# CHAPTER 1

# Install and Configure ESX 4 Classic

Currently, two versions of ESX are available: ESX 4.*x* and ESXi 4. Functionally, they offer the same hypervisor and kernel. The ESX 4.*x* version ships with an installer, which copies the source code to disk. ESXi is generally distributed on a USB stick held inside the physical server. The front end of the physical console of the two versions differ substantially. ESX 4.*x* management's front end is called the Service Console, and it is based on Red Hat Linux. ESXi offers a configuration shell based on a system called BusyBox.

VMware's stated long-term plan is eventually to discontinue the 4.*x*.*x* version in preference to ESXi, which offers a smaller footprint, simplifies patch management, and provides other advantages (discussed in the next chapter). However, for the moment, we have two versions available: the "full-fat" ESX 4.*x* version, which comes with the Service Console. I refer to this as *ESX Classic*. Additionally, we have the "skinny latte" ESXi 4 version, which has a more appliance-based front end, rather than an interactive command-line shell.

Despite the advantages of ESXi, the Classic edition of ESX with the Service Console is still the dominant platform, which is why I've started with its installation and configuration. Chapter 2 covers ESXi installation and configuration.

In this book, I will be using both versions to validate the technical content against the two flavors, and show you how to manage the ESXi 4 host in practically the same way as ESX Classic. Although they may be managed by the same vCenter server, I will deliberately separate them into different VMware clusters. Blending ESX Classic and ESXi together is not recommended; you risk creating surprising new features. (I'm using "features" in the same way as Microsoft!)

Before we get to the installation ESX Classic installation procedure, let's take a moment to consider just what ESX is.

**What's New in ESX?**

- 64-bit only
- Support for 512GB of RAM, 64 cores, and any combination up to a maximum 512 vCPUs includes1-, 2-, 4-, and 8-way virtual machines. There is experimental support for 1TB of memory. To configure a VM with 8 vCPU you will need an Enterprize Plus license.
- Support for up to 512 vCPUs on single ESX host with an increase of supported number of VMs to 320
- Native SATA disk support
- Support for IPv6
- Service Console as a virtual machine that uses a virtual disk
- Memory increased to 300MB for the Service Console
- Setting of NTP in the installation
- Built-in scripted installation option for the installer
- No longer licensed by a .lic file; licensed through locally held license strings or licenses held in the VMware vCenter database

## What Is ESX?

As the years roll by, the function of the server platform, VMware ESX, has been subtly changing. When I first started teaching ESX (we didn't even formally cover Virtual Center back then), we had a PowerPoint slide that clearly stated what ESX wasn't. This slide said that ESX wasn't just Linux with bells on or an operating system. This distinction has been somewhat muddied in recent years, even by VMware. If you take a look at the new VMware architecture and "big-picture" PowerPoint slides, you'll see they have inserted the concept of the Virtual Datacenter Center Operating System (VDC-OS).

With the advent of cloud computing and the introduction of vSphere and VDC-OS, there has been a significant repositioning of the VMware offering. Additionally, the ESX host now has an increasing number of APIs that other organizations can plug into—projects like VMsafe, vNetwork, and vStorage, which allow companies like McAfee, Cisco, and EMC to plug into ESX. This increasingly makes ESX more like an operating system than it has been in the past.

So, is ESX actually an operating system? Personally, I don't think it is. ESX does carry out one aspect of an operating system, which is to be the arbitrator to hardware. ESX "owns" the physical hardware and sits at ring 0, the most privileged ring that exists in most x86 processors. Fundamentally, if any process needs access to hardware resources (CPU, disk, network, memory, and so on), it must liaise with ESX's VMkernel for that resource.

However, ESX is not an operating system in the sense that it does not provide a rich collection of programming APIs such that an independent software vendor (ISV) could develop an application to run inside the VMkernel as a process. ESX will not run Microsoft Word or Microsoft Exchange Services—well, not directly. Apart from its own internal processes and drivers, the only thing ESX executes is the virtual machine, sometimes referred to as the virtual machine monitor. Now, of course, you could install any supported guest operating system on the virtual machine, and then run any supported application within it. For ESX, a virtual machine is just another process, albeit a very important one. Everything that ESX does is an attempt to make that process more stable, reliable, and faster than the competition offers.

Of note is that the guest operating system's role has been depreciated significantly. Applications that can run as the guest operating system now manage only *virtual* hardware, not *physical* hardware. VMware terms this depreciated guest operating system JeOS (pronounced "juice"), which stands for Just Enough Operating System. The concept basically is one of stripping down a fat operating system to a much slimmer one, leaving behind just enough to run the application. The point is that it's a mistake to take a general-purpose operating system and retrofit a virtualization piece to a kernel that is decades old. You inherit all the vulnerabilities and design flaws of the original, and try to make it do a job for which it wasn't designed. It is better to start from scratch with a brand-new file system designed for storing virtual machines. For many, ESX represents a return to that super-skinny, super-efficient coding of the past that we lost somewhere between an operating system needing 64KB of memory to run to one needing 2GB.

Let's get back to the question of what ESX *is*. ESX is what is called a *hypervisor*. The hypervisor—ESX's VMkernel—sits between the hardware and the virtual machine. Its primary task is to run virtual machines, as well as carry out other important ancillary tasks such as monitoring the performance of the processes it manages. This VMkernel is incredibly slim, leaving less of a "surface" area for vulnerabilities that could make it unstable or open to attack by intruders.

So, ESX is like an appliance that sits in your rack. Indeed, VMware could have taken the original equipment manufacturer (OEM) model of buying servers with truckloads of CPU and RAM, and then yanked the bezels off the front, attached a VMware bezel, and sold ESX as a VMware appliance. Fortunately, VMware didn't use this hardware route to market. Instead, the company partnered with existing OEMs and leveraged the OEMs pre-sale, support, and access to clients. For example, ESX is shipped on a USB stick inside certain IBM servers. To boot from the ESX USB stick, you merely enable it in the BIOS. This is done with all servers of this type, despite the fact that the customer might not use ESX. It's a good indication of how ESX has become the de facto platform in the corporate datacenter for running virtual machines.

I guess the safest thing to say about ESX is like a lot of systems, it's a work in progress. From its very creation, it has been endless evolving. And by the end of this book, you might indeed regard it as the VDC-OS.

Now let's get ready to install ESX.

## Preparing for ESX Installation

You will be pleased to know that ESX is very easy to install. You are asked even fewer questions than for the ESX 3.*x* installation, and the biggest part of the installation is the file-copy event.

You can download ESX 4.*x* from VMware's web site once you have been provided with the correct serial numbers and logins.

### Confirming Physical Settings

Before you install ESX, you should check the physical configuration of the server. To be more specific, you should run updates against your main system BIOS and firmware upgrades for devices such as host bus adapters (HBAs). As you know, servers tend to ship with factory software that isn't always up to date, and you may encounter unpredictable performance and lack of stability without the prerequisite updates to your system.

Additionally, I recommend that you check your BIOS settings to make sure additional features are enabled. Certain advanced features of vSphere 4 require special options to be enabled in the BIOS. For example, for VMware fault tolerance to work, you must have "virtualization hardware assist" options like Intel Virtualization Technology (Intel VT) or AMD Virtualization (AMD-V) enabled. Enabling these options before you begin your installation will save you time in the long run.

Finally, check that all your cables and connections are good by using the LEDS that appear on network interface cards (NICs) and HBAs. I know this sound very obvious—indeed, even patronizing to anyone who has worked in IT for some time—but in reality, the obvious is frequently overlooked.

### Checking Your Source Code ISO

After downloading the source ISO file, you should check that file before you burn it to a CD. You can do this by confirming the MD5 checksum. An md5sum program allows you to confirm your ISO file has not been corrupted by the download process. The source MD5 sum value is normally put under the link to the ISO or file you are downloading.

