originally designed to create boot disks, but you can use it to create hard disk partitions on any disk. When you create hard disk partitions, you must decide how many partitions to create and what sizes they should be. Changing the partitioning after creation wipes out all data on the disk. If you need more flexibility, use logical volumes, which provide soft partitioning that can be adjusted after creation.

After making these decisions, use an editor to create an input file to the `idisk` command. The input file begins with the total number of partitions to create, followed by the partition types and sizes. The partitions types of EFI, HPUX, and DUMP do not matter to Integrity VM and have no effect on the data stored on the partitions. Because the `idisk` command was originally designed for creating a boot disk, it forces the first partition type to be an EFI type. There are no restrictions on partition sizes based on partition types. For example, create an input file as follows:

```
# vi vdisk_part_file
2
EFI 4096MB
HPUX 8192MB
```

After you complete the input file, use it with the `idisk` command to create the partitions on a VM Host disk. For example:

```
# idisk -w -f vdisk_part_file /dev/rdsk/c4t0d0
```

After the partitions are created, run the `ioscan`, `rmsf` and `insf` utilities to create the `sdisk` device files, as follows:

```
# ioscan -funC disk


disk 4 0/3/1/0.0.0 sdisk CLAIMED DEVICE HP 36.4GMAP3367NC
/dev/dsk/c4t0d0     /dev/rdsk/c4t0d0

# rmsf -H 0/3/1/0.0.0

# ioscan

# insf -H 0/3/1/0.0.0

# ioscan -funC disk

disk 4 0/3/1/0.0.0 sdisk CLAIMED DEVICE HP 36.4GMAP3367NC
/dev/dsk/c4t0d0     /dev/rdsk/c4t0d0
/dev/dsk/c4t0d0s1   /dev/rdsk/c4t0d0s1
/dev/dsk/c4t0d0s2   /dev/rdsk/c4t0d0s2
```

The new `sdisk` partitions files (those ending with "s"), are numbered in order with respect to the `idisk` input file:

```
# diskinfo /dev/rdsk/c4t0d0s2

SCSI describe of /dev/rdsk/c4t0d0s2:
             vendor: HP 36.4G
         product id: MAP3367NC
               type: direct access
               size: 8388608 Kbytes
   bytes per sector: 512
```

The Virtual PartDisk resource statement form is:

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

Where the `/dev/rdsk/cXtYdZsV` is an HP-UX character `sdisk` device file for a hard disk partition.

```
# ioscan -funC disk

disk 4 0/3/1/0.0.0 sdisk CLAIMED DEVICE HP 36.4GMAP3367NC
/dev/dsk/c4t0d0     /dev/rdsk/c4t0d0
/dev/dsk/c4t0d0s1   /dev/rdsk/c4t0d0s1
/dev/dsk/c4t0d0s2   /dev/rdsk/c4t0d0s2
```

In this example, the Virtual PartDisk Resource Statement is `disk:scsi::disk:/dev/rdsk/c4t0d0s`.

If you are using EMC PowerPath or HP SecurePath/Autopath for a Virtual PartDisk, use their respective commands to ensure the sdisk device files chosen are enabled for use by the multipath product. Consult the multipath vendor's documentation for more information.

6.2.2.3.3 Virtual LvDisks

A Virtual LvDisk is an emulated SCSI disk whose virtual media is provided by a 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 "Using Integrity VM Storage" (page 82)).

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. See "VM Storage Management" (page 67) for information about tracking virtual media allocation.

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 top of volume group types, which are created on top of disk partitions or 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.

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                      16
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/dsk/c4t1d0
   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 form of:

`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       -        71680564 -        -        -       -

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 multipath solutions for Virtual LvDisks, see "VM Storage Multipath Solutions" (page 66).

6.2.2.3.4 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 "Using Integrity VM Storage" (page 82)).

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 machines. Additionally, it may help to allocate file directories from complete logical volumes or whole disks to make the accounting even easier. See "VM Storage Management" (page 67) for more details.

The Virtual FileDisk resource statement takes the form of:

`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, disk partition, or logical volume. For files over 2GB, 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, empty files for Virtual FileDisk can be created using `hpvmdevmgt`.

```
# mkfs -F vxfs -o largefiles /dev/dsk/c1t2d0
# mount /dev/dsk/c1t2d0 /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 "VM Storage Multipath Solutions" (page 66).

6.2.2.3.5 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 "Using Integrity VM Storage" (page 82)).

The Virtual DVDs, being read-only, do not require management to prevent conflicts writing to the device. However, to prevent potentially 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 Virtual DVD resource statement takes the form of:

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

Where /dev/rdsk/cXtYdZ is an HP-UX character device file representing a VM Host CD/DVD drive.

Typically, the HP-UX sdisk character file will already be created before booting the VM Host. If it is not, it can be created and managed using the ioscan, insf, and rmsf utilities. For example:

```
# ioscan -funC disk

disk 0 0/0/2/0.0.0.0 sdisk CLAIMED DEVICE HL-DT-STDVD+RW GCA-4040N
/dev/dsk/c0t0d0   /dev/rdsk/c0t0d0

# diskinfo /dev/rdsk/c0t0d0
SCSI describe of /dev/rdsk/c0t0d0:
             vendor: HL-DT-ST
         product id: DVD+RW GCA-4040N
               type: CD-ROM
               size: 4300800 Kbytes
   bytes per sector: 2048
```

In this example, the Virtual DVD resource statement is dvd:scsi::disk:/dev/rdsk/c0t0d0.

For a Virtual DVD to be recognized by a virtual machine, physical media must be present inside the VM Host CD/DVD drive. If media is not added at virtual machine start time, it may 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 "Virtual NullDVDs" (page 78)). 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 "Guest Administrator" (page 83)).

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 will turn into a Virtual NullDVD and the VM Host CD/DVD drive will unlock. 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.

6.2.2.3.6 Virtual FileDVDs

A Virtual FileDVD is an emulated SCSI DVD-ROM 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 "Modifying VM Storage Devices" (page 84)). 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 16:59 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 1629978624 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 "VM Storage Multipath Solutions" (page 66) for details.

Virtual FileDVDs are read-only and are sharable across active virtual machines. Use the `hpvmdevmgmt` command to mark them sharable.

To prevent media conflicts, you must manage Virtual FileDVDs carefully (see "VM Storage Management" (page 67)). 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.

6.2.2.3.7 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 "Virtual DVDs" (page 76)) or a Virtual FileDVD (see "Virtual FileDVDs" (page 77)) 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 "Guest Administrator" (page 83)). 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 is set up with the following form of the resource specification:

`dvd:scsi::null:`*`/dev/rdsk/cXtYdZ`*

Where *`/dev/rdsk/cXtYdZ`* is an HP-UX character `sdisk` file that points to the VM Host CD/DVD drive.

This is the same as setting up a Virtual DVD (see "Virtual DVDs" (page 76)), 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 -funC disk

disk 0 0/0/2/0.0.0.0 sdisk CLAIMED DEVICE HL-DT-STDVD+RW GCA-4040N
/dev/dsk/c0t0d0   /dev/rdsk/c0t0d0

# diskinfo /dev/rdsk/c0t0d0

SCSI describe of /dev/rdsk/c0t0d0:
             vendor: HL-DT-ST
         product id: DVD+RW GCA-4040N
               type: CD-ROM
               size: 0 Kbytes
   bytes per sector: 0
```

In this example, the Virtual NullDVD resource statement is `dvd:scsi::null:/dev/rdsk/c0t0d0`.

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 "Virtual FileDVDs" (page 77)), 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 16:59 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 1629978624 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 "VM Storage Multipath Solutions" (page 66)). 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 "Managing the Device Database" (page 107).

Virtual NullDVDs require no additional management beyond that required for the Virtual DVD (see "Virtual DVDs" (page 76)) or Virtual FileDVD (see "Virtual FileDVDs" (page 77)) types they become.

6.2.2.3.8 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 `sctl` device files.

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. See the device's product documentation for a list of supported guest OS drivers.

The resource statements for attached devices take the following forms depending upon device type:

- For magnetic tape, use:

  `tape:scsi::attach:`*`/dev/rscsi/cXtYdZ`*

- For media changers, use:

  `changer:scsi::attach:`*`/dev/rscsi/cXtYdZ`*

- For CD/DVD burners, use:

  `burner:scsi::attach:`*`/dev/rscsi/cXtYdZ`*

Where */dev/rscsi/cXtYdZ* is an HP-UX `sctl` device file to the device type specified.

To create an HP-UX `sctl` device file, follow these steps:

1. Run `ioscan` to pick up any new devices that may have just been connected:

```
# ioscan
```

2. Locate the device designated for attachment.

   2a. Install any device special files for these new devices:

   ```
   # insf -e
   ```

   2b. Check to see if the new devices were claimed by VM Host:

   ```
   # ioscan -fun
   ```

   The following is an example of a claimed tape device:

   ```
   tape 1 0/2/1/0.5.0 stape CLAIMED DEVICE HP C7438A
   /dev/rmt/1m            /dev/rmt/c6t5d0BESTn
        /dev/rmt/1mb           /dev/rmt/c6t5d0BESTnb
        /dev/rmt/1mn           /dev/rmt/c6t5d0DDS
        /dev/rmt/1mnb          /dev/rmt/c6t5d0DDSb
        /dev/rmt/c6t5d0BEST    /dev/rmt/c6t5d0DDSn
        /dev/rmt/c6t5d0BESTb   /dev/rmt/c6t5d0DDSnb
   ```

   If the device is not seen in `ioscan -fun`, proceed to step 2c. Otherwise, go to step 3.

   2c. If the device is not claimed, make sure the device is at least seen:

   ```
   # ioscan -fk
   ```

   The following is an example of an unclaimed media changer device:

   ```
   ext_bus 6 0/2/1/0 c8xx CLAIMED INTERFACE SCSI C1010 Ultra160 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 `sctl` device files under the `/dev/rscsi/` directory using the `hpvmdevmgmt` command. For example:

   ```
   # hpvmdevmgmt -I
   ```

4. Locate a `/dev/rscsi` `sctl` device file that corresponds to the device slated for attachment.

   4a. If the device was claimed, the `/dev/rscsi` file ends with the same *`cXtYdZ`* numbers.

   The following is an example of a tape device:

   ```
   Claimed = /dev/rmt/c6t5d0BEST
   SCTL = /dev/rscsi/c6t5d0
   ```

   The following is an example of media changer device:

   ```
   Claimed = /dev/rac/c6t0d0
   SCTL = /dev/rscsi/c6t0d0
   ```

   The following is an example of CD/DVD burner device:

   ```
   Claimed = /dev/rdsk/c4t3d2
   SCTL = /dev/rscsi/c4t3d2
   ```

   Once the `/dev/rscsi` file has been located, proceed to step 5.

   4b. If the device is unclaimed, a `/dev/rscsi` file must be created containing numbers corresponding to the hardware address.

   The following is an example of locating the hardware address for a tape device:

   ```
   ext_bus     6  0/2/1/0         c8xx       CLAIMED
   INTERFACE    SCSI C1010 Ultra160 Wide LVD A6828-60101
   unknown    -1  0/2/1/0.5.0                UNCLAIMED
   UNKNOWN     HP Ultrium Device Hardware Address = 0/2/1/0.5.0
   ```

   The following shows how the hardware address is broken down into controller, target and device numbers:

   - `c` is the instance of 0/2/1/0
   - `ext_bus` is 6
   - `t` is 5
   - `d` is 0
   - The `sctl` file to create is `/dev/rscsi/c6t5d0`

   To create the `sctl` device file, see *scsi_ctl(1M)*.

   Use the `mknod` command, substituting the values in the minor number as noted:

   ```
   # /usr/sbin/mknod  /dev/rscsi/devname c 203 0xCCTL02
   ```

   Where component parts of the minor number are constructed as follows:

Table 6-2 Minor Numbers for sctl Device Files

Minor Number	Construction
CC	Two hexadecimal digits, identifying the controlling interface card by its instance number. The instance value is displayed in `ioscan` output, under column I for the interface hardware type.
T	One hexadecimal digit identifying the drive (target) address.
L	One hexadecimal digit identifying the LUN within the device
0	Hexadecimal digit zero, for reserved portion of the minor number.

The following is an example of the tape device:

```
# /usr/sbin/mknod /dev/rscsi/c6t5d0 c 203 0x065002
```

5. Use the located or created `sctl` device file in specifying the attached device.

For this attached device	Use this resource statement
Tape	`tape:scsi::attach:/dev/rscsi/c6t5d0`
Media changer	`changer:scsi::attach:/dev/rscsi/c6t0d0`
CD/DVD burner	`burner:scsi::attach:/dev/rscsi/c4t3d0`

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 "Using Integrity VM Storage" (page 82)). 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 may 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.

6.3 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 may be played by one or more individuals depending on security requirements and skill sets. The three roles are:

- "VM Host Administrator" (page 82)
- "Guest Administrator" (page 83)
- "Guest User" (page 83)

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 "Changing Virtual Machine Configurations" (page 34))
Display information about the storage devices for a virtual machine.	`hpvmstatus` (see "Monitoring Guests" (page 101))

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 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.

Management function	Integrity VM command
Eject a virtual DVD	`vMP> ej`
Insert a virtual DVD	`vMP> in`

6.3.1.3 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 hold up to 15 storage devices and a virtual machine can use up to 30 storage 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/rdsk/c3t2d0

# hpvmstatus -P myvmm

...
[Storage Interface Details]
...
disk scsi 0 1 0 0 0 disk /dev/rdsk/c7t0d0
disk scsi 0 1 0 1 0 disk /dev/rdsk/c3t2d0
```

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/rdsk/c7t0d0
disk scsi 0 1 0 1 0 disk /dev/rdsk/c3t2d0
disk scsi 0 1 0 2 0 disk /dev/rdsk/c9t0d0
# hpvmmodify -P myvmm -d disk:scsi::disk:/dev/rdsk/c3t2d0

# hpvmstatus -P myvmm

...
[Storage Interface Details]

disk scsi 0 1 0 0 0 disk /dev/rdsk/c7t0d0
disk scsi 0 1 0 2 0 disk /dev/rdsk/c9t0d0
```

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.

The resource statement for modifying a virtual device requires virtual hardware addressing (see "VM Guest Storage Specification" (page 69)). For example:

```
# hpvmstatus -P myvmm

...
[Storage Interface Details]
...
disk scsi 0 1 0 0 0 disk /dev/rdsk/c7t0d0
disk scsi 0 1 0 1 0 disk /dev/rdsk/c3t2d0
disk scsi 0 1 0 2 0 disk /dev/rdsk/c9t0d0

# hpvmmodify -P myvmm -m disk:scsi:0,1,1:lv:/dev/lvrackA/rdisk2
# hpvmstatus -P myvmm

...
[Storage Interface Details]
...
disk scsi 0 1 0 0 0 disk /dev/rdsk/c7t0d0
disk scsi 0 1 0 1 0  lv  /dev/lvrackA/rdisk2
disk scsi 0 1 0 2 0 disk /dev/rdsk/c9t0d0
```

To complete a DVD ejection and insertion, follow the virtual console menus. However, new media selections may 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/rdsk/c1t7d0

SCSI describe of /dev/rdsk/c1t7d0:
             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/rdsk/c0t0d0

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/rdsk/c1t7d0

SCSI describe of /dev/rdsk/c1t7d0:
             vendor: HP
         product id: Virtual NullDVD
               type: CD-ROM
               size: 0 Kbytes
   bytes per sector: 0

[After inserting a new disc on the VM Host CD/DVD drive]

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/rdsk/c0t0d0

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/rdsk/c1t7d0

SCSI describe of /dev/rdsk/c1t7d0:
             vendor: HP
         product id: Virtual DVD
               type: CD-ROM
```

```
               size: 4300800 Kbytes
   bytes per sector: 2048
```

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/rdsk/c1t7d0

SCSI describe of /dev/rdsk/c1t7d0:
             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/rdsk/c1t7d0

SCSI describe of /dev/rdsk/c1t7d0:
             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/rdsk/c1t7d0

SCSI describe of /dev/rdsk/c1t7d0:
             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 stand-alone drives, and changing discs with CD/DVD burners are accomplished manually. This requires cooperation between the VM Host administrator and the guest administrator.

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 "Creating Virtual Machines" (page 27). 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:

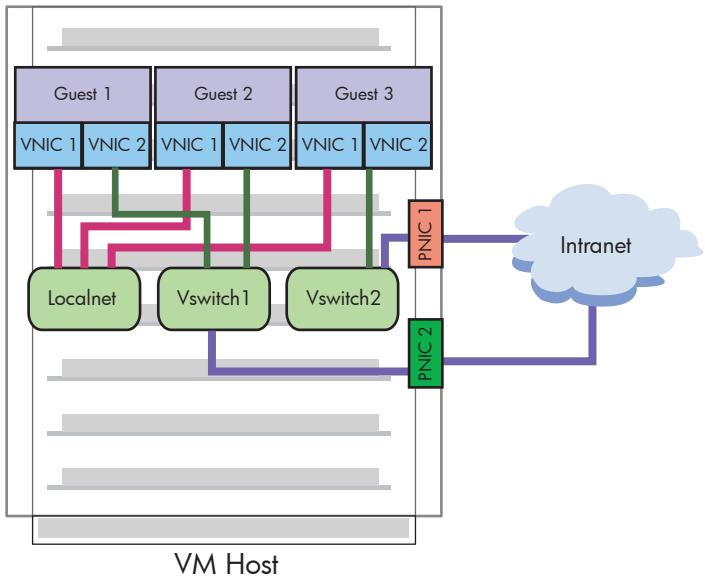
```
-a network:adaptertype:bus,device,mac-addr:vswitch:vswitch-name:portid:portnumber
```

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:

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 7–1.

