HP Integrity Virtual Machines Installation, Configuration, and Administration

Part number: T2767-90004
published October 2005, Edition 1

Legal Notices

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

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

Warranty

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

U.S. Government License

Proprietary computer software. Valid license from HP required for possession, use or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

Trademark Notices

Intel® and Itanium® are registered trademarks of Intel Corporation in the US and other countries and are used under license.

MS-DOS®, Microsoft®, and Windows® are registered trademarks of Microsoft Corporation in the United States of America and in other countries.

UNIX® is a registered trademark of The Open Group.

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.

Java™ is a US trademark of Sun Microsystems, Inc.

Publication History

The manual publication date and part number indicate its current edition. The publication date will change when a new edition is released. The manual part number will change when extensive changes are made.

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

Please direct comments regarding this guide to:

Hewlett-Packard Company
HP-UX Learning Products
3404 East Harmony Road
Fort Collins, Colorado 80528-9599

About This Guide

This Installation, Configuration, and Administration guide provides procedures to install and configure the Integrity Virtual Machines product, and 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. These terms may appear in related Virtual Server Environment documentation.

If you need information about HP-UX 11i, go to the web:

`http://docs.hp.com`

If you need to set up your system in different languages, please refer to *Configuring HP-UX for Different Languages*, available on the Instant Information DVD and on the HP documentation web site:

`http://docs.hp.com`

Intended Audience

This document is intended for system and network administrators responsible for installing, configuring, and managing Integrity 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.

Related Documents

The following documents, which are found at the *HP Technical Documentation* Web site at **http://docs.hp.com/**, may be useful to the reader of this document:

- *Ignite-UX Reference*
- *Troubleshooting Ignite-UX Installation Booting* White Paper
- *HP-UX Installation and Update Guide*
- *HP-UX Reference*

Related Information

The Integrity VM Web site is at:**http://h71028.www7.hp.com/enterprise/cache/262803-0-0-0-121.html**

This web site contains the following:

- Product description
- *Introducing HP Integrity Virtual Machines*, a white paper

Additionally, details about recent changes to Integrity VM are in the *Integrity Virtual Machines Release Notes* located on the product media and at **http://docs.hp.com**.

HP offers training for the experienced HP-UX system administrator. For details and information, go to:

http://www.hp.com/education/course-listing

You can find other HP-UX related courses at:

http://www.hp.com/education/sections/hpux.html

Additionally, HP offers technical online seminars that may be of interest located at:

http://www.hp.com/education/sections/hpux.html#tos

The *HP IT Resource Center* (ITRC) can be an invaluable source of information regarding HP products and can be found at:

http://itrc.hp.com

The IT Resource Center forums can be found at:

http://forums.itrc.hp.com/

The IT Resource Center offers services and support for your HP-UX, Linux, MPE/iX, NT, OpenVMS, and Tru64 UNIX servers and workstations, including information on patches, warranties, software, and drivers. It contains software, hardware, and network support information to help you manage your computing environment.

Additionally, the following Web sites may be of interest in obtaining a variety of information regarding the HP-UX and associated hardware architectures:

Enterprise servers, Workstations, and Systems Hardware:

http://docs.hp.com/hpux/hw/

HP Integrity Servers: **http://www.hp.com/go/integrity**

HP Software Depot: **http://software.hp.com**

HP Software Releases and Media:

http://www.software.hp.com/RELEASES-MEDIA

Software Availability Matrix: **http://software.hp.com/MATRIX/**

Software Transition Kit and Software Solutions:

http://h20214.www2.hp.com/drc/

Developer & Solution Partner Program (DSPP):

http://www.hp.com/dspp

Dev Resource Control Central: **http://devresource.hp.com**

HP Encourages Your Comments

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

Please submit comments to:

http://docs.hp.com/assistance/feedback.html

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

Typographic Conventions

We use the following typographical conventions.

audit(5)	HP-UX manpage. *audit* is the name and *5* is the section in the *HP-UX Reference*. On the web and on the Instant Information DVD, it may be a hot link to the manpage itself. From the HP-UX command line, you can enter "`man audit`" or "`man 5 audit`" to view the manpage. See man(1).
Book Title	Title of a book. On the web and on the Instant Information DVD, it may be a hot link to the book itself.
`Command`	Command name or qualified command phrase.
`ComputerOut`	Text displayed by the computer.
Emphasis	Text that is emphasized.
Emphasis	Text that is strongly emphasized.
KeyCap	Name of a keyboard key. Note that **Return** and **Enter** both refer to the same key.
Term	Defined use of an important word or phrase.
`UserInput`	Commands and other text that you type.
`Variable`	Name of a variable that you may replace in a command or function or information in a display that represents several possible values.
[ ]	Contents are optional in formats and command descriptions. If the contents are a list separated by `|`, you must choose one of the items.
{ }	Contents are required in formats and command descriptions. If the contents are a list separated by `|`, you must choose one of the items.
...	Preceding element may be repeated an arbitrary number of times.
|	Separates items in a list of choices.

Table of Contents

List of Tables

1 Introduction

This chapter describes the Integrity Virtual Machines product, including:

- "About HP Integrity Virtual Machines" describes the Integrity VM product.
- "Integrity VM Documentation" describes how to use this manual and how to find other documentation that is useful in setting up and managing Integrity Virtual Machines.

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. The Virtual Machines environment consists of two types of components:

- VM Host
- Virtual machines (also called guests)

The VM Host virtualizes the CPU, memory, and I/O devices, presenting abstractions to the guests, allowing you to allocate the resources each guest requires, providing complete control over resources, flexible management, and efficient use of the system's physical resources.

Virtual machines are abstractions of real, physical machines. The guest runs on the virtaul machine with little or no special modification, except for a small guest package provided by Integrity VM for local management of the guest's virtual machine.

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 take advantage of Integrity VM is to run multiple virtual machines on the same physical machine. Each virtual machine is isolated from the others. The VM Host administrator allocates virtual resources to the guest. A symmetric multiprocessing system can run on the virtual machine if the VM Host system has sufficient physical CPUs for it. 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.

Because multiple virtual machines share the same physical resources, I/O devices can also 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 localnet that the VM Host creates for each guest by default. The localnet allows communication among guests; the VM Host does not communicate on the 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 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.

Features of Integrity VM

- Support for a variety of HP Integrity servers, from low-end Blade servers to high-end multiprocessing systems.
- Support for single processor and multiprocessing virtual machines.
- Support for virtual machines running different operating system versions and patch levels.
- Fine-grained allocation of physical CPUs to virtual CPUs.
- Dynamic, automatic reallocation of CPU resources to virtual CPUs based on utilization.
- Guest operating system fault and security isolation.
- Virtual machine management isolation that allows you to boot, reconfigure, add, and remove virtual machines without affecting the other virtual machines.

- Centralized VM Host administration to create, remove, and modify guests from a single, central console.
- Powerful, easy-to-use command line interface to manage the VM Host and virtual machines.
- Ability to share I/O resources among guests for maximum utilization without conflicts.
- Multiple options for physical storage backing virtual disks on the guests, including:
  - RAID arrays/LUNs
  - Physical disks/partitions
  - Logical volumes
  - Files
- Virtual DVDs backed by either physical DVDs or ISO files.

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.

Manpages

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

- `hpvm(5)`, which describes the Integrity VM environment.
- `hpvmclone(1M)`, which describes how to create virtual machines.
- `hpvmcollect(1M)`, which describes how to collect virtual machine statistics.
- `hpvmconsole(1M)`, which describes how to use the virtual machine console.
- `hpvmcreate(1M)`, which describes how to create virtual machines.
- `hpvmdevmgmt(1M)`, which describes how to modify the way virtual devices are handled.
- `hpvminfo(1M)`, which describes how to get information about the VM Host.
- `hpvmmodify(1M)`, which describes how to modify virtual machines.
- `hpvmnet(1M)`, which describes how to create and modify virtual networks.
- `hpvmstart(1M)`, which describes how to start virtual machines.
- `hpvmstatus(1M)`, which describes how to get statistics about the guests.
- `hpvmstop(1M)`, which describes how to stop a virtual machine.
- `hpvmremove(1M)`, which describes how to remove a virtual machine.