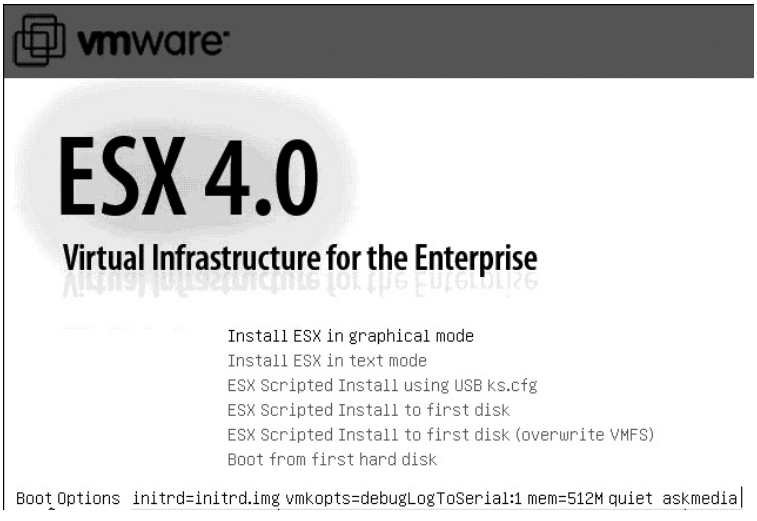
If you are running Linux on your desktop PC, you already have access to md5sum on the command line. If, you are running Microsoft Windows, you can use a free tool, such as

the one available from Nullriver Software (http://www.nullriver.com/) to check that your file is not corrupted.

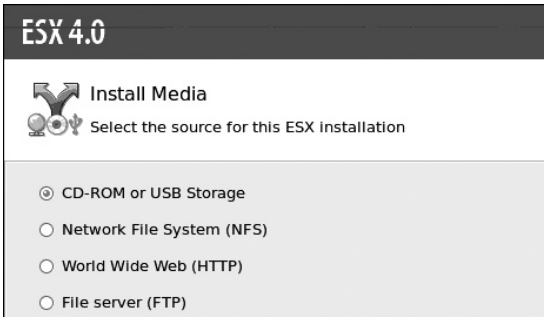
## Using an ILO

The installation medium is a DVD, rather than a CD-ROM ISO image. As such, the installation with virtual media can be incredibly slow. If you are using virtual media to start the boot process, I recommend that you use your ISO with the "virtual media" features of your server's ILO, if it has one. In the past, users have had problems with the installer engine and ILOs; however, these issues appear to have been resolved in this release.

To use this method, first copy the contents of the ISO file to a web server accessible to your ESX hosts. At the install prompt, press F2 on the keyboard to pipe a value to the installer engine, called askmedia.




This will allow you to boot from the virtual media, but midway through the installation process, specify the path to the web server for the file-copy process.




This significantly reduces the time to install ESX.

***NOTE*** *You may as want to investigate a totally scripted installation, perhaps using Pre-boot Execution Environment (PXE) booting together with a web server. Chapter 17 covers PXE-based deployment.*

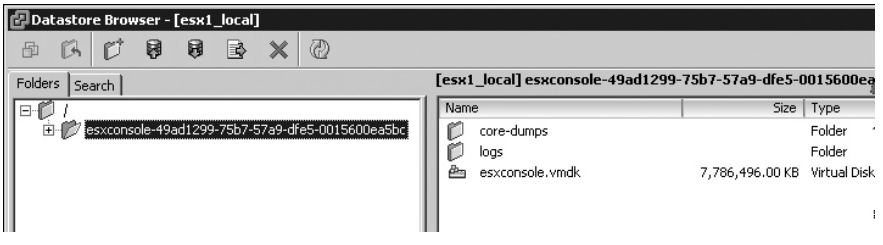
If you are using an ILO to do your installation, you may find that the mouse pointer is simply not there or does not operate correctly. This is due to the limitations of some ILOs, as opposed to the ESX installer itself. If you do use the graphical installer and your mouse does not work, you can use the keyboard. as follows:

- TAB moves forward.
- SHIFT+TAB moves backward.
- The spacebar opens pull-down lists.
- The cursor keys navigate lists.
- ENTER selects items.

## Installing ESX 4 Locally

The installation engine behind ESX is "weasel," which is a modified version of the Anaconda installation engine used for Red Hat Linux. When the system boots to the installer, it actually runs a very slim Linux distribution called BusyBox. BusyBox's job is to load the files required to install ESX, including Python scripts used by the weasel installer engine.

The first major task the BusyBox environment carries out is to load the VMkernel, which allows you to configure an ancillary virtual machine called the Service Console. The Service Console used to be very much a physical system, and was not completely virtual. For example, in previous releases, the Service Console could see a physical drive; however, in this release, it sees a virtual disk. The Service Console's virtual disk is created on a Virtual Machine File System (VMFS) volume and can be seen by using the Datastore Browser.




***NOTE*** *The Service Console is just a management front end that allows you to log in to the ESX Classic host, and carry out task through a command-line interface (CLI). It does not own or manage the virtual machines.*

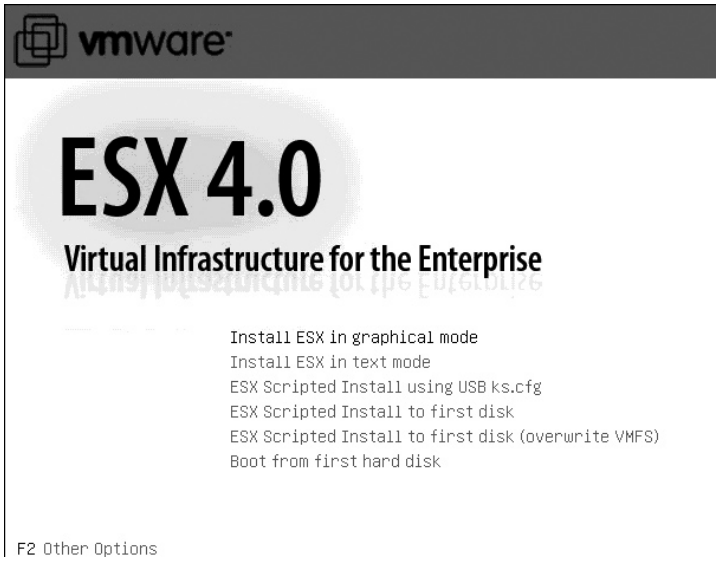
In this example, I guide you through an installation to local logical unit numbers (LUNs). Later in the chapter, I will show you how to set up to boot from a storage area network (SAN). If you are installing ESX to the local storage, I recommend disconnecting the SAN cables if you can. This prevents any chance of installing ESX to SAN unintentionally, as well

as reduces your chances of accidentally destroying terabytes of data on the SAN while installing ESX! If physical access is limited—perhaps your hardware is miles away in a colocation—you could use the BIOS of the physical host to temporarily disable the Fibre Channel card. In the installer, this level of access should be there already. However, if you also manage the storage layer, by far the easiest way to prevent access to the storage array is by using your vendor's array management software.

As I've said, ESX installation is easy. Only two areas could cause you problems: selecting the correct network card for use with the Service Console and building a robust and reliable partition table. I'll talk more about these areas as we step through the procedure.

## Starting the Installation

Begin by booting to the ESX 4.*x* CD. The opening installation screen presents six choices for completing the installation.




