Red Hat Integrated Virtualization – A quick guide

A project at the University of Cambridge named XenoServers(tm) was initiated to investigate how to deploy a global infrastructure of untrusted services on behalf of non-cooperating paying customers. Part of this project led to the Xen(tm) Hypervisor , which is a very high performance virtual machine monitor providing reliable isolation of resources between VM's (virtual machines). The Xen project is released under the GPL and in addition to the University of Cambridge also has a large community of developers including many from Red Hat. Red Hat has joined this project and is including it as part of Red Hat Enterprise Linux 5. Realizing that this project was not enough to offer customers a complete package and free alternative to commercial alternatives, Red Hat has surrounded the base Xen technology with companion modifications and additions to the Red Hat Enterprise Linux enabling simplified management, high security, and high availability in addition to the complete support and services our customers have come to trust.
The remainder of this paper will show examples of Red Hat's integrated virtualization being used to reduce costs, and improve availability, security and flexibility.

There are two styles of virtualization employed by Red Hat Enterprise Linux. One is called “paravirtualization”and the other is named “HVM”. HVM or “Hardware Virtual Machine” takes advantage of the latest Intel and AMD hardware assistance. This technique allows Red Hat Enterprise Linux to emulate a complete ia32 or ia32-64 architecture and enables the operation of any operating environment designed for those architectures. “Paravirtualization”is a highly optimized virtualization technique and only Red Hat Enterprise Linux version 5 and Red Hat Enterprise Linux version 4 Update 5 are able to operate in this way.
The benefits of paravirtualization over emulation are many.

Performance: Today paravirtualization overs significant performance benefits over HVM and emulation (Vmware) techniques. This gap will reduce as paravirtualizaed device drivers are introduced into the HVM environment, but at this time a large performance gap still exists.

CPU time accounting: If three OS instances are sharing a single virtual CPU it is possible that each OS will report that it is using 100% of their virtual CPU. Obviously they may be, but they are not each using 100% of the same physical CPU. This leads to many distorted numbers in performance, job scheduling and accounting software. Paravirtualization explicitly fixes this problem and CPU utilization is reported correctly.

Time measurement: A emulated OS must process hundreds of virtual clock interrupts. Often these are missed and there is clock skew between the guest and the host. This has been a common and loud complaint of users running traditional emulators.

Power-savings: Emulation prevents power management software from functioning properly because the communication between the guest and the host prevents CPU shutdown. Paravitualization does not have this limitation.

Hotplug: Paravirtualization allows memory to be resized dynamically. When a workload would benefit from additional memory it can have it, when it no longer benefits it can be taken away . This is not possible with emulation. This same ability extends to adding and subtracting CPU's dynamically.

Red Hat provides paravirtualization for those operating systems that can run in this mode, and Red Hat provides full virtualization for operatng systems that cannot run in the paravirtualized mode.

There are many benefits to virtualization that are not directly related to hardware savings, but do decrease costs or increase revenue. These are increased availability, increased security, significantly lower maintainance of infrastructure and applications, and overall flexibility.

Hardware consolidation: Often companies are not utilizing the full capacity of all of their servers even while some of their servers are unable to keep up with their specific workloads. This occurs because it has been too difficult to manage multiple workloads on one server or even understand the performance requirements of each workload. Usually people think in terms of CPU utilization but often memory footprint, the number of I/O channels or network traffic is the limiting factor.
Virtualization makes it easy to monitor the load each guest operating system creates on CPU, memory, I/O and networks. This load can be watched over time so that any periodic peaks can be determined whether based on time of day, end of quarter, after new product announcements or any other happening. As these workloads are understood it becomes possible to combine multiple workloads onto a smaller set of servers because of the ability to virtualize them. A reduction in the number off servers directly translates into less cost for new server purchases, floorspace, power and air conditioning, server administration, per server software license and maintenance fees and so on. A side benefit of this is the ability to migrate workloads to balance the demands on each system and to dynamically bring more systems on line when needed or shut them down when possible.
Typical experience is that over time companies can reduce their total number of servers by 25% to 75% depending on the amount of "sharing" that is possible across departments and how internal chargeback or cost allocation is set up. As long as the cost of virtualization, in money or performance, is less than the cost of acquiring and managing additional servers, consolidation is a big win.
The obvious question is What do I do with all the excess servers I have? The can be retired, re-deployed for redundancy or new projects, or used to improve performance of long running workloads.

Hardware abstraction: A common drain on IT is the need to frequently upgrade systems because of new hardware acquisitions. Each wave of new hardware is likely to be faster and better than the previous, but in oder to take advantage of this new hardware or often merely in order to utilize it at all, the operating system must be upgraded to the latest versions.
That cascades to other levels because then each ISV or home grown application must be checked to see if it has been certified on the new operating system version. Red Hat is committed to compatibility and from the date a new operating system is introduced we will support it for seven years. We have an extraordinary record formaintaining backwards compatibility, that is, each new version of our software will generally run well on older equipment. Going forward is much more difficult for us because it is not always clear how the hardware vendors will improve their products and also because they often try to differentiate their products, even their "standard" products. Sometimes it becomes necessary to modify software to run on the latest hardware.
Hardware abstraction is a property of Red Hat Integrated virtualization. Because the "real" hardware is actually controlled by the host operating system, the virtualization systems can present a standard interface to the guest systems. For example, infiniband, 10 gigabit Ethernet, RDMA controllers, fiber channel all are able to carry ip protocol messages. At the program level the applications merely open a socket or send a request to a particular ip address and port. The network layer of the operating system "abstracted" the specific hardware to a standard network programming interface so that applications can work with any ip network. But deeper in the operating system this abstraction had to eventually turn into real drivers for the specific network devices.

For a virtualized system there is an additional layer of abstraction. The virtual operating system is told that the network is on a standard NIC and there is a standard driver in the virtual system. The host system then takes the commands issued to the abstract driver in the guest and translates them to the appropriate physical hardware. You can see how the guest operating system does not need to be aware of the actual physical device, it uses the virtual device. And you can see how the host operating system must know the physical device, and also must know how to translate to the virtual device. This can also be done for graphics, for storage devices, for anything that is necessary to provide a hardware abstraction to the guest.

It is estimated that over 60% of IT resources are expended just maintaining the existing infrastructure and applications. By allowing older “stacks” to move to new servers it becomes possible to take advantage of the latest hardware without the pain of migrations. Some companies have been able to move 20 points of budget from maintenance to innovation.
The obvious question is How much overhead is all of this abstraction going to cost? That will be adressed in a following section.

”Appliance” distribution and provisioning: Many IT departments have put programs in place to standardize on a small number of “stacks” to simplify management and administration. One practice is to have a golden image that contains the basic operating system as well as any mandatory security, backup/restore, management and other standard software. The each application or set of applications only has to be qualified against the golden image. Every 18 to 24 months a new golden image is produced for each ooperating system in order to include the latest patches. Security patches are handled separately and rolled out as needed.
People outside of the IT department see this an an inhibitor to progress. A new version of an application has useful features, but also requires a new version of the operating system. IT may not be able to roll out a new golden image in time to make the departmental customer happy. Either a non-standard version is maintained which goes against the IT desire to cut costs, or the department goes without which could increase their costs or reduce revenue opportunities.
By taking an appliance viewpoint IT can take each major application and package it as a guest with its own operating system. The base host system can update when it makes sense to do that, and each appliance can be created or updated as needed and deployed as a guest.
The obvious question is How do I manage all of this? Are there tools available? And the answer is “yes”. And that will be covered in a following section.

