Chapter 18

Integrity Virtual Machines (HPVM)

HP-UX Handbook
Revision 13.00

TERMS OF USE AND LEGAL RESTRICTIONS FOR THE HP-UX RECOVERY HANDBOOK

ATTENTION: PLEASE READ THESE TERMS CAREFULLY BEFORE USING THE HP-UX HANDBOOK. USING THESE MATERIALS INDICATES THAT YOU ACCEPT THESE TERMS. IF YOU DO NOT ACCEPT THESE TERMS, DO NOT USE THE HP-UX HANDBOOK.

THE HP-UX HANDBOOK HAS BEEN COMPILED FROM THE NOTES OF HP ENGINEERS AND CONTAINS HP CONFIDENTIAL INFORMATION.

THE HP-UX HANDBOOK IS NOT A PRODUCT QUALITY DOCUMENT AND IS NOT NECESSARILY MAINTAINED OR UP TO DATE. THE HP-UX HANDBOOK IS HERE MADE AVAILABLE TO HP CONTRACT CUSTOMERS FOR THEIR INTERNAL USE ONLY AND ON THE CONDITION THAT NEITHER THE HP-UX HANDBOOK NOR ANY OF THE MATERIALS IT CONTAINS IS PASSED ON TO ANY THIRD PARTY.

Use of the HP-UX Handbook: Hewlett-Packard Company ("HP") authorizes you to view and download the HP-UX Handbook only for internal use by you, a valued HP Contract Customer, provided that you retain all copyright and other proprietary notices contained in the original materials on any copies of the materials. You may not modify the HP-UX Handbook in any way or publicly display, perform, or distribute or otherwise use them for any public or purpose outside your own business. The materials comprising the HP-UX Handbook are copyrighted and any unauthorized use of these materials may violate copyright, trademark, and other laws. If you breach any of these Terms, your authorization to use the HP-UX Handbook automatically terminates and you must immediately destroy any downloaded or printed materials.

Links To Other Web Sites: Links to third party Web sites provided by the HP-UX Handbook are provided solely as a convenience to you. If you use these links, you will leave this Site. HP has not reviewed all of these third party sites and does not control and is not responsible for any of these sites or their content. Thus, HP does not endorse or make any representations about them, or any information, software or other products or materials found there, or any results that may be obtained from using them. If you decide to access any of the third party sites linked to this Site, you do this entirely at your own risk.

Disclaimer: THE HP-UX HANDBOOK IS PROVIDED "AS IS" WITHOUT ANY WARRANTIES OF ANY KIND INCLUDING WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT OF INTELLECTUAL PROPERTY. HP further does not warrant the accuracy and completeness of the materials in the HP-UX Handbook. HP may make changes to the HP-UX Handbook at any time without notice. The HP-UX Handbook may be out of date, and HP makes no commitment to update the HP-UX Handbook. Information in the HP-UX Handbook may refer to products, programs or services that are not available in your country. Consult your local HP business contact for information regarding the products, programs and services that may be available to you.

Limitation of Liability: IN NO EVENT WILL HP, ITS SUPPLIERS, OR OTHER ANY THIRD PARTIES MENTIONED IN THE HP-UX HANDBOOK BE LIABLE FOR ANY DAMAGES WHATSOEVER (INCLUDING, WITHOUT LIMITATION, THOSE RESULTING FROM LOST PROFITS, LOST DATA OR BUSINESS INTERRUPTION) ARISING OUT OF THE USE, INABILITY TO USE, OR THE RESULTS OF USE OF THE HP-UX HANDBOOK, WHETHER BASED ON WARRANTY, CONTRACT, TORT OR ANY OTHER LEGAL THEORY AND WHETHER OR NOT ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. THIS DOES NOT APPLY IN CASE OF INTENT OR IF LIABILITY IS LEGALLY STIPULATED. IF YOUR USE OF THE HP-UX HANDBOOK RESULTS IN THE NEED FOR SERVICING, REPAIR OR CORRECTION OF EQUIPMENT OR DATA, YOU ASSUME ALL COSTS THEREOF.

The information contained herein is subject to change without notice. The only warranties for HP products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. HP shall not be liable for technical or editorial errors or omissions contained herein.

Applicable Laws: These Terms will be governed by and construed in accordance with the laws of the State of California, without giving effect to any principles of conflicts of laws.

General: HP may revise these Terms at any time by updating this posting. ***Revised Oct 2013***

FEEDBACK or QUESTIONS: please email essam.ackleh@hp.com
(please use subject syntax: *HP-UX Handbook v13.00 Chapter <YY> - <Feedback Title>*

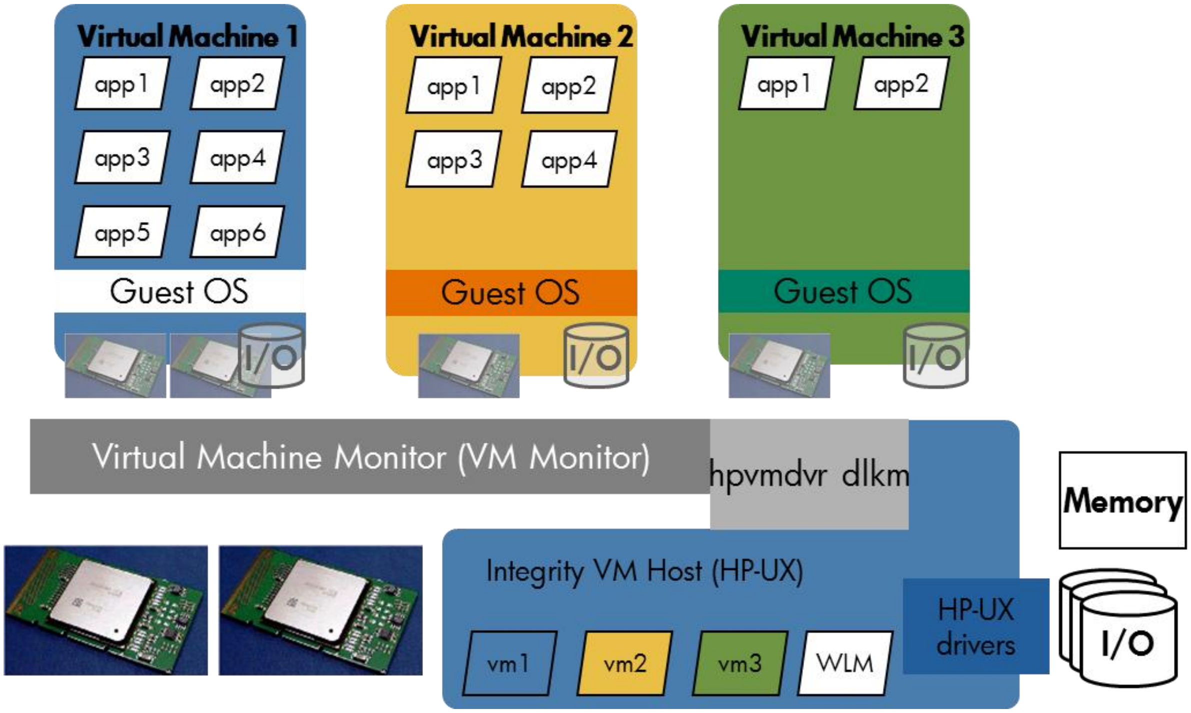
TABLE OF CONTENTS

Introduction

This chapter provides an overview of the Integrity Virtual Machine product. Integrity Virtual Machine is not included with the HP-UX Operating System. With this product it is possible to run multiple instances of HP-UX on one node. The detailed product information can be found at The Business Support Center(BCS): HP-UX Virtual Partitions (vPars) and Integrity Virtual Machines (VM).

- HP Integrity VM Installation, Configuration & Administration
- HP Integrity VM Release Notes
- White Papers

The HP Integrity Virtual Machines product is a robust soft partitioning and virtualization technology that provides operating systems isolation, with sub-CPU allocation granularity and shared I/O. Put simply, Integrity VM enables you to create a virtual machine – a software abstraction that presents all of the interfaces provided by a computer system's hardware. Integrity VM software enables virtual devices by emulating them with real hardware devices. A single HP Integrity server running Integrity VM can support multiple virtual machines, each with its own separate "guest" operating system. As a result, each virtual machine (VM) can host its own applications in a fully isolated environment. The physical resources of the Integrity server are shared amongst any of the virtual machines it hosts.



Integrity Virtual Machines is a soft partitioning and virtualization technology that provides operating system isolation, with sub-CPU allocation granularity and shared I/O.

I/O Resources

Networking

For the guest to communicate outside the VM Host system, each guest virtual network must be associated with a virtual switch (vswitch). 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 (APA is supported), or it can be local to the virtual machines on the VM Host, providing a local network between guests (`hpvmcreate` / `hpvmmodify`).

You create a virtual switch using the `hpvmnet` command. 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.

Integrity VM virtualizes the Intel Gigabit Ethernet card. The guest OS configures its own virtual network interface with an IP address using standard commands and utilities. It can also use DHCP.

Storage

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

Integrity VM supports the following virtual disk types:

Virtual Disk: emulated SCSI disk whose media comes from a VM Host disk LUN.

Virtual PartDisk: emulated SCSI disk whose media comes from a VM Host disk partition (`idisk`) (NOTE: Usage as been deprecated as of HPVM release A.03.05)

Virtual LvDisk: emulated SCSI disk whose media comes from a VM Host logical volume.

Virtual FileDisk: emulated SCSI disk whose media comes from a VM Host file (`hpvmdevmgmt`).

The following virtual DVD-ROM types are supported:

Virtual DVD: emulated SCSI DVD-ROM with virtual media comes from a disc inside of a CD/DVD drive on the VM Host.

Virtual FileDVD: emulated SCSI DVD-ROM with virtual media that comes from a VM Host ISO file.

Virtual NullDVD (empty): emulated SCSI DVD-ROM with no virtual media currently present. The next media selection may come from a VM Host CD/DVD drive or VM Host ISO file, depending on how the Virtual NullDVD is configured. Virtual DVD changes can be initiated from the virtual console (`vMP> IN` and `EJ` commands)

Attached Devices

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

The VM Host backup device types are tapes, media changers, and CD/DVD burners.

Integrity VM does not support HFS. NFS is supported but not recommended. Integrity VM virtualizes SCSI devices. Use `hpvmcreate` / `hpvmmodify` to assign storage devices to a virtual machine.

You can find more info about backing storage devices in the following white papers:

- *HP Integrity Virtual Machines Installation, Configuration and Administration Guide*
- *Best Practices for Integrity VM* (internal)
- *Best Practices for Using Files as Virtual Mass Storage* (internal)

AVIO

The Integrity VM V3.5 release introduced new accelerated storage and networking products to improve the overall I/O performance for Integrity VM. The new Accelerated Virtual I/O (AVIO) products provide up to a 60% reduction in service demand and as much as twofold improvement in throughput over the existing fully virtualized storage and networking Integrity VM solutions.

- What is AVIO?