Help Files

The virtual console is a special interface for managing guests. To invoke the virtual console after you create a guest, enter the `hpvmconsole` command, specifying the guest name. For help using the virtual console, enter the `HE` command. For more information about the virtual console, see "Providing Selective Access to Guest Consoles" in Chapter 3.

Related Documentation

Additional sources of information that you might be useful as you install Integrity Virtual Machines include:

- *HP-UX 11i Installation and Update Guide*
- *Software Distributor Administration Guide for HP-UX*
- HP Integrity Virtual Machines *Release Notes*

  Always read the release notes before installing or using HP Integrity Virtual Machines.
- *Introducing HP Integrity Virtual Machines*, a white paper available at:

`http://h71028.www7.hp.com/enterprise/downloads/Intro_VM_WP_12_Sept%2005.pdf`

- *The HP Virtual Server Environment — Making the Adaptive Enterprise Vision a Reality in Your Datacenter*, by Dan Herington and Brian Jacquot (Prentice Hall PTR, ISBN 0130855220)
- *Operation and Maintenance Guide* for your Integrity server.

Using This Manual

This manual is organized into the following chapters and appendixes:

- Chapter 1, "Introduction"

  This chapter describes the basic virtual machine concepts and the architecture of Integrity VM.

- Chapter 2, "Planning Your Virtual Machines"

  This chapter describes each of the system and network characteristics that you have to consider before setting up your virtual machine environment. It provides examples of different guest OS configurations and shows how to allocate the system resources to each guest.

- Chapter 3, "Installing Integrity VM"

  This chapter describes what you need in order to install the Integrity VM product, as well as the procedure for installing the software from media or network depots.

- Chapter 4, "Creating and Booting Guests"

  This chapter describes the procedure for creating and booting virtual machines.

- Chapter 5, "Managing Guests"

  This chapter describes how to manage and modify the VM Host and the guests.

- Appendix A provides details about the VM commands used in the procedures in this manual.
- Appendix B describes how to get information about system problems and report them to HP.
- The Glossary defines the special terms used in this manual.

2 Planning Your Virtual Machines

To achieve your goals using virtual machines, you must plan the configuration of each guest, assessing its requirements for resources on the Integrity system on which they will run. This chapter explains how to assess the resources that your system has, as well as the guests you will run on the system, and shows you how to map your guest requirements to the system resources.

- "VM Host System Resources" describes the elements of the VM Host system that you must assess as part of the configuration process and presents an example worksheet for recording the system information.
- "Guest Requirements" describes the virtual resources that guests use, and presents a worksheet for recording the guest requirements.
- "Allocating Resources to Guests" presents examples of three guests with different operating requirements and discusses how they might be set up to share a single VM Host system.
- "Running Applications in the Integrity VM Environment" describes the application environment of the VM Host and guests.

VM Host System Resources

When you install Integrity VM, the VM Host is automatically created and started. The resources of the VM Host system can be allocated to the guests, but the VM Host requires certain dedicated resources as well. Therefore, in order to install Integrity VM and the guests, you must understand what resources the VM Host system offers and what resources must be restricted to use by the VM Host. Table 2–1 describes the VM Host system resources, including those that must be restricted to the VM Host and those that can be allocated to guests. This table includes an example VM Host system, which is used later in the discussion of how to plan the guests.

Table 2-1 VM Host System Resources Information

System Element	Description	Example
VM Host name	The name of the VM Host system. It is generated from the UNIX system name.	`compass`
Operating system	The operating system running on the VM Host. You can run HP-UX 11i v2 May 2005 or later.	HP-UX 11i v2 May 2005
Number of physical CPUs	The number of physical CPUs on the VM Host system.	4
Memory (RAM)	The amount of memory on the VM Host system.	32 GB
Reserved memory	The amount of memory that is reserved for the use of the VM Host. Refer to "Installation Requirements" in Chapter 3.	5 GB
Network adapters	The number of network adapters (also called NICs, or network cards) on the VM Host system.	2
IP addresses or DHCP	The IP address for each network adapter, or served by a DHCP server.	lan0: 17.22.3.4 lan1: DHCP
Boot device	The VM Host system's boot device. This device must not be accessible to guests.	`/dev/rdsk/c0t1d0`
Unique UNIX account name	The UNIX account name for the VM Host system administrator.	hostadmin
Restricted devices	The devices that must not be accessible to guests. These include the boot device, the swap device, and any logical volumes used by the VM Host. Network devices can also be restricted to the VM Host. For more information, refer to "Restricting Devices to the VM Host" in Chapter 3.	`/dev/vg00/lvol3`

After you have recorded the information about the VM Host requirements and characteristics, you should plan each guest's requirements and characteristics to ensure that the virtual machines run as efficiently as possible without contending for resources on the VM Host system.

Guest Requirements

When you create each guest, you can specify the characteristics of the guest and the resources required by the guest. Table 2–2 describes the characteristics and resources that you can specify for each guest. Integrity VM allows you to create the guests that you describe regardless of whether the guests could actually run on the VM Host, issuing warning messages for any potential problems in the current VM Host environment. This is useful for setting up guests for future VM Host system configurations. When you start the guest, the VM Host will ensure that the guest can operate in the current VM Host system environment. If the guest cannot be started, Integrity VM provides error messages describing the specific guest characteristics that are not appropriate.

Table 2-2 Guest Requirements Information

Guest Information	Description
Guest name	The name that you specify when you create the guest. A guest name can be up to eight alphanumeric characters. If you plan to provide remote console access to the guest, its name must be a legal UNIX account name..
Operating system	The operating system that the runs on the virtual machine. Specify HP-UX 11i v2 May 2005 or later.
Virtual CPUs	The number of virtual CPUs that the virtual machine will use. Every guest has at least one virtual CPU, which is the default. A guest cannot use more than four virtual CPUs.
CPU entitlement	The minimum amount of each physical CPU guaranteed to the guest. This can be specified in either percentage or CPU clock cycles. For more information, refer to "Guest Processing Power."
Memory	The minimum amount (expressed in either MB or GB) of virtual memory required by the guest, including the operating system and the applications that run on it. For more information, refer to "Guest Memory Allocation."
MAC address	The MAC address of each network device. Use this space to record the virtual MAC address generated for the guest when it is created.
Network adapters	The number of network devices required by the guest.
Network device hardware address	The hardware address for the virtual network device (PCI bus and PCI slot). Use this space to record the network device hardware address generated for the guest when it is created.
Vswitch name	The virtual network devices (vswitches) to be used by the guest OS. Virtual network devices must be created for each network device. For more information about virtual network devices, refer to "Virtual Network Devices."
Boot device	The disk used for the guest's boot device. Each guest must have a unique, private root, and for best performance they should be on different physical devices.
Startup behavior	Whether the guest is booted automatically when Integrity VM is started, or booted manually with the `hpvmstart` command, as described in Chapter 4.
Admin account name	User account name or group name of the guest administrator. For information about access to guest virtual consoles, refer to "Providing Selective Access to Guest Consoles" in Chapter 3.
Applications	List the types and names of applications that run on the guest. For information about the application environment on guests, refer to "Running Applications in the Guest Environment."
Application (virtual) storage device	For each application, the storage media to which it requires access. For more information about the kinds of storage devices that guests can access, refer to "Virtual Storage Devices".
Virtual storage device hardware address	The hardware address of the storage device required by the guest. Use this space to record the PCI bus number, the PCI slot number, and the SCSI target number generated for the guest when it is created.
Physical backing store	The physical storage device type and the device that allocated to the guest.

The following sections describe the resources that you can allocate to guests.

Guest Processing Power

When you create a guest, you can specify the number of virtual CPUs that the guest can use, as well as the minimum amount of CPU processing power that is guaranteed to the guest for each virtual CPU. For the

purposes of this discussion, the term “physical CPU” refers to a processing entity on which a software thread can be scheduled. A guest can be allocated up to four virtual CPUs (limited by the total number of physical CPUs on the VM Host system).

Integrity VM allows you to create a guest with more CPUs than the physical system has, issuing warning messages if there are not enough physical CPUs. When you create a guest, Integrity VM checks the current CPU resources and, if insufficient resources are available to run the guest in the current configuration, issues a warning but allows you to create the guest. This allows you to create guests for future configurations. However, the guest is not allowed to boot on a system that does not have enough physical CPUs. If you do not specify the number of virtual CPUs when you create the guest, the default is one virtual CPU.