***NOTE*** *Notice the banner reads "Virtual Infrastructure for the Enterprise." The old name Virtual Infrastructure survives!*

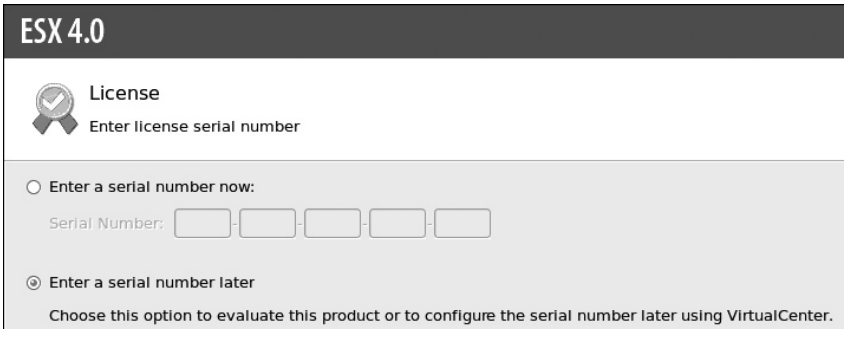
The text mode option is very different from that in the ESX 2.*x* and 3.*x* installation program. It is a purely CLI-based installation, with each choice confirmed by a 1 for yes or 0 for no type of response. In many respects, the graphic installation is *much* more friendly and intuitive.

The third option can be used for a scripted installation with the script file held on a USB stick. The options for scripted installations from to the first disk are dangerous if you are connected to a SAN. They may actually pick out a SAN LUN, rather than local storage. The final option merely bypasses the installer altogether and boots whatever operating system is located on the first hard drive. You would use this option if you accidentally left the ESX CD in the drive, for example.

Pressing F2 from this screen allows you to pass custom weasel variables to any one of the first five installation options.

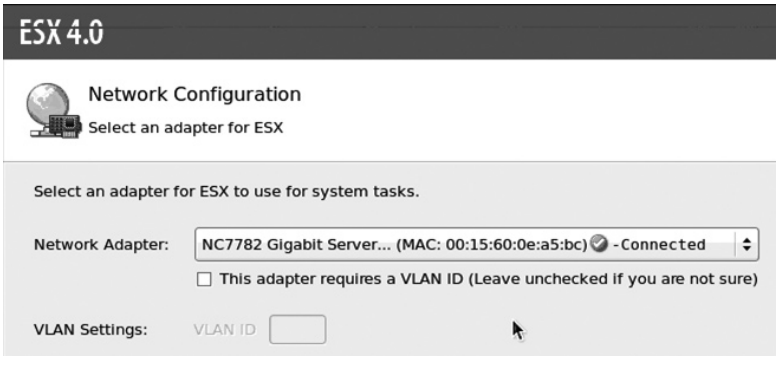
Follow these steps to install ESX in graphical mode:

1. At the initial screen, press the ENTER key to choose the graphical installation. (If you simply wait 60 seconds, the installer will enter the graphical mode by default.)
2. The installer will proceed to the Load ESX screen. Click Next.
3. Agree the end user license agreement (EULA).
4. Select your keyboard type.
5. When prompted to select custom drivers, choose No, and then select Yes to allow the installer to load the built-in drivers supplied on the ESX CD. In the rare case that the hardware you are using is so new that you need to supply a driver that is not on the ESX CD, you will then need to select custom drivers (this is like pressing F6 during a Windows installation). Once the status is 100% complete, click Next.
6. Optionally, enter your serial number. As stated earlier, VMware has abandoned the use of license files (.lic) adopted in VMware Infrastructure version 3 (Vi3), and reverted back to using license strings. If you are not using vCenter, you should enter your ESX host license now. If you are using vCenter, you can choose the "Enter a serial number later" option. Your ESX host will continue to function in an evaluation mode up to 60 days until you license it with vCenter.




## Specifying Your Network Configuration

Next, the installer asks you to select a network adapter for ESX and optionally set the virtual local area network (VLAN) ID, which will be used by the Service Console.

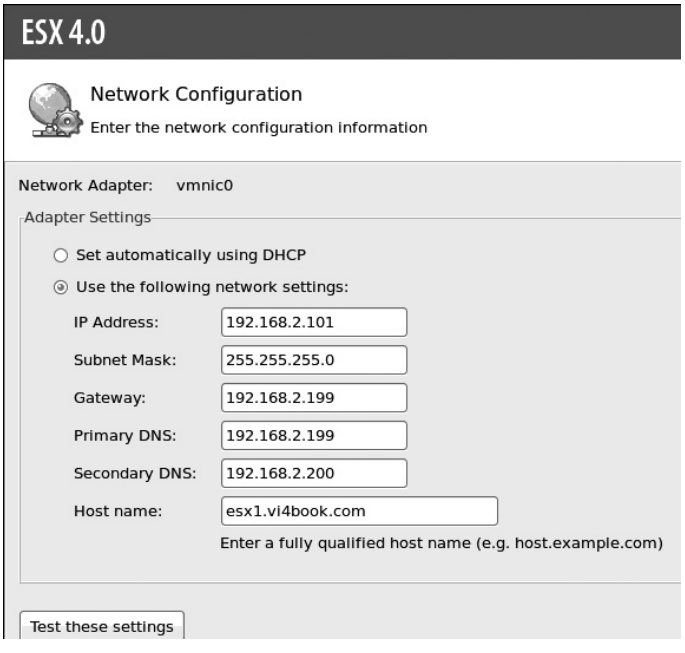


***CAUTION*** *If you select the wrong network interface card (NIC), you will not be able to communicate with the ESX host after the installation.*

How do you know you have selected the correct NIC during the installation? If after installing the ESX product, you find you cannot even ping the Service Console interface, it could be that you selected the wrong NIC during the install. If this happens to you, you may wish to consult the end of Chapter 4 entitled "Managing Service Console Networking." We used to see PCI information (Bus, Slot and Function numbers) and Speed/Duplex settings. Unfortunately, this is no longer displayed, however what is being displayed is whether the NIC is connected (there's little point in selecting a disconnected NIC now is there?) and the MAC address of the physical NIC. In fact you might prefer this data to PCI Bus:Slot :Function numbers as it is generally easier to find out the MAC address of NICS.

You should set the VLAN ID only if you're using VLAN tagging, and trunk ports have been set up on the physical switch. (If none of this makes sense, it will once you have read Chapter 4.)

Next, set your IP address, subnet mask, default gateway, primary DNS, secondary DNS, and fully qualified domain name (FQDN) for the host name.



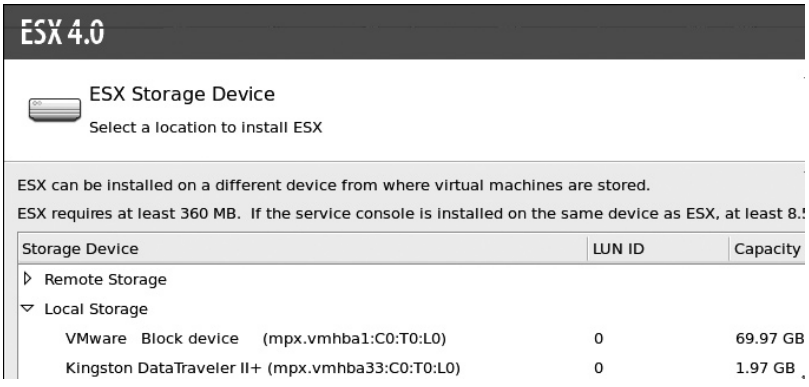

An FQDN and DNS name resolution are required for many key features. I recommend that you have your DNS infrastructure set up and in place before beginning the ESX installation. I also recommend using a static IP address, rather than using DHCP with "client reservations." This is pretty much a universally accepted standard in Europe, the Middle East, and Africa (EMEA) for server-based operating systems, although this convention may differ in different regions.

New in ESX 4.*x* is a method to test your IP configuration. It's by no means foolproof. The test button checks only for syntax, such as accidentally typing a comma instead of

a period in the fields. It does not check if the IP address of your default gateway or DNS are pingable. The result you want to see is "No problems detected with the network settings."