Security compartmentalization: One tenet of good security is to compartmentalize applications and data such that they are only accessable to appropriate systems and users. Past solutions have either put each application on its own server and added to the server proliferation, or used slow emulation techniques for isolating workloads. With Red Hat Integrated Virtualization it becomes very easy to deploy small container guests, each one holding one application or set of related applications.
Not only does this prevent a running system from spying on another, but it also servers to contain the effect of any intrusions. This compartmentalization, couples with Red Hats Secure Linux features provides for excellent security.

Checkpointing and restart: When a particular workload needs to run for an extended period of time, Red Hat Integrated Virtualization provides a number of benefits. One benefit is realized if the guest is saved to disk periodically because it can be resumed at that point in the event of a failure. A related use is for a grid environment. Desktops and servers that are only used during some hours can have guests installed that allow them to be part of a compute grid. When the grid is active, the guest is resumed and used. If the desktop or server needs to be returned to its normal use, the grid guest is paused and saved. This is called cycle harvesting and can be a very useful feature when workloads benefit from large numbers of smaller machines.

Business continuity: There are many ways to improve systems availability through effective use of Red Hat Integrated Virtualization. If a server begins to experience problems such as a bad fan, or disk errors or memory corrections, the workloads on that server can be migrated to other servers so that the faulty component can be located and replaced. Likewise, if a server completely fails, guests that have been initiated and then paused on other servers can be resumed very quickly. This is much faster than booting a replacement server and much less expensive than having mutiple servers ready to take over in the event of a machine crash.

Red Hat Cluster Manager is often used to monitor that critical workloads are executing and starting them if need be. This is most commonly seen in a failover arrangement where the cluster manager monitors a workload on a systems and if that system fails the cluster will notice that the monitored workload is not longer running and will perform any work necessary to get a replacement job running.

Efficient systems development and test

One very common use of virtualization, in fact the first use when IBM introduced VM over 30 years ago, was to create an environment that simpified software debugging and monitoring. It is possible to instrument the guest operating system by adding software to the host. The host can also be utilized to script, modify and monitor the running guest OS and its applications. Multiple guests can be created to simulate large networks on smaller systems.

Installing a Red Hat Enterprise Linux 5 Para Virtualized Guest

This section is intended to provide a quick start to install a virtual machine on a Red Hat Enterprise Linux 5.

Requirements

1. A running Red Hat Enterprise Linux 5 Server (including a configured Hypervisor/dom0)
2. Installation media
    1. For para-virtualized guest installation a Red Hat Enterprise Linux 5 Installation Tree, preferable on a web/ftp server.
    2. For fully-virtualized guest installation either DVD/CDrom based distribution media or boot.iso file and network accessible installation tree
3. For network installation server, please take reference to knowledge base: http://kbase.redhat.com/faq/FAQ_80_3592.shtm
4. At least 1GB of available RAM for guest memory creation
5. At least 8GB of free disk space for guest disk image
6. What type of storage to be used for the guest
    1. On-file container
    2. Disk partition
    3. Locally connected physical LUN
    4. LVM partition/volume
    5. iSCSI or Fiberchannel based LUN
7. All required information to configure the guest's networking/hostname
    1. Use DHCP or static IP address ?
    2. Enable Ipv6 support ?
    3. Hostname set via DHCP or static ?
8. Software Package selection/requirements
    1. Web Server
    2. Office/Productivity
    3. Software Development

Hardware pre-requisites

How to identify Intel VT/AMD AMD-V capable hardware?

If you intend to only run paravirtualized guests you can skip this section.

If you intend to run unmodified operating systems (also known as fully virtualized or HVM guests) as a guest you need to deploy a VT/AMD-V capable system. The steps below will provide an overview how you can verify whether the virtualization extensions have been enabled and RHEL5 can use them.

A list of HVM capable platforms can also be found at
http://wiki.xensource.com/xenwiki/HVM_Compatible_Processors

You need to make sure that the Intel/VT / AMD/AMD-V capabilities are enabled via the BIOS, usually named "Virtualization Technology Support".

One issue you can run into is that after re-setting or updating the BIOS you must power-cycle the system for the updated settings to take place.

Specifically for the Intel/VT platforms you should perform the steps outlined below, this will make sure you are indeed using VT technology.

Some of the menus and entry points may have slightly different names on different platforms, but you get the idea

1.Go into "BIOS Setup Utility", into "Save,Restore&Exit", select "Restore Defaults", then select "Save&Exit"

2.Power off the machine and disconnect the power supply

3.Power on the machine, go into "BIOS SETUP UTILITY" again, into "Processor", enable "Intel (R) Virtualization Technology" , then select "Save&Exit", then power off the machine and disconnect the power supply

4.Power on the machine, boot xen0 and create vmx guest The power cyle is the magic step.

Verify whether the virtualization extensions have been enabled

Once the kernel-xen has been booted you need to verify that your system has indeed enabled the Intel/VT / AMD/AMD-V extensions.

You can use the following commands to verify

```
  [root@woodie ~]# xm dmesg | grep VMX
  (XEN) VMXON is done
  (XEN) VMXON is done
  (XEN) VMXON is done
  (XEN) VMXON is done
```

or use

```
[root@woodie ~]# cat /proc/cpuinfo |grep vmx
flags		: fpu tsc msr pae mce cx8 apic mtrr mca cmov pat pse36 clflush dts acpi mmx fxsr sse sse2 ss ht tm syscall lm constant_tsc pni monitor ds_cpl vmx est tm2 cx16 xtpr lahf_lm
```

On an AMD based server you have to look for “svm” instead of “vmx”

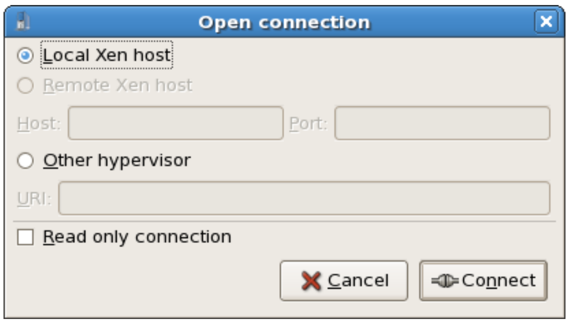
You should have a VMXON for each reported processor. If you have any other messages visit your BIOS settings.

There is no reason to go any further until you have VMXON reported - it just isn't going to work.

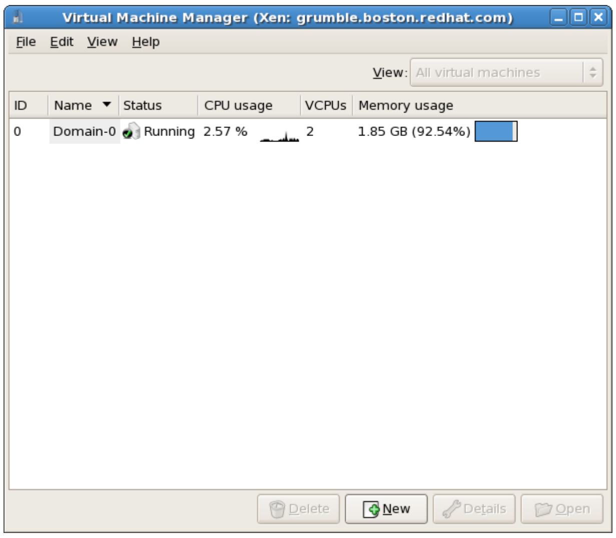
Installation of guest OS using Virtual Machine Manager

To start virt-manager simply type 'virt-manager' at the shell prompt, or start it from the GUI menu item "Virtual Machine Manager".