Figure 7-1 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*.)
- 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.

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 destinations 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 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*(1M) command. If you do not include the `-n` option, a local vswitch is created, as described in "Local Networks" (page 91).

The `hpvmnet` command also allows you to display and manage the vswitches on the VM Host. Table 7-1 describes the options to the `hpvmnet` command.

Table 7-1 Options to the `hpvmnet` Command

Option	Description
`-b`	Starts a vswitch. The vswitch must be started before it can accept network traffic. All vswitches are started automatically when Integrity VM is started.
`-c`	Creates a new vswitch.
`-h`	Halts one or all vswitches. You are asked to confirm this action.
`-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`.
`-s`	Retrieves statistics.
`-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 "Configuring VLANs" (page 94) for more information.
`-V`	Enables verbose mode, displaying information detailed information about one or all vswitches.
`-v`	Displays the version number of the `hpvmnet` command in addition to the vswitch information.

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 -n1
# 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 start a vswitch, enter the `hpvmnet` command with the `-b` option. For example, to start 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 will not be the MAC address of any virtual machine connected to this vswitch).

For information about connecting vswitches to guests, refer to Chapter 3 (page 27). For information about modifying virtual networks, refer to "Configuring Guest Virtual Networks" (page 91).

You can create multiple vswitches associated with the same host physical NIC. However, you cannot start (`hpvmnet –b`) more than one of them at the same time.

7.2.1 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 folowing commands:For example, to create a local network vswitch named `clan0`, enter the following command:

```
# 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.

If you omit the `-n` option when you create a vswitch, the default is to use `localnet`. The `localnet` vswitch can be used as a local network and vNICs can be specified for a guest in the usual way. For example:

```
# hpvmmodify -P compass1 -a network:lan::vswitch:clan0
```

This command adds a vNIC to the guest `compass` which can be used to communicate with any virtual machine connected to the `localnet` vswitch.

7.2.2 Configuring Guest Virtual Networks

You can define a vNIC for a guest using the `hpvmmodify` command. For example, the following command adds a vNIC to the guest named `compass1`.

```
# hpvmmodify -P compass1 -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
.
.
.
```

NOTE: Never modify the guest configuration files directly. Always use the Integrity VM commands to modify virtual devices and virtual machines.

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.
`0x00306E39F70B`	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 `compass1` 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  0x00306E39F70B 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
```

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 Deleting Vswitches

To delete a vswitch, first stop the vswitch using the –h option to the `hpvmnet` command. Then delete the vwitch 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

# hpvmnet -S clan1 -d

hpvmnet: Remove the vswitch 'clan1'? [n] y
```

```
# 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
```

When an active vswitch is deleted, the VM Host automatically determines that the vswitch is gone. When the vswich is re-created, the guest network automatically becomes functional again.

7.4 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 -d
   ```

2. Create a new vswitch associated with `lan1`. For example:

   ```
   # hpvmnet -S myswitch -c -n lan1
   ```

3. Add a new vNIC to your guest using the new vswitch. For example:

   ```
   # hpvmmodify -P guestname -a network:lan:,,:vswitch:myswitch
   ```

7.5 Starting Vswitches

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 starts 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/NOCKO on or off.

7.6 Halting Vswitches

Use the `hpvmnet -h` command to halt the vswitches. 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.

7.7 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:lan:[hardware-address]:vswitch:vswitch-name
  ```

- To create a new virtual machine based on the configuration of an existing virtual machine:

  ```
  # hpvmclone -P vm-name –a network:lan:[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:lan:[hardware-address]:vswitch:vswitch-name
  ```

  The –a option adds the specified vNIC to the virtual machine.

As with virtual devices, you use the –a *rsrc* option to associate a guest virtual network device with a vswitch. Before you can associate the virtual network device with a vswitch, you must create the vswitch using the hpvmnet command. The format of the *rsrc* for network devices is:

network:lan:[*hardware-address*]:vswitch:*vswitch-name*

The guest virtual network device information consists of the following fields, separated by colons:

- network
- 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 and to make sure that the locally administered bit is set and that the multicast and broadcast bits are clear.
- *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)

7.7.1 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:lan:[*hardware-address*]:vswitch:*vswitch-name*

After making this change, start the guest using the hpvmstart command.

7.8 Configuring VLANs

A 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.

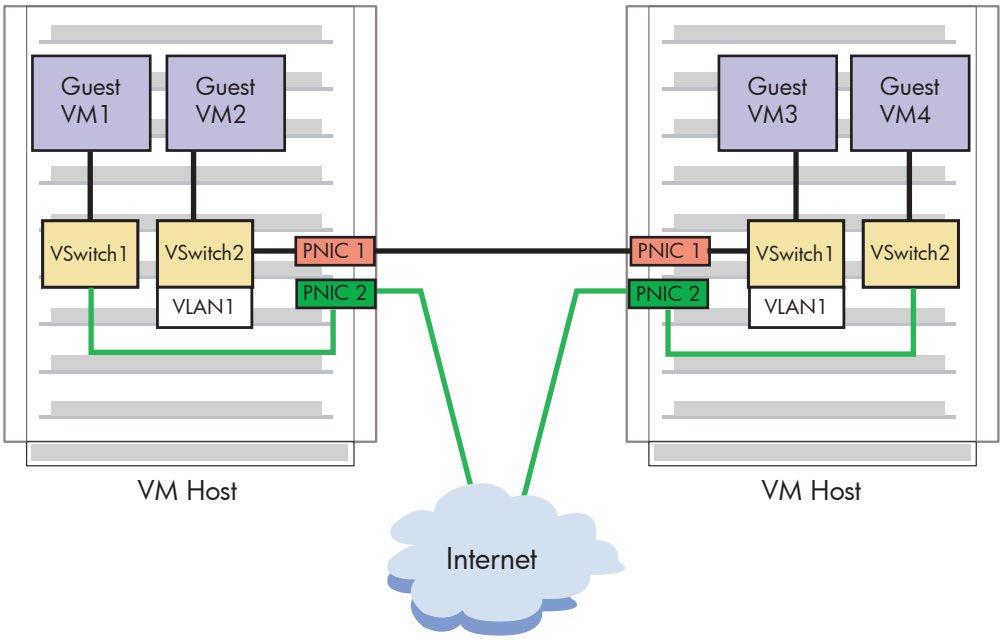
A virtual LAN (VLAN) defines logical connectivity instead of the physical connectivity defined by a LAN. A VLAN provides a way to partition a LAN logically such that the broadcast domain for a VLAN is limited to the nodes and switches that are members of the VLAN.

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

Figure 7-2 illustrates a basic virtual machine VLAN that allows guests on different VM Host systems to communicate.

Figure 7-2 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.

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 A.02.00, only tag-unaware guests are supported.

To configure a VLAN, follow this procedure:

1. Create and start the vswitch. For example, to create and start 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 7-2 describes the possible VLAN port states:

Table 7-2 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

7.8.1 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.

7.8.2 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 16-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     :  100
  Port Number             :  2
    Port State            :  Active
    Active VM             :  vm1
    Untagged VlanId       :  100
    Reserved VMs          :  vm1
```

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     :  100
  Configured Ports        :  4
  Port Number             :  1
    Port State            :  Active
    Active VM             :  vm1
    Untagged VlanId       :  none
    Reserved VMs          :  vm1
   Port Number             :  2
    Port State            :  Active
    Active VM             :  vm1
    Untagged VlanId       :  100
    Reserved VMs          :  vm1
   Port Number             :  3
```

```
 Port State            :  Active
 Active VM             :  vm2
 Untagged VlanId       :  none
 Reserved VMs          :  vm2
Port Number             :  4
 Port State            :  Active
 Active VM             :  vm2
 Untagged VlanId       :  100
 Reserved VMs          :  vm2
```

7.8.3 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*(1M) 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
```

7.9 Troubleshooting Network Problems

This section describes some commonly encountered problems using virtual networks.

- **The hpvmnetd daemon is killed**

  The following error message indicates that the `hpvmnet` daemon has been killed:

  ```
  hpvmnetd: Switch 0000564d4c414e31 already exists
  ```

  The `hpvmnetd` daemon is used by the vswitch driver. It is part of the vswitch internal infrastructure. If the `hpvmnetd` daemon is removed through the `kill` command, the vswitch driver might be in an unstable state.

7.9.1 Redefining PNICs

Changing the hardware address of a vswitch has the same effect as moving a nework 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.

After you restart the vswitch, you must initiate communication from the guest. For example, enter the `ping` command on the guest. It is not necessary to reboot the guest.

7.9.2 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, refer to 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, where 4 is the 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
```

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 macines from the VM Host and from inside the guest. This chapter describes how to manage guests using Integrity VM commands and utilities, including:

8.1 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
winguest1                5 WINDOWS Off           1     5     1    1 GB         0
winguest2                9 WINDOWS On (OS)       1     3     1    2 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 8–1 describes the options to the `hpvmstatus` command.

Table 8-1 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.

Table 8-1 Options to the hpvmstatus Command (*continued*)

Option	Description
–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	Displays information about the multiple-server environment, if Serviceguard is installed.

For example, to see detailed information about the compass1 virtual machine, enter the following command:

```
# hpvmstatus -V -P compass1

[Virtual Machine Details]
Virtual Machine Name     : compass1
Virtual Machine UUID     : 17e4af4c-34fc-11da-94e3-00306e39f70b
Virtual Machine ID       : 15
Virtual Machine Label    :
VM's Model Name          : server Integrity Virtual Machine
VM's Serial Number       : VM00540000
VM's Version Number      : 0.16.0
VM's Version Label      : HPVM V0.16.0 clearcase opt Thu Sep 29 2005 05h12m13s
 T
Operating System         : HPUX
OS Version Number        :
State                    : On
Boot type                : Manual
Console type             : vt100-plus
Guest's hostname         :
Guest's IP address       :
EFI location             : /opt/hpvm/guest-images/common/efi
Pattern File location    : /opt/hpvm/guest-images/common/patterns.vmmpat

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

[Virtual CPU Details]
Number Virtual CPUs      : 1
Minimum Virtual CPUs     : 1
Maximum Virtual CPUs     : 32
Percent Entitlement      : 5.0%
Maximum Entitlement      : 100.0%

[Memory Details]
Total memory             : 1 GB
Minimum memory limit     : 32 MB
Maximum memory limit     : 128 GB
Reserved memory          : 64 MB
```

```
Minimum reserved limit  : 32 MB
Maximum reserved limit  : 128 GB
VHPT Size               : 1 MB

[Storage Interface Details]

[Network Interface Details]
Interface               : vswitch
Guest Adaptor type      : lan
Backing                 : clan1
Bus                     : 0
Device                  : 0
Function                : 0
Mac Address             : 12-40-62-b4-99-61


[Misc Interface Details]
Guest Device type       : serial
Guest Adaptor type      : com1
Interface               : 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 a Windows virtual machine = 2
        Available entitlement for a 1 way virtual machine = 1400 Mhz
        Available entitlement for a 2 way virtual machine = 1260 Mhz
```

8.2 Creating Guest Administrators and Operators

Integrity VM provides secure access to guest consoles. When you create the guest, you can specify the group account or user account that will have guest administration privileges. These users are allowed to log in to the guest under their own user accounts and to use the `hpvmconsole` command to perform system administration tasks on the guest virtual machine.

There types of console users are specified as `admin` and `oper`. Use the `hpvmcreate`, `hpvmmodify`, and `hpvmclone` commands with the `-g` and `-u` options to assign administrator and operator privileges. The user name for the guest administrator account must be the same as the virtual machine name. Therefore, the guest admin account for virtual machine `compass1` must have the user name `compass1`.

You cannot use the `su` command to change from one privilege level to another. Per-user checks are based on login account identifiers, not UUIDs.

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/compass1 -c 'compass1 console' -s /opt/hpvm/bin/hpvmconsole guest1
   ```

   This example uses the following options:
   - -d specifies the home directory for the guest1 account.
   - -c specifies a comment text string that describes the account.
   - -s 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 guest1
   ```

A guest administrator can now access the compass1 virtual console by using the ssh command or telnet command on the VM Host and logging in to the compass1 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 compass1 virtual console on the VM Host myhost:

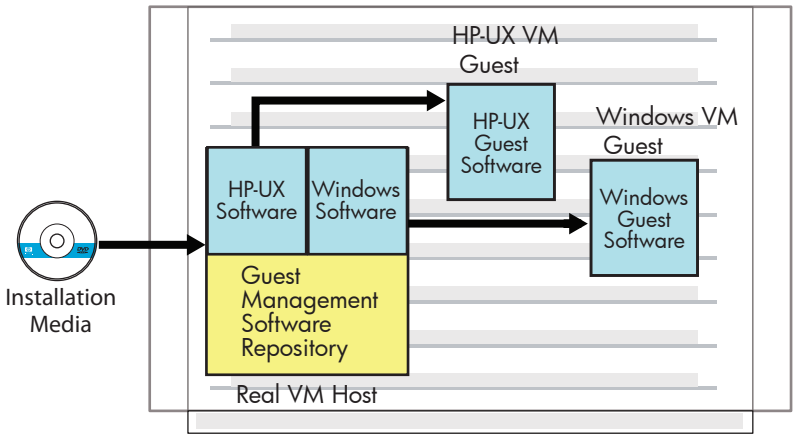
```
# telnet compass1

Trying 16.xx.yy.zz...
Connected to compass1.rose.com.
Escape character is '^]'.

HP-UX compass 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

[compass1] 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. This situation can be a problem, however, for noninteractive or scripted use of the console.

8.3 Creating the Guest Management Software Repository

After you install the guest OS, you must install special Integrity VM guest management software that runs on every virtual machine. Installing the guest management software on each guest provides patches for performance improvements and allows you to use the `hpvmcollect` command on the guest. Figure 8-1 illustrates the procedure.

Figure 8-1 Installing Guest Management Software



When Integrity VM is installed, the guest managment software is loaded into the following directory on the VM Host system: `/opt/hpvm/guest-images/`. A subdirectory is installed for each type of guest operating system, including a README file that contains instructions for applying the guest management software to the guests.

The guest management software is specific to the type of guest operating system. From the VM Host, install the guest management software on each guest as described in the appropriate chapter of this manual.

Refer to the *Integrity VM Release Notes* for information about any additional software updates that you should also install on your guests.

8.4 Using the Virtual Console

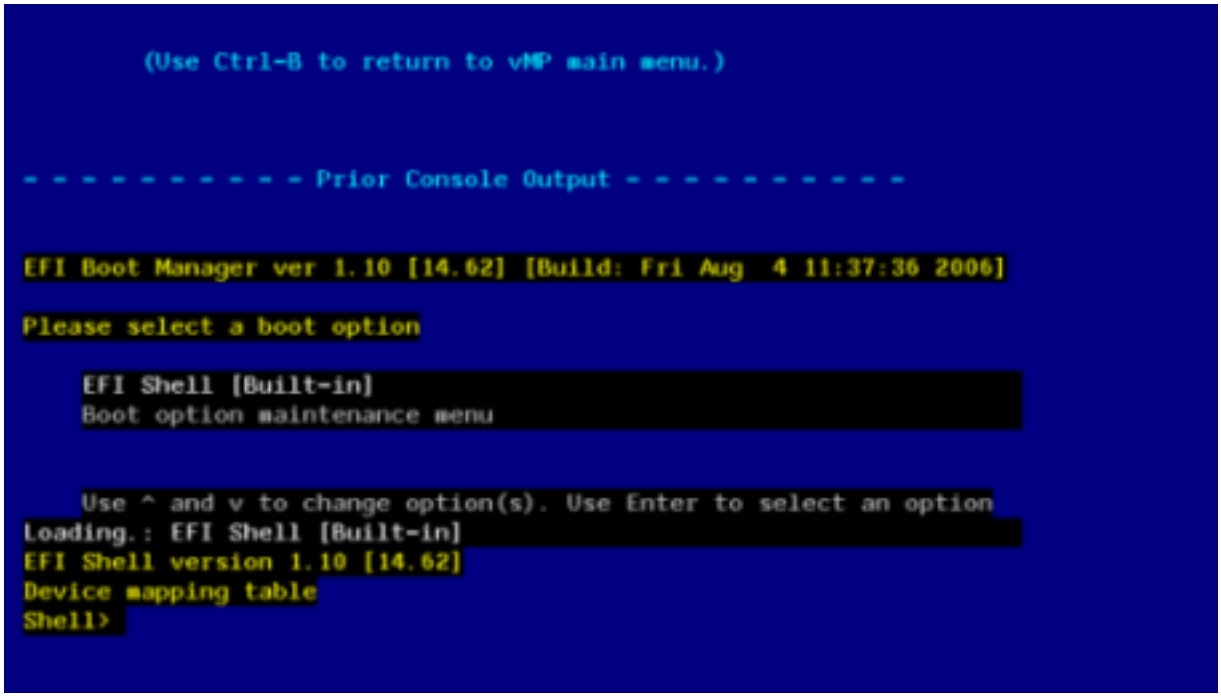
Each virtual machine has its own virtual console, from which the virtual machine can be powered on or off, the guest operating system can be booted or shut down, and so forth. The `hpvmconsole` command connects to the virtual console of a specified virtual machine.

