

Red Hat Enterprise Linux

Para-virtualized Drivers Guide

5.1



Jan Mark Holzer

Don Dutile

Barry Donahue

ISBN:

Publication date:

This document will explain how to install, configure and troubleshoot the para-virtualized drivers for networking and storage I/O in Red Hat Enterprise Linux HVM (Hardware Virtual Machine) full-virtualized guests. Para-virtualized drivers offer increased performance for fully virtualized Red Hat Enterprise Linux guests. It is assumed the reader is familiar with system administration tasks associated with software installation and configuration in Red Hat Enterprise Linux environment.

Red Hat Enterprise Linux: Para-virtualized Drivers Guide

Author	Jan Mark Holzer	<jmh@redhat.com>
Author	Don Dutile	<ddutile@redhat.com>
Author	Barry Donahue	<bdonahue@redhat.com>
Editor	Chris Curran	<ccurran@redhat.com>

Copyright © 2008 Red Hat, Inc.

Copyright © 2008 Red Hat, Inc.. This material may only be distributed subject to the terms and conditions set forth in the Open Publication License, V1.0 or later with the restrictions noted below (the latest version of the OPL is presently available at <http://www.opencontent.org/openpub/>).

Distribution of substantively modified versions of this document is prohibited without the explicit permission of the copyright holder.

Distribution of the work or derivative of the work in any standard (paper) book form for commercial purposes is prohibited unless prior permission is obtained from the copyright holder.

Red Hat and the Red Hat "Shadow Man" logo are registered trademarks of Red Hat, Inc. in the United States and other countries.

All other trademarks referenced herein are the property of their respective owners.

The GPG fingerprint of the security@redhat.com key is:

CA 20 86 86 2B D6 9D FC 65 F6 EC C4 21 91 80 CD DB 42 A6 0E

1801 Varsity Drive
Raleigh, NC 27606-2072
USA
Phone: +1 919 754 3700
Phone: 888 733 4281
Fax: +1 919 754 3701
PO Box 13588
Research Triangle Park, NC 27709
USA

Preface	vii
1. Document Conventions	vii
2. We Need Your Feedback	viii
1. Introduction to Para-virtualized Drivers	1
1. System requirements	2
2. Para-virtualization Restrictions and Support	5
3. Installation and Configuration of Para-virtualized Drivers	9
1. Common installation steps	9
4. Installation and Configuration of Para-virtualized Drivers on Red Hat Enterprise Linux 3	11
5. Installation and Configuration of Para-virtualized Drivers on Red Hat Enterprise Linux 4	17
6. Installation and Configuration of Para-virtualized Drivers on Red Hat Enterprise Linux 5	21
7. Para-virtualized Network Driver Configuration	25
8. Additional Para-virtualized Hardware Configuration	31
1. Virtualized Network Interfaces	31
2. Virtual Storage Devices	32
9. Troubleshooting Para-virtualized Drivers	35
1. Red Hat Enterprise Linux 5 Virtualization log file and directories	35
2. Para-virtualized guest fail to load on a Red Hat Enterprise Linux 3 guest operating system	37
3. A warning message is displayed while installing the para-virtualized drivers on Red Hat Enterprise Linux 3	37
4. What to do if the guest operating system has been booted with <code>virt-manager</code> or <code>virsh</code>	38
5. Manually loading the para-virtualized drivers	40
6. Verifying the para-virtualized drivers have successfully loaded	41
7. The system has limited throughput with para-virtualized drivers	41
10. Compiling para-virtualized driver packages from source code	43
11. Additional Information Links	45
A. Revision History	47

Preface

1. Document Conventions

Certain words in this manual are represented in different fonts, styles, and weights. This highlighting indicates that the word is part of a specific category. The categories include the following:

Courier font

Courier font represents commands, file names and paths, and prompts .

When shown as below, it indicates computer output:

```
Desktop      about.html   logs         paulwesterberg.png
Mail         backupfiles mail         reports
```

bold Courier font

Bold Courier font represents text that you are to type, such as: **service jonas start**

If you have to run a command as root, the root prompt (#) precedes the command:

```
# gconftool-2
```

italic Courier font

Italic Courier font represents a variable, such as an installation directory:

```
install_dir/bin/
```

bold font

Bold font represents **application programs** and **text found on a graphical interface**.

When shown like this: **OK** , it indicates a button on a graphical application interface.

Additionally, the manual uses different strategies to draw your attention to pieces of information. In order of how critical the information is to you, these items are marked as follows:



Note

A note is typically information that you need to understand the behavior of the system.



Tip

A tip is typically an alternative way of performing a task.



Important

Important information is necessary, but possibly unexpected, such as a configuration change that will not persist after a reboot.



Caution

A caution indicates an act that would violate your support agreement, such as recompiling the kernel.



Warning

A warning indicates potential data loss, as may happen when tuning hardware for maximum performance.

2. We Need Your Feedback

If you find a typographical error in the Para-Virtualized Drivers for Red Hat Enterprise Linux, or if you have thought of a way to make this manual better, we would love to hear from you! Please submit a report in Bugzilla: <http://bugzilla.redhat.com/bugzilla/> against the component *Red_Hat_Enterprise_Linux*.

When submitting a bug report, be sure to mention the manual's identifier, *Para-Virtualized_Drivers*.

If you have a suggestion for improving the documentation, try to be as specific as possible when describing it. If you have found an error, please include the section number and some of the surrounding text so we can find it easily.

Introduction to Para-virtualized Drivers

Para-virtualized drivers provide increased performance for fully virtualized Red Hat Enterprise Linux guests. Use these drivers if you are using fully virtualized Red Hat Enterprise Linux guests and require better performance.

The RPM packages for the para-virtualized drivers include the modules for storage and networking para-virtualized drivers for the supported Red Hat Enterprise guest operating systems. These drivers enable high performance throughput of I/O operations in unmodified Red Hat Enterprise Linux guest operating systems on top of a Red Hat Enterprise Linux 5.1 (or greater) host.

The supported guest operating systems are:

- Red Hat Enterprise Linux 3
- Red Hat Enterprise Linux 4
- Red Hat Enterprise Linux 5



Architecture support for para-virtualized drivers

The minimum guest operating system requirements are architecture dependent. Only x86 and x86-64 guests are supported.

The drivers are not supported on Red Hat Enterprise Linux guest operating systems prior to Red Hat Enterprise Linux 3 .

Using Red Hat Enterprise Linux 5 as the virtualization platform allows System Administrators to consolidate Linux and Windows workloads onto newer, more powerful hardware with increased power and cooling efficiency. Red Hat Enterprise Linux 4 (as of update 6) and Red Hat Enterprise Linux 5 guest operating systems are aware of the underlying virtualization technology and can interact with it efficiently using specific interfaces and capabilities. This approach can achieve similar throughput and performance characteristics compared to running on the bare metal system.

As this approach requires modifications in the guest operating system not all operating systems and use models can use para-virtualized virtualization. For operating systems which can not be modified the underlying virtualization infrastructure has to emulate the server hardware (CPU, Memory as well as IO devices for storage and network). Emulation for IO devices can be very slow and will be especially troubling for high-throughput disk and network subsystems. The majority of the performance loss occurs in this area.

The para-virtualized device drivers part of the distributed RPM packages bring many of the performance advantages of para-virtualized guest operating systems to unmodified operating systems because only the para-virtualized device driver (but not the rest of the operating system) is aware of the underlying virtualization platform.

After installing the para-virtualized device drivers, a disk device or network card will continue to appear as a normal, physical disk or network card to the operating system. However, now the device driver interacts directly with the virtualization platform (with no emulation) to efficiently deliver disk and network access, allowing the disk and network subsystems to operate at near native speeds even in a virtualized environment, without requiring changes to existing guest operating systems.

The para-virtualized drivers have certain host requirements. 64 bit hosts can run:

- 32 bit guests.
- 64 bit guests.
- a mixture of 32 bit and 64 bit guests.

The para-virtualized drivers only work on 32 bit Red Hat Enterprise Linux hosts for 32 bit guests.

1. System requirements

This section provides the requirements for para-virtualized drivers with Red Hat Enterprise Linux.

Installation.

Before you install the para-virtualized drivers the following requirements (listed below) must be met.

You will need the following RPM packages for para-virtualized drivers for each guest operating system installation.

Red Hat Enterprise Linux 5 requires:

- `kmod-xenpv`.

Red Hat Enterprise Linux 4 requires:

- `kmod-xenpv`,
- `modules-init-tools` (for versions prior to Red Hat Enterprise Linux 4.6z you require `modules-init-tools-3.1-0.pre5.3.4.e14_6.1` or greater), and
- `modversions`.

Red Hat Enterprise Linux 3 requires:

- `kmod-xenpv`.

Minimum host operating system version

- Red Hat Enterprise Linux 5.1 or higher

Minimum guest operating system version

- Red Hat Enterprise Linux 5.1 and higher
- Red Hat Enterprise Linux 4 Update 6 and higher
- Red Hat Enterprise Linux 3 Update 9 and higher

You require at least 50MB of free disk space in the `/lib` file system

Para-virtualization Restrictions and Support

This section outlines support restrictions and requirements for using para-virtualized drivers on Red Hat Enterprise Linux. What we support and the restrictions put upon support can be found in the sections below.