You will see a dialog box as the one below. Simply select the 'Connect' button and the main virt-manager window will appear.

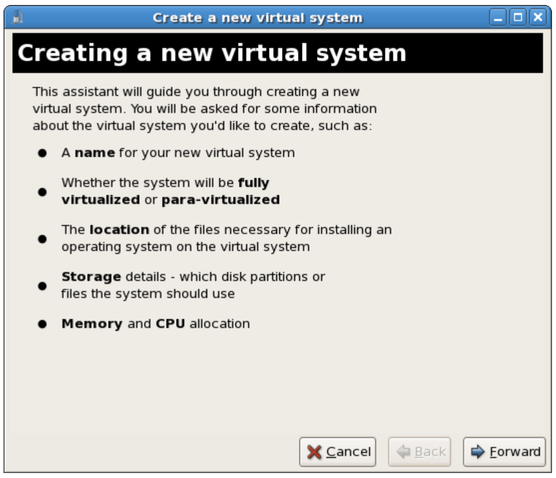


The main window will allow you to create a new virtual machine using the 'New' button.


Virtual Machine Manager (Xen: grumble.boston.redhat.com)
File
Edit
View
Help
View: All virtual machines
Delete
New
Details
Open

ID	Name	Status	CPU usage	VCPUs	Memory usage
0	Domain-0	Running	2.57 %	2	1.85 GB (92.54%)

The next window will provide a summary of the information you will need to provide in order to create a virtual machine.



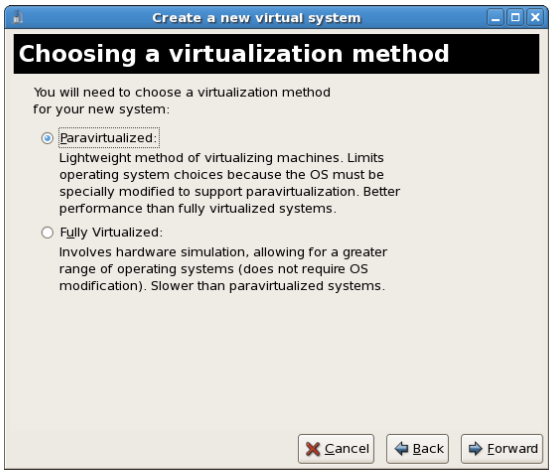
Creating a new virtual system

This assistant will guide you through creating a new virtual system. You will be asked for some information about the virtual system you'd like to create, such as:

- A **name** for your new virtual system
- Whether the system will be **fully virtualized** or **para-virtualized**
- The **location** of the files necessary for installing an operating system on the virtual system
- **Storage** details - which disk partitions or files the system should use
- **Memory** and **CPU** allocation

After you're reviewed all of the information required for your installation you can continue to the next screen.

Depending on your systems capabilities (ie whether your system has intel/VT or AMD/AMD-V capable processors/sockets) the next window will either display a single choice to create a para-virtualized guest or two choice. Where one choice will be para-virtualized (modified and optimized operating system for virtualization) guest creation and the second will be for a fully-virtualized (unmodified operating system) guest creation.



Choosing a virtualization method

You will need to choose a virtualization method for your new system:

[✓ Paravirtualized:]
Lightweight method of virtualizing machines. Limits operating system choices because the OS must be specially modified to support paravirtualization. Better performance than fully virtualized systems.

Fully Virtualized:
Involves hardware simulation, allowing for a greater range of operating systems (does not require OS modification). Slower than paravirtualized systems.

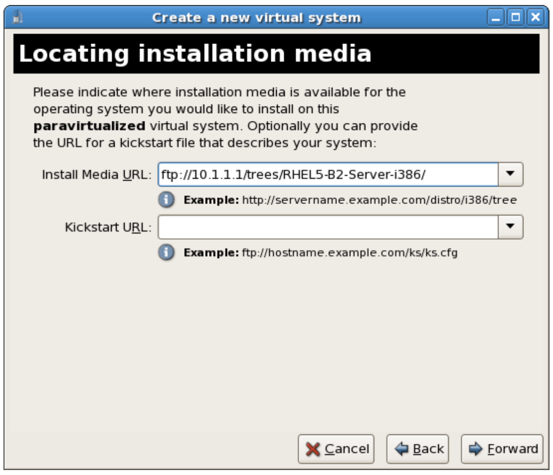
The next screen will ask for the installation media for the type of installation you selected .

The para-virtualized installation will require an installation tree accessible either via HTTP, FTP or NFS (can be setup on the same system as where you install the guest).

You can easily create an installation by either mounting the installation media DVD to a local directory and exporting it via NFS or making it available via FTP or HTTP.

If your media is an .iso file you can loopback mount the file and extract the files onto a local directory.

If SELinux is enabled, all disk images, iso files should be placed in /var/lib/xen/images/.



Locating installation media

Please indicate where installation media is available for the operating system you would like to install on this **paravirtualized** virtual system. Optionally you can provide the URL for a kickstart file that describes your system:

Install Media URL: ftp://10.1.1.1/trees/RHEL5-B2-Server-i386/

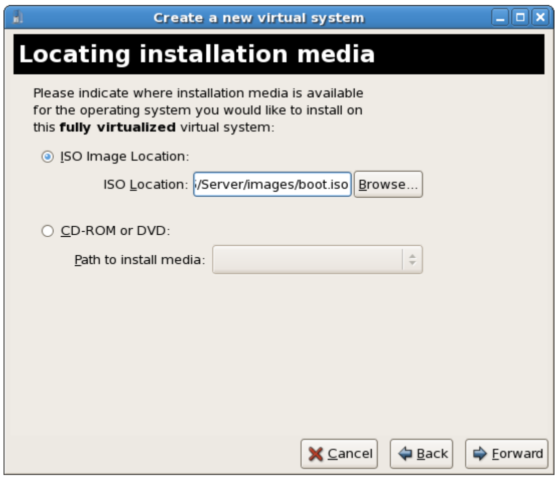
Example: http://servername.example.com/distro/i386/tree

Kickstart URL:

Example: ftp://hostname.example.com/ks/ks.cfg

The fully-virtualized installation will ask for the location of a boot media (ISO image or DVD/CD drive).

Depending on your installation media/process you can either perform a network based installation after booting your guest off the .iso file or perform the whole installation off a DVD .iso file (typically Windows installations are using DVD/CD .iso files, Linux/UNIX operating systems such as RHEL often will only use a .iso for bootstrap and the install using a network based tree)



Locating installation media

Please indicate where installation media is available for the operating system you would like to install on this **fully virtualized** virtual system:

(●) ISO Image Location:

ISO Location: /Server/images/boot.iso

( ) CD-ROM or DVD:

Path to install media:

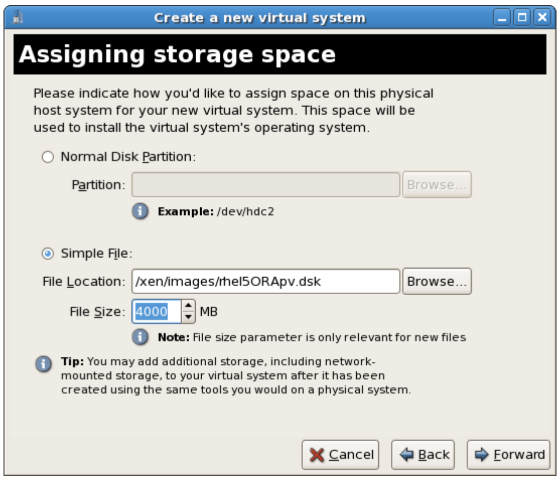
The following screen will allow you to select the storage to be used for the guest.