  - A high performance Networking and Storage solution for Integrity VM
- Details
  - Storage and network AVIO device drivers on guests and VM Host
  - Avoids traversing SCSI, network stack twice (in guest and VM Host)
  - Shared I/O channel through the VM Monitor improves latency & BW
- Performance
  - Improves bandwidth by up to 2X compared to existing Virtual I/O
  - Reduces CPU usage by up to 50% compared to existing Virtual I/O
- Support
  - Co-exists with existing Virtual I/O
  - Supports virtual disks mapped to LUNs, whole disks, LVM logical volumes
  - HP-UX 11iv2 guest support in HPVM 3.5, Windows and 11iv3 in near future

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

```
# hpvmcreate -P aviotest -O hpux -a network:aviolan::vswitch:swlan1 \
-a disk:aviostor::disk/dev/rdsk/c1t2d0
# hpvmcreate -P aviotest -O hpux -a network:avio_lan::vswitch:swlan1 \
-a disk:avio_stor::disk/dev/rdsk/c1t2d0
```

For current AVIA patches and known issues, see WTEC's AVIO page

Supported Limits

See WTEC's HPVM Supported Limits page.

Installation

Host System Requirements

Required Resources	Description
System	Any Integrity Server
Operating System	Refer to the Integrity VM Release Notes or to the WTEC Product Version page for supported Host HP-UX and HPVM releases.
Disk Storage	Sufficient disk space for the following:

	- VM Host operating system - VM Host software (50MB) - Swap space size should be at least as large as physical memory plus 4GB. - 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: - 750MB + 7.5% of remaining memory - Physical memory required for each guest. - Additional 7% of guest memory for overhead.
Software	**T2767AC/T2767BC/T2767CC** – the integrity VM product **VMProvider** (optional) – software that allows you to use System Insight Manager (SIM) to manage virtual machines **VMMigrate** (optional) – software that allows you to move a virtual machine from one VM Host to another Patches: Refer to the Integrity VM Release Notes or to the WTEC Product Version page.
Restrictions	Integrity VM version 4.3 requires non HFS file systems. You must remove any entries before installing Integrity VM. Check for HFS entries with the following command: `# grep -i hfs /etc/fstab`

Guest Requirements

HP-UX Guest

Required Resources	**Description**
Operating System	Refer to the Integrity VM Release Notes or to the WTEC Product Version page for supported Guest HP-UX and HPVM releases.
Software	**VMProvider** (optional) – software that allows you to use System Insight Manager (SIM) to manage virtual machines **HPVM-Guest** (optional) - It is technically not necessary to have the bundle installed on guests but it's HIGHLY RECOMMENDED. The

	bundle includes HPVM tools such as hpvminfo and hpvmcollect, and a set of kernel tunes that improve I/O and network performance and behavior of the guest. Patches: Refer to the Integrity VM Release Notes or to the WTEC Product Version page.

Windows Guest *(Only supported Integrity VM versions A.03.05 thru A.04.02.05)*

Required Resources	**Description**
Operating System	HP Integrity Windows 2003 software media with Service Pack 1.
Software	**Integrity VM Windows guest management software** **VM Provider** (optional) – For management from VM Manager and HP VSE. It is available on the Windows Smart Setup Media.

Linux Guest (Only supported Integrity versions A.03.05 thru A.04.02.05)

Required Resources	**Description**
Operating System	Integrity VM A.02.05: SUSE SLES 10 Update 2 Red Hat Enterprise Edition Advanced Server Release 4 Update 5
Software	**Integrity VM Linux Guest Management Software** Located at: /opt/hpvm/guest-images/linux

Software Installation

Use the swinstall command to install Integrity VM and specify the path to the depot, e.g.:

```
# swinstall -x autoreboot=true -s myserver.foo.com:/depot_path T2767CC
```

Installation files:

- Integrity VM to /opt/hpvm/
- Integrity VM data files to /var/opt/hpvm/
- Integrity VM commands to /opt/hpvm/bin/

See Using Integrity Virtual Machine section to get info about how to configure virtual machines and switches.

For OS Installation on Guests see *HP Integrity Virtual Machines Installation, Configuration and Administration Guide.*

Software Upgrade

1. Back up the `/var/opt/hpvm` directory to retain existing Integrity VM configuration files.

2. Log in to each guest on the VM Host and gracefully shut down the operating system (it is preferable even when installation procedure will stop any running guests)

3. Use the swinstall command to install the software (T2767CC) VMProvider and VMMigrate bundles should also be installed or upgraded.

Configuration files

When you install Integrity VM, the file `/etc/rc.config.d/hpvmconf` is created to record the product configuration (enable virtual machine at boot).

On the other hand virtual switches and virtual machines configuration is under the `/var/opt/hpvm/` directory.

a) virtual switches (`/var/opt/hpvm/hpvmnet`)

There is one directory for any virtual switch. It contains:

.netid	*vswitch_number* (for using with -s option)
hpvmnet_config	virtual switch name and associated nic (if exists)
hpvmnet_status	`hpvmnetd` process number created for this switch

b) virtual machines (/var/opt/hpvm/guests & /var/opt/hpvm/uuids)

`/var/opt/hpvm/guests/`*<vmname>* - it is a symbolic link to the *<vm_uuid>* directory
`/var/opt/hpvm/uuids/`*<vm_uuid>* - contains information about configuration, logs...

.vmid	*vm_number* (for using with -p option)
FPL	virtual Forward Progress Log
SEL	virtual System Event Log
console/conslog	virtual machine console logs
log	virtual machine log
guest_status	`hpvmapp` process number created for this virtual machine and the hostname of the server where it is running
vmm_config.current	current virtual machine configuration

vmm_config.prev	previous virtual machine configuration
nvram	nvram
vm_dev	virtual machine device file (hpvmdvr driver)

c) common files (/var/opt/hpvm/common)

command.log	virtual machine commands used
guest_id	last *vm_number* assigned
hpvmnet_id	last *vswitch_number* assigned
hpvm_mon_log	virtual machine monitor log
hpvm_mgmtdb	virtual machine database, with information about global devices, restricted devices, device options,…

Network Installation / Recovery (HPUX Guests)

Setting up the Ignite-UX server

The Ignite-UX server is a system with the Ignite-UX software installed and a network connection to systems (clients) intended to use Ignite-UX. Integrity servers with HP-UX 11i v3 are recommended for Ignite-UX servers to be used with Integrity VMs. The Ignite-UX software bundle is available for all HP Integrity and PA-RISC servers. The steps outlined here are based on the *HP-UX Installation Utilities (Ignite-UX)*, bundle B5725AA. For the latest Ignite-UX installation information, please reference the *Installing and Configuring an Ignite-UX Server* section of the *Ignite-UX Administration Guide*.

The configuration files /etc/dhcptab or /etc/bootptab must be directly modified on the Ignite-UX server to prepare it for use with Integrity servers, including Integrity VMs. Which of these files require modification depends on whether IP addresses are assigned statically or the network provides DHCP services.

a) Setting up for a static IP environment

For Ignite-UX clients that have a static IP address assigned to them, you will need to modify the /etc/bootptab configuration file. A defaults entry simplifies the process of adding new client entries (including those for Integrity VMs). An example of a defaults entry and two host entries are shown below.

Entries in /etc/bootptab to accommodate Integrity VM clients

```
System-IPF:\
  tc=ignite-defaults:\
  bp=15.180.3.215:\
  sm=255.255.240.0:\
  gw=15.180.0.1:\
  vm=rfc1048:\
  dn=bpo.hp.com:\
         ds=16.6.64.51:

        vmguest1:tc=System-IPF:ip=15.180.3.217:ha=9258eeee1516:
        vmguest2:tc=System-IPF:ip=15.180.3.218:ha=42de2d02b285:
```

System-IPF specifies a set of default values and two for Ignite-UX clients (vmguest1 and vmguest2) that use those defaults. This particular default entry has four values that must be modified for your configuration:

- `bp` – The IP address of the Ignite-UX server to be used to respond to clients.
- `sm` – The subnet mask being used by the clients.
- `gw` – The network gateway address.
- `ds` – The domain name server address.

Typographical errors in any of these will cause no end of enigmatic problems. Check the entries carefully.

To add another entry for a new VM, you need to specify an entry similar to that of vmguest1 and vmguest2 above, where the first attribute is the host name of the new VM. The other three attributes are defined as follows:

- tc – The defaults to be used for this group of clients (in the example above, this will be System-IPF).
- ip – The (fixed) IP address of the client.
- ha – The client hardware (MAC) address.

The MAC address for a VM can be obtained from `hpvmstatus` or from the VM's console (the VM console will be discussed later). For example, the MAC address is the last field of its LAN entry displayed in the output of `hpvmstatus`:

```
vmhost:/>hpvmstatus -P vmguest1 | grep lan
vswitch   lan         vswa          0    0    0 92-58-ee-ee-15-16
```

Uncomment the following line on `/etc/inetd.conf`:

```
bootps       dgram  udp  wait   root /usr/lbin/bootpd   bootpd
```

and force inetd to reread the `/etc/inetd.conf`:

```
# inetd -c
```

b) Setting up a DHCP environment

If Ignite-UX clients do not have static IP addresses assigned to them, as in a DHCP environment, then host-specific entries in `/etc/bootptab` are not practical. To accommodate such *anonymous* clients, HP-UX 11i v3 provides an option for DHCP configuration that enables the device_pool_group feature to be used. To use this feature, set up an entry in the `/etc/dhcptab` configuration file.

Entry in `/etc/dhcptab` to facilitate Ignite-UX server use with anonymous clients

```
dhcp_device_group:\
re:\
ncid:\
class-id="PXEClient:Arch:00002:.*":\
lease-time=300:\
subnet-mask=255.255.248.0:\
addr-pool-start-address=75.99.87.6:\
addr-pool-last-address=75.99.87.253:\
bf=/opt/ignite/boot/nbp.efi
```

The `attributes` of the entry are described as follows:

- dhcp_device_group - This starts a DHCP device pool group that provides a pool of IP addresses to a set of clients that all have the same class-id in their boot messages.

- class-id - All IPF clients send boot messages that contain a class id with a 32-character string:

- PXEClient:Arch:00002:UNDI:xxxyyy. Where "xxxyyy" are major and minor numbers for the Universal Network Device Interface revision. This line tells bootpd to respond only to clients that match a string that starts with PXEClient:Arch:00002, that is, IPF clients

- re - This is a binary option that tells bootpd to perform a regular expression match of the class-id rather than a default literal match..

- ncid - This is a binary option that tells bootpd to not send back a class-id on the message responses. This option is necessary because bootpd does not support the full PXE protocol.

- lease-time - This option indicates the time in seconds for leases given for IP addresses.

- addr-pool-start-address – Identifies the start of a range of IP addresses offered to clients.

- addr-pool-last-address – Identifies the end of a range of IP addresses offered to clients.
- subnet-mask – The subnet mask used by clients.
- bf – The EFI network boot program to use.

After this entry is made in the `/etc/dhcptab` file on the Ignite-UX server, restart the `bootpd` daemon. It is typically started by `inetd` and you can verify this by examining the `/etc/inetd.conf` configuration file. If so, kill any instance of `bootpd` and `inetd` will restart `bootpd` for you.