To start the virtual console for the guest named `compass1`, enter the following command:

```
# hpvmconsole -P compass1


   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

[compass1] vMP>
```

When the display is in the EFI, to return to the virtual console, press **Ctr/B**. Use the `co` command to open the virtual console. For example::

```
[compass1] 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 `compass1`, enter the following command:

```
# hpvmconsole -P compass1 -c "pc on"
```

Table 8-2 lists the options to the `hpvmconsole` command.

Table 8-2 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**.
`-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:

```
[compass1] vMP> he

==== vMP Help: Main Menu ================================== (Admin) ============

HPVM A.02.00.02 clearcase recorder-debug Tue Aug 15 2006 09h19m39s EDT
(C) Copyright 2000 - 2006 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
      - 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).

8.5 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.

8.6 Integrity VM Log Files

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

The VM Host log files are stored as `/var/opt/hpvm/common/command.log` and `hpvm_mon_log`.

8.7 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/rdsk/c0t1d2` 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/rdsk/c3t2d0 and /dev/rdsk/c4t2d0`, 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 or disk partitions used by the volume group on which the logical volume is made (for example, `/dev/vg01`) cannot be used as backing stores.

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.

8.7.1 The Device Database File

Integrity VM device management stored Integrity VM device mapping information in the device database file (`/var/opt/common/hpvm_mgmtdb`). This file is divided into three sections:

- The header, which states that the file should not 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 those devices, both storage and network, that guests have been 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.

8.7.2 Using the hpvmdevmgmt Command

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

If a guest is set up to use a virtual disk backed by a logical volume, do not make changes to the logical volume while the guest is running. First, stop the guest by using the `hpvmstop -g` command. If you modify a logical volume that contains a guest's root, you must recreate the guest.

If you extend a logical volume used as a guest virtual device while the guest is on, the guest does not automatically see the size increase. If the logical volume contains the guest's root device, the guest may crash. Remove the guest and recreate it if you modify the disk containing the guest's root device.

The `hpvmdevmgmt` command supports many operations, but not all operations pertain to all entry types. For example, you can replace a device only with a guest device. The `-I` and `-S` options to the `hpvmdevmgmt` command do not work on the device database.

For example, to initialize raw device special files, enter the following command:

```
# hpvmdevmgmt -I
```

This command creates the `/dev/rscsi/*` devices for the devices found in `/dev/rdsk/*`. This command runs when Integrity VM is started. If you dynamically add devices by attaching to SAN or through some other mechanism, use the `hpvmdevmgmt -I` command to create the associated raw devices in `/dev/rscsi`.

To create a large file backing store, enter the following command:

```
# hpvmdevmgmt -S
```

This command option creates file backing store devices from guests. This command allocates the space designated to the file. To create a 12 GB file, enter the following command:

```
# hpvmdevmgmt -S 12G
```

Table 8-3 describes the options to the `hpvmdevmgmt` command.

Table 8-3 Options to the hpvmdevmgmt Command

Option	Description
`-l {server|rdev|gdev}:`*`entry_name`*`:attr:`*`attr_name`*`=`*`attr_value`*	Lists an entry. To list all entries, enter the following command: `# hpvmdevmgmt -l all`
`-v`	Displays the version number of the `hpvmdevmgmt` output format. The version number is followed by the display specified by other options.
`-V`	Increases the amount of information displayed (verbose mode).
`-S` *`size filename`*	Creates a file for use as a virtual device. The size argument must end in either M for megabyte or G for gigabyte.
`-I`	Creates passthrough device files (for example, `/dev/rscsi`). Passthrough devices are used by attached devices, such as tape devices, media changers, and CD/DVD burners.
`-m {server|rdev|gdev}:`*`entry_name`*`[:attr:`*`attr_name`*`=`*`attr_value`*`]`	Modifies an existing attribute or adds the attribute if it does not already exist.
`-a {server|rdev|gdev}:`*`entry_name`*`[:attr:attribute_name=attr_value]`	Adds an entry.
`-d {server|rdev|gdev}:`*`entry_name`*`[:`*`param`*`:`*`arg`*`]`	Deletes an entry.
`-n gdev:`*`oldentry_name`*`:`*`newentry_name0`*`[,`*`newentry_name1`*`]`	Replaces a device.

For example, to use the `hpvmdevmgmt` command to display a list of the restricted devices, enter the following command:

```
# hpvmdevmgmt -l rdev

/dev/rdsk/c10t0d4:CONFIG=rdev,EXIST=YES,DEVTYPE=DISK,SHARE=NO::6005-08b4-0001-15d0-0001-2000-003a-0000
```

8.7.2.1 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. 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 only checked when booting a guest. If one guest is running with a non-shared device and another guest attempts to boot using that same device, it is blocked. If multiple guests require sharing 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, if a device like a physical DVD drive, `/dev/rdsk/c0t0d0`, must be set up as shared to allow multiple guests access for Ignite installs, then you can use the following command to modify that entry after it has been placed into the database by a guest creation or modification:

```
# hpvmdevmgmt -m gdev:/dev/rdsk/c0t0d0:attr:SHARE=YES
```

Only read-only devices can be shared among guests. Virtual DVDs and virtual network devices can be shared. DVDs are not shareable unless you specify otherwise. Sharing virtual devices or the 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 sharable, enter the following command:

```
# hpvmdevmgmt -m gdev:myswitch:attr:SHARE=NO
```

This command restricts the vswitch called `myswitch` to use by one guest only.

8.7.2.2 Replacing Devices

If a backing storage device malfunctions, replace it using the `hpvmdevmgmt -n` option. The `-n` option works only for 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 the guests to use the new pNIC through the old vswitch names without modifying the guests.

8.7.2.3 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, for some reason, 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
```

8.7.2.4 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 “Creating Virtual Storage Devices” (page 61).

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/rdsk/c2t0d0` as a restricted device, enter the following command:

```
# hpvmdevmgmt -a rdev:/dev/rdsk/c2t0d0
```

To delete the restricted device `/dev/rdsk/c2t0d0`, enter the following command:

```
# hpvmdevmgmt -d rdev:/dev/rdsk/c2t0d0
```

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.

9 Migrating Virtual Machines

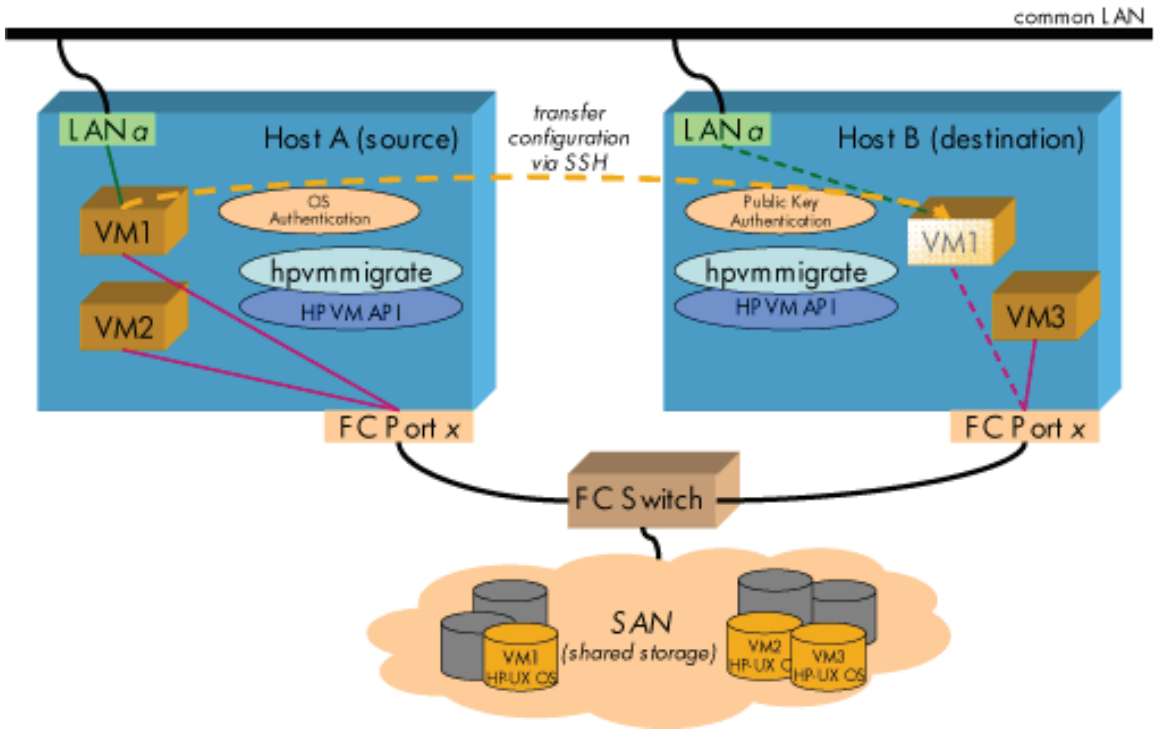
The `hpvmmigrate` command allows you to move a virtual machine from a source VM Host system to a destination VM Host system. The `hpvmmigrate` command is available with HP Integrity Virtual Machines A.01.20 and later. For information about installing the optional VMMigrate bundle, which provides the `hpvmmigrate` command, see "Installing Integrity VM" (page 21).

This chapter includes the following sections:

9.1 Introduction to Virtual Machine Migration

Figure 9-1 illustrates the process of moving a guest from Host A to Host B.

Figure 9-1 Symmetric Hosts Configured for VM Guest Migration



The basic virtual machine migration configuration 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. Similary, 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 the source and destination hosts must be configured symmetrically. That is, all the network and storage resources must be configured the same on both hosts. A symmetric configuration includes:

- A common local area network (LAN)
- Identical network interfaces configurations
- Storage Area Network (SAN) based boot disks
- Identical Fibre Channel port configurations

For guidelines about setting up storage for migrating virtual machines, see "Network and Storage Migration Considerations" (page 113).

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.

9.2 Performing a Guest Migration

To perform a guest migration:

1. Set up SSH keys on both the source and destination hosts, as described in "Security Considerations" (page 113).
2. Stop the guest on the source host, using the `hpvmstop` or `hpvmconsole` command.
3. On the source host, enter the `hpvmmigrate` command, as described in "Using the hpvmmigrate Command" (page 112).
4. Start the guest on the destination host using the `hpvmstart` or `hpvmconsole` command.

For information about starting and stopping guests, see Chapter 8 (page 101).

The `hpvmmigrate` command verifies that the destination host has sufficient resources (such as memory, network switches and storage devices) for the guest to boot. If the resources are insufficient or do not exist, or if other errors occur, the guest is not migrated to the destination host.

After successfully migrating the guest, the `hpvmmigrate` command automatically deletes the guest on the source host.

9.2.1 Using the hpvmmigrate Command

When you enter the `hpvmmigrate` command, you must specify the name of the guest to be migrated and the destination VM Host system.

Specify the guest using one of the following options:

- `-P` *`vm-name`* to specify the guest name
- `-p` *`vm_number`* to specify the virtual machine number

Specify the destination host by including the `–h` option and specifying one of the following:

- The destination host name
- The destination host IP address

Table 9-1 lists the options to the `hpvmmigrate` command.

Table 9-1 Options to the hpvmmigrate Command

Option	Description
`-P` *`source_vm-name`*`-p` *`source_vm_number`*	Specifies the name of the guest to migrate.
`-h` *`dest_hostname`*`-h` *`dest_IP_address`*	Specifies the destination VM Host system.
`–F`	Forces the migration.
`–v`	Displays the `hpvmmigrate` command version.
`–H`	Displays the `hpvmmigrate` command usage information.

You can force the guest to be migrated regardless of whether sufficient resources exist on the destination host by using the `–F` option. When you use this option, any problems found during resource validation are ignored, and the guest is migrated to the destination host.

The guest on the source host is deleted after it is successfully migrated to the destination host.

9.2.2 Example of the hpvmmigrate Command

The following example shows how to migrate the guest named VM1, residing on the host named HostA, to the destination 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 destination host (`-h HostB`)

9.3 Network and Storage Migration Considerations

Effective migration of VM Host systems depends on proper configuration of the networks and storage used by the source and destination hosts. The `hpvmmigrate` command verifies that the source and destination 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 will be much easier and faster.

9.3.1 Network Configuration Considerations

The source and destination hosts should be on the same subnet. The `hpvmmigrate` command preserves the MAC address of the guest being migrated. Thus, having the hosts on the same subnet prevents problems that can occur from changing the guest's host name or IP address. With both hosts on the same subnet, the guest boots properly on the destination host.

In addition, ensure that all pNICs are symmetrically configured on both the source and destination hosts. For example, if `lan0` on HostA is connected to subnet A, and `lan1` is connected to subnet B, make sure that, on HostB, `lan0` is connected to subnet A and `lan1` is connected to subnet B.

9.3.2 Storage Configuration Considerations

Both the source and destination hosts must share access to symmetrically configured storage devices. Specifically, both hosts must use the same character disk-device file name for each disk device. For example, both the source and destination hosts would refer to the same disk device as `/c0t1d0`. (To configure the source and destination systems with the same device file names, use the `ioinit` command to reassign instance numbers to the `ext_bus` class.)

Also, the same storage devices must be visible to both the source and destination hosts. The `hpvmmigrate` command uses the Fibre Channel worldwide identifier (WWID) to determine whether the storage allocated to a guest on the source host is also reachable on the destination host.

The `hpvmmigrate` command assumes that guests use storage area network (SAN) resources specified as whole-disk backing stores (for example, `/dev/rdsk/c26d5t2`). Although you can create virtual machines with direct attached storage (DAS), guests that use DAS cannot be migrated.

SAN logical units (LUNs) are presented to both the source and destination hosts. However, it is not necessary to present LUNs to the guests; they are made available by the VM Host when the virtual machine is booted. This configuration allows you to migrate guests without having to reconfigure the SAN.

To avoid inadvertently using the disk devices associated with a guest on more than one host, mark as restricted all the disk devices used for guest storage on all hosts, except the disk that contains the guest. To mark a disk as restricted, use the `hpvmdevmgmt` command. For example:

```
# hpvmdevmgmt -a rdev:entry_name
```

The `-a` option accepts the name of the device to be restricted. For example:

```
# hpvmdevmgmt -a rdev:/dev/rdsk/c4t1d0
```

For more information about the `hpvmdevmgmt` command, see Chapter 6 (page 61).

After the guest is successfully migrated, the `hpvmmigrate` command marks as restricted all the disk devices allocated to the guest on the source system to prevent any other guests from using them. On the destination host, the disk devices allocated to the migrated guest are marked as unrestricted.

9.3.3 Security Considerations

The `hpvmmigrate` command requires HP-UX Secure Shell (SSH) to be set up on both the source and destination host systems. SSH provides a secure communication path between hosts and is installed on HP-UX 11.23 systems by default. To enable secure communication between the source and destination hosts, you must generate SSH keys on both systems.

The hpvmmigrate command uses SSH public-key based authentication between the source and destination hosts. Password and hostbased authentication are not supported.

You need root privileges to generate and set up the SSH keys required for guest migration.

9.3.3.1 SSH Key Setup

HP recommends that you use the HP-UX Distributed Systems Administration Utilities (DSAU) tools to set up the SSH keys on the source and destination hosts, which is installed by default on HP-UX 11.23 (0512 release). The bundle name is DSAUtilities.

You use the /opt/dsau/bin/csshsetup command to set up SSH keys between hosts. The csshsetup command simplifies the task of setting up SSH public-key authentication trust relationships between hosts. The -r (round-robin) option is used to set up bidirectional authentication. Round-robin key exchange establishes "any-member-to-any-member" authentication. Refer to *csshsetup(1M)* for more information.

Alternatively, SSH keys can be generated manually on the individual systems and then copied to the remote system's $HOME/.ssh/authorized_keys2 file by using the ssh_keygen command. The ssh_keygen command generates, manages, and converts authentication keys for SSH. It also creates RSA keys for use by the SSH protocol.

To use SSH with RSA or DSA authentication, the ssh_keygen command creates the authentication key in one of the following files:

- $HOME/.ssh/identity
- $HOME/.ssh/id_dsa
- $HOME/.ssh/id_rsa

The system administrator may also use the ssh_keygen command to generate host keys, as seen in /etc/rc. See *ssh-keygen(1M)* for more information about SSH key generation.

Table A-2 lists the files that are modified or created for RSA key generation.

Table 9-2 RSA Key Files

File Name	File Contents
$HOME/.ssh2/id_rsa	Default RSA private key for the user
$HOME/.ssh2/id_rsa.pub	Default RSA public key for the user
$HOME/.ssh/authorized_keys	Names of the host RSA public keys that can authenticate to this account

9.3.3.2 SSH Key Setup Troubleshooting

If the SecureShell is installed on both the source and destination systems, you can run the ssh command on the source host, establishing a connection to the destination host. This ensures that SSH keys are set up between the two hosts. The following error message can result from having SSH keys set up improperly:

```
Error: hpvmmigrate: SSH execution error.