You can either select a disk parition/LUN

or select a file based guest image

- The convention with Red Hat Enterprise Linux 5 is for all guest images to reside in the /var/lib/xen/images/ directory as other SElinux may block access to images located in a different directory (if you run SElinux in enforcing mode)

- When using a file based guest image you will have to decide on the size of the guest image on this screen. The size of your guest image will be dependent on memory size (the installation process will size swap based on the guest memory size), the packages you want to install and other file system size standards you may want to deploy at your local site.


Create a new virtual system

Assigning storage space

Please indicate how you'd like to assign space on this physical host system for your new virtual system. This space will be used to install the virtual system's operating system.

○ Normal Disk Partition:

Partition: [ ] Browse...

Example: /dev/hdc2

◉ Simple File:

File Location: /xen/images/rhel5ORApv.dsk Browse...

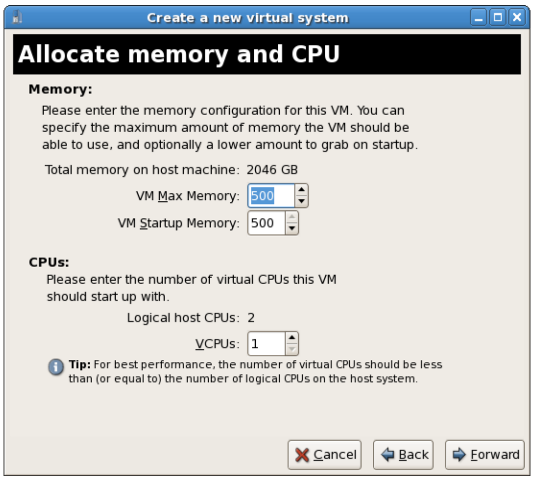
File Size: 4000 MB

Note: File size parameter is only relevant for new files

Tip: You may add additional storage, including network-mounted storage, to your virtual system after it has been created using the same tools you would on a physical system.

The last configuration information to enter is the memory size of the guest you are installing and the number of virtual CPUs you'd like to assign to your guest

- As Red Hat Enterprise Linux 5 / Virt will require physical memory to back a guest's memory you must ensure your system is configured with sufficient memory in order to accommodate the guest you'd like to run / configure
- A good practice for virtual CPU assignments is to not configure more virtual CPUs in a single guest then available physical processors in the host
    - However you can have more virtual CPUs across a number of virtual machine than there are physical processors in your server (ie oversubscribe processor resources)


Create a new virtual system

Allocate memory and CPU

Memory:

Please enter the memory configuration for this VM. You can specify the maximum amount of memory the VM should be able to use, and optionally a lower amount to grab on startup.

Total memory on host machine: 2046 GB

VM Max Memory: 500

VM Startup Memory: 500

CPUs:

Please enter the number of virtual CPUs this VM should start up with.

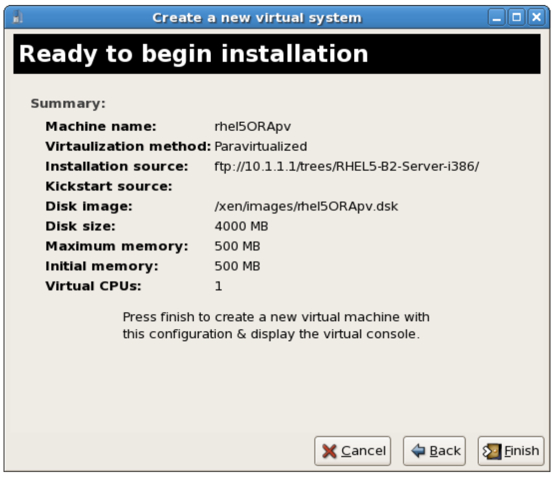
Logical host CPUs: 2

VCPUs: 1

Tip: For best performance, the number of virtual CPUs should be less than (or equal to) the number of logical CPUs on the host system.

As the last step you will be presented with a summary screen of all configuration information you've entered. Review the information and if you want to make any changes simply use the 'Back' button to get back to the screen you want to make modifications to.

Once you are satisfied with the data entered simply click the 'Finish' button and the installation process will commence.



Summary:

Machine name:	rhel5ORApv
Virtaulization method:	Paravirtualized
Installation source:	ftp://10.1.1.1/trees/RHEL5-B2-Server-i386/
Kickstart source:	
Disk image:	/xen/images/rhel5ORApv.dsk
Disk size:	4000 MB
Maximum memory:	500 MB
Initial memory:	500 MB
Virtual CPUs:	1

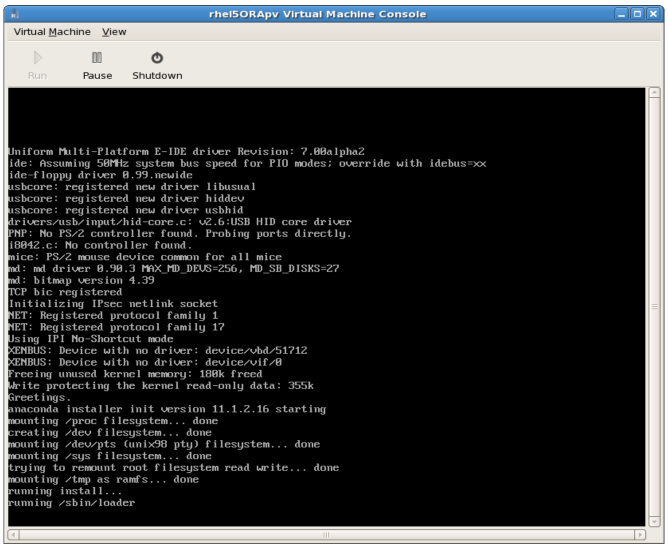
Press finish to create a new virtual machine with this configuration & display the virtual console.

Guest Operating System Installation process

Installing Red Hat Enterprise Linux 5 (para-virtualized)

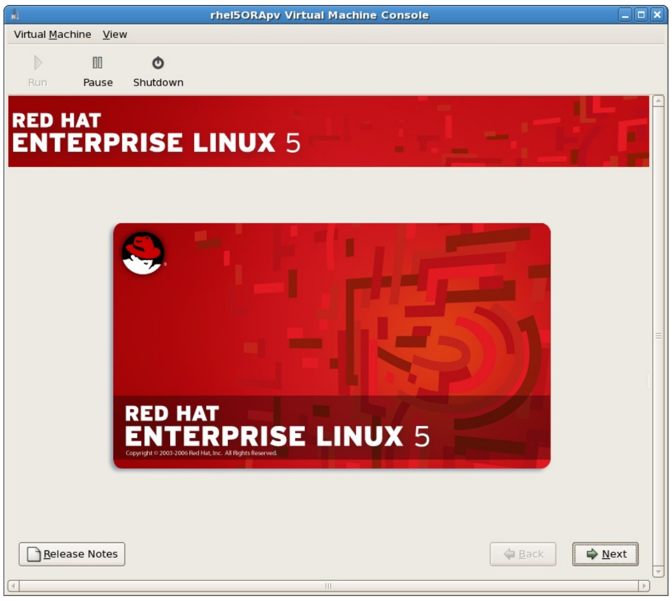
After you pressed the 'Finish' button virt-manager will automatically launch a VNC based window for the installation process.