Using depots

a) Create HP-UX 11i v3 depots from appropriate media

In this example we're using a CDROM device (`/dev/dsk/c0t3d0`) that is assumed to contain the HP-UX 11i v3 media (`/var/opt/ignite/depots/Rel_B.11.31/core`)

```
# make_depots -r B.11.31 -s /dev/dsk/c0t3d0
```

b) Add the HPMV application software for VM Hosts and Guests to the depot

Here's an example using a tar-format depot file: (`/var/opt/ignite/depots/Rel_B.11.31/apps`)

```
# make_depots -r B.11.31 -s /tmp/depots/T2767CC,r_A.01.20,a_HP-
UX_B.11.31_IA.tar

# make_depots -r B.11.31 -s /tmp/depots/VMProvider,r_A.01.20,a_HP-
UX_B.11.31_IA.tar

# make_depots -r B.11.31 -s /tmp/depots/VMMigrate,r_A.01.20,a_HP-
UX_B.11.31_IA.tar

# make_depots -r B.11.31 -s /tmp/depots/VMMGR,r_A.01.20,a_HP-
UX_B.11.31_IA.tar
```

c) Create an Ignite-UX config file for the above depots

```
# make_config -r B.11.31
```

The above command creates the `/var/opt/ignite/data/Rel_B.11.31/apps_cfg` and `/var/opt/ignite/data/Rel_B.11.31/core_cfg` configuration files. Edid `apps_cfg` file to mark not to install (FALSE) software you do not want to install by default.

d) Manage the Ignite-UX index file for applications

```
# manage_index -a -f /var/opt/ignite/data/Rel_B.11.31/apps_cfg
```

`/var/opt/ignite/INDEX` file should contain the following lines:

```
cfg "HP-UX B.11.31 Default" {
        description "This selection supplies the default system configuration that HP
supplies for the B.11.31 release."
        "/opt/ignite/data/Rel_B.11.31/config"
        "/opt/ignite/data/Rel_B.11.31/hw_patches_cfg"
        "/var/opt/ignite/data/Rel_B.11.31/core_cfg"
        "/var/opt/ignite/data/Rel_B.11.31/apps_cfg"
        "/var/opt/ignite/config.local"
}
```

Note: Make sure all of the created configuration files are in the B.11.31 cfg clauses you want in `/var/opt/ignite/INDEX`. For example, if you create a golden image `core_cfg` and `apps_cfg` will be added also but you do not need them. Instead, core software will be marked to install and generate an error about disk space available when trying to install/recovery from golden image.

Using a golden image

If you are in a position where you are or will be installing the same operating system and software configuration on multiple VM systems, it may be prudent to create a *golden image*. A golden image is a snapshot of a known, good system installation and configuration (including operating system and additional software) that is archived for use in installing other clients. Ignite-UX creates such a golden image in the form of a compressed tar or cpio archive of the operating system configuration that Ignite-UX can recognize and use to install on other machines.

The basic steps for creating such a configuration for a virtual machine that can be used with Ignite-UX are:

a) Prepare a system to be used in creating a golden image.

The first step in creating a *golden system* (i.e., the system configuration used in creating the golden image), is OS installation. Using an HP-UX Operating Environment (OE) is a good starting point as they contain combinations of HP software well-suited for many data centers.

b) (Optional) Install Integrity VM software to be included in the golden image.

There are three optional software bundles that, while optional, should be considered when creating an Integrity VM golden system:

- *VMGuestLib* – A set of libraries that provide basic information about a system in the context of Integrity VMs.

- *HPVM-Guest* – A combination of tools for tuning the HP-UX operating system for a VM and two useful commands for collecting status and state of a VM – `hpvminfo` and `hpvmcollect`.

- *VMProvider* – The Integrity VM WBEM provider which is used by HP's Virtual Server Environment (VSE) tools such as Virtualization Manager (vman) and the Integrity VM graphical user interface (vmmgr). The VMProvider requires either the VMGuestLib or HPVM-Guest.

Both *VMGuestLib* and *VMProvider* will be provided with HP-UX OEs in early 2006. So, if you are using a later HP-UX OE for your golden system configuration, you need not explicitly install these two bundles. Otherwise, these bundles will be available with the Integrity VM installation media and can be copied (with `swcopy`) or installed directly (with `swinstall`) from that media.

The HPVM-Guest bundle may be obtained from the Integrity VM host in the form of a SD-UX distribution tape file: `/opt/hpvm/guest-images/hpux/hpvm_guest_depot.sd`. This SD distribution tape file contains the HPVM-Guest and VMGuestLib software bundles. The file may be moved directly to the VM so that the bundle can be installed directly from that file with swinstall.

Alternatively, the file may be used to `swcopy` the HPVM-Guest bundle to a directory depot on a system from which the software can be installed in a more typical fashion with `swinstall`. Use the command `swcopy` to copy it to the depot:

```
# swcopy -s /opt/hpvm/guest-images/hpux/11iv3/hpvm_guest_depot.11iv3.sd \
\* @ <host>:/<depot path>/
```

c) Make the operating system archive (i.e., the golden image).

Once the golden system is configured the way you want it, either install the Ignite-UX software or copy the file `/opt/ignite/data/scripts/make_sys_image` to `/tmp` on the golden system. If you've copied the above file, change the permissions of `/tmp/make_sys_image` to be executable. On the Ignite-UX server make sure there is sufficient space in the target file system to hold the golden image archive that will be created. By default, the directory is `/var/tmp`, but you may want to store the golden image on another file system because `/var` is typically more difficult to extend than other file systems.

On the golden system, run:

```
# /tmp/make_sys_image -s <Ignite-UX server IP address> \

-d /<path to target directory> -n <archive file name>
```

For example, executing:

```
# /tmp/make_sys_image -s vmhost \
-d /var/opt/ignite/data/Rel_B.11.31 -n B.11.31_VMGuest_archive.gz
```

Note that the golden system must be listed in the */.rhosts* file on the Ignite-UX server. If not, you will see the following error:

> ERROR: Cannot `remsh` server system_name (check server `.rhosts` file).

Typically `make_sys_image` will run for several minutes.

Any system that is creating a golden image should be quiescent, that is, no applications should be up and running and no users should be logging into the system. When `make_sys_image` is creating a golden image some system files are replaced with newconfig versions. When these files are replaced it may negatively impact any applications that have not been shutdown.

You will need to make sure that the directory where the golden image resides is available for an NFS mount. In our example, the path `/var/opt/ignite/data/Rel_B.11.31` is added to the Ignite-UX server's `/etc/exports` file as follows:

```
/var/opt/ignite/data/Rel_B.11.31 -anon=2
```

and is subsequently exported by running `exportfs -a`.

d) Configure the Ignite-UX server to recognize the golden image.

Once the golden image has been created, you'll need to finish setting up the configuration for use on the Ignite-UX server. The steps to do this are as follows:

1. Capture the necessary impacts statements to be used in the configuration file by executing:

   ```
   # /opt/ignite/lbin/archive_impact -t -g \
   B.11.31_VMGuest_archive.gz > impacts.txt
   ```

   in the directory where the archive was created and capturing the output.

2. Create a configuration file by copying the example configuration file to the directory with the system archive. For example:

   ```
   # cp /opt/ignite/data/examples/B.11.31.golden_image.cfg \
   /var/opt/ignite/data/Rel_B.11.31/vm_guest.cfg
   ```

   You'll need to customize the configuration file for your archive as follows.

3. In the configuration file, modify the `nfs_source` definition in the `sw_source` section so that it references the Ignite UX server and the correct path. That is, using the example systems here, change:

   ```
   nfs_source = "10.2.72.150: /var/opt/ignite/archives/Rel_B.11.31"
   ```

to:

```
nfs_source = "15.180.3.215: /var/opt/ignite/data/Rel_B.11.31"
```

4. Continuing in the configuration file, modify the init `sw_sel` section(s) so that it corresponds to your golden image. Note that the example configuration specifies two different images – one for Itanium-based systems (ia64) and another for PA-RISC systems (`hppa)`. Assuming you didn't create an image for a PA-RISC systems version of HP-UX, you may simply delete the init sw_sel section for the PA-RISC system image `("is_hppa"`). Follow these steps:

   a. Change the description parameter to better describe your golden image. For example, change:

   ```
   description = "B.11.31 IA golden image archive"
   ```

   to:

   ```
   description = "HP-UX 11.31 image for VMs"
   ```

   b. Change the archive_path to contain the name of the actual archive file. Using our example, you would change:

   ```
   archive_path = "B.11.31_archive_IA.gz"
   ```

   to:

   ```
   archive_path = "B.11.31_VMGuest_archive.gz"
   ```

   c. Replace the 'impacts' lines with those captured in step 1 above. For example, replace the lines:

   ```
   impacts = "/" 7659Kb
   impacts = "/dev" 11Kb
   impacts = "/etc" 76229Kb
   impacts = "/home" 1Kb
   impacts = "/opt" 2223298Kb
   impacts = "/sbin" 109002Kb
   impacts = "/stand" 51127Kb
   impacts = "/usr" 2379518Kb
   impacts = "/var" 848454Kb
   ```

   with lines from `impact.txt` file.

5. Edit the file `/var/opt/ignite/INDEX` to install the new configuration for Ignite-UX. This is done by adding a new cfg section to that file. For our example, the new cfg section would read as follows:

```
cfg "HP-UX B.11.31 VM Config" {
description "This selection supplies the VM configuration."
"/opt/ignite/data/Rel_B.11.31/config"
"/var/opt/ignite/data/Rel_B.11.31/vm_guest.cfg"
"/var/opt/ignite/config.local"
}
```

6. Verify the syntax of your newly-entered configuration information with the following command:

```
# /opt/ignite/bin/instl_adm -T
```

If this command is successful, your configuration is now ready to use.

Cloning a system

The recovery configurations and archives created by `make_net_recovery` are stored in a separate directory on the Ignite-UX server for each client. Using the configuration and archive created by `make_net_recovery` on one system to install a different system involves manually copying some configuration files, and allowing NFS access to the source system's archive.

- Use `make_net_recovery` or ignite-UX to create a system recovery archive of the source system.

- Login to the Ignite-UX server.

- If the target system to be installed does not currently have a directory in `/var/opt/ignite/clients` but is up and running, then use the ignite to create that directory using **`Actions-> Add New Client for Recovery`**. If the system is not running, you will either need to boot the client from the Ignite-UX server

- Copy the `CINDEX` and `recovery` directory from the source client to the target client directory. If the target client has previously used `make_net_recovery` then it will already have a INDEX file. If the CINDEX file for the target system exists already, you may want to save a copy, and/or edit the file to add the desired entries from the source client. The commands below copy the required files. You may specify src_client and target_client using either the LAN addresses (such as 0x0060B04AAB30), or by using the client's hostname (which is a symlink to the LAN address):

```
# cd /var/opt/ignite/clients/src_client \
find CINDEX recovery | cpio -pdvma ../target_client
```

- Give the *`target_client`* NFS access to the archive of the source system. To do this, login to the server that holds the archive (normally the Ignite-UX server). Typically, each client has its own directory for storing the archives, and the directory is exported only to the individual client. In this case, you will need to edit the /etc/exports file to allow access to both the source and target clients:

  - Enter: `vi /etc/exports`
  - Append: target-client to the end of the source-client's line.
  - Enter: `exportfs -av`

- Boot the target-client from the Ignite-UX server. Then when you install the system, you can select from the recovery configurations of the source system.