## Setting Up Your Storage and Datastore

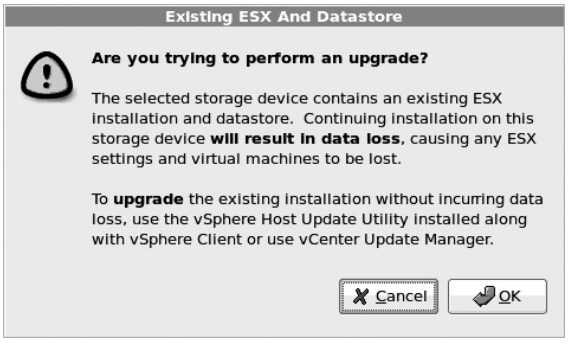
After completing your network settings, select Advanced as your partition scheme, and then select the LUN/disk on which you wish to install ESX. The installer will then enumerate your LUNs.




If you left the SAN connected (which is not best practice), you will see both remote and local storage. In the screenshot shown here, the "VMware block device" is actually a Smart Array P600 SAS controller inside a HP ProLiant G1 DL385. C0-T0-L0 is the first controller (C0), first target (T0), and first and only LUN (L0). The Kingston DataTraveler is a copy of ESXi 4, which is held on a USB memory stick. I will be booting to this in Chapter 2 to show you how the ESXi product is configured.

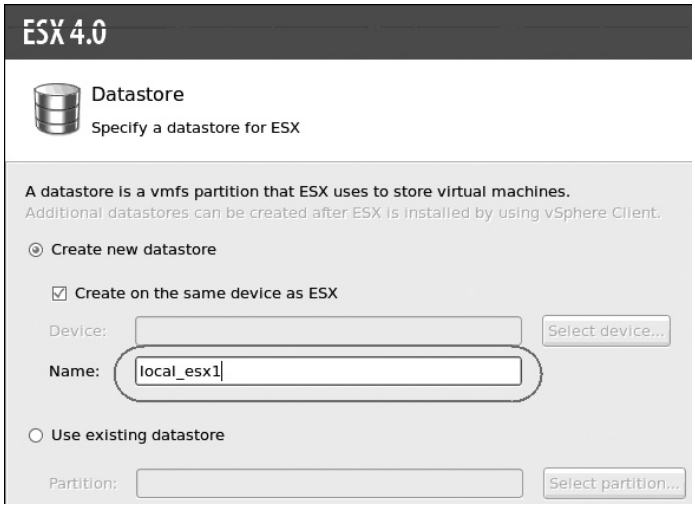
At this stage, you are selecting a physical disk that will hold your first VMFS volume. This is mainly necessary to give ESX a location in which to create the virtual disk of the Service Console.

You may receive a warning that the /dev/sdN is unreadable. This reference to /dev/sdN indicates the SCSI disk (SD), which is found first (A, B, C). Also, the dialog box should indicate on which controller this disk was found. Additionally, if you have previously installed ESX and are trying to reinstall the product, you will receive this warning message:



As the dialog box indicates, VMware no longer supports using the DVD media to upgrade an ESX host. Instead, you must use either the vSphere Host Update Utility (which is limited to ESXi only) or vCenter Update Manager (which can upgrade both ESXi 3 and ESX 3).

After selecting your storage device, specify a datastore for ESX. By default, the installer will assume you require a VMFS volume, and that this volume will be created on the same LUN as selected in the previous part of the installation, because of the default option "Create on the same device as ESX." The default volume name of "storage" is not a good one. I suggest that you use a more unique naming convention, something like local_ *hostname* (such as local_ esx1).




As the partition table doesn't exist, the installer will wish to initialize the disk. This operation is potentially dangerous if there is data on the drive. It is fine to do this if you know the LUN is blank. If your ESX host is connected to a SAN, be very careful in accepting yes!

## Creating the Partitions

Like most operating systems, the reliability of the build will come from the installer's decisions. It's critical that your partition scheme be able to protect itself from rapidly filling event log files and users copying large files to the wrong location.

Rather than using drive letters (C:, D:, E:, and so on) to address these partitions, folders are used as mounting points, because ESX is based on Linux/UNIX, where this is a standard. (You have been able to do something similar to this with Microsoft operating systems since Windows 2000 was released.) As for the physical disks themselves, I usually recommend two 36GB or 72GB disks in a mirror.

Remember that now that the Service Console is fully virtualized, you have no control over the partitions that are created on the physical disk (in particular, the /boot and vmkcore partitions). You can control only the destination of these partitions and the VMFS volume used to hold the Service Console's virtual disk. If you use autopartitioning, the installer will place the first three partitions (boot, vmkcore, and local vmfs) on a physical drive selected earlier in the installer. The remaining partitions (swap, var/log, and /) are created inside a virtual disk. Table 1-1 summarizes the physical and virtual disk partitions and their purposes.

| Physical Disk | Purpose |
|---|---|
| /boot | Holds boot .img files called by the GRUB boot loader |
| vmkcore | Used to collect diagnostics information in the event of a VMkernel crash or PSOD |
| extended partition | Takes up the whole of the volume selected during the installation |
| vmfs | Creates a VMFS volume within the extended partition using all remaining free space |
| **Virtual Disk** | **Purpose** |
| swap | Areas of swap space used by the Service Console |
| /var/log | Used to keep your log files out of the / partition |
| / | Main "operating system" partition, which is where the Service Console is installed |

**Table 1-1** Physical and Virtual Disk Partitions

Few in the VMware community use the automatic, "Recommended" partition scheme offered by the installer. Nearly everyone in the VMware community uses their own manual partition scheme, based on their personal experience and company standards.

The VMware community has debated the advantages and disadvantages of various partitioning approaches in a long and interesting forum thread, which was started by Steve Beaver, a very good friend of mine. If this interests you, check out http://communities.vmware.com/thread/46345. There are as many partition schemes as there are people on this forum. The one I recommend here is just one example.

Table 1-2 shows a summary of the partition scheme I suggest for the Service Console virtual disk. This partition scheme uses about 13GB of space. You could easily increase these values and add more partitions. Generally, I create three primary partitions, and the rest are treated as logical drives in an extended partition. The disk management utility (called Disk Druid) automatically creates an extended partition for you using the remainder of the disk, once the installer has created the /boot and vmkcore partitions.

Here are some notes on the size and purpose of each partition:

- **ext3** EXT3 is the primary file system of Linux, which is an appropriate choice, since the Service Console is based on Red Hat Linux. Also available are two proprietary file systems, which are accessible only by the VMkernel: vmkcore and VMFS version 3 (VMFS is covered in the Chapter 6).

| Mount Point | File System | Size in MB | Purpose |
|---|---|---|---|
| N/A | swap | 1600 | Swap for Service Console |
| / | ext3 | 5120 | Main OS location |
| /var | ext3 | 2048 | Log files |
| /tmp | ext3 | 2048 | Temporary files |
| /opt | ext3 | 2048 | VMware HA logging |
| /home | ext3 | 2048 | Location of users' storage |