Supported Guest Operating Systems.

Support for para-virtualized drivers is available for the following operating systems and versions:

- Red Hat Enterprise Linux 5.1
- Red Hat Enterprise Linux 4 Update 6
- Red Hat Enterprise Linux 3 Update 9

You are supported for running a 32 bit guest operating system with para-virtualized drivers on 64 bit Red Hat Enterprise Linux 5 Virtualization.

The table below indicates the kernel variants supported with the para-virtualized drivers. You can use the command shown below to identify the exact kernel revision currently installed on your host. Compare the output against the table to determine if it is supported.

```
# rpm -q --queryformat '%{NAME}-%{VERSION}-%{RELEASE}.%{ARCH}\n' kernel
```

The Red Hat Enterprise Linux 5 i686 and x86_64 kernel variants include Symmetric Multiprocessing(SMP), no separate SMP kernel RPM is required.

Take note of processor specific kernel requirements for Red Hat Enterprise Linux 3 Guests in the table below.

Kernel Architecture	Red Hat Enterprise Linux 3	Red Hat Enterprise Linux 4	Red Hat Enterprise Linux 5
athlon	Supported(AMD)		
athlon-SMP	Supported(AMD)		
i32e	Supported(Intel)		
i686	Supported(Intel)	Supported	Supported
i686-PAE			Supported
i686-SMP	Supported(Intel)	Supported	
i686-HUGEMEM	Supported(Intel)	Supported	
x86_64	Supported(AMD)	Supported	Supported

Kernel Architecture	Red Hat Enterprise Linux 3	Red Hat Enterprise Linux 4	Red Hat Enterprise Linux 5
x86_64-SMP	Supported(AMD)	Supported	
x86_64-LARGESMP		Supported	
Itanium (IA64)			Supported

Table 2.1. Supported kernel architectures for para-virtualized drivers



Note

The table above is for guest operating systems. AMD and Intel processors are supported for the Red Hat Enterprise Linux 5.1 host.



Take note

Write the output of the command below down or remember it. This is the value that determines which packages and modules you need to download.

```
# rpm -q --queryformat '%{NAME}-%{VERSION}-%{RELEASE} .%{ARCH}\n'
kernel
```

Your output should appear similar to this:

```
kernel-PAE-2.6.18-53.1.4.el5.i686
```

The name of the kernel is PAE(Physical Address Extension), kernel version is 2.6.18, the release is 53.1.4.el5 and the architecture is i686. The kernel rpm should always be in the format **kernel-name-version-release.arch.rpm**.

Important Restrictions.

Para-virtualized device drivers can be installed after successfully installing a guest operating system. You will need a functioning host and guest before you can install these drivers.



Para-virtualized block devices and GRUB

GRUB can not presently, access para-virtualized block devices. Therefore, a guest can not be booted from a device that uses the para-virtualized block device drivers. Specifically, the disk that contains the Master Boot Record(MBR), a disk

containing a boot loader (GRUB), or a disk that contains the kernel `initrd` images. That is, any disk which contains the `/boot` directory or partition can not use the para-virtualized block device drivers.

After installing the para-virtualized drivers on a guest operating system you should only use the `xm` command or `virsh` to start the guests. If `xm` is not used the network interfaces (for example, `eth1` and so on) will not be connected correctly during boot. This problem is known and the Bugzilla number is 300531 and a bug fix is in progress. The bug connects the network interface to `qemu-dm` and subsequently limits the performance dramatically.

Red Hat Enterprise Linux 3 kernel variant architecture dependencies.

For Red Hat Enterprise Linux 3 based guest operating systems you must use the processor specific kernel and para-virtualized driver RPMs as seen in the tables below. If you fail to install the matching para-virtualized driver package loading of the `xen-pci-platform` module will fail.

The table below shows which host kernel is required to run a Red Hat Enterprise Linux 3 guest on if the guest was compiled for an Intel processor.

Guest kernel type	Required host kernel type
ia32e (UP and SMP)	x86_64
i686	i686
i686-SMP	i686
i686-HUGEMEM	i686

Table 2.2. Required host kernel architecture for guests using para-virtualized drivers on Red Hat Enterprise Linux 3 for Intel processors

The table below shows which host kernel is required to run a Red Hat Enterprise Linux 3 guest on if the guest was compiled for an AMD processor.

Guest kernel type	Required host kernel type
athlon	i686
athlon-SMP	i686
x86_64	x86_64
x86_64-SMP	x86_64

Table 2.3. Required host kernel architectures for guests using para-virtualized drivers on Red Hat Enterprise Linux 3 for AMD processors



Note

After installing the para-virtualized drivers on a guest operating system you should only use the `xm` command or the `virsh` command to start the guests. If `xm` or `virsh` are not used the network interfaces (for example, `eth1`) will not be correctly connected during boot. This problem is known and the [Bugzilla¹](#) number is 300531. The bug connects the network interface to `qemu-dm` and subsequently limits the performance dramatically. A bug fix is presently in development and will be released via RHN.

Installation and Configuration of Para-virtualized Drivers

The following three chapters describe how to install and configure your fully virtualized guests to run on Red Hat Enterprise Linux 5.1 or above with para-virtualized drivers.



Verify your architecture is supported before proceeding

Para-virtualized drivers are only supported on certain hardware and version combinations. Verify your hardware and operating system requirements are met before proceeding to install para-virtualized drivers.



Maximizing the benefit of the para-virtualized drivers for new installations

If you are installing a new guest system, in order to gain maximal benefit from the para-virtualized block device drivers, you should create the guest with at least two disks.

Specifically, use the first disk to install the MBR and the boot loader (GRUB), and to contain the `/boot` partition. (This disk can be very small, as it only needs to have enough capacity to hold the `/boot` partition.)

Use the second disk and any additional disks for all other partitions (e.g. `/`, `/usr`) or logical volumes.

Using this installation method, when the para-virtualized block device drivers are later installed after completing the install of the guest, only booting the guest and accessing the `/boot` partition will use the virtualized block device drivers.

1. Common installation steps

The list below covers the high level steps common across all guest operating system versions.

1. Copy the RPMs for your hardware architecture to a suitable location in your guest operating system. Your home directory is sufficient. If you do not know which RPM you require verify against the table at [Chapter 2, Para-virtualization Restrictions and Support](#).
2. Use the `rpm` utility to install the RPM packages. The `rpm` utility will install the following four new kernel modules into

- :
- the PCI infrastructure module, `xen-platform-pci.ko`,
 - the ballooning module, `xen-balloon.ko`,
 - the virtual block device module, `xen-vbd.ko`,
 - and the virtual network device module, `xen.vnif.ko`.
3. If the guest operating does not support automatically loading the para-virtualized drivers (for example Red Hat Enterprise Linux 3) perform the required post-install steps to copy the drivers into the operating system specific locations.
 4. Shutdown your guest operating system.
 5. Reconfigure the guest operating system configuration file on the host to use the installed para-virtualized drivers.
 6. Remove the “`type=ioemu`” entry for the network device.
 7. Add any additional storage entities you want to use for the para-virtualized block device driver.
 8. Restart your guest using the “`xm create YourGuestName`” command where *YourGuestName* is the name of the guest operating system.
 9. Reconfigure the guest network

Installation and Configuration of Para-virtualized Drivers on Red Hat Enterprise Linux 3

This section contains detailed instructions for the para-virtualized drivers in a Red Hat Enterprise 3 guest operating system.



Please note

These packages do not support booting from a para-virtualized disk. Booting the guest operating system kernel still requires the use of the emulated IDE driver, while any other (non-system) user-level application and data disks can use the para-virtualized block device driver.

Driver Installation.

The list below covers the steps to install a Red Hat Enterprise Linux 3 guest with para-virtualized drivers.

1. Copy the `kmod-xenpv` rpm corresponding to your hardware architecture and kernel variant to your guest operating system.
2. Use the `rpm` utility to install the RPM packages. Make sure you have correctly identified which package you need for your guest operating system variant and architecture.

```
[root@rhel3]# rpm -ivh kmod-xenpv*
```

3. You need to perform the commands below to enable the correct and automated loading of the para-virtualized drivers. `%kvariant` is the kernel variant the para-virtualized drivers have been build against and `%release` corresponds to the release version of the para-virtualized drivers.

```
[root@rhel3]# mkdir -p /lib/modules/'uname -r'/extra/xenpv
[root@rhel3]# cp -R
/lib/modules/2.4.21-52.EL[%kvariant]/extra/xenpv/%release \
/lib/modules/'uname -r'/extra/xenpv
[root@rhel3]# cd /lib/modules/'uname -r'/extra/xenpv/%release
[root@rhel3]# insmod xen-platform-pci.o
[root@rhel3]# insmod xen-balloon.o`
[root@rhel3]# insmod xen-vbd.o
[root@rhel3]# insmod xen-vnif.o
```



Note

Warnings will be generated by `insmod` when installing the binary driver modules due to Red Hat Enterprise Linux 3 having **MODVERSIONS** enabled. These warnings can be ignored.