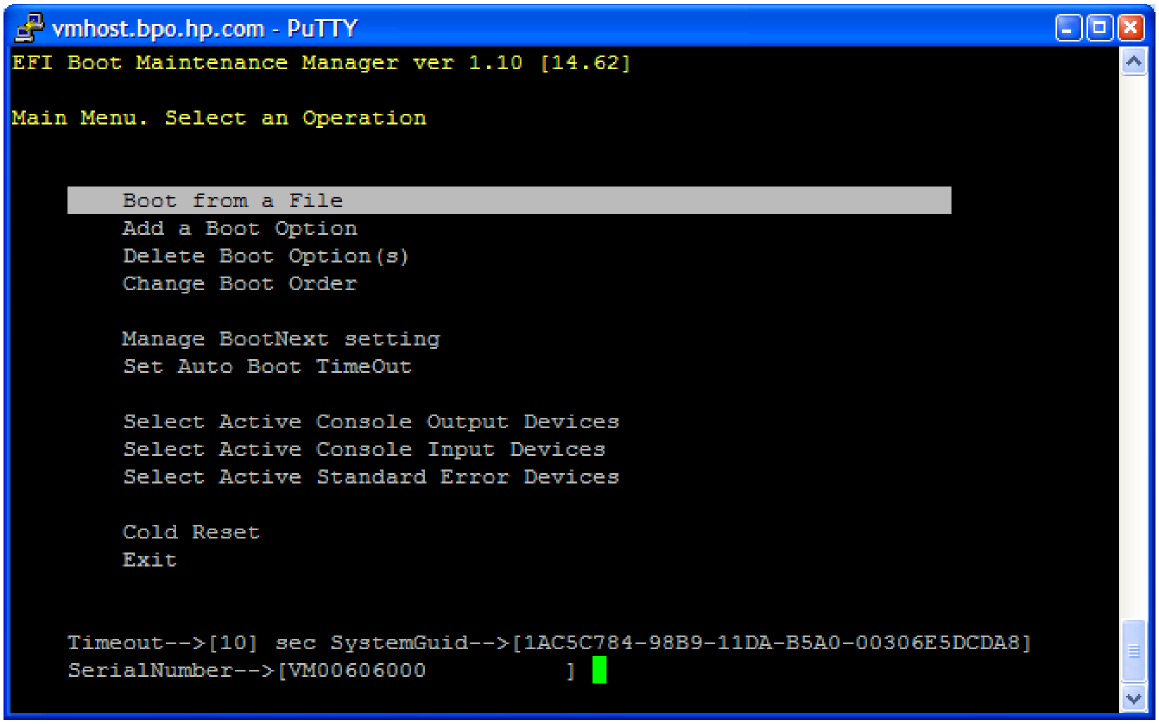
- Change the system networking parameters for the target system during the installation.

Loading VM Guest from Ignite

Once the Ignite-UX server is configured for your VM, you'll need to access its virtual console. To do this, first power on the VM (e.g., with `hpvmstart -P vmguest2`) and use `hpvmconsole` to access the console (e.g., `hpvmconsole -P vmguest2 -fi`).

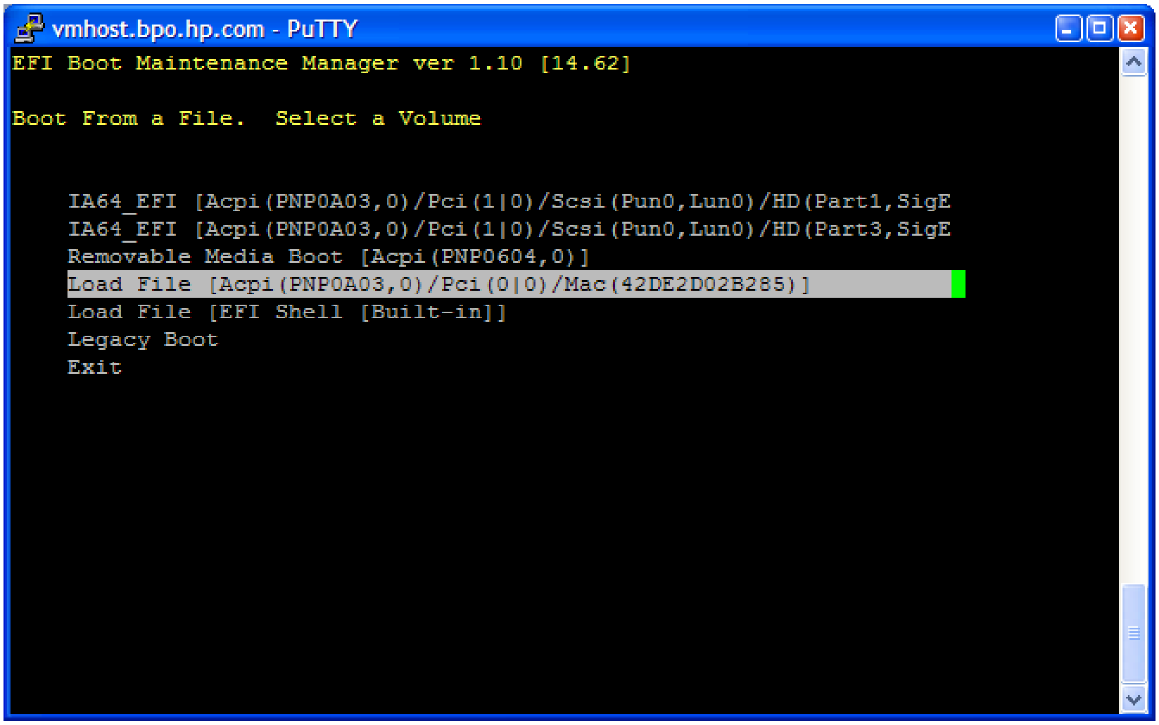
The first release of Integrity VM's console and Extensible Firmware Interface (EFI) does not have lanboot implemented. It is easy to work around this by following the sequence of instructions that follow.

Select **Boot option maintenance menu** and press enter. At the Boot Maintenance Manager menu select **Boot from a File** and press enter.



```
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


    Timeout-->[10] sec SystemGuid-->[1AC5C784-98B9-11DA-B5A0-00306E5DCDA8]
    SerialNumber-->[VM00606000           ]
```

Select the `Load File` entry with the MAC address you’re going to use for the Ignite-UX installation.



```
EFI Boot Maintenance Manager ver 1.10 [14.62]

Boot From a File.  Select a Volume


    IA64_EFI [Acpi(PNP0A03,0)/Pci(1|0)/Scsi(Pun0,Lun0)/HD(Part1,SigE
    IA64_EFI [Acpi(PNP0A03,0)/Pci(1|0)/Scsi(Pun0,Lun0)/HD(Part3,SigE
    Removable Media Boot [Acpi(PNP0604,0)]
    Load File [Acpi(PNP0A03,0)/Pci(0|0)/Mac(42DE2D02B285)]
    Load File [EFI Shell [Built-in]]
    Legacy Boot
    Exit
```

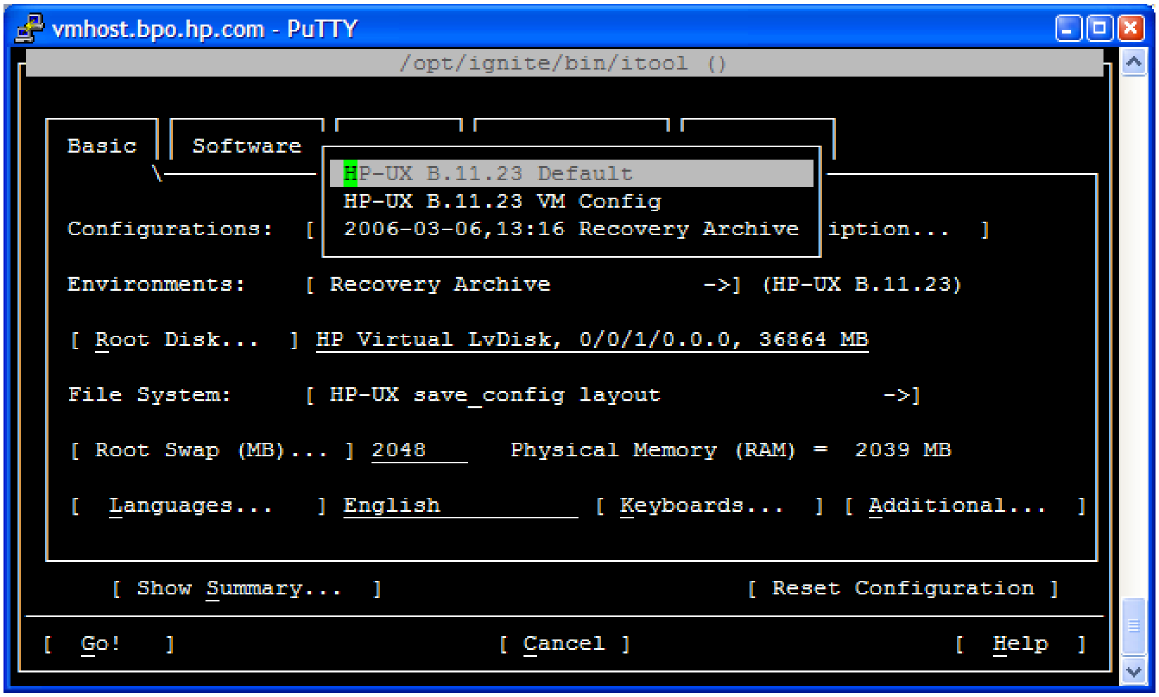
If there is more than one such `Load File` entry and your VM has a fixed IP, then be sure to select the entry with the MAC address that corresponds to the entry made for this VM in the

Ignite-UX server's `/etc/bootptab` file. Note that one can obtain the MAC address for the VM here rather than from hpvmstatus.

After selecting the appropriate *`Load File`* entry, the client will broadcast a request and the Ignite-UX Server will, if everything is configured correctly, respond. A successful connection and initiation of communication between the client and the Ignite-UX server is shown below.

```
Running LoadFile()

CLIENT MAC ADDR: 42 DE 2D 02 B2 85
CLIENT IP: 15.180.3.218  MASK: 255.255.240.0  DHCP IP: 15.180.3.215
GATEWAY IP: 15.180.0.1
Running LoadFile()

TFTP.
@(#) HP-UX IA64 Network Bootstrap Program Revision 1.0
Downloading HPUX bootloader
Starting HPUX bootloader
…
```

A few moments later you will arrive at the HP-UX installation menu. From there, OS and software installation proceeds as with any other HP-UX installation/recovery.



```
                         /opt/ignite/bin/itool ()

   Basic  |  Software  |
           \
                         HP-UX B.11.23 Default
                         HP-UX B.11.23 VM Config
   Configurations:  [    2006-03-06,13:16 Recovery Archive   iption...   ]

   Environments:    [ Recovery Archive           ->] (HP-UX B.11.23)

   [ Root Disk...  ] HP Virtual LvDisk, 0/0/1/0.0.0, 36864 MB

   File System:     [ HP-UX save_config layout                  ->]

   [ Root Swap (MB)... ] 2048       Physical Memory (RAM) =  2039 MB

   [  Languages...   ] English          [ Keyboards...  ] [ Additional...  ]

      [ Show Summary...  ]                         [ Reset Configuration ]

 [  Go!   ]                         [ Cancel ]                      [  Help  ]
```

Example

- `Default` -> installation from depots (cold install)
- `VM Config` -> installation from golden image

- `Recovery` -> recovery/cloning

DVD Installation / Recovery

Using VM Host internal DVD

You will be able to install a virtual machine from the host local DVD, you just need to add the guest configuration a virtual dvd associated with the local DVD and insert HPUX media.

Using an ISO golden image

You will need to install the Ignite-UX utilities software on the golden system. Version C.6.5.61 or later of B5725AA (*Ignite-UX*) and Ignite-UX-11-31 (*HP-UX Installation Utilities for Installing 11.31 Systems*) is highly recommended so that larger ISO images (up to 4GB) can be accommodated.