The window will show the initial boot of your guest

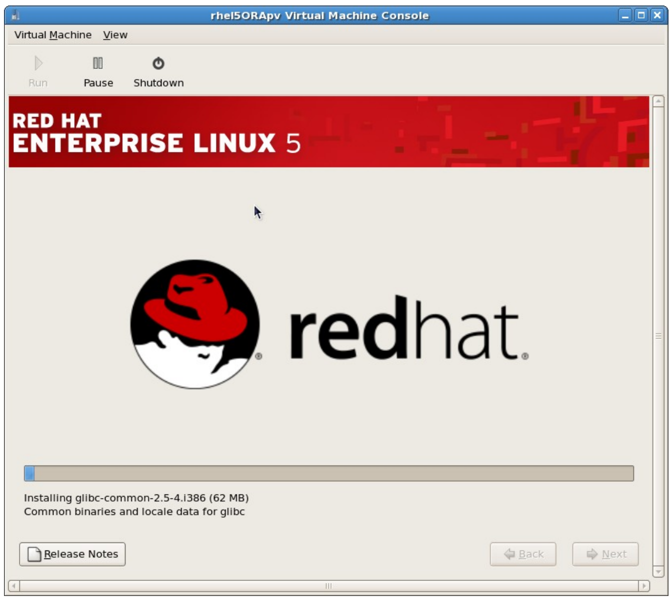


```
Uniform Multi-Platform E-IDE driver Revision: 7.00alpha2
ide: Assuming 50MHz system bus speed for PIO modes; override with idebus=xx
ide-floppy driver 0.99.newide
usbcore: registered new driver libusual
usbcore: registered new driver hiddev
usbcore: registered new driver usbhid
drivers/usb/input/hid-core.c: v2.6:USB HID core driver
PNP: No PS/2 controller found. Probing ports directly.
i8042.c: No controller found.
mice: PS/2 mouse device common for all mice
md: md driver 0.90.3 MAX_MD_DEVS=256, MD_SB_DISKS=27
md: bitmap version 4.39
TCP bic registered
Initializing IPsec netlink socket
NET: Registered protocol family 1
NET: Registered protocol family 17
Using IPI No-Shortcut mode
XENBUS: Device with no driver: device/vbd/51712
XENBUS: Device with no driver: device/vif/0
Freeing unused kernel memory: 180k freed
Write protecting the kernel read-only data: 355k
Greetings.
anaconda installer init version 11.1.2.16 starting
mounting /proc filesystem... done
creating /dev filesystem... done
mounting /dev/pts (unix98 pty) filesystem... done
mounting /sys filesystem... done
trying to remount root filesystem read write... done
mounting /tmp as ramfs... done
running install...
running /sbin/loader
```

After your guest has completed it's initial bootstrap you'll be dropped into the standard installation process for Red Hat Enterprise Linux or the operating system specific installer you select to install.



The installation process is same as the physical machine installation.

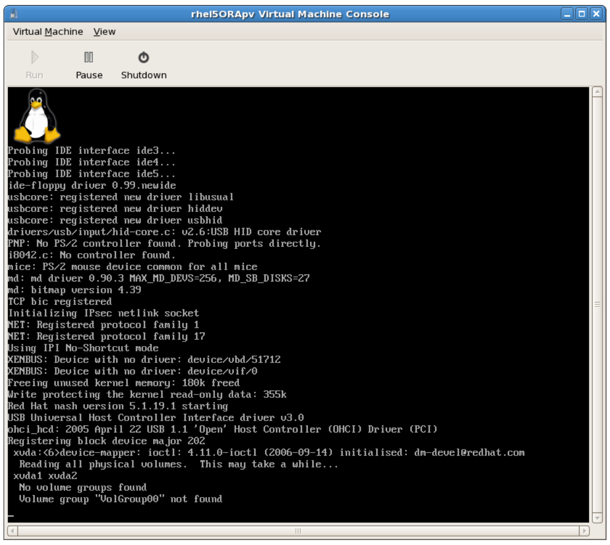


Installing glibc-common-2.5-4.i386 (62 MB)
Common binaries and locale data for glibc

But the newly installed guest will actually not reboot but just shutdown after installation.

First boot after guest installation (Red Hat Enterprise Linux 5)

- In order to reboot your new guest simply type the command "# xm create GuestName" where 'GuestName' is the name you entered during the initial installation.
- Guest configuration files are located in /etc/xen
- Once you started your guest using the CLI command above you can use virt-manager to (re)open a graphical console window for your guest

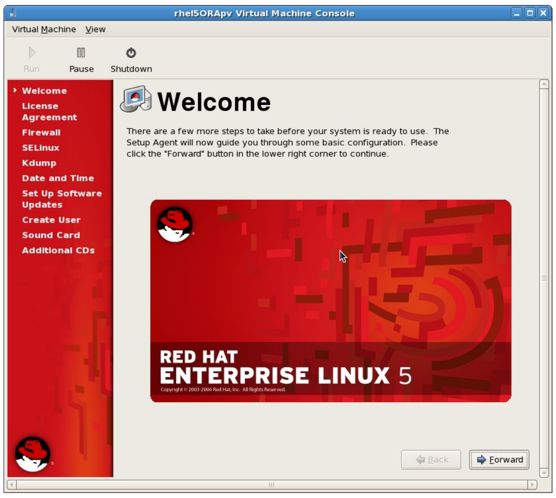


```
Probing IDE interface ide3...
Probing IDE interface ide4...
Probing IDE interface ide5...
ide-floppy driver 0.99.newide
usbcore: registered new driver libusual
usbcore: registered new driver hiddev
usbcore: registered new driver usbhid
drivers/usb/input/hid-core.c: v2.6:USB HID core driver
PNP: No PS/2 controller found. Probing ports directly.
i8042.c: No controller found.
mice: PS/2 mouse device common for all mice
md: md driver 0.90.3 MAX_MD_DEVS=256, MD_SB_DISKS=27
md: bitmap version 4.39
TCP bic registered
Initializing IPsec netlink socket
NET: Registered protocol family 1
NET: Registered protocol family 17
Using IPI No-Shortcut mode
XENBUS: Device with no driver: device/vbd/51712
XENBUS: Device with no driver: device/vif/0
Freeing unused kernel memory: 180k freed
Write protecting the kernel read-only data: 355k
Red Hat nash version 5.1.19.1 starting
USB Universal Host Controller Interface driver v3.0
ohci_hcd: 2005 April 22 USB 1.1 'Open' Host Controller (OHCI) Driver (PCI)
Registering block device major 202
 xvda:<6>device-mapper: ioctl: 4.11.0-ioctl (2006-09-14) initialised: dm-devel@redhat.com
  Reading all physical volumes.  This may take a while...
 xvda1 xvda2
  No volume groups found
  Volume group "VolGroup00" not found
_
```

First boot configuration

During the first boot after the installation has finished you will be see the 'First Boot' configuration screen

This screen will walk you through some basic configuration choices for your guest.



Welcome

There are a few more steps to take before your system is ready to use. The Setup Agent will now guide you through some basic configuration. Please click the "Forward" button in the lower right corner to continue.

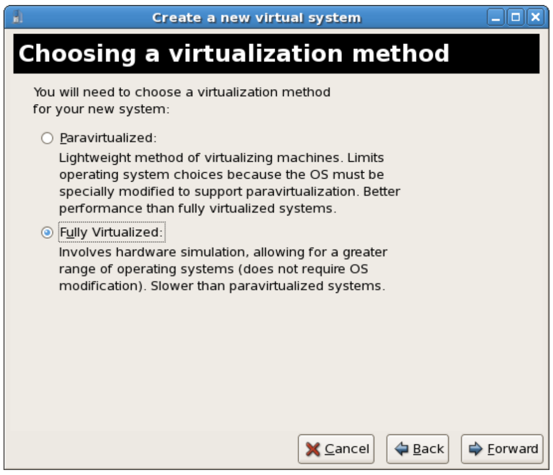
The remaining processes are just same for virtual and real machine, so we won't mention here.