Error: hpvmmigrate: Remote execution error on destination-host.
```

10 Using HP Serviceguard with Integrity VM

After you have installed Integrity VM and created the guest, you can install 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, including the following topics:

This chapter 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 with Integrity VM

After you set up Integrity VM, you can install HP Serviceguard A.11.16 or later either on the VM Host or on the HP-UX guest. Do not use Serviceguard on both the VM Host and the guest at the same time.

- To protect guest applications, install Serviceguard on the HP-UX guest. Applications on a guest can fail over to any of the following:
  - Another guest configured as a Serviceguard node that is running on the same VM Host system (see "Cluster in a Box" (page 116))
  - Another guest configured as a Serviceguard node running on a different VM Host system (see "Virtual/Virtual Cluster")
  - Another server or nPartition that is not running Integrity VM (see "Virtual/Physical Cluster" (page 118))

  Windows guests do not support HP Serviceguard; therefore, Windows guest applications cannot be configured as Serviceguard packages.
- To protect guests, install HP Serviceguard on the VM Host system. Guests configured as Serviceguard packages (*distributed guests*) are subsequently managed using HP Serviceguard commands. If the VM Host system fails, the distributed guest automatically fails over to another node in the Integrity VM *multiserver environment*. Integrity VM guests which can relocated between Integrity VM Hosts are configured into an Integrity VM multiserver environment that contains the same set of servers as is in the Serviceguard cluster. (For more information, see "Serviceguard in VM Host Configuration" (page 119)). Guests of any operating system (HP-UX and Windows) can be configured as Serviceguard packages.

Each Serviceguard configuration provides a level of protection against failure. Choose the configuration that best meets your needs, keeping the following requirements in mind:

- Storage Requirements

  To make sure the Serviceguard configuration is managable, use identical backing stores on both the primary node and alternate nodes. To use Serviceguard in Guest configurations, the backing storage units must be whole disks. Integrity VM does not support using other types of backing stores on primary and alternate nodes for applications that are configured as Serviceguard packages.

  The VM Host system storage configurations must comply with both Integrity VM and Serviceguard product requirements. For information about the Integrity VM storage subsystem, see "Creating Virtual Storage Devices" (page 61).
- Network Requirements

  To make sure network communication with guests is always available, provide identical network devices on both the primary and alternate nodes. Physical NICs (pNICs) and vswitches must be the same on both the original and adoptive nodes for virtual NICs (vNICs) to function after the failover. For more information about the Integrity VM networking subsystem, see "Creating Virtual Networks" (page 89).

In the Serviceguard with Integrity VM environment, you can use the following network configurations:

— Heartbeat LAN

  Serviceguard nodes use heartbeat LANs to maintain communication with one another. Whether Serviceguard is installed on the VM Host system or on the guest, HP recommends that you configure every LAN as a heartbeat LAN.

— Primary and standby LANs

  For local LAN failover, a Serviceguard node must have both a primary and standby LAN. In both Serviceguard in Guest and Serviceguard in Host configurations, use vswitches or hubs to connect two pNICs to the same network broadcast domain.

  For Serviceguard in Host configurations, Serviceguard monitors the physical connections and the vswitch monitor moves the vswitch between pNICs automatically.

  In a Serviceguard in Guest configuration, the pNICs are connected to vswitches, which are configured as vNICs in the guest. In this configuration, Serviceguard running in the guest determines the primary and standby LANs and performs the failover in the guest.

— Autoport Aggregation (APA)

  You can use HP-UX APA in the Serviceguard configuration on the VM Host systems. Use APA in MANUAL or AUTO-FEC modes when running on the VM Host system. Do not use LACP_AUTO mode link aggregates. For more information about APA, see the *HP Auto Port Aggregation (APA) Support Guide*.

— Virtual LANs (VLANs)

  VLANs can be configured on the vswitches or the physical switches. You can use VLANs on the VM Host system (as described in the *Using HP-UX VLANS* manual). You can configure the VLAN on vswitches used by guests (as described in "Configuring VLANs" (page 94)).

The following sections describe the configuration procedures and the specific requirements for each of the Serviceguard configurations.

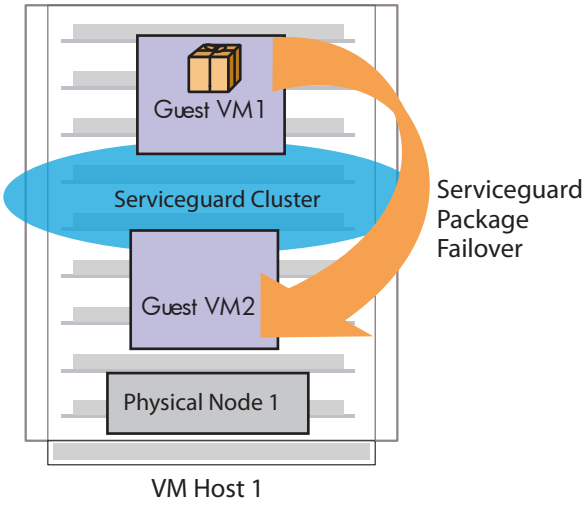
10.2 Serviceguard in Guest Configurations

You can install Serviceguard on an HP-UX guest to provide high availability for the applications running on the guest. In this type of configuration, the guest is configured as a node in a Serviceguard cluster. Depending on the configuration of the cluster, the application package can fail over from one guest to another guest in the same VM Host system, from one guest to another guest in a VM Host system, or from the guest on a VM Host system to a separate physical server or nPar. You can even mix and match Serviceguard in Guest configurations to meet your specific requirements. The following sections describe the Serviceguard in Guest configurations.

10.2.1 Cluster in a Box

Figure 10-1 shows the configuration of an application package that can fail over to another guest on the same VM Host system.

Figure 10-1 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, as follows:

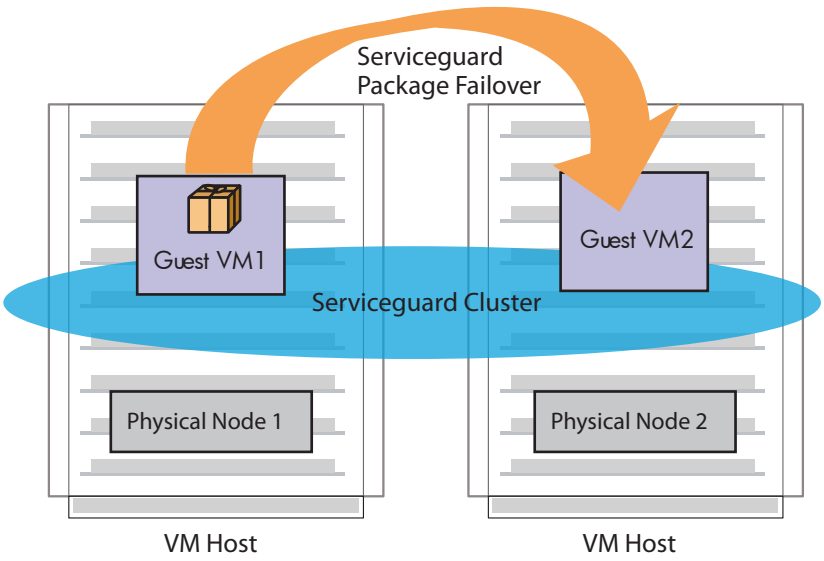
```
# hpvmdevmgmt -m gdev:/dev/rdsk/c6t1d4:attr:SHARE=YES
```

For more information about using the `hpvmdevmgmt` command, see "Managing the Device Database" (page 107).

10.2.2 Virtual/Virtual Cluster

Figure 10-2 shows the configuration of an application package that can fail over to a guest running on a different VM Host system.

Figure 10-2 Guest Application Failover to a Guest on a Different VM Host

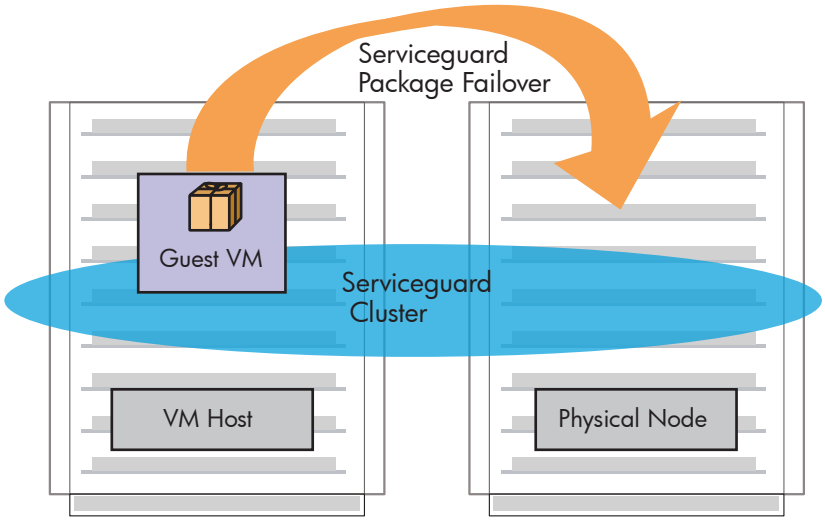


In this configuration, the Serviceguard nodes are guests running on separate nPars or HP Integrity servers.

10.2.3 Virtual/Physical Cluster

Figure 10-3 shows the configuration of an application package that can fail over to a dedicated HP Integrity server or nPartition.

Figure 10-3 Guest Application Failover to an HP Integrity Server



In this case, 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 Configuring Serviceguard in Guests

To configure a Serviceguard cluster that allows an application to fail over from one guest to another, complete the following procedure

1. Install Serviceguard on the HP-UX guests that may run the application.
2. For the virtual/physical cluster, install Serviceguard on the physical node.
3. Ensure that each guest has access to a quorum server or cluster lock disk.
4. Use the `hpvmstatus` command to make sure the guest is running and to verify the guest name.
5. 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.
6. 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.
7. 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.
8. Generate the binary configuration file and distribute it using the following command:

   ```
   # cmapplyconf -k -v -C /etc/cmcluster/gcluster.config
   ```
9. Start the cluster using the following command:

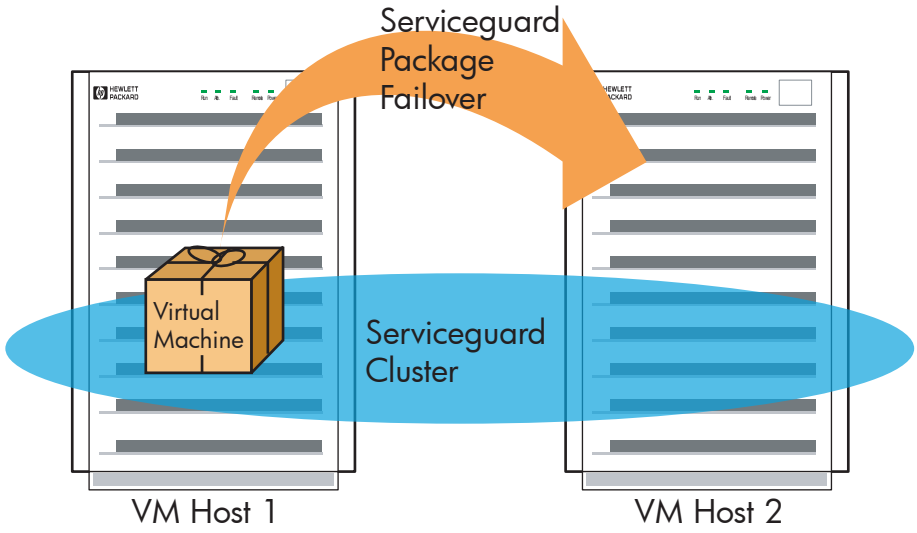
```
# cmruncl
```

This procedure provides a simple example of creating guest application packages. For information about how to set up your Serviceguard configuration, see the *Managing Serviceguard* manual.

10.3 Serviceguard in VM Host Configuration

The following sections describe how to configure a VM Host as a Serviceguard node. In this configuration, if any of the resources used by a guest fail on the primary VM Host system, the guest fails over to an adoptive VM Host system, as illustrated in Figure 10-4.

Figure 10-4 Virtual Machine Failover to Another Cluster Member



To configure Serviceguard in Host:

1. Configure the Integrity VM multiserver environment, as described in Section 10.3.1 (page 119).
2. Create the Serviceguard package, as described in Section 10.3.2 (page 120).
3. Modify the Serviceguard package configuration files to match your guest environment, as described in Section 10.3.3 (page 123).
4. Start the Serviceguard package, as described in Section 10.3.4 (page 123).

10.3.1 Configuring the Integrity VM Multiserver Environment

The Integrity VM multiserver environment provides an integrated environment when guests are configured as Serviceguard packages. In a cluster that is configured as a multiserver environment, each VM Host is aware that the guests are Serviceguard packages and gives control of these distributed guests to ServiceGuard.

For example, two VM Hosts (`host1` and `host2`) make up a Serviceguard cluster. A guest running on `host1` is configured as a Serviceguard package that can fail over to `host2`. After you set up the multiserver environment, you can use the `hpvmstatus` command on each VM Host to display consistent guest package status information. When the guest package is running on `host1`, you can use the `hpvmstatus` command on `host2` to display the guest's current status.

After you configure a guest as a Serviceguard package, you cannot use Integrity VM commands to start and stop the guest. Only Serviceguard commands control the packaged guest. This provides consistent control of the guests and also protects shared whole disk backing stores. Serviceguard ensures exclusive access for shared storage only when the disks are configured with a volume manager like LVM or VxVM. The Integrity VM multiserver environment extends this protection to whole disk backing stores and prevents accessing the same backing store from more than one guest at the same time.

The multiserver environment requires that Serviceguard be running on the VM Host, and allows you to configure guests as Serviceguard packages. The multiserver environment does not apply to configurations where Serviceguard is installed on the guest.

Set up the multiserver environment on each VM Host that is part of the multiserver environment, as follows:

1. Register each VM Host system that will be a member of the multiserver environment. Enter the following commands on each VM Host system:

   ```
   # hpvmdevmgmt -a server:host-name
   # hpvmdevmgmt -m server:host-name:attr:SERVERID=n
   # hpvmdevmgmt -m server:host-name:attr:SERVERADDR=ip-address
   ```

   Where:

   - *host-name* is the unqualified name of the VM Host system.
   - *n* is a unique number identifying this VM Host system in the multiserver environment. Enter a number from 1 through 255.
   - *ip-addr* is the IP address of the VM Host system. Use the IP address of the network connection that serves the multiserver environment.

   Enter these commands on each VM Host system in the multiserver environment, specifying appropriate values. For example, to set up two nodes in the multiserver environment (`host1` and `host2`), enter the following commands on `host1`:

   ```
   # hpvmdevmgmt -a server:host1
   # hpvmdevmgmt -m server:host1:attr:SERVERID=1
   # hpvmdevmgmt -m server:host1:attr:SERVERADDR=1.2.3.4
   # hpvmdevmgmt -a server:host2
   # hpvmdevmgmt -m server:host2:attr:SERVERID=2
   # hpvmdevmgmt -m server:host2:attr:SERVERADDR=1.2.3.5
   ```

   Enter these same commands on `host2`.

2. Verify the registration by entering the following command on each member of the multiserver environment:

   ```
   # hpvmdevmgmt -l server
   host1:CONFIG=SERVER,SERVERADDR=16.116.9.0,SERVERID=1::WWID_NULL
   host2:CONFIG=server,EXIST=NO,DEVTYPE=UNKNOWN,SHARE=NO,SERVERADDR=16.116.8.91,
   SERVERID=2::WWID_NULL
   ```

The list of servers in the multiserver environment must match the list of nodes in the Serviceguard cluster configuration.

10.3.2 Creating Guest Packages

On the VM Host, create a package configuration file and control script for the guest using the following procedure:

1. Install Integrity VM and create the guest with all necessary virtual storage devices and vswitches. Repeat this procedure on each node in the multiserver environment.
2. Install, configure, and run HP Serviceguard on every node in the multiserver environment.
3. Configure the Integrity VM multiserver environment on the primary node and the alternate nodes as described in "Configuring the Integrity VM Multiserver Environment" (page 119).
4. Start the guest on the primary node using the `hpvmstart` command. Use the `hpvmstatus` command to verify the guest name and to make sure that it's running.
5. Create a Serviceguard package by running the `hpvmsg_package` script from the HP Serviceguard for Integrity VM Toolkit, which is installed in the `/opt/cmcluster/toolkit/hpvm/` directory when you install Integrity VM. Specify the guest name as the argument to the command, as follows:

   ```
   # /opt/cmcluster/toolkit/hpvm/hpvmsg_package.sh compass1

   This is the HP Virtual Machine Serviceguard Toolkit Package Template Creation
   script.
   ```

```
This script will assist the user develop and distribute a set of Serviceguard
package configuration template files and associated start, stop and monitor scripts.