After the golden system is prepared, the ISO image is created with the following steps.

a) Making the operating system archive

The operating system archive is created locally on the golden system itself with the following command:

```
# /opt/ignite/data/scripts/make_sys_image -s local \
-d /scratch/staging -n vm.gz
```

You can also create it on the ignite server:

```
# /opt/ignite/data/scripts/make_sys_image -s <ignite_server> \
-d /var/opt/ignite/data/Rel_B.11.31.VM/scratch/staging -n vm.gz
```

It may take several minutes to complete.

b) Creating an archive configuration file

An archive configuration file must be created for the golden image. First, capture the necessary impacts statements to be used in the configuration file by executing:

```
# /opt/ignite/lbin/archive_impact -t -g ./vm.gz > impacts.txt
```

in the directory where the system archive file was created. This may take several minutes to complete.

Now create a configuration file by copying the example configuration file to the directory with the system archive. For example:

```
# cp -p /opt/ignite/data/examples/B.11.31.archives.cfg \
/scratch/staging/vm.cfg
```

You'll need to customize the configuration file for your archive. Several modifications of the (is_ia64) section are necessary. For example, in the vm.cfg file, change:

```
archive_path = "B.11.31_archive_IA.gz"
```

to:

```
archive_path = "vm.gz"
```

Then replace all of the impacts entries with those from those from the impacts.txt file (created above). Unless you are also creating a configuration for PA-RISC, you can remove the (is_hppa) section from the configuration file.

c) Creating a copy of the LIF volume

The ISO image must contain a LIF volume. Execute the following to create it:

```
# /opt/ignite/bin/make_medialif -a -r B.11.31 \
-f /opt/ignite/data/Rel_B.11.31/config \
-f /scratch/staging/vm.cfg \
-l /scratch/staging/vm.lifimage
```

Note the specification of two configuration files with the `-f` option. These are concatenated together for use by `make_medialif`.

d) Copying the EFI image

Itanium-based systems, including Integrity VMs, require an EFI image. Thus, the EFI image must be copied to the directory where the ISO image is being staged:

```
# cp -p /opt/ignite/boot/Rel_B.11.31/EFI_CD_image /scratch/staging
```

e) Creating the ISO image

Now that everything is in place, the ISO image is created with the `mkisofs` command. Note that we use the Itanium-based variation of this command.

```
# /opt/ignite/lbin/mkisofs \
-D -R -U -max-iso9660-filenames \
-no-emul-boot \
-b EFI_CD_image \
-eltorito-alt-boot \
-no-emul-boot \
-b vm.lifimage \
-o /scratch/B.11.31.iso \
/scratch/staging/
```

Note the repeat of the option `'-no-emul-boot'` is required for the subsequent `'-b'` option.

You will see two warnings that are expected (please ignore them):

```
Warning: creating filesystem that does not conform to ISO-9660.
Warning: ISO-9660 filenames longer than 31 may cause buffer overflows in the
OS.
```

The command will take several minutes to complete and will be fairly verbose.

f) Rendering the ISO mountable and bootable

The final ISO image (virtual DVD) will be both a bootable DVD (via the LIF image) and a file system that can be mounted by HP-UX. To enable the ISO image to function both ways, `instl_combine` is used as follows:

```
# /opt/ignite/lbin/instl_combine -C /scratch/B.11.31.iso
```

g) Validating the ISO image

Now that the ISO image is complete, verify that it is a valid LIF image:

```
# lifls /scratch/B.11.31.iso
```

which will simply list its contents. If it fails or produces error messages, then your ISO image is either too large for lifls to read (over 2GB) or it is corrupt.

In the event that the ISO image is larger than 2GB, lifls will produce the following message:

```
lifls(open): Value too large to be stored in data type
lifls: Can't list /scratch/B.11.31.iso; file not opened
```

To check whether this (large) image is, in fact, valid use the workaround for this problem - copy the first 600MB to another file with dd and run lifls on the resulting file.

Loading VM Guest from DVD/ISO

A VM may be configured with a virtual DVD by specifying the virtual DVD definition with commands such as `hpvmcreate` and `hpvmmodify`. To check what virtual DVD devices (if any) are defined for a VM, execute the following:

```
# hpvmstatus -P vmguest2 | grep dvd
dvd scsi 0 1 0 1 0 file /hpvm/ISO/B.11.31.iso
```

From the above we deduce that there is, indeed, a virtual DVD device associated with `vmguest2` which is associated with the file `/hpvm/ISO/B.11.31.iso`. If no such device exists, it can be added as follows:

```
# hpvmmodify -P vmguest2 -a dvd:scsi::file:/hpvm/ISO/B.11.31.iso
```

Also you can add a virtual DVD associated with the host local DVD.

```
      # hpvmmodify -P vmguest2 -a dvd:scsi::disk:/dev/rdsk/c0t0d0
```

Once the virtual DVD is associated with the VM, it will automatically identify it when powered. Note that the VM's EFI automatically identifies the file system (fsX) on the virtual DVD (media should be into DVD if using host DVD). Subsequently, the installation media is located on the virtual DVD (i.e., the ISO image) and the installation boot sequence begins. From this point, the HP-UX installation is the same as if you are using physical installation media.

Sharing media among virtual machines

Once you have the ISO image on the Integrity VM Host, you may want to use it with multiple VMs to start the installation process. As before, the virtual DVD must be defined for the new VM.

```
      # hpvmdevmgmt -l gdev:/hpvm/ISO/B.11.31.iso
      /hpvm/ISO/B.11.31.iso:CONFIG=gdev,EXIST=YES,DEVTYPE=FILE,SHARE=NO:…
```

Among the attributes listed here, note the attribute SHARE is set to NO. By setting this attribute to YES, multiple VMs may share this ISO image. This is accomplished by executing:

```
      # hpvmdevmgmt -m gdev:/hpvm/ISO/B.11.31.iso:attr:SHARE=YES (ISO)

      # hpvmdevmgmt -m gdev:/dev/rdsk/c0t0d0:attr:SHARE=YES (local DVD)
```

Using Integrity Virtual Machine

This chapter describes the usage of Virtual Machine Software and the commands.

HPVM Commands

hpvmcreate	Create a new virtual machine
hpvmremove	Remove a virtual machine
hpvmstart	Boot a virtual machine
hpvmstop	Stop a virtual machine
hpvmconsole	Connect to the console of a virtual machine
hpvmstatus	Display information about one or more virtual machines
hpvmmodify	Modify the attributes of a virtual machine
hpvmnet	Create and control an Integrity Virtual Machines virtual network switch (vswitch)
hpvmdevmgtm	Manage the devices that are associated with the VM host and the guests

hpvmclone	Create a VM that is copy of an existing VM
hpvmmigrate	Migrate a VM to a different host
hpvmcollect (*)	Collect crash dumps, logs, system status, and configuration on the VM host or guests for analysis
hpvminfo (*)	Display information about VM environment

(*)Available on guests if HPVM-Guest software is installed (recommended)

Note: Man pages are available at: WTEC SharePoint

hpvmcreate

The hpvmcreate command creates a new virtual machine (a guest), and assigns the specified attributes and resources to it. This command creates an association between the virtual devices seen by the guest and the physical devices managed by the VM Host.

Use the corresponding hpvmcreate command line options:

Guest Characteristic	Syntax	Default	Notes

Guest name	`-P vm_name`
	`-p vm_number`
		required	

Operating system	`-O os_type`	HPUX	
Virtual CPUs	`-c number_vcpus`	1	max virtual cpus = min(# physical cpus, 4)

CPU entitlement	`-e percent`
	`-E cycles`
		10%	sum of VMs cpu entitlement <= 100% (no oversubscribing)

Memory	`-r amount`	1GB	must be a multiple of 64MB
I/O Resources: network, disks, dvds	`-a rsrc`	none	see hpvmmodify info
Startup behavior	`-B start_attr`	manual	

Admin account name	`-u usergroup:[kind]`
	`-g group:[kind]`
		none	

hpvmremove

The hpvmremove command deletes a virtual machine´s configuration information and frees any resources associates with it. VM must be down.

Example:

```
    vmhost# hpvparremove -P vmguest2
```

```
hpvmremove: Remove the virtual machine 'vmguest2'? [n]:
```

hpvmstart

The hpvmstart command is used to boot a guest OS. It does all resource checks for VM and will prevent it from starting it they are not met (You can force it with the –F option but it is no recommended and can cause data corruption, virtual machine hangs,..)

Example:

```
vmhost# hpvmstart -P vmguest2
(C) Copyright 2000 - 2006 Hewlett-Packard Development Company, L.P.
Opening minor device and creating guest machine container
Creation of VM, minor device 2
Allocating guest memory: 1024MB
  allocating low RAM (0-40000000, 1024MB)
/opt/hpvm/lbin/hpvmapp (/var/opt/hpvm/uuids/1ac5c784-98b9-11da-b5a0-
00306e5dcda8/vmm_config.current): Allocated 1073741824 bytes at
x6000000100000000
  allocating firmware RAM (ffaa0000-ffab5000, 84KB)
/opt/hpvm/lbin/hpvmapp (/var/opt/hpvm/uuids/1ac5c784-98b9-11da-b5a0-
0306e5dcda8/vmm_config.current): Allocated 86016 bytes at 0x6000000140000000
Loading boot image
Image initial IP=102000 GP=5E4000
Initialize guest memory mapping tables
Starting event polling thread
Starting thread initialization
Daemonizing....
hpvmstart: Successful start initiation of guest 'vmguest2'
```

hpvmstop

The hpvmstop command stops a running virtual machine by simulating the operations performed at the system console on a physical system. It can perform a hard stop, which functions like a power failure, or a graceful stop, in which the guest operating system receives notification and time to perform cleanup operations before the stop. You can also connect to the virtual console to perform a hard stop, reset, toc or graceful shutdown of the VM (see hpvmconsole command)

Example:

To perform a graceful stop

```
vmhost# hpvmstop -P vmguest2 -g
```

To perform a a hard stop

```
vmhost# hpvmstop -P vmguest2 -h
```

hpvmconsole

Integrity VM virtual machine console is similar in appearance to the maintenance processor of an Integrity System. Each virtual machine can be powered on or off, the guest operating system can be booted or shut down, and so forth. The hpvmconsole command connects to the virtual console of a specified machine.

Using Ctrl+B you can return to the virtual console main menu except if you have logged into the physical console of an VM Host and then run hpvmconsole interactively. In that case you should use Ctrl+X.

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. 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 VMHost. 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/vmguest1 -c 'vmguest1 console' \
                 -s /opt/hpvm/bin/hpvmconsole vmguest1
   ```

   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 vmguest1
   ```

A guest administrator can now access the vmguest1 virtual console using the ssh command or telnet command on the VM Host and logging in to the vmguest1 account.

hpvmstatus

The hpvmstatus command displays information about the operational state and virtual hardware configuration of the virtual machines on the VM Host. Information displayed includes the following:

- Name of the virtual machine.
- State of the virtual machine :

State	Description
`On`	The virtual machine is “powered on”. It may be at its console prompt, or it may have booted its operating system and be fully functional. This is the normal state of a running virtual machine.
`Off`	The virtual machine is fully halted.
`Invalid`	The virtual machine configuration file is corrupted or invalid. The configuration file must be corrected before this virtual machine can be started.

- Resources attached to the virtual machine.
- Attributes assigned to the virtual machine.