Installing a Windows XP or 2003 Server Guest (fully-virtualized or HVM Guest)

Virtual Machine Manager can also be used to install a Windows based guest (as a fully-virtualized guest). However there are some extra steps which must be followed in order to complete the installation successfully. This section will explain the installation process and which extra steps need to be performed during the installation.

First you start virt-manager and select the 'New' tab to create a new virtual machine

As you install a Windows based virtual machine you need to select the option to install a 'Fully-Virtualized' guest.



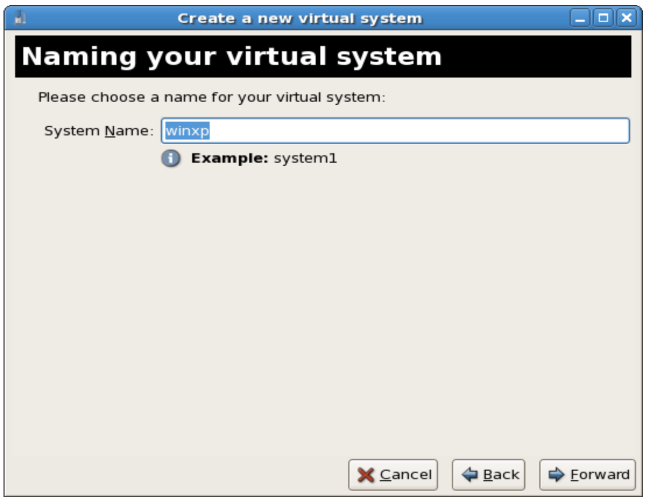
Choosing a virtualization method

You will need to choose a virtualization method for your new system:

( ) Paravirtualized:
Lightweight method of virtualizing machines. Limits operating system choices because the OS must be specially modified to support paravirtualization. Better performance than fully virtualized systems.

[✓ Fully Virtualized:]
Involves hardware simulation, allowing for a greater range of operating systems (does not require OS modification). Slower than paravirtualized systems.

Next you choose a descriptive system name



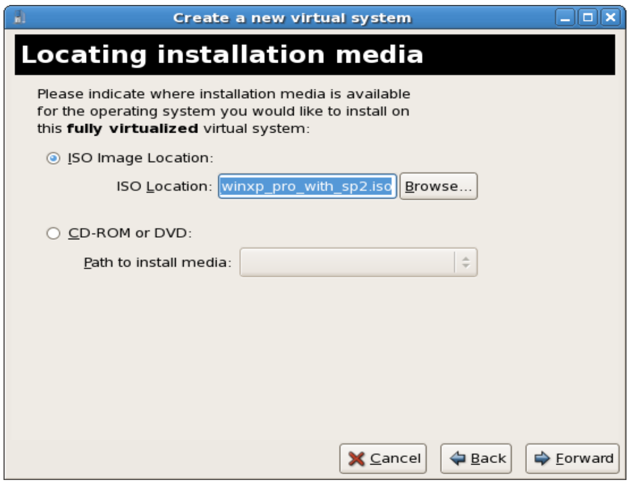
Naming your virtual system

Please choose a name for your virtual system:

System Name: winxp

Example: system1

Specify the location for the ISO image you want to use for your Windows installation.



Create a new virtual system

Locating installation media

Please indicate where installation media is available for the operating system you would like to install on this **fully virtualized** virtual system:

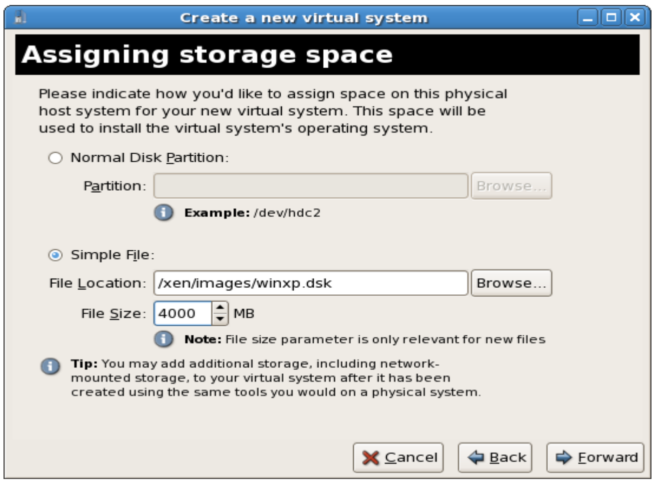
[✓ ISO Image Location:]

ISO Location: winxp_pro_with_sp2.iso

CD-ROM or DVD:

Path to install media:

Select the storage backing store, either a file based image can be used or a partition / logical volume. Next step is to specify the virtual machine resources such as CPU and Memory.



Create a new virtual system

Assigning storage space

Please indicate how you'd like to assign space on this physical host system for your new virtual system. This space will be used to install the virtual system's operating system.

Normal Disk Partition:

Partition:

Example: /dev/hdc2

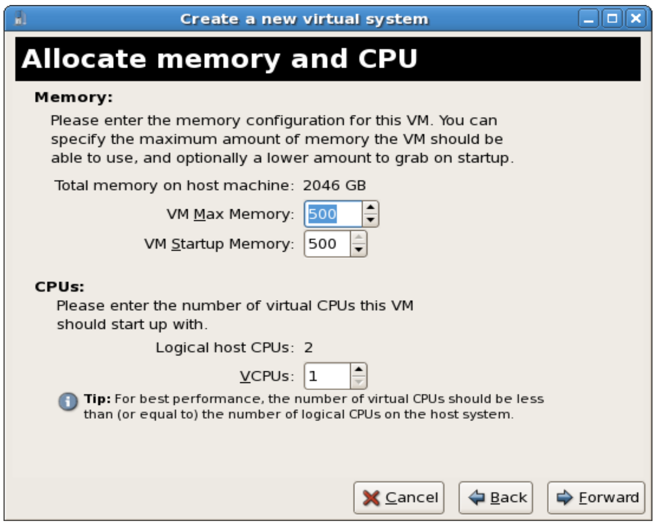
[✓ Simple File:]

File Location: /xen/images/winxp.dsk

File Size: 4000 MB

Note: File size parameter is only relevant for new files

Tip: You may add additional storage, including network-mounted storage, to your virtual system after it has been created using the same tools you would on a physical system.



Create a new virtual system

Allocate memory and CPU

Memory:

Please enter the memory configuration for this VM. You can specify the maximum amount of memory the VM should be able to use, and optionally a lower amount to grab on startup.

Total memory on host machine: 2046 GB

VM Max Memory: 500

VM Startup Memory: 500

CPUs:

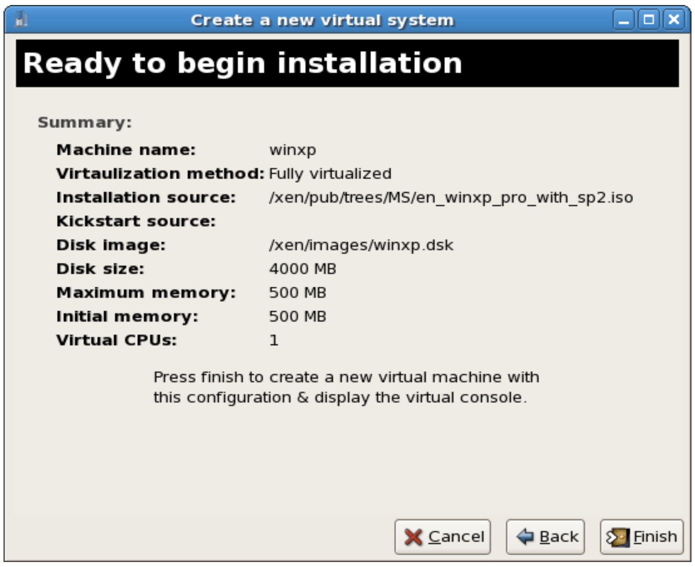
Please enter the number of virtual CPUs this VM should start up with.