**Table 1-2** Recommended Partition Scheme for the Service Console Virtual Disk

- **/boot** This is where core boot files with an .img extension (bootable image files; analogous to ISO files, which can also be bootable) are stored. After powering on ESX, the master boot record (MBR) is located and the boot loader is run. In the case of ESX 4.*x*, this is now the Grand Unified Bootloader (GRUB). GRUB then displays a menu that allows you to select which .img file to execute. The /boot mounting point is automatically created and is reserved by the installer. You cannot create this partition or change its size.
- **/swap** This is where the Service Console swaps files if memory is low. I've chosen to overallocate this partition. The default amount of memory for the Service Console is 300MB. VMware usually takes this number and doubles it to calculate the swap partition size (600MB). The maximum amount of memory you can assign to the Service Console is 800MB. This is how I derived the 1600MB value. This means if you ever choose to change the default amount of memory assigned to the Service Console, you do not need to worry about resizing the swap partition. It doesn't have a mounting point, as no files from the administrator are copied there.
- **/** Referred to as the root partition, this is the main location where the ESX operating system and configuration files are copied. If you have a Windows background, you can see it as a bit like the C: partition and folders on that drive, like C:\Windows or C:\Program directory. If this partition fills, you may experience performance and reliability issues with the Service Console, just as you would with Windows or any other operating system.
- **/var** This is where log files are held. I generally give this a separate partition just to make sure that excessive logging does not fill the / file system. Log files are normally held in /var/log, but occasionally hardware vendors place their hardware management agent log files in /var. Another reason to use a separate partition is that agents or processes that crash dump their crash information into /var/core, which can sometimes fill quite rapidly when you have a bad agent installed.
- **/tmp** In the past, VMware has recommended using a separate partition for /tmp, which I have always done. As I have plenty of disk space, I have made this larger than it really needs to be.
- **/opt** Several forum members reported that they had seen the /opt directory fill up very rapidly in ESX 3.*x*, and as a consequence, fill the / partition. This location is also sometimes used as a logging location for hardware agents. In VMware, the High Availability feature generates logging data here as well, so I created a separate partition for it to make sure it does not fill the / partition.
- **/home** Technically, you don't need a separate /home partition. In the past, VMware recommended one for ESX 2.*x* in production systems. This was due to the fact that the virtual machine's configuration files (.vmx, .nvram, and .log) were stored in the /home location. In ESX 4.*x*, all the files that make up a virtual machine are more likely to be located on external storage. I still create this partition for consistency purposes, and any users on the local ESX server are more likely to create files here than in a directory in the / partition.
- **vmkcore** This is a special partition used only if the ESX VMkernel crashes, commonly referred to as the Purple Screen of Death (PSOD). If that happens, ESX writes debugging information into this partition. After a successful boot, the system will automatically run a script to extract and compress the data to a zip file in /root.

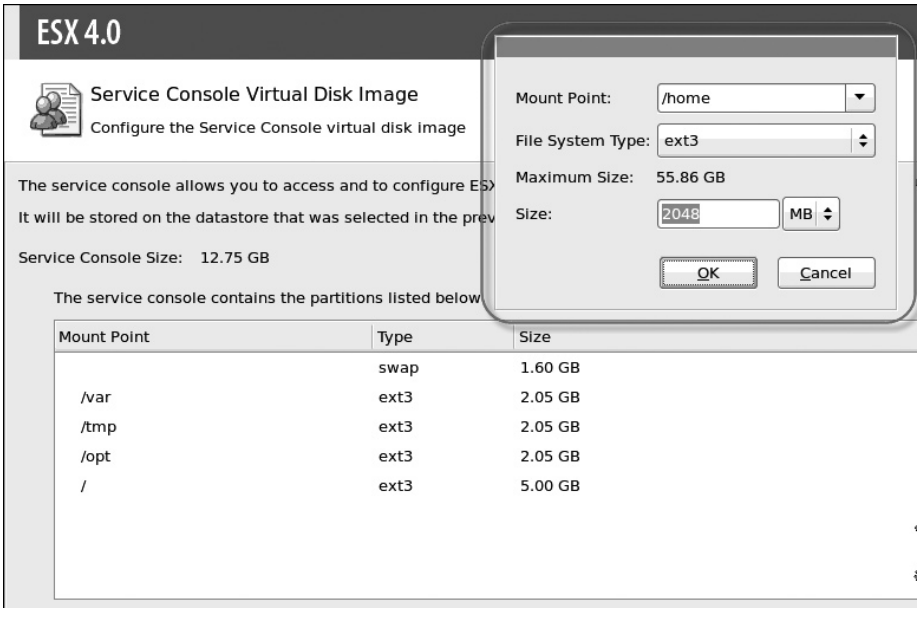
This file with a tar.gz extension can be sent to VMware Support, who will work to identify the source of the problem. In ESX 4.*x*, VMware automatically creates a vmkcore partition for you with a fixed size of 100MB. It is no longer possible to create the vmkcore partition type from the installation routine, since this partitioning routine is now being carried out within a virtual disk. (Remember that the Service Console is just a virtual machine, and as a consequence, it sees only a virtual disk; the VMkernel sees the physical disk.)

---

***NOTE*** *PSODs are normally caused by failures of RAM or CPU. You can see a rogue's gallery of PSODs at http://www.rtfm-ed.co.uk/?page_id=246.*

- **/vmimages** In ESX 2.*x*, a /vmimages partition or mounting point to a network location was used primarily for templates and ISOs. This partition is no longer required, as we now have more effective and easier ways of storing this data. Of course, if you are an ESX 2.*x* veteran who wishes to keep using /vmimages for consistency purposes, then that is fine. It's just no longer required or recommended. Personally, I still like to have a portion of disk space given over to this partition location as a "working area" when I am dealing with large files, although I've found that with my recent purchase of a SAN, this is something I use much less. If you have done a clean install, you will have a directory called /vmimages, even if you haven't created a /vmimages, which contains files used internally by VMware ESX. I will discuss this more in Chapter 7, which covers creating virtual machines.

The following screenshot shows the installer window for configuring the Service Console virtual disk image and the dialog box for creating a new partition. In this case, I selected /home from the pull-down list (you can also type in this area), chose ext3 as the partition type, and entered 2048MB as the size.

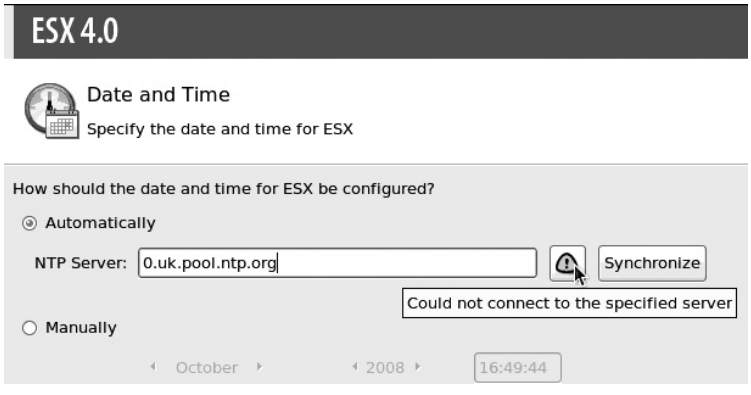


After you have built the partition table, take a few moments to confirm you created the correct number of partitions of the appropriate size. The partition table is not committed to the disk until you click Next. This would be an ideal opportunity to spot an error and correct it. Partition errors can be corrected afterward using fdisk, but it's much easier to get it right the first time. In my experience, a bad partition scheme usually results in wasted time spent reinstalling ESX.