Some useful options are:

`-D`	(Deferred) Displays information about resource assignment that will take effect the next time the virtual machine is started.
`-V`	(Verbose)
`-e`	(event) Displays event log for the VM Host.
`-r`	Displays the CPU entitlement information for the virtual machine.

Here we have some output examples:

Example:

No options:

```
vmhost:/ # hpvmstatus
[Virtual Machines]
Virtual Machine Name VM #  OS Type State     #VCPUs #Devs #Nets Memory  Runsysid
==================== ===== ======= ========= ====== ===== ===== ======= ========
vmguest1                 2 HPUX    On (OS)        1     2     2    2 GB         0
vmguest2                 3 HPUX    On (OS)        1     3     2 2048 MB         0
```

Information about vmguest1 – for verbose (V)

```
vmhost:/ # hpvmstatus -P vmguest1
[Virtual Machine Details]
Virtual Machine Name VM #  OS Type State
==================== ===== ======= ========
vmguest1                 2 HPUX    On (OS)

[Authorized Administrators]
Oper Groups:
Admin Groups:
Oper Users:   vmguest1
Admin Users:

[Virtual CPU Details]
vCPUs  Type    Entitlement Maximum
====== ======= =========== =======
     1       30.0%  100.0%

[Memory Details]
Total       Reserved
Memory      Memory
=========== ===========
   2 GB    64 MB

[Storage Interface Details]
Guest                                  Physical
Device  Adaptor    Bus Dev Ftn Tgt Lun Storage   Device
======= ========== === === === === === ========= =========================
disk    scsi         0   1   0   1   0 disk      /dev/rdsk/c4t0d0
disk    scsi         0   1   0   2   0 lv        /dev/vgimages/rvmguest1disk

[Network Interface Details]
Interface Adaptor    Name/Num   PortNum Bus Dev Ftn Mac Address
========= ========== ========== ======= === === === =================
vswitch   lan        vswa       1         0   0   0 92-58-ee-ee-15-16
vswitch   lan        vswb       1         0   2   0 76-c3-8d-f7-52-65

[Misc Interface Details]
Guest                                  Physical
Device  Adaptor    Bus Dev Ftn Tgt Lun Storage   Device
======= ========== === === === === === ========= =========================
serial  com1                           tty       console
```

CPU entitlement information

```
       vmhost:/ # hpvmstatus -r -P vmguest2
       [Virtual Machine entitlements]
                                  Percent       Cumulative
       #VCPUs Entitlement Maximum   Usage             Usage
       ====== =========== ======= ======= ================
            1       25.0%  100.0%    3.4%          9121475

       [Virtual CPU details]
       vCPU Cumulative       Guest   Host    Cycles   Sampling
       ID   Usage            percent percent achieved Interval
       ==== ================ ======= ======= ======== ===========
        0          8994609    3.3%    3.3%    49MHz  10 seconds
```

hpvmmodify

The hpvmmodify command modifies the attributes and resources of the specified virtual machine (-P vm_name / -p vm_number).

All attributes and resources can be changed statically, so that changes take effect when the virtual machine is next restarted. Some attributes and resources can also be changed dynamically. Dynamic changes take effect immediately and remain in effect when the virtual machine is next started, un less you explicitly specify otherwise with the –A option.

Task	Syntax	Allowed with guest running
I/O resource allocation	-a (add) (***) -m (modify) (**) -d (delete) For storage devices: device-type:adapter-type:[hardware-address]:storage-type:device device-type = disk/dvd adapter-type = scsi hardware-address = bus,device,target storage-type = disk/lv/file/null disk-> disk or dvd lv -> LVM or VxVM character logical device file file -> locally-mounted, nonHFS file (created with *hpvmdevmgmt* command) null-> VxFS directory containing ISO files or dvd For attached devices: device-type:adapter-type:[hardware-address]:storage-type:device device-type = tape/changer/burner adapter-type = scsi hardware-address = bus,device,target storage-type = attach The device used should be the sctl device (/dev/rscsi/cXtYdZ), these devices are created when Integrity VM starts or using `hpvmdevmgmt -I`. For network devices: network:lan:[hardware-address]:vswitch:vswitch-name hardware-address = bus,device,mac-addr mac-addr format: 0xaabbcc001122/aa-bb-cc-00-11-22 vswitch-name = name assigned to the virtual network when it is created using *hpvmnet* command	yes (*)
VM Name	-N new_name	no
Virtual	-c number_vcpus	reboot

CPUS		required to take effect
CPU Entitlement	-e percent -E cycles	yes
Memory	-r amount	reboot required to take effect
Boot options	-B start_attr	yes
Group/User auth.	-g [+|-]group:{admin|oper} -u [+|-]user:{admin|oper}	yes

(*) Before removing virtual devices with the hpvmmodify command, make sure that the guest operating system is no longer directing I/O to the device. Dismount the device if it is mounted. If you attempt to remove a device that has I/O in progress, the hpvmmodify command incorrectly removes the device from the guest configuration file. The hpvmstatus command no longer displays the device, and the hpvmmodify command does not retry the device removal, but the guest operating system sees the device as available. Reboot the guest to remove the device.

(**) When you use –m option you need to specify *hardware-address. (*If you want to modify hardware-address use –d to delete it and then –a to add with the new hardware-address)

(***) VM cannot detect all potential backing store conflicts. Care with: multiple path to the same disk, overlapping physical storage allocated for different backing store types… It can result in data corruption.

Example:

```
# hpvmmodify -P vmguest1 -a disk:scsi:0,1,2:disk:/dev/rdsk/cXtYdZ
                    -a disk:scsi::lv:/dev/vgXX/rlvolY
                    -a disk:scsi:0,1,1:file:/guestfiles/diskfile
                    -a dvd:scsi::disk:/dev/rdsk/cXtYdZ
                    -a dvd:scsi::null:/ISO-directory
                    -a tape:scsi::attach:/dev/rscsi/cXtYdZ
```

hpvmnet

A virtual machine accesses its network through a virtual LAN device connected to a virtual network switch (vswitch). The virtual network switch is connected in turn to a single physical network interface (NIC) on the VM Host. The hpvmnet command is used to create and manage vswitches.

A vswitch works like an actual network switch. It accepts outbound network traffic from all guests configured to use it and transmits the traffic over the physical interface. It accepts inbound network traffic for all guests configured to use it and directs the traffic to the appropriate guest.

A virtual switch can be shared by multiple virtual machines or dedicated to a single virtual machine. It can be associated with at most one physical network interface. The VM Host's physical network interface must be attached to a network with connectivity to the desired subnets. The network interface may optionally be configured in the VM Host with an IP address or multiple IP alias addresses, but this is only necessary if the VM Host shares the interface with the vswitch and directs its own network traffic over the card. If you alter any characteristics of a network interface associated with a running vswitch, for instance, through the ifconfig commands on the VM Host, you must stop and restart the vswitch. Otherwise, any guests using that vswitch will experience intermittent network failures. Stopping and restarting a vswitch can occur while its guests are running; no guest shutdown is required.

By default, Integrity VM creates a vswitch named localnet that is not associated with a physical interface. It can be used for communication between the guests running on the same host.

Here we have the main options:

Task	Syntax
Create / Delete a virtual switch	`-c / -d`
Start / Stop / Restart a virtual switch	`-b / -h / -r`
Designates the network interface that vswitch will use (with –c option)	`-n nic_id`

Example:

```
vmhost:/>hpvmnet
Name     Number State   Mode      PPA    MAC Address    IP Address
======== ====== ======= ========= ====== ============== ===============
localnet      1 Up      Shared           N/A            N/A
vswa          3 Up      Shared    lan0   0x00306e5dcda8 15.180.3.215
vswb          4 Up      Shared    lan1   0x00306e5dcda9
```

hpvmdevmgmt

List an entry in the Integrity VM device-management database, which tracks and validates guest-device usage, ensures that devices are only shared deliberately, and restricts guest access to devices used by the VM Host. Guest devices are added, modified, and removed from this database when you use Integrity VM commands, such as hpvmcreate, hpvmmodify, and hpvmclone. The hpvmdevmgmt command allows you to examine the database entries, alter specific device attributes, specify shared devices, and perform specialized functions associated with device management.

The device management database contains three types of entries:

- restricted devices (rdev)
- guest devices (gdev)
- VM host devices (server)

A device management database entry contains a name or alias, attributes in the form ATTRIBUTE_NAME=VALUE, a list of guest names or other device entries depending upon this entry (called its dependents), and a unique identifier.

Main options:

<table>
<tr><th>Task</th><th>Syntax</th></tr>
<tr><td>List/Add/Delete/Modify devices</td><td><code>-l {all|server|rdev|gdev}</code>
<code>-a {all|server|rdev|gdev}:entry_name</code>
<code>-d {all|server|rdev|gdev}:entry_name</code>
<code>-m {all|server|rdev|gdev}:entry_name:attr:atti_name=attr_value</code></td></tr>
<tr><td>Create a file for use as a virtual device</td><td><code>-S size filename (size must end in M for megabyte or G for gigabyte)</code></td></tr>
<tr><td>Replace a guest device</td><td><code>-n gdev:oldentry_name:newentry_name0[,newentry_name1}</code></td></tr>
<tr><td>Install sctl devices under the <code>/dev/rscsi</code> directory</td><td><code>-I</code></td></tr>
</table>

Example:

Information about all devices:

```
vmhost:>hpvmdevmgmt -l all
# HP Virtual Machine Management Database
# Created 02/23/06 16:48:21 by user root
# Version 1.20.0
#
# This file must not be modified by hand.
#
HPVM MANAGEMENT_DB START
vmhost:CONFIG=SERVER::WWID_NULL
localnet:CONFIG=gdev,EXIST=YES,DEVTYPE=SWITCH,SHARE=YES::WWID_NULL
lan0:CONFIG=gdev,EXIST=YES,DEVTYPE=NIC,SHARE=NO:vswa:WWID_NULL
vswa:CONFIG=gdev,EXIST=YES,DEVTYPE=SWITCH,SHARE=YES:vmguest1:WWID_NULL
/dev/vgguestimages/rvmguest1:CONFIG=gdev,EXIST=YES,DEVTYPE=LV,SHARE=NO:vmguest1:HPVM_STATID_2942_1073807361
/dev/rdsk/c4t0d0:CONFIG=gdev,EXIST=YES,DEVTYPE=DISK,SHARE=NO:vmguest1:6006-0b00-0009-291e-0000-0000-0000-0083
/dev/rdsk/c0t0d0:CONFIG=gdev,EXIST=YES,DEVTYPE=DISK,SHARE=YES:vmguest1,vmguest2:HPVM_STATID_127_3154116608
lan1:CONFIG=gdev,EXIST=YES,DEVTYPE=NIC,SHARE=NO:vswb:WWID_NULL
vswb:CONFIG=gdev,EXIST=YES,DEVTYPE=SWITCH,SHARE=YES:vmguest1,vmguest2:WWID_NULL
/dev/vgguestimages/rvmguest2:CONFIG=gdev,EXIST=YES,DEVTYPE=LV,SHARE=NO:vmguest2:HPVM_STATID_2973_1073807362
HPVM MANAGEMENT_DB EOF
```

- Modify dvd SHARED attribute so several virtual machines could use it.

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

hpvmclone

The hpvmclone command creates a copy of an existing virtual machine and its configuration information. This command copies the configuration files of the existing guest. It does not copy the actual data and software associated with the guest.

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

Command syntax is like in hpvmmodify command but –N option is required in this case to indicate new virtual machine name.