4. Verify `/etc/modules.conf` and make sure you have an alias for `eth0` similar to the one below. If you are planning to configure multiple interfaces add an additional line for each interface.

```
alias eth0 xen-vnif
```

Edit `/etc/rc.local` and add the line:

```
insmod /lib/modules/`uname -r`/extra/xenpv/%release/xen-vbd.o
```



Note

Substitute “%release” with the actual release version (for example 0.1-5.el) for the para-virtualized drivers. If you update the para-virtualized driver RPM package make sure you update the release version to the appropriate version.

5. Shutdown the virtual machine (use “`#shutdown -h now`” inside the guest).
6. Edit the guest configuration file in `/etc/xen/YourGuestsName` in the following ways:
 - Remove the “`type=ioemu`” entry from the “`vif=`” entry.
 - Add any additional disk partitions, volumes or LUNs to the guest so that they can be accessed via the para-virtualized (`xen-vbd`) disk driver.
 - For each additional physical device, LUN, partition or volume add an entry similar to the one below to the “`disk=`” section in the guest configuration file. The original “`disk=`” entry might also look like the entry below.

```
disk = [ "file:/var/lib/xen/images/rhel3_64_fv.dsk,hda,w" ]
```

- Once you have added additional physical devices, LUNs, partitions or volumes; the para-virtualized driver entry in your XML configuration file should resemble the entry shown below.

```
disk = [ "file:/var/lib/xen/images/rhel3_64_fv.dsk,hda,w",  
"tap:aio:/var/lib/xen/images/UserStorage.dsk,xvda,w" ]
```



Note

Use "tap:aio" for the para-virtualized device if a file based image is used.

7. Boot the virtual machine using the `xm` command:

```
# xm create YourGuestName
```



Note

You must use "`xm create <virt-machine-name>`" on Red Hat Enterprise Linux 5.1. The para-virtualized network driver(`xen-vnif`) will not be connected to `eth0` properly if you are using Red Hat Enterprise Linux 5.1 and the `virt-manager` or `virsh` interfaces. This issue is currently a known bug, BZ 300531.

Red Hat Enterprise Linux 5.2 does not have this bug and the `virt-manager` or `virsh` interfaces will correctly load the para-virtualized drivers.



Be aware

The para-virtualized drivers are not automatically added and loaded to the system because `weak-modules` and `modversions` support is not provided in Red Hat Enterprise Linux 3. To insert the module execute the command below.

```
insmod xen-vbd.ko
```

Red Hat Enterprise Linux 3 requires the manual creation of the special files for the block devices which use `xen-vbd`. The steps below will cover how to create and register para-virtualized block devices.

Use the following script to create the special files after the para-virtualized block device driver is loaded.

```
#!/bin/sh  
module="xvd"
```

```
mode="664"
major=`awk "\\$2==\"$module\" {print \\$1}" /proc/devices`
# < mknod for as many or few partitions on xvd disk attached to FV guest >
# change/add xvda to xvdb, xvdb1, etc. for 2nd, 3rd, etc., disk added in
# in xen config file, respectively.
mknod /dev/xvdb b $major 0
mknod /dev/xvdb1 b $major 1
mknod /dev/xvdb2 b $major 2
chgrp disk /dev/xvd*
chmod $mode /dev/xvd*
```

For each additional virtual disk, increment the minor number by 16. In the example below an additional device, minor number 16, is created.

```
mknod /dev/xvdc b $major 16
mknod /dev/xvdc1 b $major 17
```

This would make the next device 32 which can be created by:

```
mknod /dev/xvdd b $major 32
mknod /dev/xvdd1 b $major 33
```

Now you should verify the partitions which you have created are available.

```
[root@rhel3]# cat /proc/partitions
major  minor  #blocks  name
3       0       10485760  hda
3       1       104391   hda1
3       2       10377990  hda2
202    0       64000    xvdb
202    1       32000    xvdb1
202    2       32000    xvdb2
253    0       8257536  dm-0
253    1       2031616  dm-1
```

In the above output, you can observe that the partitioned device “xvdb” is available to the system.

The commands below mount the new block devices to local mount points and updates the `/etc/fstab` inside the guest to mount the devices/partitions during boot.

```
[root@rhel3]# mkdir /mnt/pvdisk_p1
[root@rhel3]# mkdir /mnt/pvdisk_p2
[root@rhel3]# mount /dev/xvdb1 /mnt/pvdisk_p1
[root@rhel3]# mount /dev/xvdb2 /mnt/pvdisk_p2

[root@rhel3]# df /mnt/pvdisk_p1
Filesystem            1K-blocks    Used   Available Use%   Mounted on
```

/dev/xvdb1	32000	15	31985	1%	/mnt/pvdisk_p1
------------	-------	----	-------	----	----------------



Performance tip

Using a Red Hat Enterprise Linux 5.1 host(dom0), the "noapic" parameter should be added to the kernel boot line in your virtual guest's `/boot/grub/grub.conf` entry as seen below. Keep in mind your architecture and kernel version may be different.

```
kernel /vmlinuz-2.6.9-67.EL ro root=/dev/VolGroup00/rhel4_x86_64
rhgb noapic
```

A Red Hat Enterprise Linux 5.2 dom0 will not need this kernel parameter for the guest.



Please note

The Itanium (ia64) binary RPM packages and builds are not presently available.

Installation and Configuration of Para-virtualized Drivers on Red Hat Enterprise Linux 4

This section contains detailed instructions for the para-virtualized drivers in a Red Hat Enterprise 4 guest operating system.



Please note

These packages do not support booting from a para-virtualized disk. Booting the guest operating system kernel still requires the use of the emulated IDE driver, while any other (non-system) user-level application and data disks can use the para-virtualized block device driver.

Driver Installation.

The list below covers the steps to install a Red Hat Enterprise Linux 4 guest with para-virtualized drivers.

1. Copy the `kmod-xenpv`, `modules-init-tools` and `modversions` RPMs corresponding to your hardware architecture and kernel variant to your guest operating system.
2. Use the `rpm` utility to install the RPM packages. Make sure you have correctly identified which package you need for your guest operating system variant and architecture. An updated `module-init-tools` is required for this package, it is available with the Red Hat Enterprise Linux4-6-z kernel and beyond.

```
[root@rhel4]# rpm -ivh modversions
[root@rhel4]# rpm -Uvh module-init-tools
[root@rhel4]# rpm -ivh kmod-xenpv*
```



Note

There are different packages for UP, SMP, Hugesmem and architectures so make sure you have the right RPMs for your kernel.

3. Execute `cat /etc/modules.conf` to verify you have an alias for `eth0` similar to the one below. If you are planning to configure multiple interfaces add an additional line for each interface. If it does not look like the entry below change it.

```
alias eth0 xen-vnif
```

4. Shutdown the virtual machine (use “#shutdown -h now” inside the guest).
5. Edit the guest configuration file in `/etc/xen/YourGuestsName` in the following ways:
 - Remove the “`type=ioemu`” entry from the “`vif=`” entry.
 - Add any additional disk partitions, volumes or LUNs to the guest so that they can be accessed via the para-virtualized (`xen-vbd`) disk driver.
 - For each additional physical device, LUN, partition or volume add an entry similar to the one shown below to the “`disk=`” section in the guest configuration file. The original “`disk=`” entry might also look like the entry below.

```
disk = [ "file:/var/lib/xen/images/rhel4_64_fv.dsk,hda,w" ]
```

- Once you have added additional physical devices, LUNs, partitions or volumes; the para-virtualized driver entry in your XML configuration file should resemble the entry shown below.

```
disk = [ "file:/var/lib/xen/images/rhel3_64_fv.dsk,hda,w",  
"tap:aio:/var/lib/xen/images/UserStorage.dsk,xvda,w" ]
```



Note

Use “`tap:aio`” for the para-virtualized device if a file based image is used.

6. Boot the virtual machine using the `xm` command:

```
# xm create YourGuestName
```



Note

You must use “`xm create <virt-machine-name>`” on Red Hat Enterprise Linux 5.1. The para-virtualized network driver (`xen-vnif`) will not be connected to `eth0` properly if you are using Red Hat Enterprise Linux 5.1 and the `virt-manager` or `virsh` interfaces. This issue is currently a known bug, BZ 300531.

Red Hat Enterprise Linux 5.2 does not have this bug and the `virt-manager` or

`virsh` interfaces will correctly load the para-virtualized drivers.

On the first reboot of the virtual guest, `kudzu` will ask you to "Keep or Delete the Realtek Network device" and "Configure the xen-bridge device". You should configure the `xen-bridge` and delete the Realtek network device.



Performance tip

Using a Red Hat Enterprise Linux 5.1 host(`dom0`), the "`noapic`" parameter should be added to the kernel boot line in your virtual guest's `/boot/grub/grub.conf` entry as seen below. Keep in mind your architecture and kernel version may be different.

```
kernel /vmlinuz-2.6.9-67.EL ro root=/dev/VolGroup00/rhel4_x86_64
rhgb noapic
```

A Red Hat Enterprise Linux 5.2 `dom0` will not need this kernel parameter for the guest.