## Completing the Installation

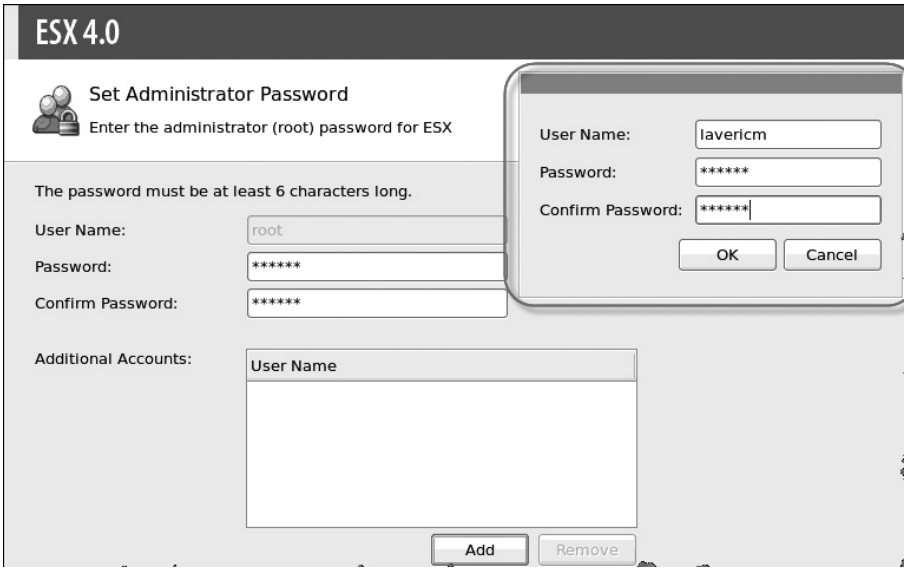
Once you have partitioned the virtual disk, the remainder of the installation procedure is very simple. It asks you many ancillary but important questions, such as your time configuration and user account configuration. Follow these steps to complete the installation:

1. Set your time zone. You can set the time by clicking the map. This interface can be difficult to use via an ILO because of mouse latencies. You might find it easier to click the Advanced button and use the PAGE DOWN key to set your time zone.
2. Set your time and date. In ESX 4.*x*, you can now set the Network Time Protocol (NTP) settings during the installation. This can save some time in the postconfiguration phase. Select the option Automatically, and type the name or IP address of your NTP server. The Synchronize button will attempt to sync your local time to the NTP service. If it cannot communicate, you will see an alert:




3. Set the root password. The root account is the user with highest privileges. You can create other ESX users within the installer. It's good practice to create a user for yourself now, as you will then be able to use a Secure Shell (SSH) client like

PuTTY immediately, without the need to load the vSphere Client to create a user account first. (Using PuTTY and installing the vSphere Client are discussed in the next section.)




4. You will see a summary of your choices. At this stage, you can go back and change some settings, such as your IP address, if you feel you have set them incorrectly. Clicking Next initiates the copy process.
5. At the end of the installation process, click Finish. This normally causes the CD-ROM to eject. When the system reboots, you get your first boot of the ESX server. If you are using virtual media to access the ESX 4.*x* ISO file, remember to disconnect it before the reboot. You might find that BIOS settings cause the server to boot to the CD again.

## Connect and Manage the ESX host

You can use an SSH client to access ESX's command line. For access to the graphical user interface (GUI), you use the vSphere Client.

### Gaining Command-Line Access

The SSH protocol allows you to gain access to ESX's CLI without the use of an ILO. A very popular tool used to create a remote SSH session is PuTTY, created by Simon Tatham. This tool is available from http://www.chiark.greenend.org.uk/~sgtatham/putty/.

Why would you want this command-line style access to ESX? One reason is that there are some tasks that can be done only via the command line. In other situations, because you have a mix of CLI and GUI tasks, it is easier to choose one interface rather than work with two.

However, in ESX 4.*x*, some new security settings disable SSH access to the root account. By default, in a clean install, there is no access to the ESX Service Console for the root account except via the ILO. This security change was introduced to enforce more user account traceability and an audit trail. User activity such as logons and changes in privilege are logged in /var/log/messages.

To get around this limitation, you can create a local account on ESX, use PuTTY to gain access, and then finally switch to the root account. To gain command-line access as root with SSH, open a SSH/PuTTy session to the ESX host and log in with your recently created account. To levitate to a higher plane of privileges type **`su -`**. Enter the root account password assigned during installation.

The `su` command allows you to "switch user" and assumes, unless otherwise specified, that you would like to change to root. The minus sign indicates you would like to use the root account's environmental settings, such as path variables. This is very important if you want to run commands properly and not receive "command not found" error messages.

A more sophisticated way to allow elevated access without using the `su` command or having knowledge of the root account's password is to use the sudo utility, which allows you to specify which commands an ordinary user can run. This allows you to avoid disclosing the root password to allow a user to run commands normally used by root. I present a simple sudo configuration in the free PDF guide on the ESX Classic command-line available from my web site:

http://www.rtfm-ed.co.uk/docs/vmwdocs/ESX4.x-VC4.x-ServiceConsole-Guide.pdf

## Installing the vSphere Client

The main graphical tool used to administer an ESX server directly or via vCenter is the vSphere Client. You can download and install this tool from the ESX host. You can then use this tool to create additional ESX users that can log in remotely with SSH. You will need to do this if you neglected to create a user during the installation.

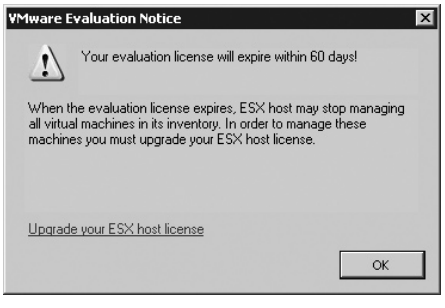
To download the vSphere Client, from your management PC, open a web browser and type in the FQDN of your ESX host (such as https://esx1.vi4book.com). Choose Yes to continue to access the ESX host, despite the fact its certificate is not trusted. (ESX creates a server certificate for each installation you do. These are autogenerated using OpenSSL and are not signed from a recognized root certificate authority.) Finally, select the link to download the vSphere Client.

Now you can run the VMware-viclient.exe file to install the client to your management PC. Select your required language. Next, enable the option to "Install vSphere Host Update Utility 4.0." This is a utility that allows you to patch your ESX hosts, even if you don't have vCenter and VMware Update Manager.

After installing the vSphere Client, you should be able to open it from the shortcut on your desktop by logging on to your ESX host by its FQDN as the root user. You'll notice in this release that VMware has added "pass-through" authentication to the vSphere; that is, if you enable the option, it will attempt to use the credentials you supplied to your management PC and pass them through to the vSphere. This is great if you're using vCenter, but it's not valid

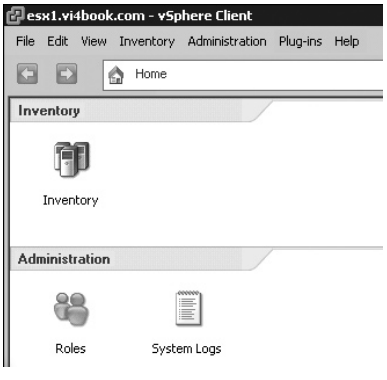
if you're logging in to the ESX host directly, where only the root account exists by default. ESX does automatically resolve your Windows users to the ESX users.


Until you license the ESX host, you will always receive a message stating how many days you remaining in the evaluation period.

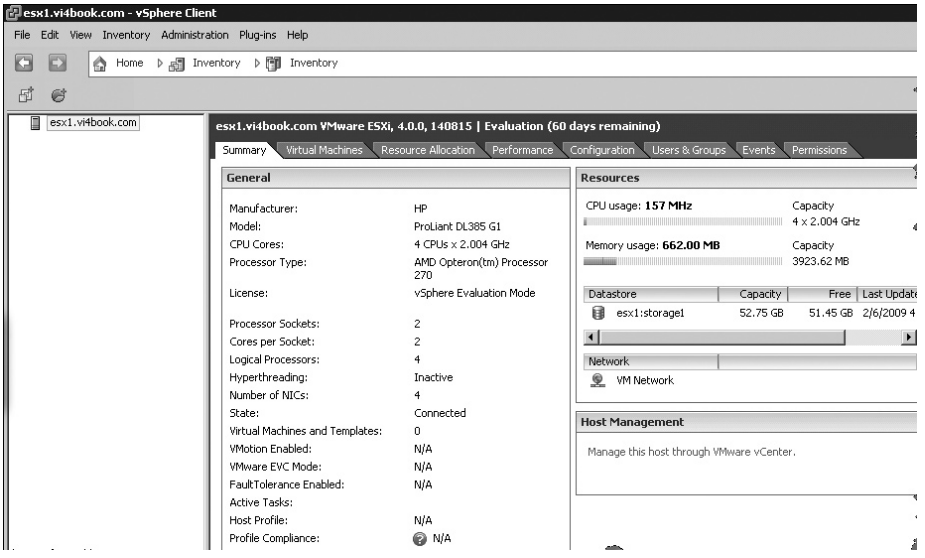



## Creating Additional Local Users on an ESX Host

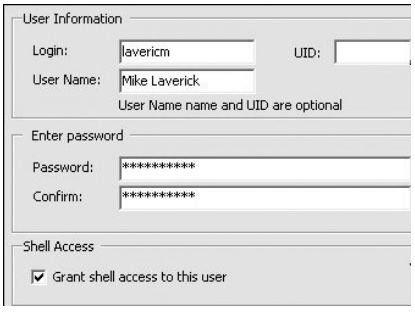
When the new vSphere Client opens, you might not immediately see your ESX host, but see the Inventory view instead.