The templates generated by these scripts will handle many guest configurations,
but it is only a template 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     #VCPUs #Devs #Nets Memory  Runsysid
==================== ===== ======= ========= ====== ===== ===== ======= ========
compass1                 1 HPUX    Off            1     5     1  512 MB        0
[Storage Interface Details]
Guest                                  Physical
Device Adaptor    Bus Dev Ftn Tgt Lun Storage   Device
====== ========== === === === === === ========= ========================
disk   scsi         0   0   0   0   0 disk      /dev/rdsk/c12t0d0
disk   scsi         0   0   0   1   0 lv        /dev/vgsglvm/rlvol1
disk   scsi         0   0   0   2   0 file      /hpvm/g1lvm/hpvmnet2
disk   scsi         0   0   0   3   0 lv        /dev/vx/rdsk/sgvxvm/sgvxvms
disk   scsi         0   0   0   4   0 file      /hpvm/g1vxvm/hpvmnet2
disk   scsi         0   0   0   5   0 disk      /dev/rdsk/c12t0d5
[Network Interface Details]
Interface Adaptor    Name/Num   Bus Dev Ftn Mac Address
========= ========== ========== === === === =================
vswitch   lan        vswitch2     0   1   0 ea-5c-08-d3-70-f2
vswitch   lan        vswitch5     0   2   0 f2-c7-0d-09-ac-8f
vswitch   lan        vswitch6     0   4   0 92-35-ed-1f-6c-67

Would you like to create a failover package for this Virtual Machine summarized above? (y/n):y

Would you like to distribute the package to each cluster member? (y/n):y

The failover package template files for the Virtual Machine were successfully created.
```

The script asks you to confirm the following actions:

- Creating a failover package
- Distributing the package to all the cluster nodes

Respond to both prompts by entering `y`. The `hpvm_package.sh` script creates the virtual machine package template files in the `etc/cmcluster/`*`guest-name`*`/` directory:

- *`guest-name`*`.config`
- *`guest-name`*`.sh`
- `hpvmsg_ctrl`
- `hpvmsg_mon`
- `hpvmsg_start`
- `hpvmsg_stop`

The `hpvmsg_package` is a utility that you can use to configure a guest as a Serviceguard package. The utility uses the guest name that you supply as an argument to create and populate the `/etc/cmcluster/`*`guest-name`*`/` directory with a set of template files that contain basic Serviceguard parameter settings. HP recommends that you review and modify these template files as needed for your specific multiserver environment. For more information, see "Modifying the Package Configuration Files" (page 123) and the *Managing Serviceguard* manual.

6. Stop the guest using the appropriate operating system command, or use the `hpvmstop -F` command on the VM Host system. (Because the guest has been configured as a Serviceguard package, the –F option is necessary.) For example:

   # **hpvmstop -P *guest-name* -F**

   Alternatively, enter the following command on the guest:

   # **/usr/sbin/shutdown -h now**

   Unmount all file backing stores and deactive any LVM logical volumes or deport VxVM volumes used as backing stores for the guests.

7. Verify that the package is set up correctly by entering the following command:

   # **cmcheckconf -v -C /etc/cmcluster/*cluster-name*.config -P /etc/cmcluster/*guest-name*/*guest-name*.config**

   Where:
   - *cluster-name* is the name of the Serviceguard cluster.
   - *guest-name* is the name of the guest.

8. Update and redistributed the binary configuration files to the `/etc/cmcluster/`*guest-name*`/` directory on all cluster nodes:

   # **cmapplyconf -v -C /etc/cmcluster/*cluster-name*.config -P /etc/cmcluster/*guest-name*/*guest-name*.config**

   When prompted whether to modify the cluster configuration, enter **y**. For example:

   # **cmapplyconf -v -C /etc/cmcluster/cluster1.config -P /etc/cmcluster/compass1/compass1.config**

```
Checking cluster file: /etc/cmcluster/cluster.config
Checking nodes ... Done
Checking existing configuration ... Done
Gathering configuration information ... Done
Gathering configuration information ... Done
Gathering configuration information ..
Gathering storage information ..
Found 10 devices on node host1
Found 10 devices on node host2
Analysis of 20 devices should take approximately 3 seconds
0%----10%----20%----30%----40%----50%----60%----70%----80%----90%----100%
Found 7 volume groups on node charm
Found 7 volume groups on node clowder
Analysis of 14 volume groups should take approximately 1 seconds
0%----10%----20%----30%----40%----50%----60%----70%----80%----90%----100%
.....
Gathering Network Configuration ......... Done
Cluster cluster1 is an existing cluster
Parsing package file: /etc/cmcluster/compass1/compass1.config.
Package hpvmnet2 already exists. It will be modified.
Checking for inconsistencies .. Done
Cluster cluster1 is an existing cluster
Maximum configured packages parameter is 10.
Configuring 3 package(s).
7 package(s) can be added to this cluster.
200 access policies can be added to this cluster.
Modifying configuration on node host1
Modifying configuration on node host2

Modify the cluster configuration ([y]/n)? y
Marking/unmarking volume groups for use in the cluster
```

```
0%----10%----20%----30%----40%----50%----60%----70%----80%----90%----100%
Modifying the cluster configuration for cluster cluster1.
Modifying node host1 in cluster cluster1.
Modifying node host2 in cluster cluster1.
Modifying the package configuration for package compass1.
Completed the cluster creation.
```

If the package configuration file contains the appropriate settings, start the Serviceguard service as described in "Starting the Distributed Guest" (page 123).

10.3.3 Modifying the Package Configuration Files

The Serviceguard for Integrity VM toolkit creates templates that supply basic arguments to Serviceguard parameters. Review and modify the Serviceguard parameters based on the information for your Serviceguard cluster and the information supplied in the *Managing Serviceguard* manual. Make the appropriate changes to the *guest-name*`.config` and the *guest-name*`.sh` files.

Edit the package configuration file to add any LVM volume groups that are used by the distributed guest. Include a separate VOLUME_GROUP parameter for each cluster-aware volume group. These volume groups will be initialized with the cluster ID when the `cmapplyconf` command is used.

10.3.4 Starting the Distributed Guest

To start the distributed guest:, enter the following command:

```
# cmrunpkg -v guest-name
```

For example:

```
# cmrunpkg -v -p compass1
Running package compass1 on node host1.
cmrunpkg  : Successfully started package compass1.
cmrunpkg  : Completed successfully on all packages specified.
```

Verify that the guest is on and running. Use both the Integrity VM `hpvmstatus` command and the Serviceguard `cmviewcl` command to verify the status. For example:

```
# hpvmstatus -P compass1
[Virtual Machines]
Virtual Machine Name VM #  OS Type State     #VCPUs #Devs #Nets Memory  Runsysid
==================== ===== ======= ========= ====== ===== ===== ======= ========
compass1                 1 HPUX    On              1     5     1  512 MB        0
```

```
# cmviewcl -v compass1
CLUSTER      STATUS
cluster1     up

  NODE         STATUS       STATE
  host1        up           running

    Network_Parameters:
    INTERFACE    STATUS       PATH                 NAME
    PRIMARY      up           0/2/1/0/4/1          lan7
    PRIMARY      up           0/2/1/0/6/1          lan9
    PRIMARY      up           0/5/1/0/7/0          lan6
    STANDBY      up           0/1/2/0              lan1
    STANDBY      up           0/2/1/0/4/0          lan2
    STANDBY      up           0/2/1/0/6/0          lan8
    STANDBY      up           LinkAgg0             lan900
    STANDBY      up           0/0/3/0              lan0
```

```
  PACKAGE      STATUS       STATE        AUTO_RUN     NODE
  compass1     up           running      disabled     host1

    Policy_Parameters:
    POLICY_NAME     CONFIGURED_VALUE
    Failover        configured_node
    Failback        manual

    Script_Parameters:
    ITEM       STATUS   MAX_RESTARTS  RESTARTS   NAME
    Service    up                 0         0   host1

    Node_Switching_Parameters:
    NODE_TYPE    STATUS       SWITCHING    NAME
    Primary      up           enabled      host1     (current)
    Alternate    up           enabled      host2

NODE          STATUS       STATE
host   2      up           running

  Network_Parameters:
  INTERFACE    STATUS       PATH                 NAME
  PRIMARY      up           0/2/1/0/4/1          lan7
  STANDBY      up           0/1/2/0              lan1
  STANDBY      up           0/2/1/0/4/0          lan2
  STANDBY      up           0/2/1/0/6/0          lan8
  STANDBY      up           LinkAgg0             lan900
  PRIMARY      up           0/5/1/0/7/0          lan6
  PRIMARY      up           0/2/1/0/6/1          lan9
  STANDBY      up           0/0/3/0              lan0
```

If desired, enter the `cmmodpkg` command to enable autorun and failover.

10.3.5 Starting the Vswitch Monitor

The vswitch monitor is responsible for monitoring the activities of the Serviceguard network monitor and for moving the vswitch configuration, when appropriate, between primary and standby network interfaces. The vswitch monitor requires no user configuration and is installed as part of the Integrity VM product. If Serviceguard is running and any distributed guests are configured, the vswitch monitor is automatically started on the VM Host system when the VM Host system boots. To start the vswitch monitor manually, use the following command:

```
# /sbin/init.d/vswitchmon start
```

To verify that the vswitch monitor is running, enter the following command:

```
# ps -ef | grep vswitchmon
```

10.3.6 Verifying That Distributed Guests Can Fail Over

To verify that the guests configured as Serviceguard packages and the multiserver environment are working properly, use the following commands to perform a manual failover:

1. On the original node (`host1`), verify that the package named `compass1` is running:

   ```
   host1# cmviewcl -v -p compass1
   ```

2. Halt the `compass1` package on `host1`:

```
host1# cmhaltpkg compass1
Halting package compass1.
```

3. Start the package on the other VM Host system (`host2`):

```
host2# cmrunpkg -n host2 compass1
```

4. Enable the package:

```
host2# cmmodpkg -e compass1
```

5. On the adoptive node, verify that the `compass1` package has started:

```
host2# cmviewcl -v -p compass1
```

6. On the adoptive node, verify that the guest named `compass1` is on:

```
host2# hpvmstatus -P compass1
```

10.3.7 Managing Distributed Guests

To start, stop, and monitor distributed guests, use the Serviceguard commands described in this section. Do not use the Integrity VM commands (`hpvmstart`, `hpvmstop`, and `hpvmmigrate`) to manage distributed guests.

10.3.7.1 Starting Distributed Guests

To start a distributed guest, enter the following command:

```
# cmrunpkg guest-name
```

10.3.7.2 Stopping Distributed Guests

To stop a distributed guest, enter the following command:

```
# cmhaltpkg guest-name
```

10.3.7.3 Monitoring Distributed Guests

To monitor the distributed guest, enter the following command:

```
# cmviewcl -v -p guest-name
```

10.3.7.4 Modifying Distributed Guests

You can modify the resources for the distributed guest using the `hpvmmodify` command. However, if you modify the guest on one VM Host server, you must make the same changes on the other nodes in the multiserver environment.

After you modify vswitches, logical volumes, or file backing stores used by distributed guests, make sure that Serviceguard can continue to monitor the guests. Run the `hpvmsg_package` script and restart the guest packages to update the Serviceguard information.

10.3.8 Monitoring Network Connections

The vswitch monitor runs the `vswitchmon.sh` script on the VM Hosts in the multiserver environment and monitors the Serviceguard Network Manager by monitoring the `syslog.log` file. When it detects that Serviceguard is failing over a primary network to a standby network, the vswitch monitor halts, deletes, creates, and boots the vswitch associated with the primary network onto the the standby network. When the primary network is restored, Serviceguard and the vswitch monitor move the network and associated vswitch back to the primary network.

10.4 Upgrading from Integrity VM A.01.20 Toolkit

Extensive changes to the packaging of the Integrity VM A.02.00 ServiceGuard for Integrity VM Toolkit require that customers remove the Integrity VM A.01.20 toolkit before installing the Integrity VM A.02.00 version. After you install Integrity VM A.02.00, repackage the distributed guests, as described in this section.

To upgrade to Integrity VM A.02.00, perform the following sequence of steps on each VM Host node in the multiserver environment:

1. Move the distributed guest to the adoptive node using the `cmhaltpkg` and `cmrunpkg` commands.
2. Remove the old toolkit from the VM Host system as described in "Removing the Serviceguard for Integrity VM Toolkit" (page 126).
3. Install the Integrity VM A.02.00 product as described in Chapter 2 (page 21).
4. Move the distributed guest back to the VM Host system using the `cmhaltpkg` and `cmrunpkg` commands.
5. Log in to the distributed guest and remove the old toolkit as described in "Guest Toolkit Removal" (page 126).
6. Repackage the guest using the procedure described in "Repackaging Guests" (page 127).

10.4.1 Removing the Serviceguard for Integrity VM Toolkit

To remove the Serviceguard for Integrity VM toolkit, perform the following steps on the VM Host system:

1. Remove the toolkit package template files and scripts.

   The old toolkit files should be removed from every VM Host in the multiserver environment. Distributed guests and required resources are not affected by removing the toolkit. Remove the toolkit by removing the toolkit installation directory using the `rm` command, as follows:

   ```
   # rm -rf /var/opt/hpvm/cluster
   ```

2. Remove the vswitch monitor script and files.

   The vswitch monitor script consists of three files: the actual `vswitchmon` script and the symbolic links used to start and stop the script during system boot and shut down. Distributed guests and required resources are not affected when you remove these files. If a reboot occurs during the interval between the removal of `vswitchmon` scripts and the installation of the A.02.00 software, vswitch monitoring is temporarily interrupted. To delete the vswitch monitor, delete the associated files using the `rm` command as show in the following example:

   ```
   # rm /sbin/init.d/vswitchmon /sbin/rc2.d/K004vswitchmon  /sbin/rc3.d/S802vswitchmon
   ```

3. Remove the `sepd` tunable in every distributed guest configuration.

   Removal of the `sepd` tunable is optional for Integrity VM A.02.00. The `sepd` tunable is specific to Integrity VM A.01.20; it is not used in Integrity VM A.02.00. To remove the `sepd` tunable, remove the `sepd` entry in the guest configuration file at `/var/opt/hpvm/`*`guestname`*`/`, where *`guestname`* is the name of the guest that was packaged under Integrity VM A.01.20.

10.4.2 Guest Toolkit Removal

To remove the Integrity VM A.01.20 guest toolkit, remove the `hpvmsgping` script from guests created with the older toolkit. The `hpvmsgping` monitor script consists of the `hpvmsgping` script, a configuration file, and the links used to start and stop the script during system boot and shut down. Removal of these files does not affect the currently running guests or the availability of the system. Delete the associated files using the `rm` command as shown in the following example:

```
# rm /sbin/init.d/hpvmsgping /sbin/rc2.d/K003hpvmsgping /sbin/rc3.d/S801hpvmsgping /etc/hpvmsgping.conf
```

10.4.3 Repackaging Guests

After you upgrade all the nodes in the multiserver environment to Integrity VM A.02.00, repackage the guests. To repackage a guest,:

1. On the original node where the guest is running, run the `hpvmsg_package` command.
2. Apply the package configuration using the `cmapplyconf` command.
3. To provide Serviceguard protection immediately, restart the guest package after repackaging it.

For more information about using these commands, see "Creating Guest Packages" (page 120).

10.5 Troubleshooting Serviceguard with Integrity VM

This section describes how to solve some of the problems that can occur using Serviceguard and Integrity VM.

10.5.1 Serviceguard in Host Troubleshooting

If the distributed guest does not start or failover, check both the `/var/adm/syslog/syslog.log` file and the package log file (`/etc/cmcluster/guest–name/guest–name.sh.log`).

If a package fails to start, ServiceGuard performs a package halt. The log files include a Halting package section after the Starting package section where, the actual starting failure messages are found. Look at the Halting package section as well as the Starting package section when you view package log files after a package start failure.

If the distributed guest does not fail over, take the package down using the `cmhaltpkg` command. Make sure the guest has the resources it needs to run on the adoptive node by manuall starting the package on the adoptive node with the same workload using the `cmrunpkg` command.

If the package does not start under manual control, stop the cluster and test the guest named `compass1`.

1. Use the `hpvmmodify` command to set the guest to be not distributed. For example:

   ```
   # hpvmmodify -P compass1 -i NONE
   # hpvmmodify -P compass1 -j 0
   ```

2. Use the `hpvmstart` command to start the guest with the same VM Host system and workload. Use the virtual console (`hpvmconsole`) to make sure the the guest OS is installed and applications are running properly.

After testing the guest, create the Serviceguard package again.

If the guest does not start and displays errors about storage problems, and you are using logical volumes, the storage units might not be available to the VM Host. To make the storage units available , enter the appropriate commands, as follows:

- For LVM logical volumes, enter the following commands:

  ```
  # vgchange -c n /dev/vgxx
  # vgchange -a y /dev/vgxx
  ```

- For VxVM logical volumes, enter the following commands:

  ```
  # vxdg import diskgroup-name
  # vxvol -g diskgroup-name startall
  ```

- If you are using files on a logical volume, also enter the following command:

  ```
  # mount /dev/vgxx /mount-point
  ```

After making sure the backing storage devices are available, restore them to their original state.

Some problems that arise from improper storage configuration include:

- Whole disks - Verify that the VM Host has access to the disks. This may be traced to a hardware or storage subsystem issue.
- LVM - Before starting a package, ServiceGuard requires that all volume groups associated with the package are inactive. See the *Managing Serviceguard* manual for details on deactivating LVMs.
- VxVm - Before starting a package, ServiceGuard requires that all disk groups associated with the package are deported. See the *Managing ServiceGuard* manual for details.
- Files - Before starting a package, ServiceGuard requires that filesystems of file backing stores associated with the package are unmounted.

If the guest has problems accessing network, make sure the network devices are available on the VM Host system. Packages do not start if any of their defined subnets are unavailable. This causes multiple failures if no standby LANs are available, or when one or more switches, hubs, interfaces or cables fail.

A common issue when starting a package is the lack of available memory. See "Creating Virtual Machines" (page 27) for more information about providing the required memory resources.

10.5.2 Creating Distributed Guests

This manual describes how to use the `hpvmsg_package.sh` script to help you configure guests as Serviceguard packages. If you create the Serviceguard package configuration and control scripts manually instead, use the following options to the `hpvmcreate`, `hpvmmodify`, or `hpvmclone` command to identify the Serviceguard package name and to mark the guest as a distributed guest.

- Use the `–i` option to specify the Serviceguard package. (For example, `–i SG_package_name`.)
- Use the `–j 1` option to specify that the guest is a distributed guest.

For more information, read the `hpvmsg_package.sh` file.

10.5.3 Networking

If the guest has network problems after failover:

- Make sure the vswitches are properly 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` parameters are system- and load-dependent. Specifically:
  - 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 may decide that the other node has become unavailable and initiate cluster reformation. This parameter is entered in microseconds; the default value is 2,000,000 microseconds. 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 due to 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 defects through your support channel. Use the following 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.

Table 11–1 describes the options to the `hpvmcollect` command.

Table 11-1 Options to the hpvmcollect Command

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 hostname 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.
`-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.

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 compass1 on the VM Host, enter the following command:

```
# hpvmcollect -P compass1
```

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 compass1, enter the following command:

```
# hpvmcollect -c -n 21 -d /tmp/hpvm_collect_archive compass1
```

This command collects information about the guest called compass1 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 compass1

  HPVM host crash/log collection tool version 0.8
  Gathering info for post-mortem analysis of guest 'test' 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/sornson/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.

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 `compass1`:

```
compass1# hpvmcollect -c

HPVM guest crash/log collection tool version 0.8
Gathering info for post-mortem analysis on guest (hostname 'compass1')

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/compass1_Sep.29.05_122453PST.tar"
```

NOTE: To use the `hpvmcollect` command on the guest, you must first install the guest management software on the guest as described in "Creating the Guest Management Software Repository" (page 105).

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 Managing the Size of the VMM Driver Log File

The monitor log file (`/var/opt/hpvm/common/hpvm_mon_log`) is 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 grow up 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 kill it:

```
# cat /var/run/hpvmmonlogd.pid
5052
# kill -HUP 5052
```

Integrity VM Manpages

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} -N clone_name[-F | -s] [-l vm_label] [-B
start_attr] [-O os_type[:version]][-c number_vcpus]
[-e percent | -E cycles]
[-r amount ][-S]
[-g -group]...[-g [+]group[:{admin|oper}]]...
[-u -user]...[-u [+]user[:{admin|oper}]]...[-a rsrc]...[-m rsrc]...[-d rsrc]...
```

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. It does not copy the actual data and software associated with the 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 this will cause resource conflicts if both the original and clone virtual machines are booted together.*

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

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 would be running at the same time the new virtual machine would be running, you can use the following command to check a device for dependents:

```
hpvmdevmgmt -l 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 vm_number using the `hpvmstatus` command.

  You must specify either the `-P` or the `-p` option.

`-e 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 will be 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 will be allocated the minimum percentage of 5%. The default is 10%.

  The entitled CPU resources inherited from the existing virtual machine, specified in cycles or percentages, will be replaced in the new virtual machine by this percentage.

  The `-e` and the `-E` options are mutually exclusive.

`-E 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 will be that of the existing virtual machine.

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 value of entitlement inherited from the existing virtual machine (specified in either cycles or percentages) will be replaced in the new virtual machine by the new value in CPU clock cycles.

The `-e` and the `-E` options are mutually exclusive.

`-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 character (_), and the period (.).

The virtual machine name must not start with a dash (-).

`-l vm_label`

Specifies a descriptive text string for the new virtual machine. This 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 character (_), 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 normally occurs when the VM Host is booted, but also occurs if Integrity Virtual Machines is stopped and restarted on a running VM Host. Integrity VM 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.

`-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.

operating-system-type can have the following (case-insensitive) values:

- `HPUX` - Specifies the HP-UX operating system.
- `Windows` - Specifies the Windows operating system.

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 character (_), and the period (.). If white space is desired then `version` must be quoted.

`-a rsrc`

Adds an I/O resource to the new virtual machine. The resource specification (*`rsrc`*) is described in *hpvmresources(1M)*.

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(1M)*.

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, by disk partitions, 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 `hpvmnet` command and backed by physical LAN cards. See the `hpvmnet` manpage for more information on vswitches.

For information about specifying storage and network resources for guests, see *hpvmresources(1M)*.

`-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.

`-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.

`-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 the following letters:

- 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, use the `-g -group` syntax.

This option can be specified more than once.

`-u [+]user[:{admin|oper}]`

Specifies user authorization. Remove user authorization by using the `-u username` syntax. The virtual machine user account specified here can use the `hpvmconsole` command to manage the virtual machine.

Add user authorization using the `-u [+]username[:{admin|oper}]` syntax, where + is optional. When adding a group authorization, the default authorization type is `oper`.

`-s`

Sanity-checks the new virtual machine configuration and returns warnings or errors, but does not create the virtual machine.

RETURN VALUES

The `hpvmclone` command exits with one of the following values:

0: Successful completion.
1: One or more error conditions occurred.

DIAGNOSTICS

hpvmclone 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.
- clone_vm_name already exists.
- vm_name or vm_number does not exist, cannot be accessed, is not a virtual machine, or is corrupt.
- The hpvmclone command and Integrity Virtual Machines 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 compass2, to create a new virtual machine named compass5.

```
# hpvmclone -P compass2 -N compass5
```

Following are sample warning messages returned when hpvmclone is executed with various configuration problems on the guest compass5:

```
HPVM guest compass5 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/rdsk/c2t1d0: Device file '/dev/rdsk/c2t1d0' 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/rdsk/c1t1d3 backing device does not exist.
Warning 8 on item /dev/rdsk/c3t1d0: Device file '/dev/rdsk/c3t1d0' 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 compass5 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         Untagged Number of    Active VM
 Number  state        VLANID   Reserved VMs
 ======= ============ ======== ============ ============
 1       Active       none     2            vm1
 2       Active       100      2            vm1
 3       Active       none     1            vm2
 4       Active       100      1            vm2
```

AUTHORS

The hpvmclone command was developed by the Hewlett-Packard Company.

SEE ALSO

hpvm(5), hpvmcollect(1M), hpvmconsole(1M), hpvmcreate(1M), hpvmdevmgmt(1M), hpvminfo(1M), hpvmmigrate(1M), hpvmmodify(1M), hpvmnet(1M), hpvmremove(1M), hpvmresources (1M), hpvmstart(1M), hpvmstatus(1M), hpvmstop(1M)

hpvmcollect(1M)

NAME

hpvmcollect -- Collects crash dumps, logs, system status, and configuration on the VM Host and guests for post-mortem analysis.

SYNOPSIS

```
hpvmcollect [-cfhl] [-b #] [-d dir] [-n #] [-s host] {-P vm_name | -p vm_number}
```

DESCRIPTION

The `hpvmcollect` command collects log files, system status, device information, system and Integrity Virtual Machines configuration, guest information, and, optionally, crash dumps.

When run on a VM Host, it collects systemwide information as well as information for a specified guest. In this case, you may specify a guest using the virtual machine name or the virtual machine number.

When run in a guest, it collects only the information associated 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

No options can be specified more than once.

`hpvmcollect` recognizes the following command-line options and arguments:

`-b` *bug_report_number*

Overrides the default archive name with `bug_` 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.

`-c`

Includes the latest crash dump directory in the archive. This option is used if the guest or the VM Host crashes or hangs.

`-d` *directory*

Specifies a target directory in which to create the `hpvmcollect_archive` directory.

`-f`

Forces an archive to be overwritten, if it exists, rather than renamed with an appended timestamp.

`-n` *crash dump number*

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 only be used with the `-c` option.

`-l`

Leaves the collected information in a directory rather than an archive file. The directory name follows the same naming convention as the archive name.

`-s` *hostname*

Specifies a hostname to receive the archive, which is copied using `scp`. Verify that you can login to the host without a password.

`-h`

Prints out the help message.

`-P` *vm_name*

Specifies the unique name of the virtual machine to be archived.

The `-P` and `-p` options are mutually exclusive.

`-p` *vm_number*

Specifies the unique number of the virtual machine to be archived. The `vm_number` is displayed by the `hpvmstatus` command.

The `-P` and `-p` options are mutually exclusive.

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 0.8
  Gathering info for post-mortem analysis of guest 'myguest' 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 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 VMM image ..................................................... OK
  Copying hpvmdvr image ................................................. OK
  Copying hpvmntdvr image ............................................... OK
  Copying NVRAM image ................................................... OK
  Copying guest VM crash image .......................................... N/A
  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 log ..................................... N/A

  Finished with the collection

  Tar archiving and compressing ................................... TGZ
  Remote copying the archive  ..................................... NO

  The collection is
  "//hpvmcollect_archive/myguest_Sep.07.06_134528EDT.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
```

AUTHORS

The hpvmcollect command was developed by the Hewlett-Packard Company.

SEE ALSO

hpvm(5), hpvmclone(1M),hpvmconsole(1M), hpvmcreate(1M), hpvmdevmgmt(1M), hpvminfo(1M), hpvmmigrate(1M), hpvmmodify(1M), hpvmnet(1M), hpvmremove(1M), hpvmresources(1M), hpvmstart(1M), hpvmstatus(1M), hpvmstop(1M)

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 the virtual machine can be powered on or off, the guest operating system can be booted or shut down, and so forth. The `hpvmconsole` command connects to the virtual console of a specified virtual machine.

If you have logged into the physical console of an VM Host and then run `hpvmconsole` interactively:

To return to the physical console, use **control-B**.

To return to the virtual console main menu, use **Ctrl/X**.

Options

`hpvmconsole` recognizes the following standard Integrity VM options and arguments:

`-P` *vm_name*

Specifies the name of the virtual machine to be booted.

You may 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 may 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 the **Ctrl/-B** character 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 was encountered.

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 (**Ctlr/B**) and use **Ctrl/t** instead:

```
hpvmconsole -e ^t -P guestname
```

AUTHORS

The hpvmconsole command was developed by the Hewlett-Packard Company.

SEE ALSO

hpvm(5), hpvmclone(1M), hpvmcollect(1M), hpvmcreate(1M), hpvmdevmgmt(1M), hpvminfo(1M), hpvmmigrate(1M), hpvmmodify(1M), hpvmnet(1M), hpvmremove(1M), hpvmresources(1M), hpvmstart(1M), hpvmstatus(1M), hpvmstop(1M)

hpvmcreate(1M)

NAME

hpvmcreate -- Create a new Integrity Virtual Machines virtual machine.

SYNOPSIS

```
hpvmcreate -P vm_name [-F | -s] [-l vm_label] [-B start_attr] [-O
os_type[:version]][-c number_vcpus]
[-e percent | -E 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]]
```

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 will necessarily be running at the same time or on the current VM Host. Therefore, the `hpvmcreate` command will allow the creation of virtual machines that cannot boot on the current system. A guest configuration will receive a warning at creation, and an error at start time, for any issues that would 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 character (_), and the period (.). The virtual machine name cannot start with a dash (-). The `-P` option is required.

`-e 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 will be allocated the minimum percentage of 5%. The default entitlement is 10%.

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`

Specifies the virtual machine's CPU entitlement in CPU cycles.

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 `-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.

`-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`, or both. 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 character (_), and the period (.). If white space is desired, the label must be quoted ("").

`-B start_attr`

Specifies the startup behavior of the virtual machine. Starting a virtual machine is equivalent to powering on a physical system. To cause the guest operating system to boot automatically, the guest must have autoboot set, and a default boot device must be specified at its virtual console.

`start_attr` can have the following (case-insensitive) values:

- `auto`: Automatically start the guest when Integrity VM is initialized on the VM Host.
- `manual`: Manually start the guest using the `hpvmstart` command or the `hpvmconsole` command.

If the `start_attr` attribute is set to `auto`, the virtual machine is started when Integrity VM is initialized. This normally occurs when the VM Host is booted, but also occurs if Integrity VM is stopped and restarted on a running VM Host. Integrity VM 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.

`-a rsrc`

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, disk partitions, 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, by disk partitions, or by whole disks.
- Virtual DVDs, which can be backed by filess in a VM Host file system or by physical DVD drives.
- 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(1M)*.

`-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.

operating-system-type can have the following (case-insensitive) values:

- `HPUX` - Specifies the HP-UX operating system.
- `Windows` - Specifies the Windows operating system.

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 character (_), and the period (.). If white space is desired then `version` must be quoted.

`-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 four.

`-r amount`

Specifies the amount of memory available to this virtual machine.

The size is expressed as an integer, optionally followed by one of the following letters:

- 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[:kind]`

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 `kind` 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[:kind]`

Specifies the user authorization. A VM Host user account specified here can use the `hpvmconsole` command to manage this guest. The `kind` 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`

Sanity-checks 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.

The `-F` and `-s` options are mutually exclusive.

RETURN VALUES

The `hpvmcreate` command exits with one of the following values:

0: Successful completion.

1: One or more error conditions occurred.

DIAGNOSTICS

`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 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` has been 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

Create a virtual machine named myguest1, specifying four virtual CPUs, and two GB of memory, and /dev/disk/c1t2d0 as a SCSI disk device:

```
# hpvmcreate -P myguest1 -c 4 -r 2G -a disk:scsi::disk:/dev/rdsk/c1t2d0
```

Create a virtual machine named myguest2, specifying two 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/rdsk/c2t2d0 \
-a network:lan::vswitch:switch1
```

Create a virtual machine named cougar with two 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 a virtual network interface backed by virtual switch hostnet:

```
# hpvmcreate -P cougar -c 2 -r 2G \
-a disk:scsi::disk:/dev/rdsk/c3t1d0 \
-a disk:scsi::disk:/dev/rdsk/c2t1d0s1 \
-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:lan::vswitch:hostnet
```

Following are sample warning messages 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/rdsk/c2t1d0: Device file '/dev/rdsk/c2t1d0' 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/rdsk/c1t1d3 backing device does not exist.
Warning 8 on item /dev/rdsk/c3t1d0: Device file '/dev/rdsk/c3t1d0' 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.
```

AUTHORS

The hpvmcreate command was developed by the Hewlett-Packard Company.

SEE ALSO

hpvm(5), hpvmclone(1M), hpvmcollect(1M), hpvmconsole(1M), hpvmdevmgmt(1M), hpvminfo(1M), hpvmmigrate(1M), hpvmmodify(1M), hpvmnet(1M), hpvmremove(1M), hpvmresources(1M), hpvmstart(1M), hpvmstatus(1M), hpvmstop(1M)

hpvmdevmgmt(1M)

NAME

hpvmdevmgmt – Manage the devices that are associated with the VM Host and the guests.

SYNOPSIS

```
hpvmdevmgmt [-V] -a {server|rdev|gdev}:entry_name
hpvmdevmgmt [-V] -d {server|rdev|gdev}:entry_name
hpvmdevmgmt [-V] -d gdev:{all|entry_name}:depend:dependent_name
hpvmdevmgmt [-V] -d {server|rdev|gdev}:entry_name:attr:attr_name
hpvmdevmgmt [-V] -m {server|rdev|gdev}:entry_name:attr:attr_name=attr_value
hpvmdevmgmt [-V] -n gdev:oldentry_name:newentry_name0[,newentry_name1]
hpvmdevmgmt [-V] -l {all|server|rdev|gdev}[:entry_name]
hpvmdevmgmt [-V] -l {all|server|rdev|gdev}:depend:dependent_name
hpvmdevmgmt [-V] -l {all|server|rdev|gdev}:attr_name=attr_value
hpvmdevmgmt [-V] -I
hpvmdevmgmt -v
hpvmdevmgmt [-V] -S file_size file_name
```

DESCRIPTION

Lists an entry 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, alter specific device attributes, specify shared devices, and perform specialized functions associated with device management. You can use the `hpvmdevmgmt` command to create database entries for restricted devices (to which guest access is prohibited). such as creating raw device files, and for pre-extending files used as virtual devices.

The device management database contains three types of entries:

- Restricted devices (`rdev`)
- Guest devices (`gdev`)
- 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 or other device entries depending upon this entry (called its dependents), and a unique identifier.

Only superusers can execute the `hpvmdevmgmt` command.

If you have the Serviceguard environment set up, you can use this command to add the VM Host to the Serviceguard cluster environment. To establish a multiserver entry, enter the following commands:

```
# hpvmdevmgmt -a server:srvr_hostname
# hpvmdevmgmt -m server:srvr_hostname:attr:SERVERADDR=ip-address
# hpvmdevmgmt -m server:srvr_hostname:attr:SERVERID=server-id
```

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. To delete a multiserver entry, enter the following command:

```
# hpvmdevmgmt -d server:srvr_hostname
```

Options

No options can be specified more than once.