Example:

-Clone virtual machine vmguest2 to a new virtual machine call vmguest3

```
vmhost:/>hpvmclone -P vmguest2 -N vmguest3
HPVM guest vmguest3 configuration problems:
    Warning 1 on item /dev/vgguestimages/rvmguest2: Device file
'/dev/vgguestimages/rvmguest2' in use by another guest.
These problems may prevent HPVM guest vmguest3 from booting.
hpvmclone: The cloning process is continuing.
```

hpvmmigrate

The hpvmmigrate command takes an existing virtual machine and moves its configuration information to a different host.

Note: The hpvmcreate command is included in the HP Integrity Virtual Machines product and is packaged in the separate VMMigrate bundle.

The hpvmmigrate command checks to make sure that the destination host has sufficient resources (such as memory, network switches and storage devices) for the guest to boot. If the resources are insufficient or do not exist, or other error occur, the guest is not migrate to the destination host.

After successfully migrating the guest, the hpvmmigrate command automatically deletes the guest on the source host. In HPVM 2.0 hpvmmigrate command does not delete guest if this is a distributed guest (service guard package).

Command Characteristic	Syntax
Guest to migrate	-P source_vm_name -p source_vm_number
Destination VM Host system	-h dest_hostname or dest_IP_address
New name (If needed)	-N new_vm_name

(See "Migrating Virtual Machines" for more information)

Example:

- To migrate vmguest2 to virtual host vmhostB

```
vmhost:/>hpvmmigrate -P vmguest2 -h vmhostB
```

hpvmcollect

The `hpvmcollect` command collects log files, system status, device information, system and Integrity Virtual Machines configuration, guest information, and, optionally, crash dumps.

When run on a VM Host, it collects systemwide information as well as information for a specified guest.

When run in a guest, it collects only the information associated with the guest.

The `hpvmcollect` command creates a directory and produces a tar archive or a compressed tar archive containing the collected information and places it in your current directory (default) or the directory you specify with the –d option on the host you specify with –h option.

Example:

Collect host and guest data for a single guest name “vmguest1” into /tmp (/tmp/hpvmcollect_archive/ gets created)

```
# hpvmcollect -d /tmp -P vmguest1
```

Note: Please use it prior to logging a call.

hpvminfo

Allows you to determine whether you are running in a guest or on the VM Host. When run in a guest, this command returns information to identify the VM Host.

Example:

Run in a VM host

```
vmhost:/>hpvminfo
hpvminfo: Running on an HPVM host.
```

- Run in a guest:

```
vmguest1:/>hpvminfo -V
hpvminfo: Running inside an HPVM guest.
Host chassis information
  Host model string                : ia64 hp server rx4640
  Host serial number               : DEH4547A7B
  Host partition ident             : f00974b0-7bb3-11da-954d-08663df866e2
  Host machine ident               : f00974b0-7bb3-11da-954d-08663df866e2
Host Inet information
  Hostname                         : vmhost
  Number of host IPv6 Addresses    : 0
  Number of host IPv4 Addresses    : 1
    IP Address                     : 15.180.3.215
```

HPVM Troubleshooting

Logging:

Log File	HPVM Component Usage
/var/adm/syslog/syslog.log	Virtual switch (hpvmntdvr) CPU Scheduling agent (vm_fssagt)
/var/opt/hpvm/common/hpvm_mon_log (*)	Virtual Machine Monitor
/var/opt/hpvm/common/command.log	Host command line interface
/var/opt/hpvm/guests/<guest>/log	Guests (hpvmapp)
/var/opt/hpvm/guests/<guest>/console/conslog	Guest consoles
/var/opt/hpvm/guests/<guest>/FPL (binary) /var/opt/hpvm/guests/<guest>/SEL (binary)	Virtual Firmware

(*) Global HPVM monitor log:

Troubleshooting tips:

- Check *Release Notes* for known issues and WTEC known problems web page

- Integrity VMs and vPars First Pass Guide (maintained by WTEC)

- Use `hpvmcollect (1m)` to collect several log files.

- To TOC a Guest

```
# hpvmconsole -P myguest -q -c tc

# hpvmconsole -P myguest -q -c tc -f  (to view console messages)
```

- Extracting other logs with `hpvmconsole (1m)`

```
# hpvmconsole -P myguest -q -c cl > cons.log

# hpvmconsole -P myguest -q -c ‘ed -init’ > init.log

# hpvmconsole -P myguest -q -c ‘ed -mca’ > mca.log

# hpvmconsole -P myguest -q -c ‘rec -view’ > op.log
```

- Virtual firmware logging, Forward Progress Log and System Event Log

```
# /usr/sbin/diag/contrib/slview -p 0 -f FPL (or SEL)
# hpvmconsole -P <guest> → vMP>SL command

# tail -f /var/opt/hpvm/common/hpvm_mon_log
```

- To change the size of the log file:
  - Use the ch_rc command to change the file size:

    ```
    # ch_rc -a -p VMMLOGSIZE=4096
    ```

  - Kill the monitor log daemon (it respawns):

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

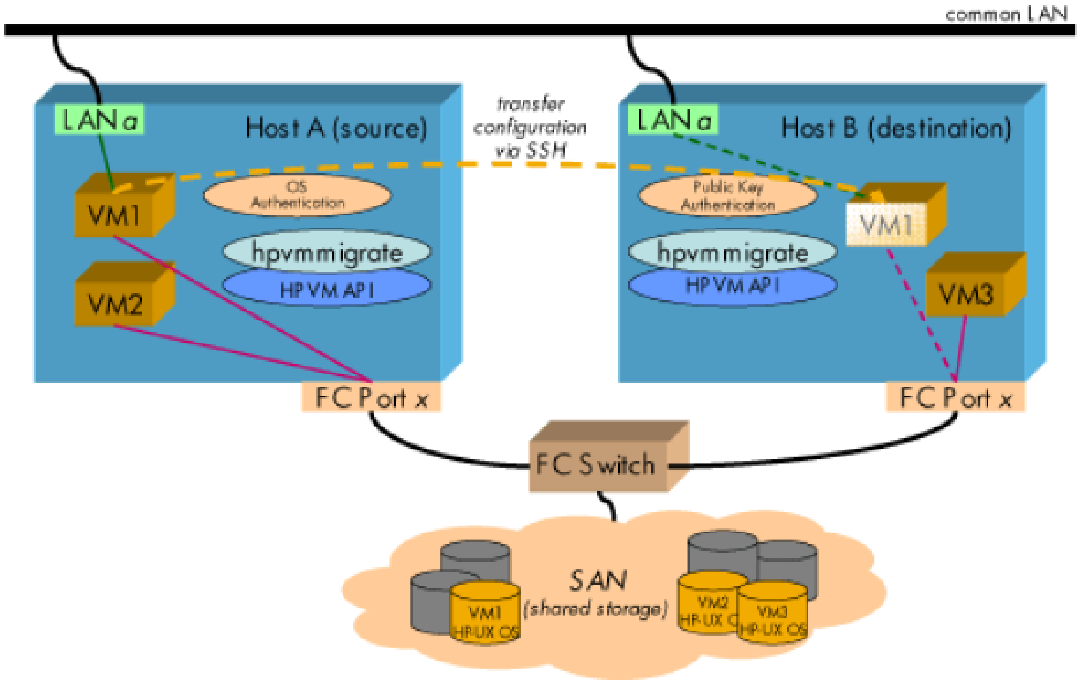
    (`/etc/rc.config.d/hpvmconf VMMLOGSIZE=1024kb` by default)

Migrating Virtual Machines

A – Both VM Hosts Active

The `hpvmmigrate` command allows you to move a virtual machine, also known as guest, from a source VM Host to a destination VM Host system. The `hpvmmigrate` command is available with HP Integrity VM, version A.01.20.00 and later. It is installed with the optional VMMigrate bundle.

Symmetric Hosts Configured for VM Guest Migration



To enable migration the source and destination hosts must be configured symmetrically. That is, all the network and storage resources must be configured the same on both hosts. A symmetric configuration includes:

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

Refer to LAN/SAN Configuration Considerations for additional details.

Performing a Guest Migration

- Set up SSH keys on both the source and the destination hosts, as described in Security Considerations.
- Stop the guest on the source host, using the hpvmstop or hpvmconsole command.
- On the source host, enter the hpvmmigrate command.

- Start the guest on the destination host using the hpvmstart or hpvmconsole command.

Networking Configuration Considerations

The source and destination hosts should be on the same subnet. The `hpvmmigrate` command preserves the MAC address of the guest being migrated, so having the hosts on the same subnet prevents problems where it would be necessary to change to the guest´s host name or IP address

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

Storage Configuration Considerations

Both the source and destination hosts must share access to symmetrically configured storage devices. Specially, both hosts must use the same character disk device file name for each disk device (see `ioinit(`1M) for information about how to reassign instance numbers to the *`ext_bus`* class.)

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

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

```
# hpvmdevmgmt –a rdev:entry_name
```

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

Security Considerations

The `hpvmmigrate` command requires HP-UX Secure Shell (SSH) to be set up on both the source and destination host systems. SSH provides a secure communication path between hosts and is installed on HP-UX 11.31 systems by default. To enable secure communication between the source and the destination hosts, you must generate SSH keys on both systems.

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

SSH Key Setup:

HP recommends that you use the HP-UX Distributed System Administration Utilities (DSAU) tools to set up the SSH keys on the source and destination hosts, which is installed by default on HP-UX 11.31. The bundle name is DSAUtilities.

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

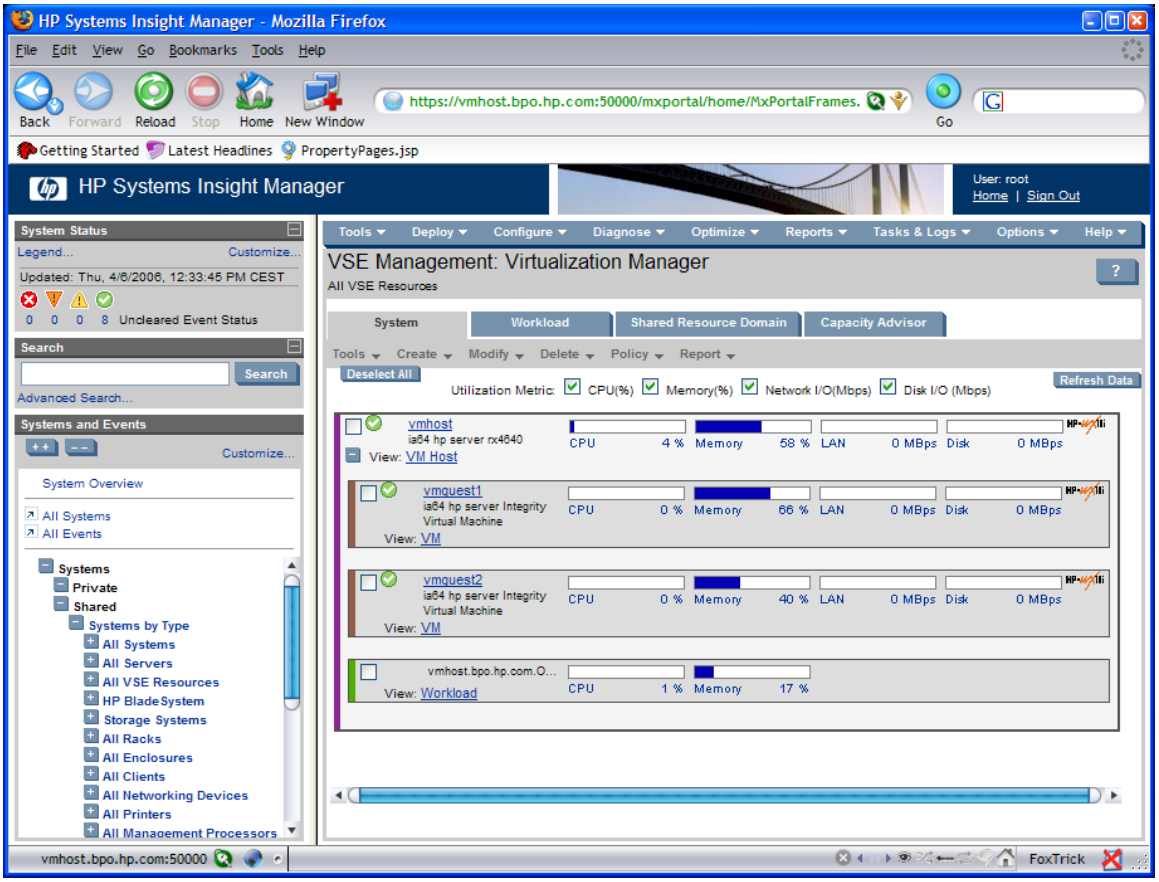
Alternatively, SSH keys can be generated manually on the individual systems and then copied to the remote system's `$HOME/.ssh/authorized_keys2` file by using the `ssh_keygen` command.