Select the Inventory icon, and you will then see your ESX host, in a view that looks very similar to the previous version of VMware Virtual Infrastructure client.




From this view, click the User & Groups tab, right-click the window, and choose Add. Fill in the dialog box to create the new user.



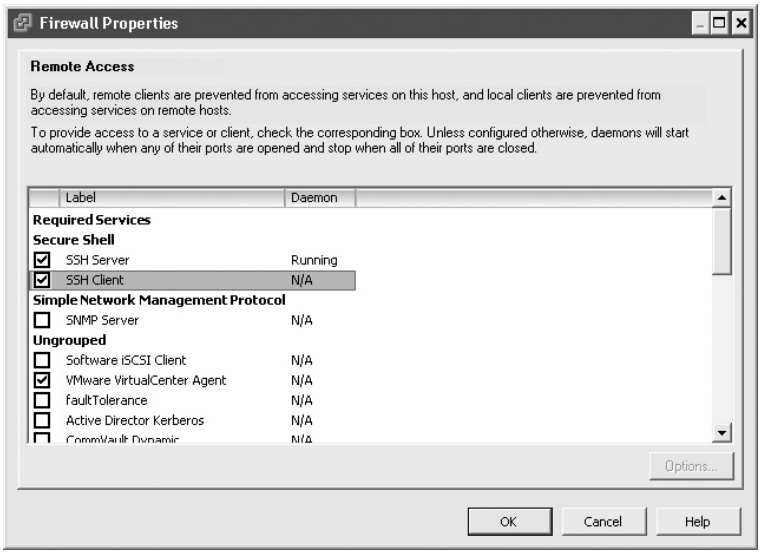

You must enable "Grant shell access" to allows this user to have access to the Service Console via SSH. Otherwise, the user will have access to the ESX host only with vSphere. (In ESXi 4, this "Grant shell access" option is not available, as technically, there is no command-line shell available.)

## Enabling SSH from Your ESX Host to Other Hosts

I like to connect to one ESX host and then use SSH from that session to get to my other servers. This saves me from needing to open repeated PuTTY sessions. However, you will be unable to use the `scp` (Secure Copy) command to copy files to ESX servers from an ESX server. Under the default settings, this is not allowed in ESX 4.*x*, as the firewall denies the client (although every ESX host is an SSH server), with an error like "ssh: connect to host esx2.vi4book.com port 22: Connection refused."

To enable this kind of access, you need to adjust the firewall settings, as follows:

1. In the vSphere Client, click the Configuration tab.
2. Under the Software panel, select Security Profile.
3. Click the Properties link in the far-right corner.
4. In the Firewall Properties dialog box, enable the SSH Client option, and then click OK.




This is all that's required for SSH. Now you could type something like `ssh -l lavericm esx2.vi4book.com` and then use `su -` to elevate your privileges to root rights, as described earlier.

If you ever get confused about who you are, try the command `whoami`. If you are unsure which host you're connected to, try the command `hostname`. You can use the command `exit` to leave SSH sessions.

If you are using the command line, you can enable the SSH client using the command-line version of the ESX firewall:

```
esxcfg-firewall -e sshClient
```

Note that this command, like all ESX CLI commands, is case sensitive.

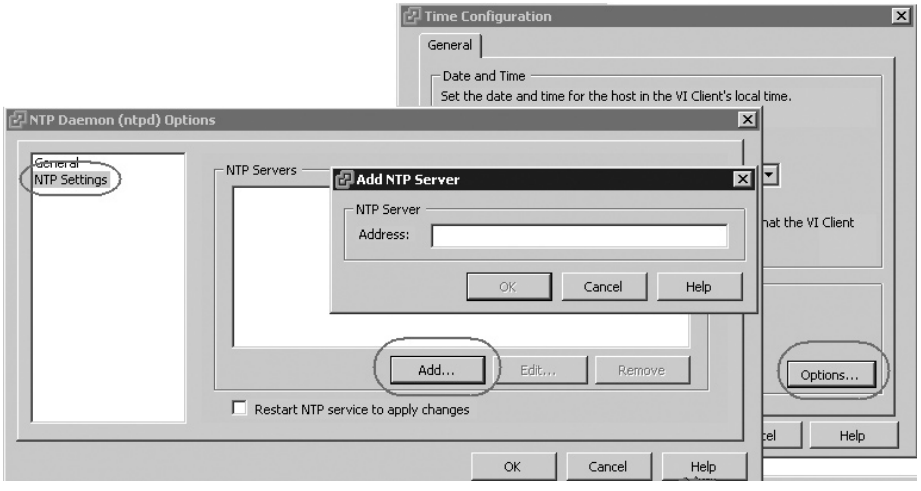
## Configuring NTP

Whether your virtual machines do or don't get their time from the ESX hosts is separate and distinct from the time configuration of the ESX host itself. Even if your Windows virtual machines still receive time updates from a physical domain controller, time still needs to be correctly configured for the ESX host. Without a correct time configuration of an ESX host,

your log files will have incorrect date and time stamps relative to actual events themselves, and performance statistics will be skewed at best; at worst, they simply not be displayed at all.

You can configure NTP from the GUI from the VSphere Client, as follows:

1. In the vSphere Client, click the Configuration tab.
2. Under the Software panel, select Time Configuration.
3. Click the Properties link in the far-right corner.
4. In the Time Configuration dialog box, click the selection box to choose NTP Client Enabled.
5. Click the Options button.
6. In the NTP Daemon (ntpd) Options dialog box, select NTP Settings in the list on the left.
7. Click the Add button to add the name or IP address of an NTP service/server. Ideally, you should be able to point to an internal NTP server, which in turn synchronizes its time with an external time source with a public NTP server.




8. Click OK to close all the dialog boxes.

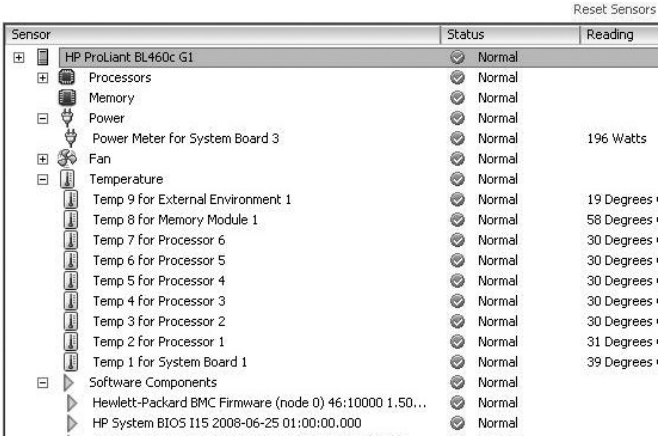
## Monitoring Hardware

Historically, we have needed to install hardware monitoring agents (commonly referred to as vendor CIM agents) to allow us to monitor the physical server in the hardware vendor's management and monitoring tools. For example, it's not unusual to install the HP Insight Management Agents to an HP ProLiant Server so we can monitor the hardware in HP management tools. We would do the same with Dell (OpenManage) and IBM (Director). Since Vi3.5, we have also been able to monitor the hardware status of an ESX host using the Health Status option.