You can also specify the minimum amount of processing power guaranteed to the guest from each virtual CPU. This is the guest's “entitlement.” When you create a guest, you can specify the entitlement as a percentage from 5% to 100%. If you do not specify the entitlement, the guest receives 5% entitlement by default. Alternatively, you can specify the entitlement as the number of CPU clock cycles per second to be guaranteed to each virtual CPU on the guest.

When the guest is booted, the VM Host ensures that sufficient processing power is available for each guest to receive its entitlement. For a guest with multiple virtual CPUs, it verifies that the guest entitlement is available on the same number of physical processors as the guest as virtual CPUs. The sum of the all the entitlements for all the running guests cannot total more than 100% for each physical processor. If insufficient CPU resources are available, the guest is not allowed to boot; error messages are displayed to indicate the specific problem.

If a guest is busy, and sufficient processing power is available on the system, that guest can receive more than its entitlement. When there is contention for processing power (as on an oversubscribed system with busy guests) each guest is limited to its entitlement. You can modify the number of virtual CPUs and entitlement for a guest, as described in “Changing Guest Configurations” in Chapter 5.

For guests with multiple virtual CPUs, the entitlement is guaranteed on all the virtual CPUs in the guest's configuration. To prevent contention, the VM Host schedules work across all the virtual CPUs in the guest's configuration.

Guest Memory Allocation

When you create a guest, you can specify the amount of virtual memory (in either gigabytes or megabytes) to be allocated to the guest. 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 you specify when you create the guest 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 guest. This allows you to create guests for future configurations. When the guest is started, the VM Host makes sure that there is sufficient memory to run the guest. In addition to the amount of memory you specify for the guest, the VM Host requires a certain amount overhead for booting the guest. 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, refer to “Installation Requirements,” in Chapter 3.

Guest Virtual Networks

The guest virtual network consists of:

- Virtual network device
- Virtual switch

For the guest to communicate outside the VM Host system, each guest virtual network must be associated with a virtual switch (vswitch). If you start a guest without any vswitch, the guest has no network communication channel. This is like booting up a single system for the first time using the console.

For each network adapter accessible to a guest, you must create a vswitch. A vswitch functions like a physical network switch, 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 physical network device, or it can be local to the virtual machines on the VM Host, providing a local network between guests.

You create a virtual switch using the `hpvmnet` command, as described in "Creating Virtual Switches" in Chapter 3. You can create virtual switches any time. If you create the guest before creating a specific vswitch, the guest is created and warning messages display the specific problem. This allows you to create guests for future configurations. In the examples in this manual, the vswitch is created before the guest.

You can also restrict physical network devices to use by the VM Host only.

Integrity VM always creates a vswitch named localnet. This network is not associated with a physical interface. It is used only for communication between the guests running on the same VM Host. This interface does not use a name server or router, and the VM host does not access the localnet.

The guest OS configures its own virtual network interface with an IP address using standard commands and utilities. It can also use DHCP.

Virtual Storage Devices

When you create a guest, you specify the virtual storage devices that the guest uses. Guest virtual storage devices are backed by physical devices on the VM Host system. You must have sufficient physical storage for the VM Host and for all of the guests.

When you create a guest with the `hpvmcreate` command, you can specify both the virtual devices that the guest recognizes and the physical backing stores on the VM Host system. The guest virtual storage devices are either disks or DVDs. Virtual disks are read/write, so they cannot be shared between guests or with the VM Host. Virtual DVDs can be specified as shareable, so they can be used as installation golden images. However, sharing DVDs among guests should be carefully planned to avoid poor performance and overly complex management.

When you allocate a storage device to guests, be careful not to cause conflicts in uses of a backing store. For example, if a file in a file system on /dev/dsk/c8t2d0 is used as a backing store, the raw device /dev/rdsk/v8t2d0 cannot also be used as a backing store. Conflicts are not always obvious, so it is important to be careful when allocating backing stores to guests. HP recommends that you create a matrix of all the VM Host's resources and the guests' usage of them. The worksheets presented in this chapter are helpful for organizing this information.

The physical devices on the VM Host system that can be used as backing stores are:

- Disks and DVDs
- Logical volumes
- Locally mounted files

Integrity VM does not support HFS. NFS is supported but not recommended.

When you create the guest, Integrity VM checks the current physical configuration. If the guest uses backing stores that are not available, the guest is created and warning messages provide details. If you start a guest that requires physical resources that are not available on the VM Host system, the guest is not allowed to start, and error messages provide detailed information about the problem.

The physical backing store that you associate with a guest virtual device can affect the performance of the guest. 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 guests across the physical backing stores.

Each type of backing store type has benefits and drawbacks:

- Files are easy to create and change, but they can be slow for guests to access.
- Disks are fast but expensive.
- Disk partitions are fairly fast but difficult to manage.
- Logical volumes provide good performance and are fairly easy to manage.

When you create logical volumes as backing stores, create them with no file system and do not mount them. Integrity VM uses them as raw devices.

Some devices should be restricted to use by the VM Host and to each guest. The VM Host requires restricted devices, as described in "Controlling Access to Devices" in Chapter 3. Guests also require dedicated storage devices for their guest operating system boot device and swap device.

Allocating Resources to Guests

The way you allocate the physical resources to the guests determines the ultimate success of your configuration. For both performance and safety, spread the workload across devices. The VM Host reads the guest configuration from the guest configuration file at `/var/opt/hpvm/guests/guestname`. You can check the current configuration against your plans, adjusting the configuration for better performance and easier management.

The guests must all share the same physical system with the VM Host. Therefore, it is useful to look at the resource requirements of all guests that will be running at the same time. Table 2–3 lists the types of information you need in order to create a guest, with space to enter your own guest information. Use the information from Table 2–1, VM Host System Resources Information to help you assess the resource requirements for your Integrity VM environment.

Table 2-3 Guest Planning Worksheet

Guest name	
Operating system	
Virtual CPUs	
CPU entitlement	
Memory	
Network adapters	
Network device hardware address	
IP address	
Vswitch name	
Boot device	
Startup behavior	
Admin account name	
Applications	
Application (virtual) storage device	
Virtual storage device hardware address	
Physical backing store	

Note that this table reflects Table 2–2. Use the information from Table 2–2 to help you fill out your planning worksheet for each guest.

The following sections present three example guests designed to run on the VM Host named `compass`, which is presented in Table 2-1. The three example guests are:

- `compass1`, a software distribution server with high network and disk storage requirements.
- `compass2`, a research and development system. For security purposes, it can access only the local network. This guest is a heavy CPU and memory user.
- `compass3`, an operations server. It has high disk-storage requirements, requires network access, and has regular spikes of high CPU and memory usage.

In the following sections, each guest is added to the planning chart, allowing you to assess the total requirements of all the guests running at the same time on the same VM Host system.

The Distribution Server (compass1)

The first example guest, `compass1`, has one virtual CPU, two virtual network devices, one of which is a dedicated network device. This distribution server requires both virtual disk storage (a logical volume) and a virtual DVD storage device. Table 2-4 shows the planning worksheet with the data for the first guest included.

Table 2-4 Planning Worksheet for compass1

Guest name	**compass1**		
Operating system	HP-UX 11i v2 May 2005		
Virtual CPUs	1		
CPU entitlement	5%		
Memory	1 GB		
Network adapters	2		
Network device hardware address	lan(0,1)		
IP address	DHCP served		
Vswitch name	clan1 (shared) clan2 (dedicated)		
Boot device	/dev/rdsk/c0t1d0		
Startup behavior	automatic		
Admin account name	guest1		
Applications	Oracle 10*g*		
Application (virtual) storage device	disk (/dev/rdsk/c1t1d0) DVD		
Virtual storage device hardware address	disk: default hardware address DVD: PCI bus: 0 PCI slot: 0 SCSI target: 1		
Physical backing store	disk: /dev/vg01/rlv022 DVD: /null:/root		

If the guest runs multiple applications with specific requirements for virtual devices, you might need to expand this chart.

The R&D System (compass2)

Information about the second guest, `compass2`, is entered into the next blank column in the worksheet, as shown in Table 2–5. This guest is a research and development system. It runs a multiprocessing operating system and has large memory requirements. It is a highly secure environment, and network usage is restricted to local machines. This virtual machine is an isolated environment for patching, upgrading, and testing software changes.