`hpvmdevmgmt` recognizes the following command-line options and arguments:

`-l {server|rdev|gdev}:entry_name:attr:attr_name=attr_value`
Lists an entry. The option can perform the following actions:

- List all entries. To list all entries, use the following command format: `hpvmdevmgmt -l all`.
- List all with the specified attribute or dependency. To list all the devices with a specific attribute, use the following command format: `hpvmdevmgmt [-V] -l {all|server|rdev|gdev}:attr_name=attr_value`. To list all the devices with a specific dependency, use the following command format: `hpvmdevmgmt [-V] -l {all|server|rdev|gdev}:depend:dependent_name`.
- List a single entry by name. To list a specific entry by name, use the following command format: `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 pathname of the file to be created. An error is returned on an attempt to overwrite an existing file.

`-I`

Creates passthrough device files (for example, `/dev/rscsi/c0t0d0`). Passthrough device files are used for attached devices (tape devices, media changers, and CD/DVD burners).

`-m {server|rdev|gdev}:entry_name[:attr:attr_name=attr_value]`

Modifies an existing attribute or adds the attribute if it does not already exist.

`-a {server|rdev|gdev}:entry_name[:attr:attribute_name=attr_value]`

Adds an entry. The 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}:entry_name[:param:arg]`

Deletes an entry. The deletion option can process the following deletions:

1. Deletion of an entry. An entry cannot be deleted if it has dependents. To delete a specific entry, use the following command format: `-d {server|rdev|gdev}:entry_name`
2. Deletion of a dependent from one or all entries of a certain type. To delete all entries with a specific dependent, use the following command format: `d gdev:{all|entry_name}:depend:dependent_name`
3. Deletion of an attribute from an entry. To delete all entries with a specific attribute, use the following command format:`-d {server|rdev|gdev}:entry_name:attr:attr_name`

To delete a cluster entry, specify the server host name for the *`entry_name`*.

`-n gdev:oldentry_name:newentry_name0[,newentry_name1]`

Replaces a device. Typically used when a device goes bad.

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 Virtual Machines are at different revision levels.

EXAMPLES

List a guest-device entry:

```
hpvmdevmgmt -l gdev:/dev/rdsk/c2t1d0s2
```

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, that is, 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 all necessary raw device files:

```
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/rdsk/c2t1d0s4
```

AUTHORS

The hpvmdevmgmt command was developed by the Hewlett-Packard Company.

SEE ALSO

hpvm(5), hpvmclone(1M), hpvmcollect(1M), hpvmconsole(1M), hpvmcreate(1M), hpvminfo(1M), hpvmmigrate(1M), hpvmmodify(1M), hpvmnet(1M), hpvmremove(1M), hpvmresources(1M), hpvmstart(1M), hpvmstatus(1M), hpvmstop(1M)

hpvminfo(1M)

NAME

hpvminfo -- Display information about the Integrity VM environment.

SYNOPSIS

```
hpvminfo [-V | -M | -X] [-v]
```

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

No options can be specified more than once.

`hpvminfo` 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.

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 run on the VM Host.

```
# hpvminfo
```

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

The following example demonstrates the command run inside a guest.

```
# hpvminfo

hpvminfo: Running inside an HPVM guest.
```

The following example shows the 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 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 the Hewlett-Packard Company.

SEE ALSO

hpvm(5), hpvmclone(1M), hpvmcollect(1M), hpvmconsole(1M), hpvmcreate(1M), hpvmdevmgmt(1M), hpvmmigrate(1M), hpvmmodify(1M), hpvmnet(1M), hpvmremove(1M), hpvmresources(1M), hpvmstart(1M), hpvmstatus(1M), hpvmstop(1M)

hpvmmigrate(1M)

NAME

hpvmmigrate -- Migrate a virtual machine to a different VM host.

SYNOPSIS

```
hpvmmigrate {-P source_vm_name | -p source_vm_number} -h {dest_hostname |
dest_IP_addr} [-F] [-d] [-B]
hpvmmigrate [-v]
hpvmmigrate [-H]
```

DESCRIPTION

The `hpvmmigrate` command moves an existing virtual machine to the destination VM Host.

In order to move a virtual machine from a source VM Host to a destination VM Host, both VM Hosts must be configured to allow common access to all of the required resources of the migrating virtual machine. In addition, the migration of a virtual machine is controlled by a set of secure remote operations which must be enabled on both systems. After the VM Host makes sure that the virtual machine has the resources to start on the destination host, it is stopped on the source VM Host.

If the virtual machine is not set up as a distributed guest (that is, as a Serviceguard package), it is deleted on the VM Host system after it is successfully started on the destination VM Host.

The resources that are defined in the virtual machine's configuration file are checked to determine whether the migrated virtual machine could boot on the destination VM Host. If there is a problem, it is reported and the virtual machine is not migrated. You can specify the `-F` (force) option to suppress the errors and force the virtual machine migration to the destination VM Host. The `-F` option should be used with caution; some errors can prevent a virtual machine from booting on the destination VM Host.

Only superusers can execute the `hpvmmigrate` command.

Options

No options can be specified more than once.

`hpvmmigrate` recognizes the following command-line options and arguments:

`-P source_vm_name`
: Specifies the unique name of the virtual machine to be migrated.

`-p source_vm_number`
: Specifies the unique number of the virtual machine to be migrated. The vm_number is reported via the `hpvmstatus` command.

  You must specify either the `-P` option or the `-p` option.

`-h {dest_hostname|dest_IP_addr]`
: Specifies the host name or IP address of the destination machine to which the virtual machine is being migrated. The destination machine must be a valid VM Host and must be accessible by the source VM Host.

`-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). This option ignores all resource validation errors, including oversubscribing of resources. It is important to note that these errors may prevent the virtual machine from booting on the destination VM Host. Any validation errors will be logged in the Integrity VM command log.

`-v`
: Displays the version number of the `hpvmmigrate` command.

`-H`
: Displays the usage of the `hpvmmigrate` command.

`-d`
: Causes `hpvmmigrate` to automatically shut down the target guest before the migration process, after the resource test in the target host.

`-B`

Causes `hpvmmigrate` to boot the target guest automatically after the migration process is complete.

RETURN VALUES

The `hpvmmigrate` command exits with one of the following values:

0: Successful completion.

1: One or more error conditions occurred.

DIAGNOSTICS

`hpvmmigrate` 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.
- *`source_vm_name`* or *`source_vm_number`* does not exist, cannot be accessed, is not a virtual machine, or is corrupt.
- The `hpvmmigrate` command and Integrity Virtual Machines are at different revision levels.
- Guest already exists on the destination VM Host.
- Guest is running.
- Invalid guest configuration.
- Remote execution error.
- Guest resource validation error.
- The version of the `hpvmmigrate` command is incompatible with the version on the destination VM Host.

EXAMPLES

Migrate the virtual machine named compass1, to the host pman.hp.com.

```
# hpvmmigrate -P compass1 -h pman.hp.com
```

Migrate the virtual machine named compass1 to the VM Host pman.hp.com, ignoring resource validation errors.

```
# hpvmmigrate -P compass1 -h pman.hp.com -F
```

Display the version number of the `hpvmmigrate` command.

```
# hpvmmigrate -v
hpvmmigrate: Version A.02.00.00
```

AUTHORS

hpvmmigrate was developed by the Hewlett-Packard Company.

SEE ALSO

hpvm(5), hpvmclone(1M), hpvmcollect(1M), hpvmconsole(1M), hpvmcreate(1M), hpvmdevmgmt(1M), hpvminfo(1M), hpvmmodify(1M), hpvmnet(1M), hpvmremove(1M), hpvmresources(1M), hpvmstart, hpvmstatus(1M), hpvmstop(1M)

hpvmmodify(1M)

NAME

hpvmmodify - Rename a virtual machine or modify the attributes of a virtual machine.

SYNOPSIS

```
hpvmmodify {-P vm_name | -p vm_number} [-F | -s] [-N new_vm_name][-l vm_label]
[-B start_attr] [-O os_type[:version]]
[-c number_vcpus][-e percent | -E cycles][-r amount]
[-g -group]...[-g -group [+]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]]...
hpvmmodify -A {-P vm_name | -p vm_number} [-F] [-l vm_label] [-B start_attr]
[-e percent | -E cycles][- -a rsrc]...[-m rsrc]...[-d rsrc]
```

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.

Virtual machine modification is designed for flexibility, and assumes that all existing virtual machines will not necessarily be running at the same time or on the current VM Host. Therefore, the `hpvmmodify` command will allow virtual machines to be modified in such a way that they cannot boot on the current system. A guest configuration will receive a warning at modification, and an error at start time, for any issues that would 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.

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.

`-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.

`-e 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, the virtual machine will be allocated the minimum percentage of 5%. The default is 10%.

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.

`-E cycles`

Specifies the virtual machine's CPU entitlement in CPU cycles.

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.

`-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.

`-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 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 character (_), and the period (.). If white space is desired, the label must be quoted ("").

`-B start_attr`

Specifies the startup behavior of the virtual machine. Starting a virtual machine is equivalent to powering on a physical system. For the virtual machine to boot automatically, it must also have autoboot set and a default boot device specified at its virtual console. `start_attr` can have one of the following (case-insensitive) values:

- `auto`: Automatically start the VM when Integrity Virtual Machines is initialized on the host.
- `manual`: Require manual start of the VM.

If the `start_attr` attribute is set to `auto`, the virtual machine is started when Integrity VM is initialized.

This normally occurs when the VM Host is booted, but also occurs if Integrity VM is stopped and restarted on a running VM Host. Integrity VM 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 automatically be started 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.

`-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.

operating-system-type can have the following (case-insensitive) values:

- `HPUX` - Specifies the HP-UX operating system.
- `Windows` - Specifies the Windows operating system.

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 character (_), and the period (.). If white space is desired then `version` must be quoted.

`-a rsrc`

Adds an I/O resource to a virtual machine. The resource is specified as described in the `--m` option.

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 the `-m` option. 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 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 will replace the one in use.

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, disk partitions, 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, by disk partitions, 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. See the `hpvmnet` manpage for more information about virtual network devices.

For information about specifying storage and network resources for guests, see *hpvmresources(1M)*.

`-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 character (_), and period (.). The virtual machine name must not start with a dash (-).

The virtual machine name can only be changed by using the `-N` option.

The name change takes effect immediately.

`-c number_vcpus`

Specifies the number of virtual CPUs this virtual machine sees at boot time. If unspecified, the number defaults to one.

`-r amount`

Specifies the amount of memory available to this virtual machine.

The sizes are expressed as integers, optionally followed by one of the following letters:

- M -megabytes
- G -gigabytes

If the letter is omitted, the unit defaults to 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. `{admin|oper}` 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. {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`

Sanity-checks the virtual machine configuration and returns warnings or errors, but suppresses the action that the command would normally perform. This option is used to invoke resource checking for the specified virtual machine configuration without actually modifying the virtual machine. In the normal case, where -s not specified, the virtual machine is modified even if resource warnings occur.

The -F and -s options are mutually exclusive.

RETURN VALUES

The hpvmmodify command exits with one of the following values:

0: Successful completion.

1: One or more error conditions occurred.

DIAGNOSTICS

hpvmmodify 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.
- 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 Virtual Machines are at different revision levels.

Using colon (:), semicolon (;), or comma (,) when entering device names will cause 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/rdsk/c2t2d1:

```
# hpvmmodify -P myguest2x -m disk:scsi:0,0,4:disk:/dev/rdsk/c2t2d1
```

Change the network device at hardware address 0,2 to a different vswitch, myswitch, preserving its original virtual MAC address

```
# hpvmmodify -P myguest2 -m network: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:lan:0,1
```

Change the CPU entitlement to 50%:

```
# hpvmmodify -P myguest2 -e 50
```

Temporarily change the CPU entitlement to 50% until virtual machine myguest2 is rebooted:

```
# hpvmmodify -A -P myguest2 -e 50
```

Following are sample warning messages returned when `hpvmmodify` 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/rdsk/c2t1d0: Device file '/dev/rdsk/c2t1d0' 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/rdsk/c1t1d3 backing device does not exist.
Warning 8 on item /dev/rdsk/c3t1d0: Device file '/dev/rdsk/c3t1d0' 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.
```

AUTHORS

The hpvmmodify command was developed by the Hewlett-Packard Company.

SEE ALSO

hpvm(5), hpvmclone(1M), hpvmcollect(1M), hpvmconsole(1M), hpvmcreate(1M), hpvmdevmgmt(1M), hpvminfo(1M), hpvmmigrate(1M), hpvmnet(1M), hpvmremove(1M), hpvmresources(1M), hpvmstart(1M), hpvmstatus(1M), hpvmstop(1M)

hpvmnet(1M)

NAME

hpvmnet -- Create and control an Integrity Virtual Machines virtual network switch (vswitch).

SYNOPSIS

```
hpvmnet [-S vswitch_name | -s vswitch_number] [-V | -M | -X] [-v]
hpvmnet -c -S vswitch_name [-n nic_id]
hpvmnet -d {-S vswitch_name | -s vswitch_number} [-F]
hpvmnet -b [-S vswitch_name | -s vswitch_number]
hpvmnet -h [-S vswitch_name | -s vswitch_number] [-F]
hpvmnet -r [-S vswitch_name | -s vswitch_number] [-F]
hpvmnet [-S vswitch_name | -s vswitch_number] -u
portid:portid[,...]:vlanid:[vlan-id | none]
hpvmnet [-S vswitch_name | -s vswitch_number] [-p all | portid] [-M | -X] [-v]
```

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. The `hpvmnet` command is used 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 may optionally 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 will experience intermittent network failures. Stopping and restarting a vswitch can occur while its guests are running; no guest shutdown is required.

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).

There is no 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.

Options

No options can be specified more than 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.

`-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.

`-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.

`-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).

`-c`

Creates a new vswitch.

`-d`

Deletes an existing vswitch.

`-b`

Starts an existing vswitch. Vswitches must be started before they will accept guest traffic. Note that Integrity VM attempts to automatically start all existing vswitches when Integrity VM itself is started.

`-h`

Stops a vswitch.

`-r`

Stops and restarts a vswitch.

`-n nic_id`

Designates the network interface on the VM Host that this vswitch will use. Network interfaces are displayed by the `lanscan` command. If `-n` is not specified when creating a vswitch, a local vswitch will be created.

`-u portid:portnumber:vlanid:{vlan-id|none}`

Specifies the VLAN identifier for the specific vswitch and port. Configure VLANs by specifying the number of the port on the vswitch (*`portnumber`*) to use for VLAN communication, and the VLAN identifier (*`vlan-id`*). 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.

`-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. Do not use the `-p` option with the `-V` 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.
- `vswitch_name` or `vswitch_number` 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      PPA    MAC Address    IP Address
======== ====== ======= ========= ====== ============== ===============
localnet      1 Up      Shared           N/A            N/A
myswitch      2 Up      Shared    lan0   0x00306e4aa30d 10.116.9.246
sw_one        3 Up      Shared    lan1   0x00306e4a929c
sw_two        4 Down    Shared    lan2
sw_three      5 Down    Shared    lan3
```

Display the attributes of vswitch *myswitch*:

```
hpvmnet -S myswitch

Name     Number State   Mode      PPA    MAC Address    IP Address
======== ====== ======= ========= ====== ============== ===============
myswitch      2 Up      Shared    lan0   0x00306e4aa30d 10.116.9.246
[Port Configuration Details]
Port    Port         Untagged Number of    Active VM
Number  state        VLANID   Reserved VMs
======= ============ ======== ============ ============
1       Reserved     none     1
2       Reserved     100      1
3       Reserved     none     1
4       Reserved     100      1
```

Display the verbose attributes of vswitch number 2:

```
#hpvmnet -s 2 -V

Name                    : myswitch
number                  : 2
PID                     : 25616
State                   : Up
Mode                    : Shared
PPA                     : lan0
Host MAC Address        : 0x00306e4aa30d
Host IP Address         : 10.116.9.246
Number of guests        : 1
Guest's Name            : hpux1
  MAC Address           : 0xEE4C7872D330
  IP Address            : 1.2.3.4
Packets in              : 7617014
Packets out to stream   : 1858
Packets out to guest    : 1173072
Packets dropped         : 6442084
Broadcasts              : 1171407
Counter 0               : 0
Counter 1               : 0
Counter 2               : 0
Counter 3               : 0
Cycles in               : 2919
Cycles out to stream    : 33105
Cycles out to guest     : 11776
Cycles switch packet    : 1097
```

Delete the vswitch named *switch2*:

```
# hpvmnet -d -S switch2

hpvmnet: Remove the vswitch switch2? [n]: y
```

Delete the vswitch with the vswitch id of 6, skipping 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
```

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      PPA    MAC Address    IP Address
======== ====== ======= ========= ====== ============== ===============
myswitch      2 Up      Shared    lan1   0x00306ef3120c 1.2.3.4
```

```
[Port Configuration Details]
Port    Port         Untagged Number of    Active VM
Number  state        VLANID   Reserved VMs
======= ============ ======== ============ ============
1       Reserved     none     1
2       Active       none     1            config2
3       Reserved     none     1
4       Active       none     1            winguest2

# hpvmnet -S myswitch -p 4
Vswitch Name            :  myswitch
Max Number of Ports     :  100
Port Number             :  4
  Port State            :  Active
  Active VM             :  winguest2
  Untagged VlanId       :  none
  Reserved VMs          :  winguest2
```

AUTHORS

The hpvmnet command was developed by the Hewlett-Packard Company.