Now, verify the partitions which you have created are available.

```
[root@rhel4]# cat /proc/partitions
major    minor    #blocks  name
   3         0    10485760  hda
   3         1     104391  hda1
   3         2    10377990  hda2
  202         0     64000  xvdb
  202         1     32000  xvdb1
  202         2     32000  xvdb2
  253         0    8257536  dm-0
  253         1    2031616  dm-1
```

In the above output, you can see the partitioned device "`xvdb`" is available to the system.

The commands below mount the new block devices to local mount points and updates the `/etc/fstab` inside the guest to mount the devices/partitions during boot.

```
[root@rhel4]# mkdir /mnt/pvdisk_p1
[root@rhel4]# mkdir /mnt/pvdisk_p2
[root@rhel4]# mount /dev/xvdb1 /mnt/pvdisk_p1
[root@rhel4]# mount /dev/xvdb2 /mnt/pvdisk_p2
```

```
[root@rhel4]# df /mnt/pvdisk_p1
Filesystem          1K-blocks    Used   Available Use%    Mounted on
/dev/xvdb1          32000        15      31985    1%    /mnt/pvdisk_p1
```



Note

This package is not supported for Red Hat Enterprise Linux 4-GA through Red Hat Enterprise Linux 4 update 2 systems and kernels.



Also note...

IA64 binary RPM packages and builds are not presently available.



A handy tip

If the `xen-vbd` driver does not automatically load. Issue the following command from the guest's terminal. Substitute `%release` with the correct release version for the para-virtualized drivers.

```
[root@rhel4]# insmod /lib/modules/'uname
-r' /weak-updates/xenpv/%release/xen-vbd.ko
```

Installation and Configuration of Para-virtualized Drivers on Red Hat Enterprise Linux 5

This section contains detailed instructions for the para-virtualized drivers in a Red Hat Enterprise 5 guest operating system.



Please note

These packages do not support booting from a para-virtualized disk. Booting the guest operating system kernel still requires the use of the emulated IDE driver, while any other (non-system) user-level application and data disks can use the para-virtualized block device driver.

Driver Installation.

The list below covers the steps to install a Red Hat Enterprise Linux 5 guest with para-virtualized drivers.

1. Copy the `kmod-xenpv` rpm corresponding to your hardware architecture and kernel variant to your guest operating system.
2. Use the `rpm` utility to install the RPM packages. Make sure you correctly identify which package you need for your guest operating system variant and architecture.

```
[root@rhel5]# rpm -ivh kmod-xenpv*
```

3. Issue the command below to disable automatic hardware detection inside the guest operating system

```
[root@rhel5]# chkconfig kudzu off
```

4. Execute `cat /etc/modules.conf` to verify you have an alias for `eth0` similar to the one below. If you are planning to configure multiple interfaces add an additional line for each interface. If it does not look like the entry below change it.

```
alias eth0 xen-vnif
```

5. Shutdown the virtual machine (use “`#shutdown -h now`” inside the guest).

6. Edit the guest configuration file in `/etc/xen/<Your GuestName>` in the following ways:

- Remove the `"type=ioemu"` entry from the `"vif="` entry.
- Add any additional disk partitions, volumes or LUNs to the guest so that they can be accessed via the para-virtualized (`xen-vbd`) disk driver.
- For each additional physical device, LUN, partition or volume add an entry similar to the one shown below to the `"disk="` section in the guest configuration file. The original `"disk="` entry might also look like the entry below.

```
disk = [ "file:/var/lib/xen/images/rhel4_64_fv.dsk,hda,w" ]
```

- Once you have added additional physical devices, LUNs, partitions or volumes; the para-virtualized driver entry in your XML configuration file should resemble the entry shown below.

```
disk = [ "file:/var/lib/xen/images/rhel3_64_fv.dsk,hda,w",  
"tap:aio:/var/lib/xen/images/UserStorage.dsk,xvda,w" ]
```



Note

Use `"tap:aio"` for the para-virtualized device if a file based image is used.

7. Boot the virtual machine using the `xm` command:

```
# xm create YourGuestName
```



Note

You must use `"xm create <virt-machine-name>"` on Red Hat Enterprise Linux 5.1. The para-virtualized network driver (`xen-vnif`) will not be connected to `eth0` properly if you are using Red Hat Enterprise Linux 5.1 and the `virt-manager` or `virsh` interfaces. This issue is currently a known bug, BZ 300531.

Red Hat Enterprise Linux 5.2 does not have this bug and the `virt-manager` or `virsh` interfaces will correctly load the para-virtualized drivers.

To verify the network interface has come up after installing the para-virtualized drivers issue the following command on the guest. It should display the interface information including an assigned IP address

```
[root@rhel5]# ifconfig eth0
```

Now, verify the partitions which you have created are available.

```
[root@rhel5]# cat /proc/partitions
major minor #blocks name
 3      0 10485760 hda
 3      1  104391 hda1
 3      2 10377990 hda2
202     0   64000 xvdb
202     1   32000 xvdb1
202     2   32000 xvdb2
253     0 8257536 dm-0
253     1 2031616 dm-1
```

In the above output, you can see the partitioned device "xvdb" is available to the system.

The commands below mount the new block devices to local mount points and updates the `/etc/fstab` inside the guest to mount the devices/partitions during boot.

```
[root@rhel5]# mkdir /mnt/pvdisk_p1
[root@rhel5]# mkdir /mnt/pvdisk_p2
[root@rhel5]# mount /dev/xvdb1 /mnt/pvdisk_p1
[root@rhel5]# mount /dev/xvdb2 /mnt/pvdisk_p2
[root@rhel5]# df /mnt/pvdisk_p1
Filesystem          1K-blocks      Used    Available Use%    Mounted on
/dev/xvdb1           32000          15        31985    1%    /mnt/pvdisk_p1
```



Performance tip

Using a Red Hat Enterprise Linux 5.1 host(dom0), the "noapic" parameter should be added to the kernel boot line in your virtual guest's `/boot/grub/grub.conf` entry as seen below. Keep in mind your architecture and kernel version may be different.

```
kernel /vmlinuz-2.6.9-67.EL ro root=/dev/VolGroup00/rhel4_x86_64
rhgb noapic
```

A Red Hat Enterprise Linux 5.2 dom0 will not need this kernel parameter for the guest.

Para-virtualized Network Driver Configuration

Once the para-virtualized network driver is loaded you may need to reconfigure the guest's network interface to reflect the driver and virtual Ethernet card change.

Perform the following steps to reconfigure the network interface inside the guest.

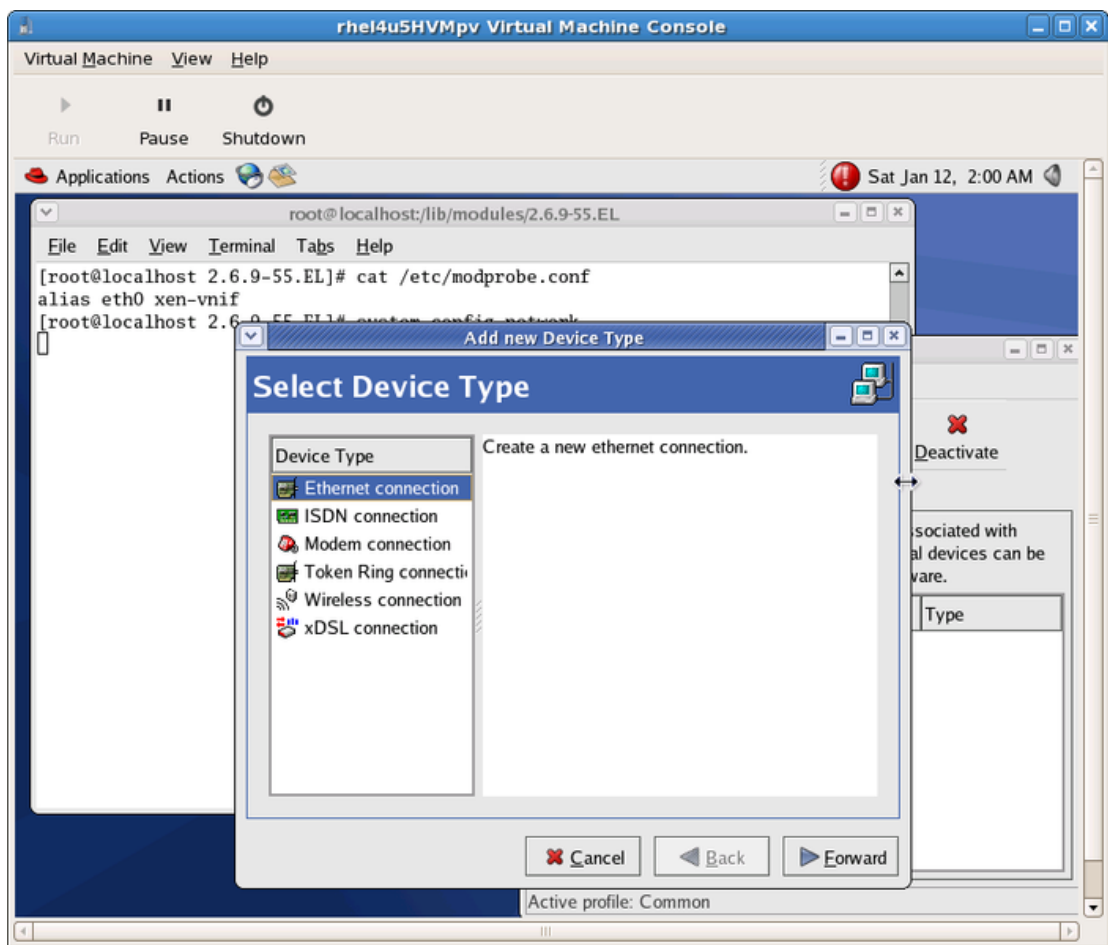
1. In `virt-manager` open the console window for the guest and log in as `root`.
2. On Red Hat Enterprise Linux 4 verify the file `/etc/modprobe.conf` contains the line `alias eth0 xen-vnif`.

```
# cat /etc/modprobe.conf
alias eth0 xen-vnif
```