Logical host CPUs: 2

VCPUs: 1

Tip: For best performance, the number of virtual CPUs should be less than (or equal to) the number of logical CPUs on the host system.

Before the installation will continue you'll be presented with a summary screen and once you're satisfied with the setup simply press 'Finish' to proceed to the actual installation.


Create a new virtual system

Ready to begin installation

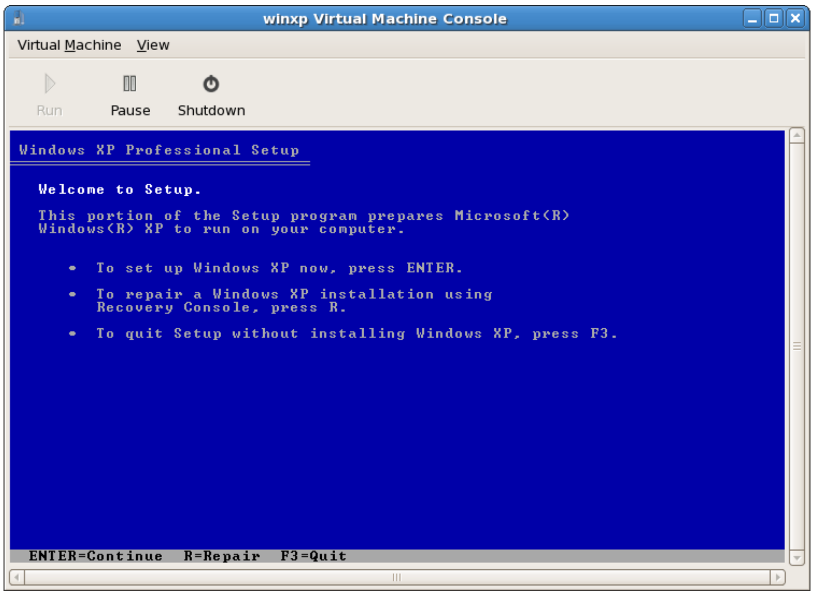
Summary:

Machine name:	winxp
Virtaulization method:	Fully virtualized
Installation source:	/xen/pub/trees/MS/en_winxp_pro_with_sp2.iso
Kickstart source:	
Disk image:	/xen/images/winxp.dsk
Disk size:	4000 MB
Maximum memory:	500 MB
Initial memory:	500 MB
Virtual CPUs:	1

Press finish to create a new virtual machine with this configuration & display the virtual console.

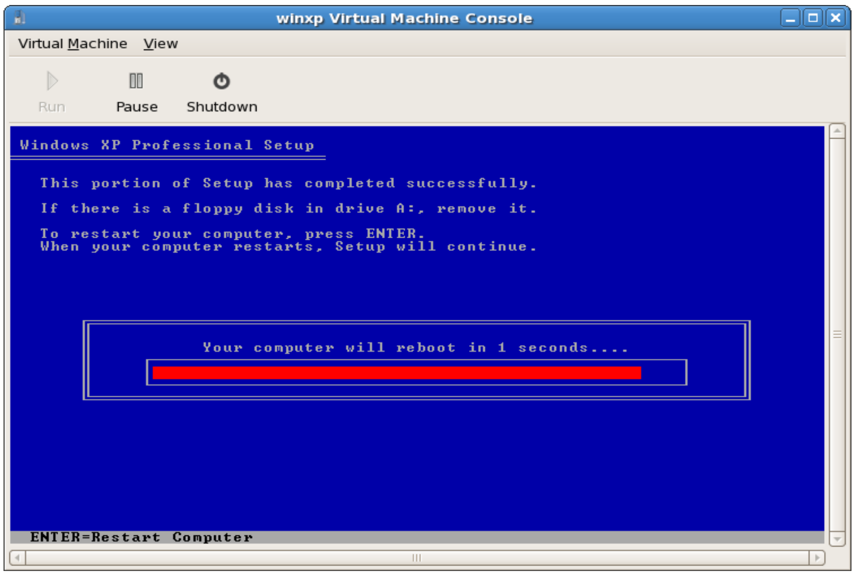
Now the actual Windows installation will start.

- As you need to make a hardware selection it is important that you open a console window very quickly after the installation has started.
- Once you press 'Finish' make sure you set focus to the virt-manager summary window and select your newly started Windows guest. Double click on the system name and a console window will open.
- Press quickly F5 to select a new HAL, once you get the dialog box in the Windows install select the 'Generic i486 Platform' tab (you can scroll through the selections using the Up/Down arrows).
- The installation will proceed like any other standard Windows installation.



```
Windows XP Professional Setup

  Welcome to Setup.

  This portion of the Setup program prepares Microsoft(R)
  Windows(R) XP to run on your computer.

      •  To set up Windows XP now, press ENTER.

      •  To repair a Windows XP installation using
         Recovery Console, press R.

      •  To quit Setup without installing Windows XP, press F3.

  ENTER=Continue   R=Repair   F3=Quit
```

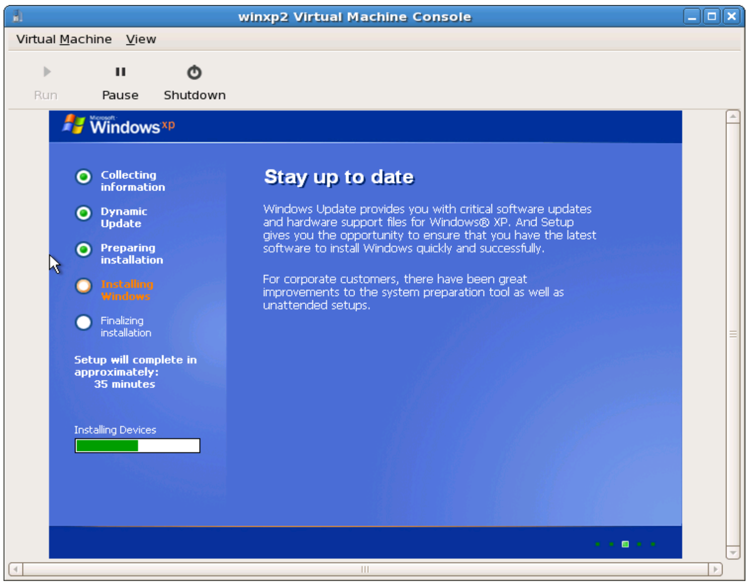
After setup has completed your Windows virtual machine will be rebooted.



```
Windows XP Professional Setup

  This portion of Setup has completed successfully.

  If there is a floppy disk in drive A:, remove it.

  To restart your computer, press ENTER.
  When your computer restarts, Setup will continue.

            Your computer will reboot in 1 seconds....

ENTER=Restart Computer
```

- You will have to halt the virtual machine after its initial reboot as you need to manually edit the virtual machine's configuration file which is located in /etc/xen/{VirtualMachineName}
- You can halt the virtual machine using the command
  # xm destroy WindowsGuest (where WindowsGuest is the name of your virtual machine)
- You will need to modify the 'disk' entry and add a 'cdrom' entry to the config file
- The 'old" en try will look similar to
  disk = ['file:/xen/images/winxp.dsk,hda,w']
  and the new entry should look like the following
  disk = ['file:/xen/images/winxp.dsk,hda,w' ,
  'file:/xen/pub/trees/MS/en_winxp_pro_with_sp2.iso,hdc:cdrom,r',]
- Now you can restart your Windows virtual machine using the command
  # xm create WindowsGuest (where WindowsGuest is the name of your virtual machine)

Once you open the console window you will see Windows continuing with the setup phase.