SEE ALSO

hpvm(5), hpvmclone(1M), hpvmcollect(1M), hpvmconsole(1M), hpvmcreate(1M), hpvmdevmgmt(1M), hpvminfo(1M), hpvmmigrate(1M), hpvmmodify(1M), hpvmremove(1M), hpvmresources(1M), hpvmstart(1M), hpvmstatus(1M), hpvmstop(1M)

hpvmremove(1M)

NAME

hpvmremove -- Remove an Integrity Virtual Machines virtual machine.

SYNOPSIS

```
hpvmremove {-P vm_name | -p vm_number} [-F]
```

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

No options can be specified more than once.

`hpvmremove` 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 via 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).

RETURN VALUES

The `hpvmremove` command exits with one of the following values:

0: Successful completion.

1: One or more error conditions occurred.

DIAGNOSTICS

`hpvmremove` 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.
- `vm_name` or `vm_number` does not exist, cannot be accessed, is not a virtual machine, or is corrupt.
- `vm_name` is in some state other than Off.
- The `hpvmremove` command and Integrity Virtual Machines are at different revision levels.

EXAMPLES

Delete the virtual machine myguest:

```
# hpvmremove -P myguest
hpvmremove: Remove the virtual machine myguest? [n]: 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.
```

AUTHORS

The hpvmremove command was developed by the Hewlett-Packard Company.

SEE ALSO

hpvm(5), hpvmclone(1M), hpvmcollect(1M), hpvmconsole(1M), hpvmcreate(1M), hpvmdevmgmt(1M), hpvminfo(1M), hpvmmodify(1M), hpvmmigrate(1M), hpvmnet(1M), hpvmresources(1M)), hpvmstart(1M), hpvmstatus(1M), hpvmstop(1M)

hpvmresources(1M)

NAME

hpvmresources -- Specifying storage and network devices.

SYNOPSIS

```
Virtual resource specification
```

DESCRIPTION

The `hpvmcreate`, `hpvmclone`, and `hpvmmodify` commands may be used 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.

For storage devices, enter the resource specification as follows:

```
devicetype:adaptertype:bus,device,target:storage:device
```

The guest virtual device information consists of the following fields, separated by colons:

- devicetype (virtual device type):
  - `disk`
  - `dvd`
  - `tape`
  - `changer`
  - `burner`
- adaptertype (virtual device adapter type): `scsi`
- 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 will be automatically generated. 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 or partition character device file (for example, `/dev/rdsk/c4t3d2`).
- `lv`. For the physical device, specify the LVM or VxVM character logical device file (for example, `/dev/vg01/rlv0l2`).
- `file`. For the physical device, specify a locally-mounted, non-NFS, VxFS file (for example, `/guestfiles/diskfile1`).
- `null`. Specifies an empty storage unit. This should not be a world-writable directory such as `tmp`. (This is useful for removable media, such as DVDs.)
- `attach`. Specifies 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 shows how to associate a guest virtual disk device with a physical disk device as follows:

```
disk:scsi:0,1,0:disk:/dev/rdsk/c1t2d0
```

- 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/rdsk/c1t2d0`.

The following example shows how to associate 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 shows how to specify a tape device:

```
tape:scsi::attach:/dev/rscsi/c6t5d0
```

Specifying Network Devices

To associate a guest virtual network device with a virtual network switch (vswitch), use the syntax described below. (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` for network devices is:

```
network:adaptertype:bus,device,mac-addr:vswitch:vswitch-name:portid:portnumber
```

The guest virtual device information consists of the following fields, separated by colons:

- `network`
- adaptertype (virtual device adapter type): `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 will be checked to make sure it is unique, because the address cannot conflict with any of the VM Host's physical network adapter MAC addresses, and to make sure that the proper bits are set. You can specify 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 will be generated automatically. If you specify only a portion of the virtual network device hardware address (for example, just the MAC address), you must include the commas (for example, `,,00:01:01:01`).

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 shows how to associate a guest virtual network device with a vswitch:

```
network:lan:0,1,02-02-03-04-05-06:vswitch:net1
```

- The guest virtual network device type is `network`.
- The virtual adapter type is `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 hpvmcreate command was developed by the Hewlett-Packard Company.

SEE ALSO

hpvm(5), hpvmclone(1M), hpvmcollect(1M), hpvmconsole(1M), hpvmcreate(1M), hpvmdevmgmt(1M), hpvminfo(1M), hpvmmigrate(1M), hpvmmodify(1M), hpvmnet(1M), hpvmremove(1M), hpvmstart(1M), hpvmstatus(1M), hpvmstop(1M)

hpvmstart(1M)

NAME

hpvmstart - Start a virtual machine.

SYNOPSIS

```
hpvmstart {-P vm_name | -p vm_number} [-F | -s]
```

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 checks to make sure that the starting virtual machine can be allocated all of the required resources defined by its configuration file. If not, the virtual machine will not be started.

Any of the following conditions could prevent the virtual machine from starting:

1. The server has fewer CPUs than what the virtual machine requires.
2. The server has insufficient free memory.
3. The server has insufficient CPU resources.
4. The server has insufficient swap resources.
5. Another virtual machine is using a specified nonshared backing device.
6. The server is using a specified backing device.
7. A specified backing device does not exist.
8. A specified vswitch is not available. The vswitch must be created using the `hpvmnet` command before the guests using it can be started.
9. The specified MAC address is in use.
10. The specified guest is a distributed guest.

Only superusers can execute the `hpvmstart` command.

Options

No option can be specified more than 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 will be issued.

  Caution: HP does not recommend using the `-F` option because it can result in poor virtual machine performance, oversubscription, data corruption, or it may hang the virtual machine.

`-s`
: Sanity-checks the specified guest configuration and reports any errors or warnings that would prevent it from starting. The guest is not started.

RETURN VALUES

The `hpvmstart` 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 some 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
```

Following are sample warning messages returned when `hpvmstart` is executed with various configuration problems on the 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/rdsk/c2t1d0: Device file '/dev/rdsk/c2t1d0' 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/rdsk/c1t1d3 backing device does not exist.
Warning 8 on item /dev/rdsk/c3t1d0: Device file '/dev/rdsk/c3t1d0' 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.
```

AUTHORS

The hpvmstart command was developed by the Hewlett-Packard Company.

SEE ALSO

hpvm(5), hpvmclone(1M), hpvmcollect(1M), hpvmconsole(1M), hpvmcreate(1M), hpvmdevmgmt(1M), hpvminfo(1M), hpvmmigrate(1M), hpvmmodify(1M), hpvmnet(1M), hpvmremove(1M), hpvmresources(1M), hpvmstatus(1M), hpvmstop(1M)

hpvmstatus(1M)

NAME

hpvmstatus -- Display status information about one or more virtual machines.

SYNOPSIS

```
hpvmstatus [-V | -M | -X] [-v]
hpvmstatus {-P vm_name | -p vm_number} [-D] [-V | -M | -X] [-v]
hpvmstatus -e [-P vm_name | -p vm_number] [-V | -M | -X] [-v]
hpvmstatus -r [-P vm_name | -p vm_number] [-V | -M | -X] [-v]
hpvmstatus -d {-P vm_name | -p vm_number} [-M | -X] [-v]
hpvmstatus -m [-M | -X]
hpvmstatus -S [-M | -X]
hpvmstatus -s [-M | -X]
```

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:

- The version of the command (if you specify the `-v` option).
- The name of the virtual machine (limited to 20 characters in summary format).
- The 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.
  - Invalid: The virtual machine configuration file is corrupted or invalid. The configuration file must be corrected before this virtual machine can be started.
- The running condition of the guest.. The machine can be in one of the following conditions:
  - EFI: The virtual machines is running normally in Extensible Firmware Interface (EFI).
  - OS: The virtual machine is running normally in the operating system.
  - ATTN! - The virtual machine may need attention because it is not responding to interrupts.
- The resources attached to this virtual machine.
- The attributes assigned to 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 that are off, the command displays the configuration that will be used when the guest in next booted.

A variety of information can be presented:

- To list all the virtual machines that are on the VM Host, enter the `hpvmstatus` command without the `-P`, `-p`, `-e`, or `-r` options,
- To display detailed information about a virtual machine, use `-P` or `-p` option to specify the virtual machine, without the `-e`, `-r`, or `-d` options.
- To display devices in the same format used on the command line, include 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 resource scheduling information, include the `-r` option.
- To display the mode the scheduler is in, include the `-S` option.

To obtain a display in machine-readable format, use the `-M` or `-X` option.

Only superusers can execute the `hpvmstatus` command.

Options

No options can be specified more than 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.

The `-V`, `-M`, and `-X` options are mutually exclusive.

`-M`

Displays verbose attribute and resource information in a machine- readable format.

Individual fields are separated by one of three delimiters:

1. The colon (:) separates each field and resource type.
2. The semicolon (;) separates subfields of a resource type.
3. 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.

`-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 will take effect the next time the virtual machine is started (deferred mode).

`-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 CPU entitlement information for the virtual machines, including:

- #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 may be slightly different than what was specified. For example, the value may be rounded down to the nearest whole percentage of CPU entitlement.
- Maximum: The maximum amount of CPU entitlement this virtual machine can use.
- 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 cycles this virtual machine has consumed since it was booted.

When you specify a virtual machine, the `hpvmstatus` command displays the following information for each virtual CPU:

- Cumulative Usage: The number of cycles this virtual CPU has consumed since the virtual machine was booted.
- Guest percent: The CPU percentage the guest has consumed.
- Host percent: The CPU percentage that the VM Host uses on behalf of the guest.
- Cycles achieved (expressed in MHz).
- Sampling Interval: The time period between samples.

`-d`

Displays the devices on the specified virtual machine in the same format used on the command line.

-S

Reports the scheduler mode (usually NORMAL). If gWLM controls the VM Host, the scheduler will be in CAPPED mode.

-s

Displays the current VM Host resources.

-m

Displays information about the multiserver environment, including the Serviceguard identifier, state, IP address, and hostname. If the VM Host is not a Serviceguard server, the following message is displayed:`No HPVM multi-server environment configured`.

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 Virtual Machines 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  Runsysid
==================== ===== ======= ========= ====== ===== ===== ======= ========
config1                  1 HPUX    Off             1     5     1  512 MB
  0
config2                  2 HPUX    Off             1     7     1    1 GB
  0
winguest1                5 WINDOWS On (OS)         1     5     1    1 GB
  0
winguest2                9 WINDOWS Off             1     3     1    2 GB
  0
```

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    Off

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

```
[Virtual CPU Details]
#vCPUs  Entitlement Maximum
======  =========== =======
    1          5.0%  100.0%

[Memory Details]
Total       Reserved
Memory      Memory
=========== ===========
   1 GB    64 MB

[Storage Interface Details]
Guest                                  Physical
Device  Adaptor    Bus Dev Ftn Tgt Lun Storage   Device
======= ========== === === === === === ========= ========================
dvd     scsi         0   0   0   0   0 file      /bigfiles/HPUX11i_0603-OE-MC.iso
disk    scsi         0   0   0   1   0 disk      /dev/rdsk/c4t9d0
disk    scsi         0   0   0   2   0 file      /bigfiles/vdisk_config2
disk    scsi         0   0   0   3   0 disk      /dev/rdsk/c5t3d0
dvd     scsi         0   0   0   4   0 file      /bigfiles/win_3790.iso
dvd     scsi         0   0   0   5   0 null      /dev/rdsk/c0t0d0
dvd     scsi         0   0   0   6   0 null      /bigfiles/

[Network Interface Details]
Interface Adaptor    Name/Num   PortNum Bus Dev Ftn Mac Address
========= ========== ========== ======= === === === =================
vswitch   lan        myswitch   2         0   5   0 02-47-41-46-01-02

[Misc Interface Details]
Guest                                  Physical
Device  Adaptor    Bus Dev Ftn Tgt Lun Storage   Device
======= ========== === === === === === ========= ========================
serial  com1                           tty       console
```

Display the mode in which the scheduler is running:

```
# hpvmstatus -S

HPVM scheduler is running in NORMAL mode.
```

Display the system resources on the VM Host system:

```
# hpvmstatus -s

                [HPVM Server System Resources]

        Processor speed = 1000 Mhz
        Total physical memory = 12276 Mbytes
        Total number of processors = 2
        Available memory = 8560 Mbytes
        Available swap space = 17782 Mbytes
        Maximum vcpus for an Hpux virtual machine = 2
        Maximum vcpus for a Windows virtual machine = 2
        Maximum entitlement for a 1 way virtual machine = 1000 Mhz
        Maximum entitlement for a 2 way virtual machine = 250 Mhz
```

Display the Serviceguard server information on the VM Host system:

```
# hpvmstatus -m

HPVM Multi-server environment
        This servers identifier=1

        Server_id=1
        Server_state=1
        Server_ipaddr=1.2.3.4
        Server_hostname=rake

        Server_id=1
        Server_state=0
        Server_ipaddr=1.3.5.7
        Server_hostname=cloud
```

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
```

AUTHORS

The hpvmstatus command was developed by the Hewlett-Packard Company..

SEE ALSO

hpvm(5), hpvmclone(1M), hpvmcollect(1M), hpvmconsole(1M), hpvmcreate(1M), hpvmdevmgmt(1M), hpvminfo(1M), hpvmmigrate(1M), hpvmmodify(1M), hpvmnet(1M), hpvmremove(1M), hpvmresources(1M), hpvmstart(1M), hpvmstop(1M)

hpvmstop(1M)

NAME

hpvmstop -- Stop a virtual machine.

SYNOPSIS

```
hpvmstop {-P vm_name | -p vm_number} [-h | -g] [-F] [-q]
```

DESCRIPTION

The `hpvmstop` command stops a running virtual machine by simulating the operations performed at the system console on a physical system. It 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.

Unless the `hpvmstop` command returns an error message, the specified virtual machine is shut down.

The `hpvmstop` command does not create a crash dump, and no automatic restart is 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

No options can be specified more than 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 either the `-P` or the `-p` 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 either the `-P` or the `-p` option.

`-g`
: Performs a graceful shutdown. The guest operating system is notified of an imminent power failure, which gives it time to perform cleanup operations. This is the default action. HP recommends stopping virtual machines using their native operating system commands.

  The `-h` and `-g` options are mutually exclusive.

`-h`
: Performs a hard stop, equivalent to a power failure. The guest operating system receives no notice and thus no opportunity to cleanup. In these circumstances, the guest operating system does not create a crashdump and no automatic restart is performed. HP recommends stopping virtual machines using their native operating system commands.

  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.

`-q`
: Makes certain scripted operations less verbose (quiet mode).

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.
- `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 are at different revision levels.
- The specified guest is a distributed guest.

EXAMPLES

Perform a graceful shutdown of the virtual machine called compass1:

```
# hpvmstop -P compass1
```

AUTHORS

The hpvmstop command was developed by the Hewlett-Packard Company.

SEE ALSO

hpvm(5), hpvmclone(1M), hpvmcollect(1M), hpvmconsole(1M), hpvmcreate(1M), hpvmdevmgmt(1M), hpvminfo(1M), hpvmmigrate(1M), hpvmmodify(1M), hpvmnet(1M), hpvmremove(1M), hpvmresources(1M), hpvmstart(1M), hpvmstatus(1M)

hpvm(5)

NAME

hpvm – HP Integrity Virtual Machines (Integrity VM).

SYNOPSIS

```
Virtualization technology
```

DESCRIPTION

Integrity Virtual Machines allows the creation and management of virtual machines (VMs) in which unmodified operating systems designed for the Itanium Processor Family (IPF) can run. Integrity Virtual Machines 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 commands and their descriptions:

- `hpvmclone`: Create a cloned copy of a virtual machine.
- `hpvmcollect`: Collect crash dumps, logs, system status, and configuration on host and guest for post-mortem analysis.
- `hpvmconsole`: Connect to the console of a virtual machine.
- `hpvmcreate`: Create a new virtual machine.
- `hpvmdevmgmt`: Manage the device database.
- `hpvminfo`: Display information about the Integrity VM environment.
- `hpvmmigrate`: Move a virtual machine from one VM Host to another.
- `hpvmmodify`: Rename or modify the attributes of a virtual machine.
- `hpvmnet`: Configure virtual network devices.
- `hpvmremove`: Remove a virtual machine.
- `hpvmstart`: Start a virtual machine.
- `hpvmstatus`: Display status of one or more virtual machines.
- `hpvmstop`: Stop a virtual machine.

All commands except `hpvmconsole` require superuser privileges.

AUTHORS

Integrity Virtual Machines was developed by the Hewlett-Packard Company.

SEE ALSO

hpvmclone(1M), hpvmcollect(1M), hpvmconsole(1M), hpvmcreate(1M), hpvmdevmgmt(1M), hpvminfo(1M), hpvmmigrate(1M), hpvmmodify(1M), hpvmnet(1M), hpvmremove(1M), hpvmresources(1M), hpvmstart(1M), hpvmstatus(1M), hpvmstop(1M)

Glossary

This glossary defines the terms and abbreviations as they are used in the Integrity VM product documentation.

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 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.

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.

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.

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 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 The local network created by Integrity VM for internal local communications. Guests can communicate on the localnet, but the VM Host cannot.

migration The operation of stopping a Serviceguard package on one cluster member and then starting it on another cluster member. This is accomplished using the `hpvmmigrate` command. 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.

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 controll 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 mltiple 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. Many customers using Serviceguard want 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.

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.
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

T

U

V

W