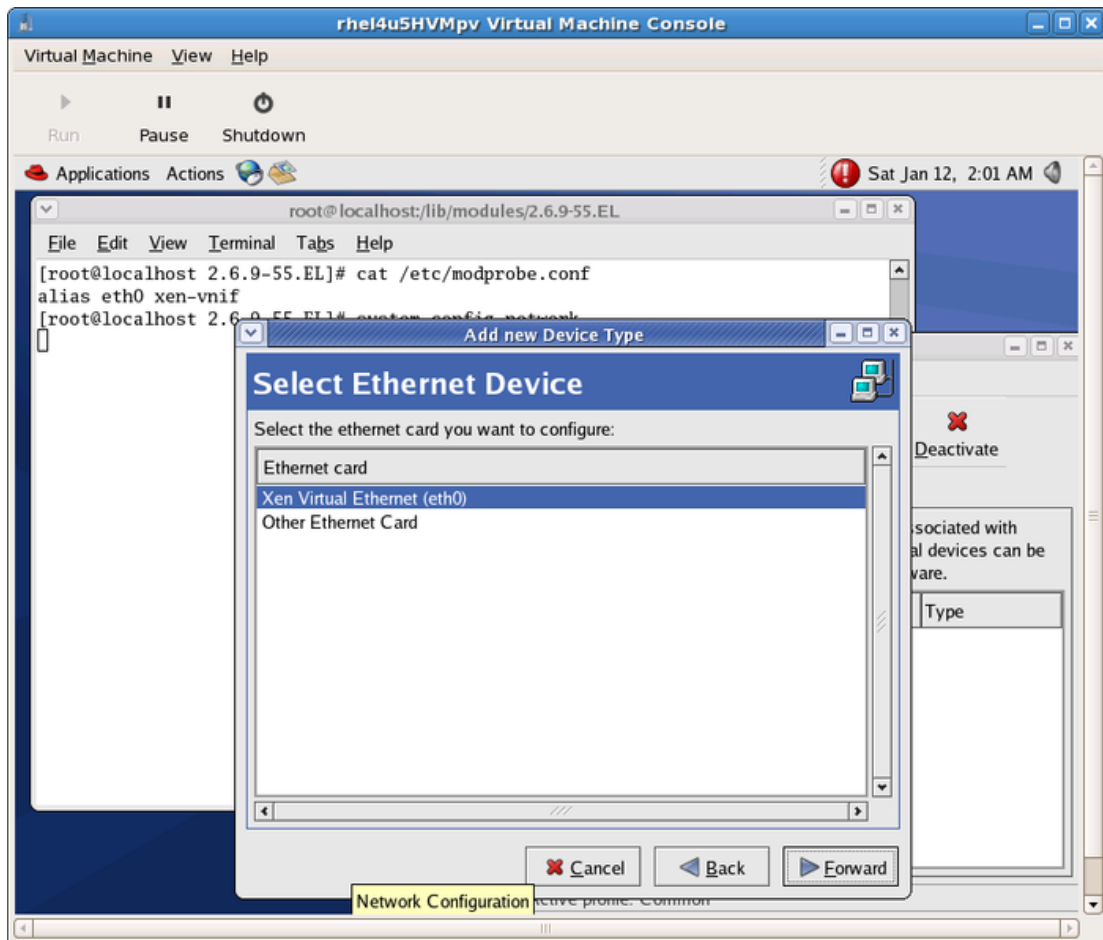
3. To display the present settings for `eth0` execute `# ifconfig eth0`. If you receive an error about the device not existing you should load the modules manually as outlined in [Section 5, "Manually loading the para-virtualized drivers"](#).

```
ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 00:00:00:6A:27:3A
          BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:630150 errors:0 dropped:0 overruns:0 frame:0
          TX packets:9 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:109336431 (104.2 MiB)  TX bytes:846 (846.0 b)
```

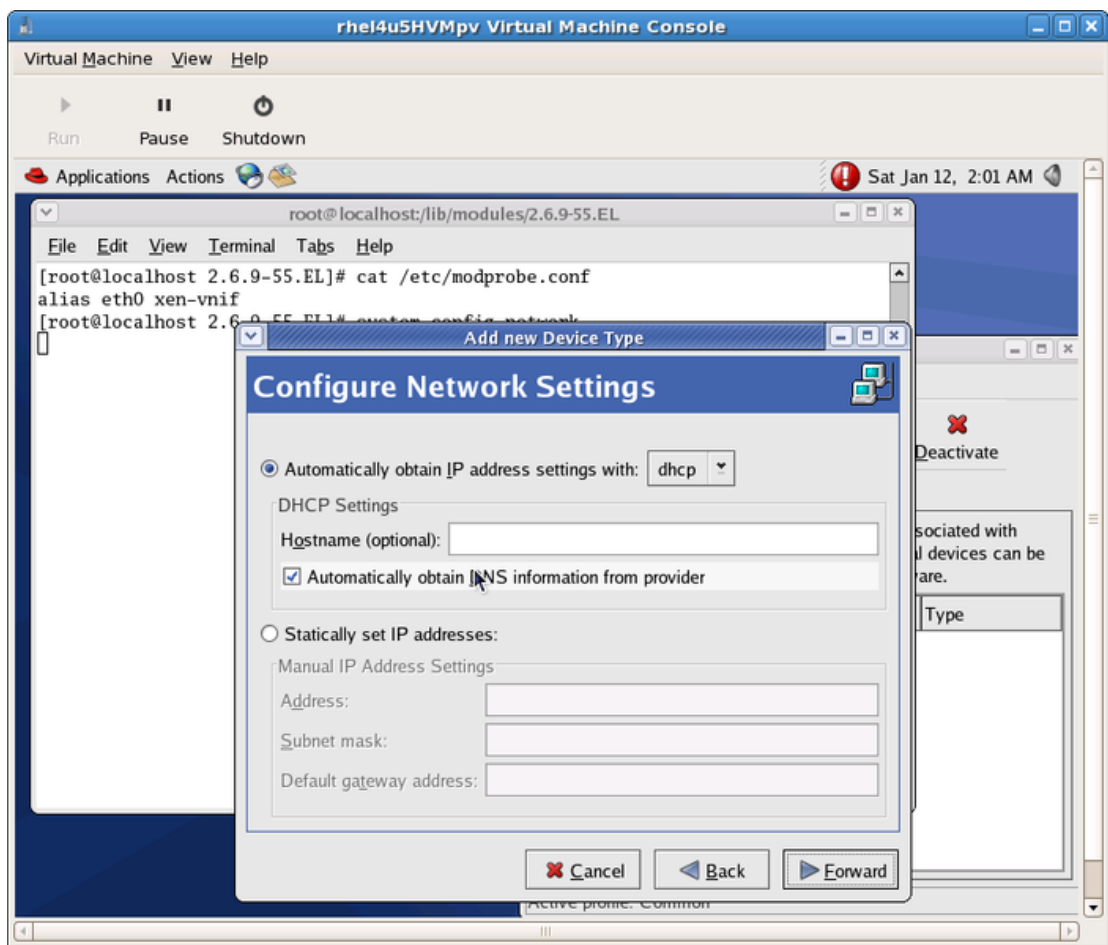
4. Start the network configuration utility(NetworkManager) with the command `# system-config-network`. Click on the **Forward** button to start the network card configuration.