Table 2-5 Planning Worksheet for compass2

Guest name	**compass1**	**compass2**	
Operating system	HP-UX 11i v2 May 2005	HP-UX 11i v2 May 2005	
Virtual CPUs	1	2	
CPU entitlement	5%	50%	
Memory	1 GB	5 GB	
Network adapters	2	1	
Network device hardware address	lan(0,1)	none	
IP address	DHCP served	none	
Vswitch name	clan1 (shared) clan2 (dedicated)	localnet	
Boot device	/dev/rdsk/c0t1d0	/dev/rsk/c0t2d0	
Startup behavior	automatic	manual	
Admin account name	guest1	guest2	
Applications	Oracle 9	C++	
Application (virtual) storage device	disk (/dev/rdsk/c1t1d0) DVD	disk (/dev/rdsk/c1t2d0)	
Virtual storage device hardware address	disk: default hardware address DVD: PCI bus: 0 PCI slot: 0 SCSI target: 1	disk: default hardware address	
Physical backing store	disk: /dev/vg01/rlv022 DVD: /null:/root	disk:/dev/vg02/rlv023	

Make sure that dedicated virtual devices (like the boot disk) are not the same on any other guests, and that the number of virtual CPUs and the amount of memory you specify do not exceed the amount that is available on the VM Host system when the guest is created. Note that the storage device for the `compass2` guest is different from the `compass1` guest, making it easier to manage the VM Host system when both guests are running at the same time, as well as providing a balanced workload.

The Operations Server (compass3)

The third guest, `compass3`, is the system used by management and corporate operations. The application demands on this virtual machine vary greatly and the resource demands spike frequently. Network access is required and there are high I/O performance requirements. Table 2–6 shows the information for `compass3` in the last column of the worksheet.

Table 2-6 Planning Worksheet for compass3

Guest name	**compass1**	**compass2**	**compass3**
Operating system	HP-UX 11i v2 May 2005	HP-UX 11i v2 May 2005 Update	HP-UX 11i v2 May 2005
Virtual CPUs	1	2	1
CPU entitlement	5%	50%	5%
Memory	1 GB	3 GB	2GB
Network adapters	2	1	1
Network device hardware address	lan(0,1)	none	lan(0,1)
IP address	DHCP	none	17.22.3.6
Startup behavior	automatic	manual	automatic
Vswitch name	clan1 (shared) clan2 (dedicated)	localnet	clan1 (shared)
Boot device	`/dev/rdsk/c0t01d0`	`/dev/rdsk/c0t2d0`	`/dev/rdsk/c0t3d0`
Startup behavior	automatic	manual	automatic
Admin account name	guest1	guest2	guest3
Applications	Oracle 9	C++	DeskMgr
Application (virtual) storage device	disk: /dev/rdsk/c1t1d0 dvd	disk: /dev/rdsk/c1t2d0	disk: /dev/rdsk/c1t3d0
Virtual storage device hardware address	disk: default hardware address DVD: PCI bus: 0 PCI slot: 0 SCSI target: 1	disk: default hardware address	disk: default hardware address
Physical backing store	disk: /dev/vg01/rlv022 DVD: /null:/root	disk: /dev/vg02/rlv023	disk: /dev/vg03/rlv024

The planning worksheet now shows all three guests side by side. Use the parameters from the VM Host system (Table 2–1) and those you record in the guest planning worksheet (Table 2–6) to assess the total requirements of the guests on the system. Total memory and disk space requirements include the guest requirements and the VM Host requirements. For more information about the total memory and disk space requirements, refer to "Installation Requirements" in Chapter 3.

Running Applications in the Integrity VM Environment

The VM Host system runs the Integrity VM software. It can also run physical resource, performance, and software management and monitoring tools. Do not run end-user applications on the VM Host. Typical software you can run on the VM Host includes the following:

- HP-UX Foundation Operating Environment (FOE)
- 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)

Software that should not be run on the VM Host system includes the following:

- Process Resource Manager (PRM)
- 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 the same access to memory, CPUs, network devices, and storage devices as if they were part of dedicated system.

Typical software to run on a guest includes the following:

- HP-UX Foundation Operating Environment (FOE)
- 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.

The following types of applications should not be run on a guest:

- Integrity VM software
- Hardware diagnostic tools and support tools (should be run on the VM Host)
- Utility pricing tools (should be run on the VM Host)
- Applications that require direct access to physical hardware (for example, disaster-tolerant solutions)
- SAN Management tools and applications that require access to serial interfaces (Integrity VM virtualizes SCSI and Ethernet devices only.)

You must purchase licenses for any software you run in 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 HP's Virtualization Licensing program. For more information, contact your HP representative.

You can install the VM Host on a system that is running HP-UX 11i v2 May 2005 and 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.

3 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:

- "Installation Requirements" describes the system requirements for the running the software and the names of the software bundles that are required.
- "Installation Procedure" describes the procedure for installing the Integrity VM software.
- "Installation Verification" describes how to make sure the Integrity VM software was installed properly.
- "Troubleshooting Installation Problems" describes how to solve problems that occur during the product installation.
- "Removing Integrity VM" describes how to remove the Integrity VM product.
- "Postinstallation Procedures" describes how to prepare the VM Host environment for the guests you will create in Chapter 4.

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 outlined in "System Requirements."

NOTE For updated information, read the Integrity VM Release Notes.

System Requirements

Table 3–1 describes the minimum configuration requirements for installing Integrity VM on the VM Host system.

Table 3-1 Requirements for Installing Integrity VM

Required Resource	Description
Computer	An Integrity server
Operating system	HP-UX 11i v2 May 2005 or later, running on an Integrity hardware platform, as well as any appropriate software patches. 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) • 1.5 times physical memory for swap space (for example, for 1 GB of RAM, swap space should be 1.5 GB) • Disk space for each guest operating system, including swap space • Disk space for the applications running on each guest
Memory	Sufficient physical memory (RAM), including the following: • 750 MB + 7.5% of additional memory • 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 guest memory for overhead To determine the total memory required for your guests, refer to "Planning Your Virtual Machines," in Chapter 2. For example, for a VM Host with 16 GB of memory, the memory requirements would be calculated as follows: • 2.1 GB for the VM Host (16 GB minus 750 MB) • 3 GB for an HP-UX guest (guest requires 2 GB, plus 1 GB operating system minimum) • 3.21 GB total guest requirement (3 GB + 7%) • 10.69 GB remaining for additional guests (16 GB — 2.1 GB — 3.21 GB = 10.69 GB)
Integrity VM software	The software bundle T2767AC. Refer to "Bundle Names" 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).

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 (includes VMGuestLib and VMKernel SW)
- VMProvider (optional — provides WBEM Provider and WBEM Services)
- PRM-Sw-Krn (required — installed automatically if necessary)

If you intend to use the HP Integrity VM Manager to manage the VM Host, install the VMProvider bundle.

Installation Procedure

Once you have read the product Release Notes and verified that you have met the proper system requirements as described in "Installation Requirements," install the Integrity VM software as described in this section.

NOTE Installing the Integrity VM software requires the system to reboot.

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

1. Prepare the depot to be used for installing Integrity VM. (For information about software depots, refer to the *Software Distributor Administration Guide*). This step is optional, but preparing a depot for the software bundles reduces the number of times the system must reboot during the installation process.
2. If you have the installation media, mount it.

   If you are installing from the network, identify the VM Host and pathname that corresponds to the SD depot that contains the T2767AC bundle (for example, `my.server.foo.com:/depot/path`).
3. 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.
4. After the installation completes, you can install the optional VMProvider bundle. To install VMProvider, enter the following command:

   ```
   # swinstall -x autoreboot=true -s my.server.foo.com:/depot/path VMProvider
   ```
5. Unmount and remove any installation media. The VM Host system automatically reboots, if necessary.

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. For a list of Integrity VM commands, see Appendix A.
- Table 3–2 lists the kernel parameters that are modified.

Table 3-2 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

Installation Verification

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

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

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

Troubleshooting Installation Problems

If the installation verification fails, report the problem using the procedures described in Appendix B.