Stay up to date

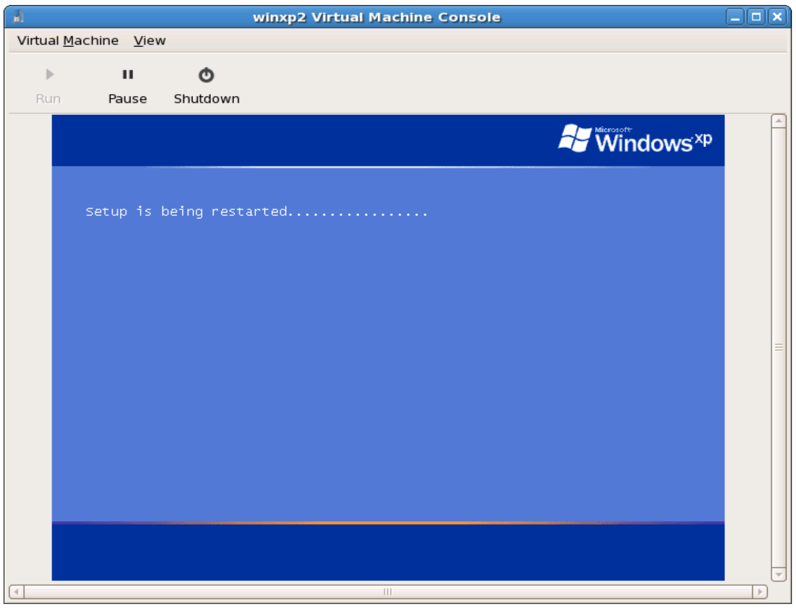
Windows Update provides you with critical software updates and hardware support files for Windows® XP. And Setup gives you the opportunity to ensure that you have the latest software to install Windows quickly and successfully.

For corporate customers, there have been great improvements to the system preparation tool as well as unattended setups.

Setup will complete in approximately: 35 minutes

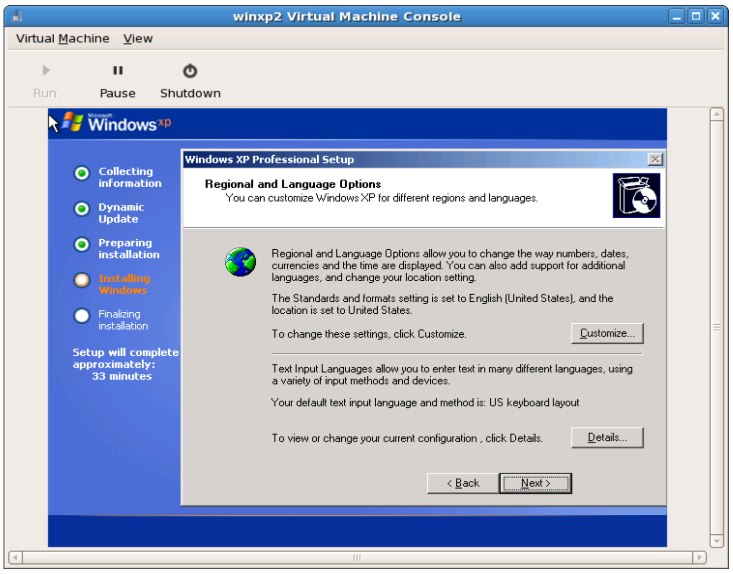
Installing Devices

If your installation seems to get stuck during the setup phase you can simply restart the virtual machine using the command mentioned above. This will usually get the installation to continue.



```
Setup is being restarted................
```

After setup has finished you will see the Windows boot screen.



Regional and Language Options
You can customize Windows XP for different regions and languages.

Regional and Language Options allow you to change the way numbers, dates, currencies and the time are displayed. You can also add support for additional languages, and change your location setting.

The Standards and formats setting is set to English (United States), and the location is set to United States.

To change these settings, click Customize. [Customize...]

Text Input Languages allow you to enter text in many different languages, using a variety of input methods and devices.

Your default text input language and method is: US keyboard layout

To view or change your current configuration , click Details. [Details...]

[< Back] [Next >]

Now you can continue with the standard setup of your Windows installation.

Configuration of live migration

To enable live migration of guest OS, we have to configure our Xen daemon to let it listen to migration request and do the live migration process, in the file /etc/xen/xend-config.sxp, edit it and make sure it got these lines uncommented:

```
(xend-relocation-server yes)
(xend-relocation-hosts-allow '')
```

Reboot the server or restart the “xend” service to let the changes take effect.

Simple live migration configuration using NFS

Live migration of guest OS requires a shared writable storage, in real life we should use SAN storage and cluster file system likes GFS which is included in RHEL 5 AP. But for experiment we can use a NFS volume to test for functionality in 2 hosts environment.

Install the paravirt guest OS on one of the hosts, the guest OS's disk images should be located at /var/lib/xen/images. Make /var/lib/xen/images exported by NFS and writable for other hosts, by adding a line in /etc/exports:

```
/var/lib/xen/images *(rw,sync,no_root_squash)
```

You need to restart the nfs service by “/etc/init.d/nfs restart” to take effect. This will make the disk images accessible on other host.

On the other host, try mount the shared disk images by:

```
mount -t nfs x.x.x.x:/var/lib/xen/images /var/lib/xen/images
```

where x.x.x.x is the IP address of the host sharing out the disk images.

Copy the configuration of the guest OS to the 2nd host, the configuration of the guest OS should be /etc/xen/[guest OS name], eg. /etc/xen/rhel5pv. Now you can start the guest OS on either host. If both of the hosts can start and shutdown the guest OS, you can now try to test the live migration.

While the guest OS are running on one host, you can migrate it to other host by:

```
xm migrate –live rhel5pv host2
```

Assuming the guest OS name is rhel5pv and the host which the guest OS need to migrate to is host2.

The live migration process should take is related to network speed and guest OS memory size, for a gigabit network connection, a guest OS of 512M RAM should be migrated in about 5-6 seconds.

Appendix

Basic commands used in a Red Hat Enterprise Linux Virtualization environment

XM

console	Attach to <Domain>'s console.
create	Create a domain based on <ConfigFile>.
destroy	Terminate a domain immediately.
domid	Convert a domain name to domain id.
domname	Convert a domain id to domain name.
dump-core	Dump core for a specific domain.
list	List information about all/some domains.
mem-set	Set the current memory usage for a domain.
migrate	Migrate a domain to another machine.
pause	Pause execution of a domain.
reboot	Reboot a domain.
rename	Rename a domain.
restore	Restore a domain from a saved state.
save	Save a domain state to restore later.
shutdown	Shutdown a domain.
sysrq	Send a sysrq to a domain.
top	Monitor a host and the domains in real time.
unpause	Unpause a paused domain.
uptime	Print uptime for a domain.
vcpu-list	List the VCPUs for a domain or all domains.
vcpu-pin	Set which CPUs a VCPU can use.
vcpu-set	Set the number of active VCPUs for allowed for the domain.
dmesg	Read and/or clear Xend's message buffer.
info	Get information about Xen host.
log	Print Xend log
sched-credit	Get/set credit scheduler parameters.
vnet-list	List Vnets.