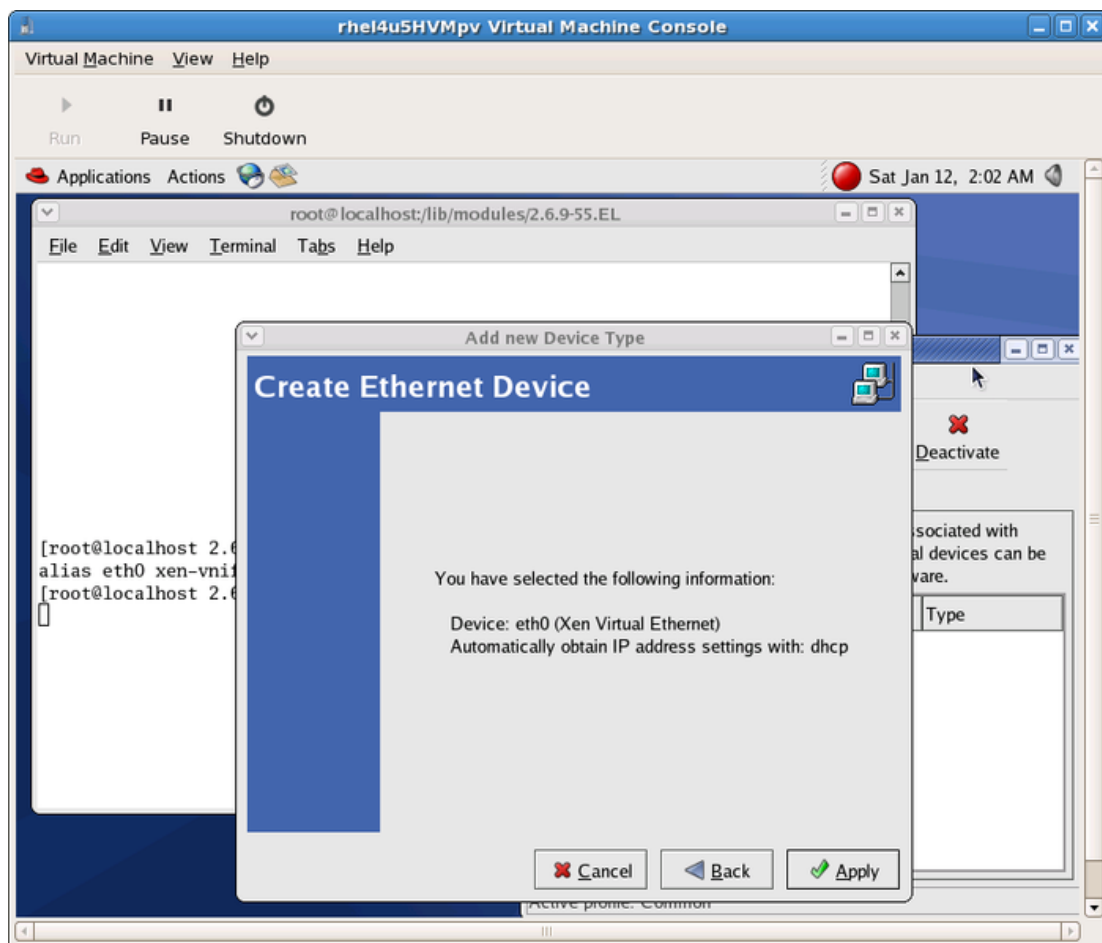
5. Select the 'Xen Virtual Ethernet Card (eth0)' entry and click 'Forward'.



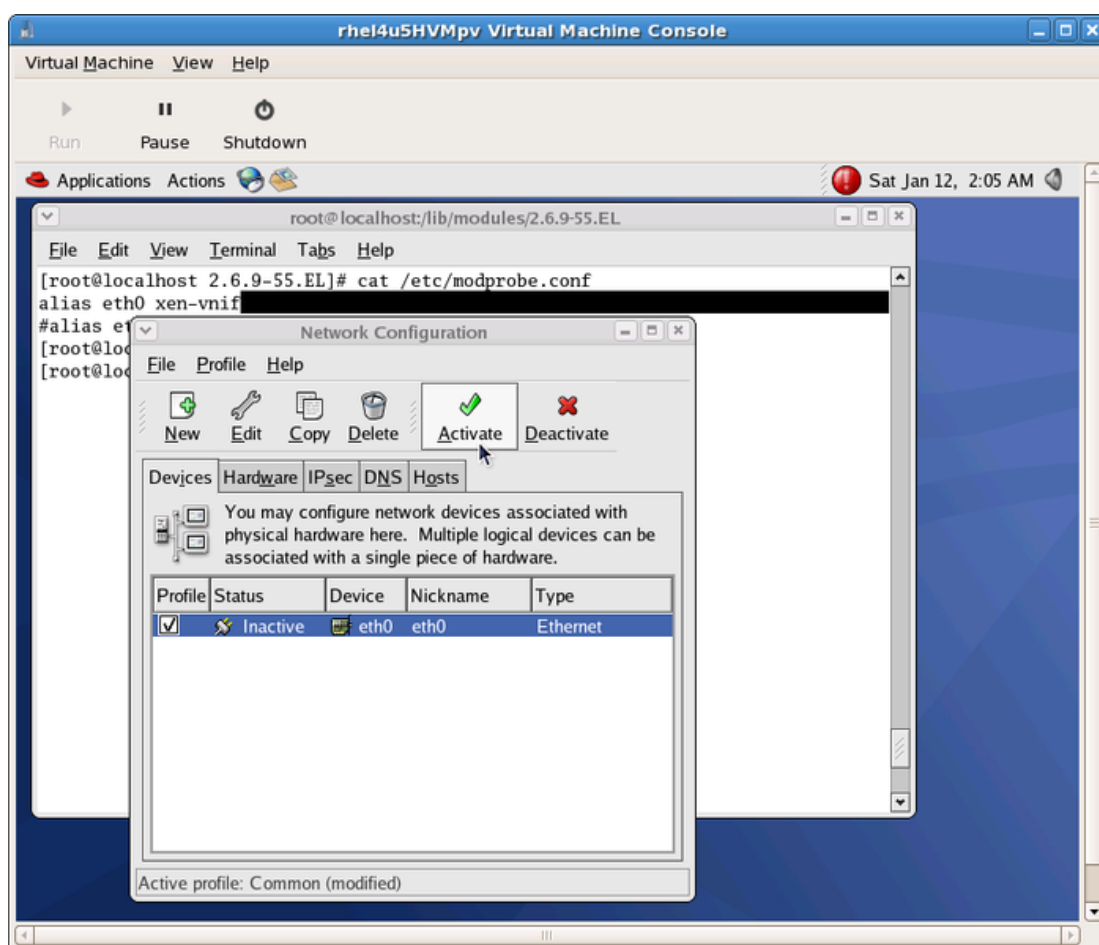
Configure the network settings as required.



6. Complete the configuration by pressing the 'Apply' button.



7. Press the **'Activate'** button to apply the new settings and restart the network.



8. You should now see the new network interface with an IP address assigned.

```
$ ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 00:16:3E:49:E4:E0
          inet addr:192.168.78.180  Bcast:192.168.79.255  Mask:255.255.252.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:630150 errors:0 dropped:0 overruns:0 frame:0
          TX packets:501209 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:109336431 (104.2 MiB)  TX bytes:46265452 (44.1 MiB)
```

Additional Para-virtualized Hardware Configuration

This section will explain how to add additional virtual network or storage to a guest operating system. For more details on configuring network and storage resources on Red Hat Enterprise Linux 5 Virtualization read the document available on [Emerging Technologies, Red Hat.com](#)¹

1. Virtualized Network Interfaces

Perform the following steps to configure additional network devices for your guest.

Edit your guest configuration file in `/etc/xen/YourGuestName` replacing `YourGuestName` with the name of your guest.

The original entry may look like the one below.

```
vif = [ "mac=00:16:3e:2e:c5:a9,bridge=xenbr0" ]
```

Add an additional entry to the “vif=” section of the configuration file similar to the one seen below.

```
vif = [ "mac=00:16:3e:2e:c5:a9,bridge=xenbr0",  
        "mac=00:16:3e:2f:d5:a9,bridge=xenbr0" ]
```

Make sure you generate a unique MAC address for the new interface. You can use the command below.

```
# echo 'import virtinst.util ; print virtinst.util.randomMAC()' | python
```

After the guest has been rebooted perform the following step in the guest operating system. Verify the update has been added to your `/etc/modules.conf` in Red Hat Enterprise Linux 3 or `/etc/modprobe.conf` in Red Hat Enterprise Linux 4 and Red Hat Enterprise Linux 5. Add a new alias for each new interface you added.

```
alias eth1 xen-vnif
```

Now test that each new interface you added make sure it is available inside the guest.

```
# ifconfig eth1
```

¹ http://et.redhat.com/~jmh/docs/Installing_RHEL5_Virt.pdf

The command above should display the properties of **eth1**, repeat the command for **eth2** if you added a third interface, and so on.

Now you can configure the new network interfaces using `redhat-config-network` or Red Hat Enterprise Linux3 or `system-config-network` on Red Hat Enterprise Linux 4 and Red Hat Enterprise Linux 5.

2. Virtual Storage Devices

Perform the following steps to configure additional virtual storage devices for your guest.