Problem:

One or more of the following messages are displayed:

```
could not write monParams: Device is busy

hpvmnet * already exists

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

Solution:

You can ignore these messages.

Problem:

The Integrity VM log file `/var/opt/hpvm/common/hpvm_mon_log` is too small.

Solution:

This log file is limited to 1024 KB in size. 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 larger than 1024 KB, include the following line in the `/etc/rc.config.d/hpvmconf` file:

```
VMMSIZE=$size
```

The value of `$size` is the amount in KB (kilobytes). For example, `VMMSIZE=10420` sets the maximum size of the monitor log to 10420 KB. After you make this change to the `hpvmconf` file, enter the following command:

```
# kill -HUP '/var/run/hpvmmonlogd.pid'
```

Removing Integrity VM

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

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

To remove these bundles, enter the following commands:

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

These procedures do not remove guests. To remove guests, refer to "Removing Guests" in Chapter 4.

Postinstallation Procedures

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.
```

You can create guests now by using the `hpvmcreate` command, as described in "Creating Guests." First, however, the following virtual machine tasks should be performed:

- Creating virtual switches

  Even though you have not created the guests, it is a good idea to create the virtual switches associated with the virtual network devices that the guests require. If you specify a vswitch name that you have not created with the `hpvmnet` command, you receive warning messages about the undefined vswitch name when you create the guest.

- Restricting access to devices

  The VM Host needs dedicated disk space for its operating system. You can also restrict a network device for use by the VM Host only. Define restricted devices before you create guests to make sure that guests do not require access to a restricted device or a portion of a restricted device.

- Providing selective access to guest consoles

  You can set up a group or user account for guest administrators, who would be responsible for system administration on a specific guest. Specify a group or a user account for guest administration when you create or modify the guest.

The following sections describe these procedures.

Creating Virtual Switches

In order to provide network access for guests, you must create virtual network switches (vswitches) for them. This section describes how to create a vswitch and verify that it has started. (For information about how virtual switches work, see "Virtual Network Devices," in Chapter 2.)

To create virtual switches, use the `hpvmnet` command. The following is the basic format of a command that creates a virtual switch:

```
hpvmnet -c -S switchname -n 0
```

where:

- `-c` indicates the creation of a vswitch.
- `-S` *switchname* specifies the name of the virtual switch.
- `-n 0` associates the new vswitch with the physical network device `lan0`.

Table 3–3 describes some of the functions of the `hpvmnet` command.

Table 3-3 Options to the `hpvmnet` Command

Option	Function
`-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 0`	Associates a VM Host network device to the vswitch. To associate a vswitch to `PPA 0` (also called `lan0`), enter `-n PPA0`. (`PPA` is the physical point of attachment as displayed by the `lanscan` command.) You cannot associate a vswitch with a vPPA (vlan).
`-s`	Retrieves statistics.
`-S vswitch_name`	Specifies the name of the virtual switch. The vswitch name is limited to eight characters and must be unique on the VM Host.
`-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.

For a complete list of `hpvmnet` command options, refer to `hpvmnet(1M)`.

To display the existing vswitches, enter the `hpvmnet` with no arguments and options.

The following command creates a virtual switch called `clan1` associated with lan1. The `hpvmnet` command is used to display 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
lan0          2 Up      Shared           N/A            N/A
clan1         5 Down    Shared    lan1
#
```

Note that `localnet` is automatically created for the local network. For a guest to use the `localnet` device, it must be added to the guest's configuration.

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
lan0          2 Up      Shared           N/A            N/A
clan1         5 Up      Shared    lan1   0x00306e3977ab
```

Note that `clan1` is associated with the network interface with the MAC address `0x00306e3977ab`.

To delete a vswitch, first stop the vswitch and then enter the `hpvmnet` command with the `-d` option. For example:

```
# hpvmnet -S clan1 -d

hpvmnet: The vswitch is currently active
hpvmnet: Unable to continue

# hpvmnet -S clan1 -h
hpvmnet: Halt the vswitch 'clan0'? [n]: y

# hpvmnet -S clan1 -d
hpvmnet: Remove the vswitch 'clan0'? [n] y
```

```
# hpvmnet -v
Name     Number State    Mode       PPA    MAC Address    IP Address
======== ====== ======= ========= ====== ============== ===============
localnet      1 Up      Shared            N/A            N/A
lan0          2 Up      Shared            N/A            N/A
```

For information about allocating vswitches to guests, refer to "Creating Guests" in Chapter 4. For information about modifying virtual networks, refer to "Configuring Virtual Networks" in Chapter 5.

Restricting Devices to the VM Host

As stated in "Planning Your Virtual Machines," some devices should be restricted for use by the VM Host. The VM Host system boot device and swap disk, as well as any logical volumes used by the VM Host should be restricted from access by guests. Define restricted devices before you create guests in order to prevent accidental conflicts.

You can also restrict network devices to the VM Host. This provides a dedicated network connection for the VM Host system.

To restrict access to storage and network devices, use the `hpvmdevmgmt` command. For example, the following command restricts a logical volume:

```
# hpvmdevmgmt -a rdev:/dev/vg00/lvol8
```

You also use the `hpvmdevmgmt` command to display a list of the restricted devices. For example:

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

To list the devices available on the VM Host system, use the `ioscan` command.

The `hpvmdevmgmt` command is also useful for defining shared devices.

Providing Selective Access to Guest Consoles

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 on to the guest under their own user accounts and use the `hpvmconsole` command to perform system administration tasks on the guest virtual machine.

There are two types of console users: admin and oper. Use the `hpvmcreate`, `hpvmmodify`, and `hpvmclone` commands with the `-g` and `-u` options to assign admin and oper privileges. You cannot use the `su` command to change from one privilege level to another. Per-user checks are based on real 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 user to have access to the VM Host, you can restrict their use of the hpvmconsole command to guest console access only by creating a restricted account for that purpose, as follows:

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 eight characters. For example:

```
# useradd -d /var/opt/hpvm/guests/compass1 -c 'compass1 console' \
> -s /opt/hpvm/bin/hpvmconsole guest1
```

   In this example, the following options are used:

   - `-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 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 it 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

[compass] vMP>
```

The virtual console interface displays raw characters for the CL and CO command, 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 your console commands. Interactive users can clear the screen. This situation can be a problem, however, for noninteractive or scripted use of the console.

4 Creating and Booting Guests

After you install Integrity VM, you can begin to create guests. This chapter describes how to create and boot guests, and how to solve problems you might encounter in the process.

This chapter contains the following sections:

- "Creating Guests" describes how to use the `hpvmcreate` command to create the three example guests (`compass1`, `compass2`, and `compass3`) that are described in Chapter 2.
- "Booting Guests" describes how to boot guests by using the `hpvmstart` command (available to the VM Host administrator) and the `hpvmconsole` command (available to the guest administrator as well as the VM Host administrator).
- "Stopping Guests" describes how to stop a guest that is running.
- "Removing Guests" describes how to remove a guest. You must stop a running guest before you can remove it.
- "Troubleshooting Guest Creation Problems" describes how to solve some of the problems that can occur when you create and boot guests.

Creating Guests

To create guests, use the `hpvmcreate` command, specifying the guest characteristics using the command options shown in Table 4–1.

Table 4-1 Options to the `hpvmcreate` Command

Guest Characteristic	Command Option
Guest name	`-P` *vm_name*
Operating system	`-O` *os_type*
Virtual CPUs	`-c` *number_vcpus*
CPU entitlement	`-e` *percent* `-E` *cycles*
Memory	`-r` *amount*
Virtual network devices	`-a` *rsrc*
MAC address	`-a` *rsrc*
Boot path	`-a` *rsrc*
Startup behavior	`-B` *start_attr*
Admin account name	`-u` *usergroup*:[*kind*] `-g` *group*:[*kind*]
Application disks	`-a` *rsrc*

Note that these items of information are almost equivalent to the information in the guest planning worksheet provided in "Allocating Resources to Guests" in Chapter 2. You can use the information you recorded in your guest planning chart to create each guest.

Specifying Virtual Devices

To allocate virtual network devices and virtual storage devices to guests, you use the `-a` option with the *rsrc* argument. The *rsrc* argument supplies all the information that the VM Host needs to allocate the device to the guest.

Use the `-a` option to the `hpvmcreate` command to specify both virtual storage devices and virtual network devices. If the required resources are not available on the current VM Host system, the guest is created and warning messages describe the resources that are not available. This allows you to create guests for future system configurations.

For storage devices, enter the resource specification in the following format:

`device-type``:`*`adapter-type`*`:[`*`hardware-address`*`]:`*`storage-type`*`:`*`device`*

The device information contains the guest virtual device information (*device-type*:*adapter-type*:[*hardware-address*]) and the physical device information (*storage-type*:*device*), separated by a colon (:).

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

- *device-type* (virtual device type): `disk` or `dvd`
- *adapter-type* (virtual device adapter type): `scsi`
- [*hardware address*] (optional). If you do not specify the virtual device hardware address, it will be generated for you (recommended). If specified, the hardware address is formatted as *bus*:*device*:*target*.
  - PCI bus number (*bus*)
  - PCI slot number (*device*)
  - SCSI target number (*target*)
- *storage-type*:*device*

  Table 4–2 lists the physical storage types and associated device specifications. You can specify the following storage types

Table 4-2 Physical Storage Types

Storage Type	Device	Example
`disk`	Disk or DVD	`/dev/rdsk/c4t3d2`
`lv`	LVM or VxVM character logical device file	`/dev/vg01/rlv0l2`
`file`	Locally-mounted, nonHFS file	`/guestfiles/diskfile`
`null`	VxFS directory containing ISO files	`/docs`

For examples of using the `hpvmcreate` command to create guests and allocate various virtual devices, see "Example Guest Creation."

You can also use the *rsrc* resource specification to associate a guest virtual network device with a virtual network switch (vswitch). 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: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 (recommended). 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)

Cloning Guests

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

Example Guest Creation

The example guests described in Chapter 2 can be created as described in this section. To create the first guest in the example configuration (`compass1`), enter the following command:

```
# hpvmcreate -Pcompass1 -c1 -r1G \
-a network:lan::vswitch:clan1\
-a network:lan::vswitch:clan2\
-a network:lan:0,1:vswitch:localnet \
-a disk:scsi::lv:/dev/vg01/rlv022\
-a dvd:scsi:0,0,1:null:/dev
#
```

The guest `compass1` has one virtual CPU, 1 GB of memory, and 5% entitlement (the default). It has network access using both vswitches `clan1` and `clan2`, and accesses the logical volume named `rlv022`.

To create the second guest (`compass2`), enter the following command:

```
# hpvmcreate -Pcompass2 -c2 -r3G -e50 -B manual\
 -a disk:scsi:1,0,2:disk:/dev/vg01/rlv023
#
```

The guest `compass2` has one virtual CPU, 3 GB of memory, and 50% entitlement. It is allocated access to the disk device associated with `/dev/vg01/rlv023`. Network access will be through the local network only.

To create the third guest (`compass3`), enter the following command:

```
# hpvmcreate -Pcompass3 -c1 -r2G \
-a disk:scsi::disk:/dev/vg01/rlv023 \
-a network:lan::vswitch:clan1
HPVM guest compass3 configuration problems:
    Warning 1: Insufficient free memory for guest.
    These problems may prevent HPVM guest compass3 from booting.
hpvmcreate: The creation process is continuing.
#
```

Note that `compass3` was created even though memory resources are insufficient to run this guest on the current VM Host. You can modify the characteristic of a guest by using the `hpvmmodify` command, as described in "Changing Guest Configurations" in Chapter 5.

If problems exist on the VM Host system when you boot the guest, the guest is not allowed to boot.

You can view information about the guest configuration using the `hpvmstatus` command:

```
# hpvmstatus
[Virtual Machines]
Virtual Machine Name VM #  OS Type State    # vCPUs # Devs # Nets Memory
==================== ===== ======= ======== ======= ====== ====== ===========
compass1                 15 HPUX    Off            1      2      2        1 GB
compass2                 16 HPUX    Off            2      1      0        3 GB
compass3                 17 HPUX    Off            1      1      1        2 GB

#
```

Booting Guests

You can boot the guest in either of the following ways:

- The `hpvmstart` command
- The `hpvmconsole` command

If you attempt to boot a guest on a VM Host system that does not have the required resources, the guest will not be booted and error messages will describes the specific resource problems.

To boot the guest the first time, you have to specify the installation media. To install from an Ignite server, set up the server with the guest VM Host name, MAC (obtained from `hpvmstatus`) and IP address, as you would for a physical system. Set up a LAN boot device from the guest console as you would for a physical system.

To install using a physical DVD device, you must first set up a virtual DVD. Follow these steps:

1. Verify the VM Host physical DVD drive location by checking the `ioscan` output:

   ```
   # ioscan -funC disk
   ```

2. Use the character device path of the VM Host's physical DVD to set a virtual DVD entry:

   ```
   # hpvmmodify -P compass1 -a dvd:scsi::disk:/dev/rdsk/c0t0d0
   ```

To start the guest from the VM Host administrator account, enter the `hpvmstart` command. For example, to start the guest called `compass1`, enter the following command:

```
# hpvmstart -Pcompass1
(C) Copyright 2000 - 2005 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 1
Allocating guest memory: 64MB
  allocating low RAM (0-4000000, 64MB)
/opt/hpvm/lbin/hpvmapp (/var/opt/hpvm/uuids/ce17ee10-3131-11da-9845-00306e39f70b
/vmm_config.current): Allocated 67108864 bytes at 00
  allocating firmware RAM (ffaa0000-ffab5000, 84KB)
/opt/hpvm/lbin/hpvmapp (/var/opt/hpvm/uuids/ce17ee10-3131-11da-9845-00306e39f70b
/vmm_config.current): Allocated 86016 bytes at 0x600
Loading boot image
Image initial IP=102000 GP=5F4000
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
==================== ===== ======= ======== ======= ====== ====== ===========
compass1                15 HPUX    On             1      2      2        1 GB
compass2                16 HPUX    Off            2      1      0        3 GB
compass3                17 HPUX    Off            1      1      1        2 GB
#
```

After you specify the installation media, OS installation continues normally, independent of the type of installation media.

To boot the guest from the guest console, enter the following command to turn on the virtual machine:

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

Press any key to interrupt the boot sequence.

```
EFI Boot Manager ver 1.10 [14.62]

Please select a boot option

    Acpi(PNP0A03,0)/Pci(3|1)/Ata(Primary,Slave)
    HP-UX Primary Boot: 0/16/1/3/1.0.0.0
    EFI Shell [Built-in]
    Boot option maintenance menu
Use ^ and v to change options. Press Enter to select an option.
```

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
```

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
```

Exit the EFI Boot Maintenance Management screen to return to the EFI Boot Manager screen. Boot from the selected entry.

When basic installation setup is complete, the software is copied from the distribution media to the guest's disk. Then the operating sytem eboots. If this reboot fails, restart it by booting from the master disk, as follows:

1. Enter the EFI shell.
2. Enter `fs0:`

```
Shell> fs0:
```

3. Enter `hpux`:

```
fs0\> hpux
```

The guest will boot from master disk (fs0:).

If the autoboot fails, use the EFI to select the correct boot option. Otherwise, you will have to boot from the master disk whenever you reboot.

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:

```
# hpvmmodify -P compass1 -d dvd:scsi:0,0,0
```

3. Reboot the guest, if necessary, by entering the following command:

```
# hpvmconsole -P $compass1 -c 'pc -on'
```

Allow the system postinstallation configuration to complete. The guest reboots automatically when the configuration process is done.

Installing a Guest Software Depot

The Integrity VM product also contains guest software that you can install on a virtual machine after you have installed its HP-UX operating system. Installing this depot improves the performance of the guest's operating system and provides the `hpvminfo` and `hpvmcollect` commands to guest administrators. After you install Integrity VM, the media is in the form of an SD (Software Distributor) distribution tape file located in `/opt/hpvm/guest-images/hpux/hpvm_guest_depot.sd`. This file can be used to create a general-purpose SD depot on the system you specify. For example, to create the guest depot on `myhost.corporate.com:/your/depot`, enter the following commands:

1. On the VM Host, create the general-purpose depot:

```
# swcopy -s /opt/hpvm/guest-images/hpux/hpvm_guest_depot.sd \* /
> @myhost.corporate.com:/your/depot
```

2. Verify that the software depot has been copied correctly:

```
 # swverify -d \* @@myhost.corporate.com:/your/depot
```

3. On the VM Host, register the new depot on *depot-host*:

```
# swreg -l depot /your/depot
```

4. On the guest, install the HPVM-Guest bundle from the new depot:

```
# swinstall -s /your/depot
```

In addition to the HPVM-Guest bundle, you may also choose to install the VMProvider bundle on the guest. This bundle, which is included in the Integrity VM depot, enables guest management using `vmmgr`, a GUI-based component of the Virtual Server Environment (VSE).

Be sure to check the product release notes for any software updates that you should also install on your guests.

Stopping Guests

You can use either the `hpvmstop` command or the `hpvmconsole` command. To stop a guest gracefully, use the `hpvmstop -g` command. You are required to confirm this command. For example:

```
# hpvmstop -g -Pcompass1
hpvmstop: Stop the virtual machine 'compass1'? [n]: y
#
```

Removing Guests

To remove a guest from the configuration, use the `hpvmremove` command. This command requires you to confirm this action. For example:

```
# hpvmremove -Pcompass1
hpvmremove: Remove the virtual machine 'compass1'? [n]: y
# hpvmstatus
[Virtual Machines]
Virtual Machine Name VM #  OS Type State    # vCPUs # Devs # Nets Memory
==================== ===== ======= ======== ======= ====== ====== ===========
compass2                16 HPUX    Off             2      1      0         3 GB
compass3                17 HPUX    Off             1      1      1         2 GB
#
```

This command removes the guest `compass1` and all its configuration files, and restores any resources allocated to that guest to the VM Host's pool of available resources. This does not affect the data and storage used by the guest's users and applications.

Troubleshooting Guest Creation Problems

When the guest boots, the VM Host performs a dynamic resource calculation to make sure there are enough resources to run the guest. If problems occur, one or more of the following messages is displayed:

```
HPVM guest badguest configuration problems:
    Warning 1: Guest's vcpus exceeds server's physical cpus.
    Warning 2: Guest's vcpus exceeds the supported maximum, 4 vcpus.
    Warning 3: Insufficient free memory for guest.
    Warning 4: Insufficient swap resource for guest.
    Warning 5: Insufficient cpu resource for guest.
    Warning 6 on item /dev/rdsk/c2t1d0: Device file '/dev/rdsk/c2t1d0' in
use by another guest.
    Warning 7 on item /dev/vg00/rhostswap: Device file '/dev/vg00/rhostswap'
in use by server.
    Warning 8 on item /dev/rdsk/foo: '/dev/rdsk/foo' backing device does not
exist.
    Warning 9 on item hostnet: Guest MAC address for switch hostnet is in
use.
    Warning 10 on item offnet: Vswitch offnet is not active.
    Warning 11 on item badnet: Vswitch badnet does not exist.
These problems may prevent HPVM guest badguest from booting.
hpvmstart: Unable to continue.
```

These warnings allow the guest to be created, but not booted.

The following error messages prevent the guest from being created or booted:

```
hpvmcreate: ERROR (badguest): Duplicate backing device  '/dev/rdsk/c2t1d0'.
hpvmcreate: ERROR (badguest): Illegal blk device '/dev/dsk/c2t1d0s5' as
backing device
```

You can modify the guest configuration by using the `hpvmmodify` command, as described in "Changing Guest Configurations."

Review the commands you used to create the virtual switches and guests, comparing your input with the messages and the `hpvmstatus` display to determine whether errors were made in the process of creating virtual switches and guests.

To view the commands that have been entered on the VM Host, look at the following file:

```
/var/opt/hpvm/common/command.log
```

You can ignore the following messages in the guest log file `/var/opt/hpvm/guests/`*`guestname`*:

```
LockInit: redundant call for living lock!

Returning QUEUE_FULL to dev = …
```

Problem:

If a guest appears to hang (that is, there is no response to the `ping` command, or the `hpvmconsole` freezes), enter `^B` (Ctrl/B) in the `hpvmconsole` session. This action returns you to the virtual machine console. Enter the `CM` command, then the `TC` command to send an `INIT` command to the guest OS. Do not reboot the guest.

On the VM Host, enter the `hpvmcollect` without the `-c` option. This command collects VM Host and Integrity VM information that is useful in analyzing the problem. Refer to "Reporting Problems with Integrity VM" in Appendix B for more information.

Problem:

The guest hangs in the EFI shell when you are starting it, and you get the following message:

```
Shell> \efi\hpux\hpux
'\efi\hpux\hpux' not found
Exit status code: Invalid Parameter
```

Solution:

The EFI boot parameters were probably not set up correctly during guest OS installation. Choose the correct EFI partition to boot. For example:

```
Shell> fs3:
fs3:\> hpux
```

Problem:

The `hpvmconsole` command hangs when you are starting a guest.

Solution:

Check the guest log file in `/var/opt/hpvm/guests/`*`guestname`*`/log`. If there is a user configuration problem, make changes to the guest configuration and restart. Otherwise, enter the `hpvmcollect` command, and report the problem through your support channel.

5 Managing Guests

When you have set up your virtual machines and your guests are running, it may be necessary to modify the configuration. This chapter contains the following sections:

- "Configuring Virtual Networks" describes how to modify the guest virtual network devices.
- "Monitoring Guests" describes how to monitor guests while they are running.
- "Making Devices Shareable" describes how to make a storage device (a DVD) shareable and how to make a vswitch nonshareable.
- "Changing Guest Configurations" describes how to modify the characteristics of guests.
- "Troubleshooting Guests" describes how to solve problems that occur while the virtual machine configuration is running.
- "Guest Log Files" describes the log files that record information about guest activity.

Configuring Virtual Networks

To get a list of vswitches currently defined for the VM Host, enter the `hpvmnet` command. For example:

```
# hpvmnet
Name     Number State   Mode      PPA    MAC Address    IP Address
======== ====== ======= ========= ====== ============== ===============
localnet      1 Up      Shared           N/A            N/A
lan0          2 Up      Shared           N/A            N/A
compnet       4 Down    Shared    lan0                  16.116.14.205
clan1         5 Up      Shared    lan1   0x00306e3977ab
clan2         6 Down    Shared    lan2
```

The guest configuration file `/var/opt/hpvm/guests/`*`guestname`*`/vmm_config.current` contains an entry for each guest virtual network device. For example:

```
.
.
.
# Virtual Network Devices
#
lan(0,0).0x00306E39F70B = switch(clan1)
.
.
.
```

NOTE When you are looking at the guest configuration file, remember that the left side of the equals sign (=) is for the guest and the right side is for the VM Host.

When the guest is booted (through `hpvmstart` or `hpvmconsole`), the guest LAN is configured as specified in the LAN entry in the guest configuration file. For example:

compass1	
`lan(0,0)`	Bus 0 and device number 0 is the guest LAN hardware path.
`0x00306E39F70B`	Guest virtual MAC address.
`switch(clan1)`	The vswitch name is `clan1`.

Entering the `lanscan` command on the guest `compass1` results in the following:

```
compass1# 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
```

Note that the hardware path from the output of `lanscan` on the guest matches what was specified in the guest configuration file. The `Station Address` in the `lanscan` output also matches the guest virtual MAC address in the guest configuration file.

Recreating a Virtual Switch

You do not need to shut down and reboot the guest if you accidentally delete its vswitch (for example, if you use the following command):

```
# hpvmnet -d -i clan1
```

The Integrity VM network stack automatically determines that the guest vswitch has disappeared. Once the guest's associated vswitch is re-created, the guest network is alive again. The following console reconnect message appears on the guest:

```
compass1# vswitch reconnect = e0000001398624c0
vswitch reconnect = e0000001398624c0...
```

The message repeats until the guest vswitch is re-created and reconnected.

To change the vswitch to use another physical NIC on the VM Host (for example, to change from `lan0` to `lan1`), delete the vswitch that was associated with `lan0`. Create another vswitch with the same name or id and specify `-n 1`. Your guest network should be back in a few seconds.

Redefining a Virtual Switch

Modifying the `bus/dev` IDs of a LAN entry in the guest configuration file has the same effect as moving a nework adapter from one hardware slot to another on a nonvirtual machine. Similar to other 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` accordingly:

```
# 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 this 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 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:

```
# /sbin/rc2.d/S340net stop
# ch_rc -a -p "INTERFACE_NAME[0] = "lan3"
# /sbin/rc2.d/S340net start
```

Your guest network is functioning again.

You must restart a vswitch after the following events:

- The MAC address is changed (either by swapping the network adapter associated with the vswitch or associating the vswitch with a different network adapter.
- The IP address associated with the network adapter associated with the vswitch is changed.
- 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.

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.