B – Source VM Host Dead

Service Guard on the VM Host is now supported (HPVM v 2.00). See *Designing High Availability Solutions with Service Guard and Integrity VM* document and "Using Service Guard with Integrity VM" chapter in *HP Integrity Virtual Machines Installation, Configuration and Administration Guide.*

VSE (Virtual Server Environment)

HP VSE Management Software runs under HP System Insight Manager and provides the visualization, configuration, workload policy, and capacity planning tools to optimize the system resources in your VSE. Use HP Integrity Essentials Virtualization Manager to manage a pool of multi-OS, dynamically sizeable virtual servers. Each virtual server can grow and shrink, based on service-level objectives and business requirements, using HP Integrity Essentials Global Workload Manager and the various VSE partitioning and utility pricing technologies. Use HP Integrity Essentials Capacity Advisor to analyze your current server resource utilization and plan for future workload expansion.


HP Systems Insight Manager - Mozilla Firefox
File Edit View Go Bookmarks Tools Help
Back Forward Reload Stop Home New Window Go
https://vmhost.bpo.hp.com:50000/mxportal/home/MxPortalFrames.
Getting Started Latest Headlines PropertyPages.jsp
HP Systems Insight Manager
User: root
Home | Sign Out
Tools Deploy Configure Diagnose Optimize Reports Tasks & Logs Options Help
System Status
Legend... Customize...
Search
Search
Advanced Search...
Systems and Events
Customize...
System Overview
All Systems
All Events
Systems
Private
Shared
Systems by Type
All Systems
All Servers
All VSE Resources
HP Blade System
Storage Systems
All Racks
All Enclosures
All Clients
All Networking Devices
All Printers
All Management Processors
System Workload Shared Resource Domain Capacity Advisor
Tools Create Modify Delete Policy Report
Deselect All
Refresh Data
vmhost.bpo.hp.com:50000
FoxTrick

Updated: Thu, 4/6/2006, 12:33:45 PM CEST

0 0 0 8 Uncleared Event Status

VSE Management: Virtualization Manager

All VSE Resources

Utilization Metric: ☑ CPU(%) ☑ Memory(%) ☑ Network I/O(Mbps) ☑ Disk I/O (Mbps)

System	CPU	Memory	LAN	Disk
vmhost ia64 hp server rx4640 View: VM Host	4 %	58 %	0 MBps	0 MBps
vmguest1 ia64 hp server Integrity Virtual Machine View: VM	0 %	66 %	0 MBps	0 MBps
vmguest2 ia64 hp server Integrity Virtual Machine View: VM	0 %	40 %	0 MBps	0 MBps
vmhost.bpo.hp.com.O... View: Workload	1 %	17 %		

The following documents, available at Business Support Center (BCS), may be useful in extending your knowledge of VSE Management Software:

- *VSE Management Software Quick Start Guide*
- *Getting Started with HP Integrity Essentials Global Workload Manager*
- *Getting Started with HP Integrity Essentials Capacity Advisor*
- *Getting Started with HP Integrity Virtual Machines Manager*
- *Getting Started with HP Integrity Virtual Machines white paper*
- *HP Integrity Essentials Global Workload Manager Administrator's Guide*
- *HP Integrity Essentials Virtual Machines Installation, Configuration, and Administration*

Release History

A.01.00 *(T2767AC)*

- First Release.
- This release of Integrity Virtual Machines supports the HP-UX 11i v2 May 2005 operating system for both the VM Host and the guest.

A.01.20 *(T2767AC)*

- hpvmmigrate command supported by installing an additional bundle (VMMigrate)
- Operational problems corrected.

A.02.00 *(T2767AC)*

- Guests can be configured as Serviceguard packages (distributed guests)
- Guests can run Serviceguard
- Guests can run Microsoft Windows 2003 (SP1)
- Enhancement to the hpvmmigrate command:
  - a running guest is automatically stopped before migration
  - distributed guests are not disabled or deleted on the source after a successful migration.
- Storage enhancements:
  - Attached I/O (tape, changer, CD/DVD burner) can be used as backing stores and can be used as backup devices.
  - A storage device can hold up to 2TB.
  - Veritas Multipath (DMP) can be used on VM Host for guest backing storage devices.
  - Guests can have up to 30 storage devices.
- Networking enhancement:
  - Support for Virtual LAN (VLAN)

A.03.00 (T2767AC)

- Guests can run any of the following operating systems:

  - HP-UX 11.23 (May 2005 [0505] or later)
  - HP-UX 11.31
  - Windows 2003 (Enterprise or Datacenter edition)
  - RedHat Linux Enterprise Edition Advanced Server Release 4 update 4.

- Administrators can dynamically change the size of memory allocated to the HP-UX guest. This feature includes:
  - Changes to the hpvmcreate, hpvmmodify, and hpvmclone commands to configure the guest.
  - Changes to the hpvmstatus command to view dynamic memory data for a configured guest.
  - A new guest-only utility (hpvmmgmt) to view guest dynamic memory data.
- Networking enhancements include:
  - Administrators can change the physical network interface card (pNIC) for a vswitch using the -C option to the hpvmnet command.
  - Administrators can clone the selected vswitch to a newly named vswitch using the —N option to the hpvmnet command. The cloned vswitch will have the same VLAN port information configuration data as the parent vswitch.
- Multiple virtual machine administrator and operator accounts can be created to manage virtual machines remotely.
- Workloads can be migrated from a physical Integrity system or nPar to a virtual machine using the P2V (physical to virtual) assistant.
- Attached I/O (tape, changer, CD/DVD burner) can be used as backing storage by all types of guests.

A.03.05 (T2767AC)

- Guests can run any of the following operating systems:
  - HP-UX 11i v2 (May 2005 [0505] or later), including HP–UX 11i v2 0712
  - HP-UX 11i v3
  - Windows 2003 (Enterprise or Datacenter edition) SP1 and SP2

  - Red Hat® Linux Enterprise Edition Advanced Server Release 4 update 4, update 5, and update 6.
  - SUSE® Linux Enterprise Server (SLES) for HP Integrity servers SLES 10 update 1.

- Integrity VM now includes the capability of Accelerated Virtual I/O (AVIO), which improves the performance of both storage and network access for virtual machines. AVIO is supported on HP-UX 11i v2 host and guests. This feature includes:
  - Changes to the *hpvmnet* command. The changes include displaying the adapter type for each port defined on the vswitch and getting port specific statistics.
  - Changes to the *hpvmresources* and *hpvmstatus* manpages to include the new AVIO names for guest creation: avio_lan, and avio_stor.
  - Change to the hpvmcollect command to add values for AVIO LANs and storage.
  - Changes to the hpvmcreate, hpvmclone, and hpvmmodify commands to add the designation for an AVIO network adapter and an AVIO storage adapter on a guest.
  - Changes to the hpvmclone command to clone guests with additional AVIO network or storage devices.
- AVIO storage support for Logical Volume backing storage devices (virtual LvDisk) on HostAVIOStor B.11.23.0712.01 and later.
- AVIO now supports Active-Passive configuration on EVA GL series (3000/5000) starting with HostAVIOStor B.11.23.0712.01.
- Hierarchy checking
- New public APIs defined in /opt/hpvm/include/hpvm_api_public.h for host and guests. The following APIs have been added to Integrity VM:
  - hpvm_api_server_check — Checks if running on an Integrity VM server system.
  - hpvm_api_virtmach_check — Checks if running on an Integrity VM virtual machine.

  - hpvm_api_version_get — Gets the version string of an Integrity VM server or virtual machine.
  - hpvm_api_my_uuid_get — Gets the uuid for this running Integrity VM server or virtual machine.
  - hpvm_api_server_uuid_get — Gets the uuid for the Integrity VM server of the virtual machine running this API.
  - hpvm_api_server_hostname_get — Gets the host name for the Integrity VM server of the virtual machine running this API.

- The EFI direct tape boot functionality has been added to Integrity VM.
- Change to the hpvminfo command to display the information returned by the supported public interfaces defined in /opt/hpvm/include/hpvm_api_public.h.
- Storage used by the VM Host system is now better protected in this release

For release history of later versions, see WTEC's Version and Patches page.

HPVMs vs vPars

	Virtual Partitions **Coarse soft partitioning with dedicated resources**	**Integrity Virtual Machines** **Fine-grained, virtualized soft partitioning with shared resources**
Servers supported	– PA-RISC cell-based servers – Integrity cell-based servers running HP-UX 11i v2 May 2005 update (or later)	– Integrity servers
CPU Resources	Dedicated, CPU (core) granularity	Shared, sub-CPU granularity
CPU Allocation	Dynamic CPU migration	Fast, automatic CPU reallocation based on demand or entitlement
Networking	Dedicated HW per partition	Shared or dedicated
Mass Storage	Dedicated HW per partition	Virtualized with multiple backing devices, HW sharing
Manageability	CLI and GUI (display only)	CLI and GUI
Security vs. flexibility	Root privilege on any vPar can reconfig vPars	Configuration of VMs occurs with privileged operator in VM host
Performance	Minimal overhead - dedicated resources	virtualization overhead – depends on workload
OS	HPUX 11i	HPUX 11i v2, Windows 2003 EE SP1 (guest OS)

References

Integrity Virtual Machines and Virtual Partitions documentation

HP Integrity Virtual Machines Installation, Configuration and Administration

Top Ten Tips for Using Integrity Virtual Machines

Top Ten Tips for Using Virtual Partitions

Using Ignite/UX with Integrity VM (whitepaper)

WTEC Integrity VM (HPVM) & vPars Home

WTEC Integrity VM Known Problems

Ignite UX documentation

Serviceguard documentation

Additional Information

BCS: A Guide to HP-UX Document Collections

Hardware Consolidation with Integrity VM

Using ServiceGuard to Manage Integrity Virtual Machines

Designing High Availability Solutions with ServiceGuard and Integrity VM

Also you can find known issues you might encounter with Integrity VM and operational details that might not be documented elsewhere on Release Notes (BCS).