Edit your guest configuration file in `/etc/xen/YourGuestName` replacing `YourGuestName` with the name of your guest. The original entry may look like the one below.

```
disk = [ "file:/var/lib/xen/images/rhel5_64_fv.dsk,hda,w" ]
```

Now, add an additional entry for your new physical device, LUN, partition or volume to the “`disk=`” parameter in the configuration file. Storage entities which use the para-virtualized driver resemble the entry below. The “`tap:aio`” parameter instructs the hypervisor to use the para-virtualized driver.

```
disk = [ "file:/var/lib/xen/images/rhel5_64_fv.dsk,hda,w",  
        "tap:aio:/var/lib/xen/images/UserStorage1.dsk,xvda,w" ]
```

If you want to add more entries just add them to the “`disk=`” section as a comma separated list.



Note

You need to increment the letter for the 'xvd' device, that is for your second storage entity it would be 'xvdb' instead of 'xvda'.

```
disk = [ "file:/var/lib/xen/images/rhel5_64_fv.dsk,hda,w",  
        "tap:aio:/var/lib/xen/images/UserStorage1.dsk,xvda,w" ],  
        "tap:aio:/var/lib/xen/images/UserStorage2.dsk,xvdb,w" ]
```

Verify the partitions have been created and are available.

```
# cat /proc/partitions  
major minor #blocks name  
3 0 10485760 hda  
3 1 104391 hda1  
3 2 10377990 hda2
```



```
202    0    64000  xvda
202    1    64000  xvdb
253    0   8257536  dm-0
253    1   2031616  dm-1
```

In the above output you can see the partition or device “xvdb” is available to the system.

Mount the new devices and disks to local mount points and update the `/etc/fstab` inside the guest to mount the devices and partitions at boot time.

```
# mkdir /mnt/pvdisk_xvda
# mkdir /mnt/pvdisk_xvdb
# mount /dev/xvda /mnt/pvdisk_xvda
# mount /dev/xvdb /mnt/pvdisk_xvdb
# df /mnt/pvdisk_p1
Filesystem      1K-blocks    Used   Available Use%   Mounted on
/dev/xvda       64000         15     63985    1%   /mnt/pvdisk_xvda
/dev/xvdb       64000         15     63985    1%   /mnt/pvdisk_xvdb
```


Troubleshooting Para-virtualized Drivers

This chapter deals with issues you may encounter with the Red Hat Enterprise Linux hosts and fully virtualized guests using the para-virtualized drivers

1. Red Hat Enterprise Linux 5 Virtualization log file and directories

Red Hat Enterprise Linux 5 Virtualization related log file.

In Red Hat Enterprise Linux 5, the log file written by the `xend` daemon and the `qemu-dm` process are all kept in the following directories:

`/var/log/xen/`

directory holding all log file generated by the `xend` daemon and `qemu-dm` process.

`xend.log`

- This log file is used by `xend` to log any events generate by either normal system events or operator initiated events.
- virtual machine operations, such as create, shutdown and destroy, are all logged in this log file.
- Usually this log file will be the first place to look at in the event of a problem. In many cases you will be able to identify the root cause by scanning the logfile and review the entries logged just prior to the actual error message.

`xend-debug.log`

- used to record error events from `xend` and its subsystems (such as framebuffer and Python scripts etc..)

`xen-hotplug.log`

- used to log events from hotplug events.
- events such as devices not coming online or network bridges not online will be logged in this file

`qemu-dm.PID.log`

- this file is create by the `qemu-dm` process which is started for each fully-virtualized guest.
- the `PID` will be replaced with the PID of the process of the related `qemu-dm` process

- You can retrieve the PID for a given `qemu-dm` process using the `ps` command and in looking at the process arguments you can identify the virtual machine the `qemu-dm` process belongs to.

If you are troubleshooting a problem with the **virt-manager** you should review the **virt-manager** log file. The log file for **virt-manager** is in the `.virt-manager` directory in the user's home directory whom ran **virt-manager**. This directory will usually be `~/ .virt-manager/virt-manager`.



Note

The log file is overwritten every time you start **virt-manager**. If you are troubleshooting a problem with **virt-manager** make sure you save the logfile before you restart **virt-manager** after an error has occurred.

Red Hat Enterprise Linux 5 Virtualization related directories.

There are a few other directories and files which may be of interest when troubleshooting a Red Hat Enterprise Linux 5 Xen environment:

`/var/lib/xen/images/`

the standard directory for file based virtual machine images.

`/var/lib/xen/xend-db/`

directory that hold the xend database which is generated every time the daemon is restarted.

`/etc/xen/`

holds a number of configuration files used to tailor your Red Hat Enterprise Linux 5 Virtualization environment to suite your local needs

- `xend-config.sxp` is the main configuration for the xend daemon. It used to enable/disable specific functionality of the Xen daemon, and to configure the callouts to Xen networking.

`/var/xen/dump/`

hold dumps generate by virtual machines or when using the `xm dump-core` command.

`/proc/xen/`

has a number of entries which can be used to retrieve additional information:

- `/proc/xen/capabilities`
- `/proc/xen/privcmd`
- `/proc/xen/balloon`

- /proc/xen/xenbus
- /proc/xen/xsd_port
- /proc/xen/xsd_kva

2. Para-virtualized guest fail to load on a Red Hat Enterprise Linux 3 guest operating system

Red Hat Enterprise Linux 3 is uses processor architecture specific kernel RPMs and because of this the para-virtualized drivers may fail to load if the para-virtualized driver RPM does not match the installed kernel architecture.

When the para-virtualized driver modules are inserted, a long list of unresolved modules will be displayed. A shortened excerpt of the error can be seen below.

```
insmod xen-platform-pci.o
Warning: kernel-module version mismatch
xen-platform-pci.o was compiled for kernel version
2.4.21-52.EL
while this kernel is version 2.4.21-50.EL
xen-platform-pci.o: unresolved symbol
__ioremap_R9eac042a
xen-platform-pci.o: unresolved symbol
flush_signals_R50973be2
xen-platform-pci.o: unresolved symbol
pci_read_config_byte_R0e425a9e
xen-platform-pci.o: unresolved symbol
__get_free_pages_R9016dd82
[...]
```

The solution is to use the correct RPM package for your hardware architecture for the para-virtualized drivers.

3. A warning message is displayed while installing the para-virtualized drivers on Red Hat Enterprise Linux 3

Installing the para-virtualized drivers on a Red Hat Enterprise Linux 3 kernel prior to 2.4.21-52 may result in a warning message being displayed stating the modules have been compiled with a newer version than the running kernel.

This message, as seen below, can be safely ignored.

```
Warning: kernel-module version mismatch
xen-platform-pci.o was compiled for kernel version
2.4.21-52.EL
while this kernel is version 2.4.21-50.EL
Warning: loading xen-platform-pci.o will taint the kernel:
forced load
```

```
See http://www.tux.org/lkml/#export-tainted for information
about tainted modules
Module xen-platform-pci loaded, with warnings
```

The important part of the message above is the last line which should state the module has been loaded with warnings.

4. What to do if the guest operating system has been booted with `virt-manager` OR `virsh`

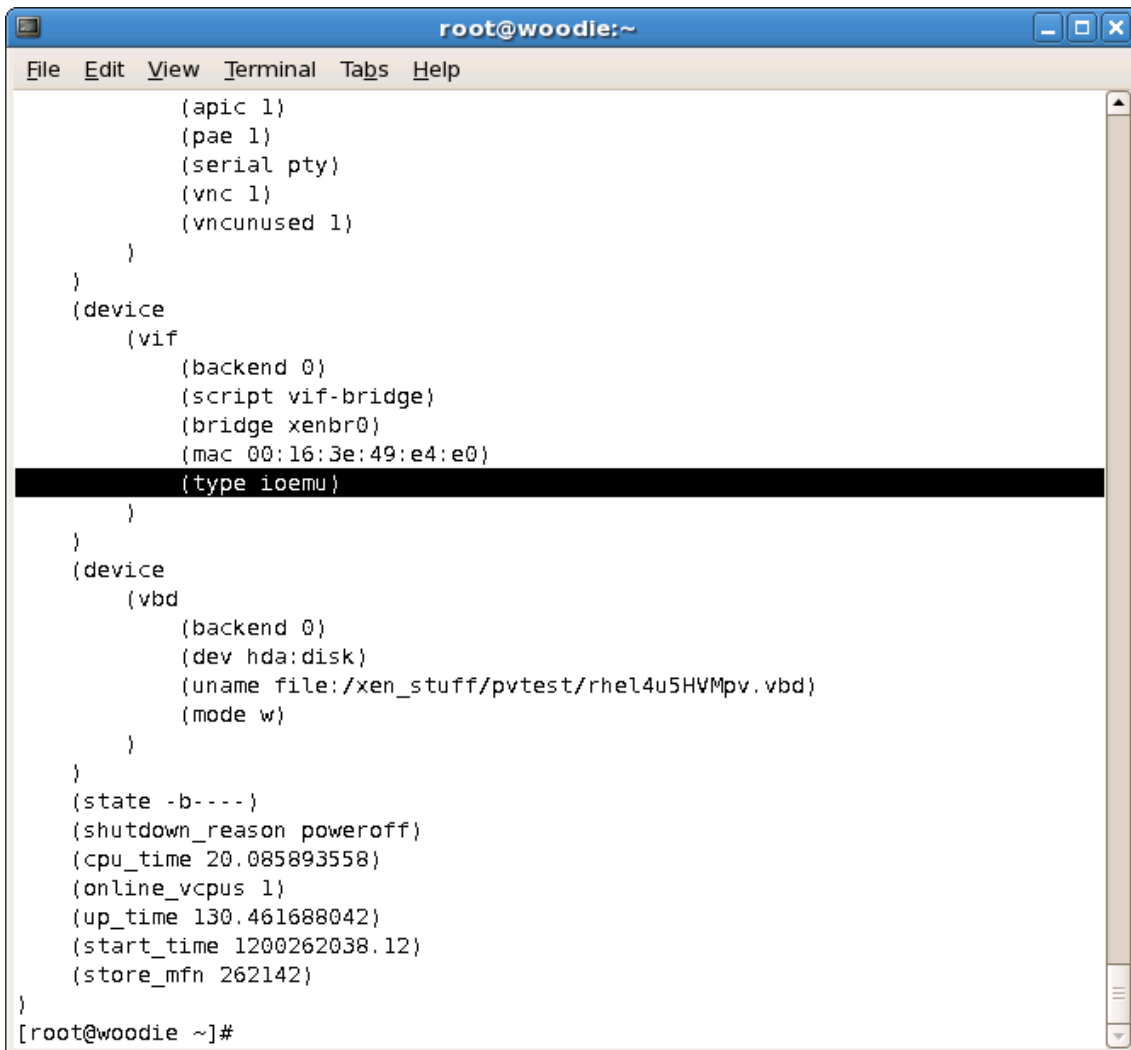
As mentioned in the installation notes, a guest operating system with network para-virtualized drivers installed must be started using the “# `xm create GuestName`” command. You can only use other methods for starting the guest in Red Hat Enterprise Linux 5.2.