Monitoring Guests

To see detailed information about guests, enter the `hpvmstatus -V` command. For example:

```
# hpvmstatus -V -Pcompass1
[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

#
```

Making Devices Shareable

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.

Virtual network devices are assumed to be shareable devices. However, the physical network devices backing them are not shareable.

You can make a virtual DVD device shareable by entering the following command:

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

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 re-create the guest. Specifically:

- 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 re-create it if you modify the disk containing the guest's root device.

To restrict a virtual network device so that it is no longer shareable, enter the following command:

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

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

Do not make read/write disks shareable.

Changing Guest Configurations

To modify the resources of a guest, use the `hpvmmodify` command. For example, guest `compass1` cannot start. The following message is generated:

```
# hpvmstart -Pcompass1
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.
#
```

To modify the configuration of the problematic guest `compass1` to remove virtual CPUs and memory, enter the following command:

```
# hpvmmodify -Pcompass3 -c1 -r1GB
# hpvmstart -Pcompass1
(C) Copyright 2000 - 2005 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 1
Allocating guest memory: 1024MB
  allocating low RAM (0-40000000, 1024MB)
/opt/hpvm/lbin/hpvmapp (/var/opt/hpvm/uuids/17e4af4c-34fc-11da-94e3
-00306e39f700
  allocating firmware RAM (ffaa0000-ffab5000, 84KB)
/opt/hpvm/lbin/hpvmapp (/var/opt/hpvm/uuids/17e4af4c-34fc-11da-94e3-
00306e39f700
Loading boot image
Image initial IP=102000 GP=5F4000
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
==================== ===== ======= ======== ======= ====== ====== ===========
compass1                15 HPUX    Off             1      0      1        1 GB
compass2                16 HPUX    Off             2      1      0        5 GB
compass3                17 HPUX    On              1      1      0        1 GB
#
```

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:

- `vmm_config.current` contains the current guest configuration currently set.
- `vmm_config.prev` contains the last known guest configuration settings.
- `vmm_config.next` contains the configuration settings that have changed since the guest was started. To initiate these changes, you must reboot the guest.

Troubleshooting Guests

To turn off a guest gracefully, follow the same procedure as if you were working with real hardware. If the guest hangs, cold reboot as you would real hardware

1. At the guest console, enter `^B` to access the virtual machine console.
2. Enter the `CM`.
3. Enter `Y`.

Then boot the guest.

The following are some common problems that can occur on guests, and how to solve them.

Problem:

Error messages regarding entitlement, such as the following:

```
unable to set the entitlement
```

Solution:

These messages usually occur because the resource allocation agent for Integrity VM is not installed or running. Refer to "Verifying the Installation of Integrity VM" in Chapter 3.

Problem:

The `hpvmstart` or `hpvmconsole` command exits, aborts, or hangs.

Solution:

Report the problem through your support channel. Refer to "Reporting Problems with Integrity VM" in Appendix B for more information.

Problem:

Your guest hangs or crashes the following message on the console:

```
"Guest punishment:""VMM panic:""Assertion failed"
```

Solution:

Report this problem through your support channel. Refer to "Reporting Problems with Integrity VM" in Appendix B for more information.

Problem:

The following message is displayed on the VM Host:

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

Solution:

- Verify that the path name to the file-backing store is correct and that the physical storage device is mounted.
- Verify that the size of the physical storage device is divisible by 512 bytes (for a disk device) or 2048 (for a DVD device).
- Modify the guest configuration using the `hpvmmodify` command, as described in "Changing Guest Configurations" in Chapter 5.

Guest Log Files

Each guest has a log file named `/var/opt/hpvm/guests/`*`guestname`*`/log`. To check all the guest logs, look at `/var/opt/hpvm/uuids/guests/*/log`.

The guest log file can grow very large, so you must periodically rotate this log. From the console, enter the following command:

```
# rec -rotate
```

Save or delete the old log files as necessary.

The command log file is stored as `/var/opt/hpvm/common/command.log`. This log file can also grow very large. To refresh it, copy the existing log file elsewhere. A new log file is automatically generated. Save or delete the old log file as necessary.

Appendix A Integrity VM Commands

Table 7–1 lists the Integrity VM commands and the sections of this manual in which they are described. Use Integrity VM commands to perform common VM management tasks such as starting a virtual machine or collecting configuration information on guest machines. Except for the `vmconsole` command, these commands are for use by the VM Host administrator only.

For detailed information about the Integrity VM commands, including description, syntax, and command-line options, see the Integrity VM manpages, which are installed in the `/opt/hpvm/man/man1m` directory.

Table A-1 Integrity VM Commands

Command	Function	For More Information
`hpvmstart`	Start a virtual machine.	"Booting Guests"
`hpvmclone`	Create a cloned copy of a virtual machine.	"Cloning Guests"
`hpvmcollect`	Collect crash dumps, logs, system status, and configuration information on VM Host and guest machines for problem analysis.	"Reporting Problems with Integrity VM"
`hpvmconsole`	Connect to the console of a virtual machine.	"Providing Selective Access to Guest Consoles"
`hpvmcreate`	Create a new virtual machine.	"Creating Guests"
`hpvmdevmgmt`	Manage the device database.	"Restricting Devices to the VM Host"
`hpvminfo`	Display information about the VM Host.	"Installation Verification"
`hpvmmodify`	Rename or modify the attributes of a virtual machine. (Certain modifications require a guest reboot.)	"Changing Guest Configurations"
`hpvmnet`	Configure virtual network devices.	"Creating Virtual Switches"
`hpvmremove`	Remove a virtual machine.	"Removing Guests"
`hpvmstop`	Stop a virtual machine.	"Stopping Guests"
`hpvmstatus`	Display status of one or more virtual machines.	"Monitoring Guests"

NOTE All commands except `hpvmconsole` require superuser privileges.

To use the `hpvminfo` and `hpvmcollect` commands on the guest, you must install the guest kit on the guest as described in "Installing a Guest Depot" in Chapter 3.

Appendix B 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 with the environment when the VM Host crashed.

   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.

If the VM Host hangs, make sure a crash dump is generated by using `TC` 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-related logs, and configuration files for later analysis.

Because the `hpvmcollect` command collects generic Integrity VM and HP-UX operating system and system information, it might not collect all the information needed to analyze the source of the problem. Make sure that all the relevant information is included in the collection. For example, if the 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 -Pcompass1
```

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 has crashed, 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 -Pcompass1

  HPVM host crash/log collection tool version 0.8
  Gathering info for post-mortem analysis of guest 'compass1' 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 messages from vmm .......................................... OK
  Collecting lv info .................................................... N/A
  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 ......................................... NA
  Copying guest's console log file ....................................... OK
  Copying hpvm configuration ............................................. OK
  Coping 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
  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
  "/var/opt/hpvm/common/hpvmcollect_archive/compass1_Oct.04.05_165043EDT.tar.gz"
```

If you get an error message such as 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, it is not necessary to specify a 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 install the guest kit on the guest as described in "Installing a Guest Depot" in Chapter 3.

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
```

Integrity Virtual Machines Glossary

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 Itanium®-based systems.

CPU Central processing unit. A single processor core, either physical or virtual.

EFI Extensible Firmware Interface (the boot firmware for all Integrity systems).

entitlement The minimum percentage of CPU processing power that a guest can use.

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 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.

host administrator The system administrator. This level of privilege provides control of the VM Host system and its resources, as well as creating and management of guests.

Ignite-UX The HP-UX Ignite server product, used as a core build image to create or reload HP-UX servers.

Integrity VM The HP Integrity Virtual Machines product.

ISSE HP Instant Support Enterprise Edition

localnet The local network created by Integrity VM for internal, local communications. Guests can communicate on the localnet, but the VM Host cannot.

LUN Logical Unit Number.

NIC Network Interface Card. Also called *network adapter*.

PMAN Platform Manager. See *VM Host*.

virtual machine Virtual hardware system. Also called *VM*.

virtual machine console The user-mode application that provides console emulation for virtual machines. Each instance of the virtual machine console is one console session for its associated virtual machine. Also called *VM console*.

Virtual Machine Manager (VMM) The management application responsible for managing and configuring HP Integrity Virtual Machines. Also called *VM Manager*.

VM See *Virtual machine*.

VM Host The virtual machine host system.

VMM See *Virtual Machine Manager*

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.

Index