If the guest operating system has been booted using the `virt-manager`(the GUI tool) or `virsh`(the command line application) interface the boot process will detect the *new* old Realtek card. This due to the fact `libvirt`, as the underlying API to `virt-manager` and `virsh`, will always add `type=ioemu` to the networking section followed by prompting the systems administrator to reconfigure networking inside the guest. It is recommend you interrupt the boot process (using `virt-manager`, `virsh` or `xm`) and to boot the guest using the `xm` command. In the event of the guest operating system has booted all the way to multi-user mode you will detect that there is no networking active as the backend and frontend drivers are not connected properly.

To fix this issue, shut down the guest and boot it using “`xm create`”. During the boot process `kudzu` (the hardware detection process) will detect the “old” Realtek card. Simply select “**Remove Configuration**” to delete the Realtek card from the guest operating system. The guest should continue to boot and configure the network interfaces correctly.

You can identify if your guest has been booted with `virt-manager`, `virsh` or “`xm create`” using the command “# `xm list -long YourGuestName`”

In the screenshot below you can see the entry “`ioemu`” highlighted in the “`device vif`” (networking) section. This would mean the guest was booted with `virt-manager` or `virsh` and networking is not configured correctly, that is, without the para-virtualized network driver.



```
root@woodie:~
File Edit View Terminal Tabs Help
    (apic 1)
    (pae 1)
    (serial pty)
    (vnc 1)
    (vncunused 1)
  )
}
(device
  (vif
    (backend 0)
    (script vif-bridge)
    (bridge xenbr0)
    (mac 00:16:3e:49:e4:e0)
    (type ioemu)
  )
)
(device
  (vbd
    (backend 0)
    (dev hda:disk)
    (uname file:/xen_stuff/pvtest/rhel4u5HVMpv.vbd)
    (mode w)
  )
)
(state -b----)
(shutdown_reason poweroff)
(cpu_time 20.085893558)
(online_vcpus 1)
(up_time 130.461688042)
(start_time 1200262038.12)
(store_mfn 262142)
)
[root@woodie ~]#
```

In the screenshot below you can see there is no “type ioemu” entry in the “device vif” section so you can safely assume the guest has been booted with “xm create YourGuestName”. This means networking is configured to use the para-virtualized network driver.

```

root@woodie:~
File Edit View Terminal Tabs Help
    (vnc 1)
    (vncunused 1)
    (display :1.0)
    (xauthority /root/.Xauthority)
    (acpi 1)
    (apic 1)
  }
}
(device
  (vif
    (backend 0)
    (script vif-bridge)
    (bridge xenbr0)
    (mac 00:16:3e:49:e4:e0)
  )
)
(device
  (vbd
    (backend 0)
    (dev hda:disk)
    (uname file:/xen_stuff/pvtest/rhel4u5HVMpv.vbd)
    (mode w)
  )
)
}
(state r-----)
(shutdown_reason poweroff)
(cpu_time 3.836914716)
(online_vcpus 1)
(up_time 5.34920191765)
(start_time 1200262268.5)
(store_mfn 262142)
}
[root@woodie ~]#

```

5. Manually loading the para-virtualized drivers

If for some reason the para-virtualized drivers failed to load automatically during the boot process you can attempt to load them manually.

This will allow you to reconfigure network or storage entities or identify why they failed to load in the first place. The steps below should load the para-virtualized driver modules.

First, locate the para-virtualized driver modules on your system.

```
# cd /lib/modules/`uname -r`/
# find . -name 'xen-*.ko' -print
```

Take note of the location and load the modules manually. Substitute {LocationofPV-drivers} with the correct location you noted from the output of the commands above.

```
# insmod \
    /lib/modules/'uname
-r'/{LocationofPV-drivers}/xen-platform-pci.ko
```



```

        # insmod /lib/modules/'uname
-r'/{LocationofPV-drivers}/xen-balloon.ko
        # insmod /lib/modules/'uname
-r'/{LocationofPV-drivers}/xen-vnif.ko
        # insmod /lib/modules/'uname
-r'/{LocationofPV-drivers}/xen-vbd.ko

```

6. Verifying the para-virtualized drivers have successfully loaded

One of the first tasks you will want to do is to verify that the drivers have actually been loaded into your system.

After the para-virtualized drivers have been installed and the guest has been rebooted you can verify that the drivers have loaded. First you should confirm the drivers have logged their loading into `/var/log/messages`

```

# grep -E "vif|vbd|xen" /var/log/messages
        xen_mem: Initialising balloon driver
        vif vif-0: 2 parsing device/vif/0/mac
        vbd vbd-768: 19 xlxbd_add at
/local/domain/0/backend/vbd/21/76
        vbd vbd-768: 19 xlxbd_add at
/local/domain/0/backend/vbd/21/76
        xen-vbd: registered block device major 202

```

You can also use the `lsmod` command to list the loaded para-virtualized drivers. It should output a list containing the `xen_vnif`, `xen_vbd`, `xen_platform_pci` and `xen_balloon` modules.

```

# lsmod|grep xen
        xen_vbd                19168  1
        xen_vnif                28416  0
        xen_balloon            15256  1 xen_vnif
        xen_platform_pci       98520  3
xen_vbd,xen_vnif,xen_balloon,[permanent]

```

7. The system has limited throughput with para-virtualized drivers

If network throughput is still limited even after installing the para-virtualized drivers and you have confirmed they are loaded correctly (see [Section 6, “Verifying the para-virtualized drivers have successfully loaded”](#)). To fix this problem, remove the `'type=ioemu'` part of `'vif='` line in your guest's configuration file.

Compiling para-virtualized driver packages from source code

You may need to rebuild the RPMs to get them working properly on your system for specific architectures. Use the source RPMs provided by Red Hat. These instructions will also allow you to install from the source RPMs.

Before compiling, the `kverrel` and `kverbase` variables in the `xenpv.spec` file must be changed to match the kernel version that binary modules are being built for. Usually, this is the value returned from `'uname -r'`.

You can then use the following command to rebuild the `kmod-xenpv` package. Change *SOURCE* to reflect the correct path to your `kmod-xenpv` RPM file.

```
cd /usr/src/redhat/SOURCE/xenpv
rpmbuild -bb kmod-xenpv
```


Additional Information Links

- [Red Hat](#)¹
- [Virtualization Infocenter](#)²
- [libvirt](#)³
- [virt-Manager](#)⁴
- [Red Hat Cluster Suite](#)⁵
- [Red Hat Emerging Technology Group](#)⁶

¹ <http://www.redhat.com/>

² <http://www.openvirtualization.com/>

³ <http://www.libvirt.org/>

⁴ <http://virt-manager.et.redhat.com/>

⁵ <http://www.redhat.com/solutions/gfs/>

⁶ <http://et.redhat.com/>

Appendix A. Revision History

Revision History

Revision 1.0 January 2008 Chris Curran

Conversion to Docbook xml and editing

Revision 1.1 January 2008 Chris Curran

Bug fixes and error corrections

Revision 2.0 July 2008 Chris Curran

Bug fixes and updates