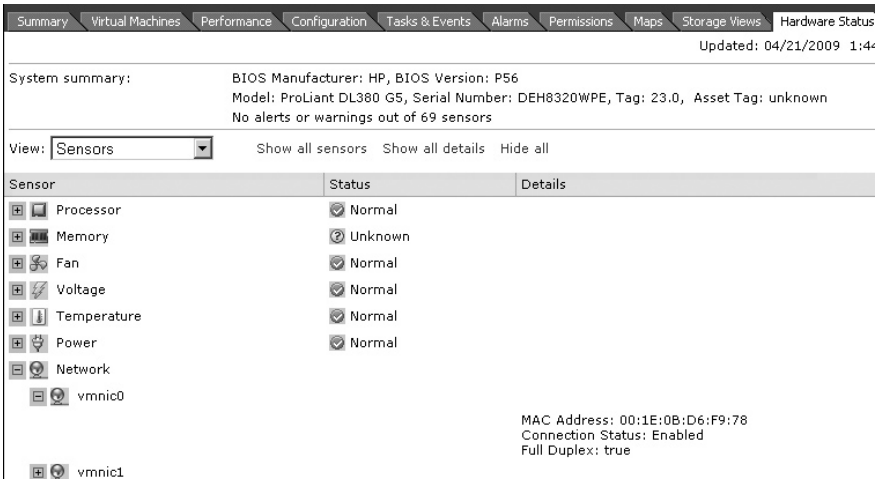
## Viewing Health Status

The Health Status feature first came into being specifically for ESXi 3, but it is also available to the ESX Classic edition as well. ESXi doesn't have the Service Console environment to allow this functionality to be installed, and so VMware partnered with the OEMs to "embed" the Common Information Model (CIM) into the ESXi product. This allows you to monitor the status of the hardware either from the vSphere Client or from your vendor's proprietary monitoring system.

To view your physical server's health status, in the vSphere Client, click the Configuration tab. Under the Hardware panel, select the Health Status option. The example shown here is taken from an HP ProLiant Blade. The health status tab only appears when you connect directly to an ESX host, and it does not appear if you manage the ESX host with vCenter. However, you do have a "Hardware Status" tab which reports very similar information.




If you are using ESX Classic configured with vCenter, you will see a Hardware Status page, which you will find on each ESX host. This can offer some very detailed information about your hardware.



### Installing Hardware Management Agents

Despite the existence of the Health Status feature in the ESX product, you might still wish to install the full hardware management agent to your ESX hosts. One reason is just to reduce their noise—most servers will run at their maximum fan speeds until you install the management agent. Perhaps that is deliberate, because you will certainly know if you haven't installed the agent!

As an example, I've chosen to document the installation of the HP Insight Management Agent for ESX 4. The procedures for other vendors' agents will vary.

I've often found locating the correct version of the hardware agent tricky. Start with a search on something like, "HP Management Agents for VMware ESX Server 4.x." At the time of writing, the latest HP Insight Management Agent isn't available on the main hp.com domain, but on the ftp site instead at this location:

ftp://ftp.hp.com/pub/softlib2/software1/pubsw-linux/p332370759/v49789/

Download the agent, and upload it to the ESX Classic host using the DataStore Browser.

Log in to the Service Console as root, and extract the download package with the `tar` command:

```
tar -zxvf hpmgmt-N.N.N-vmware4x.tgz
```

Change directory into the hpmgmt directory created via the extraction process. Execute the installer. With the HP Insight Management Agent, this is generally called installvmNNN.sh:

```
./installvmNNNvibs.sh
```

Most installers of this type can run silently with the script. This usually requires a couple of switches on the .sh file, together with some kind of answer or unattended text file. HP has a sample unattended file in its agent called hpmgmt.conf.example, which could be run as follows:

```
installvmNNNvibs.sh --silent --inputfile hpmgmt.conf.example
```

The HP installer engine checks the version of ESX you are running and will alert you if you are installing a version of the agent that is incompatible with the version of ESX you are using. At the end of the installation, you will be asked to reboot the ESX host.

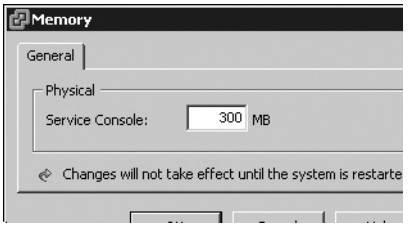
## Increasing Memory Allocation to the Service Console

The Service Console in ESX Classic is given an allocation of resources in precisely the same way as a virtual machine. More specifically, the Service Console is given an allocation of memory. In ESX 3, this was 272MB of RAM. In ESX 4, it is given 300MB of RAM.

Some environments may use a more customized ESX classic installation, which includes additional agents and daemons to the base installation. These additional agents and daemons may consume more memory than was originally expected by VMware. If necessary, you can increase the amount of memory allocated to the Service Console as follows:

1. Select the ESX host, and then click the Configuration tab.
2. In the Hardware pane, select the Memory link.

3. On the far-right side of the vSphere Client window, select Properties.
4. In the dialog box, type in the new amount of memory.




The option to increase the memory allocated to the Service Console appears only in ESX Classic, and is not available or relevant to the ESXi 4 platform.

As the dialog box clearly states, you will need to reboot the ESX host for this change to take effect. For this reason, if you do think you will need to increase the amount of memory to the Service Console, then you may wish to factor in this configuration change during the roll-out phase.

## Configuring ESX4 Classic for SAN Booting

If you are unfamiliar with SAN technology, you might wish to read Chapter 6, which covers storage, and then return to this section. It's also worth noting that with the advent of ESXi 3 and 4, the attractiveness of the configuration for SAN booting might decline. After all, if you want a diskless ESX host, you can use the embedded version of ESX. However, if your organization wants to keep on using the ESX Classic edition and has a diskless server, you can follow the instructions in this section.

Since ESX 2.*x*, VMware has supported booting from both local storage and from SAN-based storage. SAN-based booting is a tempting option, especially if you run ESX servers from blades. It allows you to quickly replace a failed blade with a working one, and bring that ESX host back online quickly. Another advantage is that you can leverage your SAN snapshot features as a way of backing up your ESX server build. One of the appeals of booting from a SAN is that nothing about the ESX host itself is physically stored locally. This makes the ESX product more like an appliance—merely an engine for running virtual machines. But as I've just noted, that same goal can be achieved with ESXi 4, and with a much simpler configuration.

### SAN-Based Booting Restrictions and Requirements

Before preparing ESX for SAN-based booting, it's important to know some key restrictions and what is actually supported by VMware:

- In previous releases booting from an SAN and using virtual machine clustering (clustering software from Microsoft or Veritas, for example) was not supported. If you want to set up clustering software, you needed local storage on the server. This restriction has now been removed and VMware does now support virtual machine clustering with the boot disks of the virtual machines stored on the SAN.

- The ESX server can see up to a maximum of 256 LUNs (0–255). (Previous versions of the ESX installer displayed only the first 128 LUNs; this limitation no longer exists.)

With ESX 2.x, you could not use a feature called Raw Device Mappings (RDMs). RDMs allow a virtual machine access to native LUNs on the SAN for existing data or storing your data using the guest operating system's native files system on an SAN LUN, rather than creating a virtual disk on VMFS partition. RDM files and SAN booting are now supported together. Also, in ESX 2, VMware supported booting only for the "lowest order" LUNs, so if an ESX host had access to LUNs 10, 12, and 15, the LUN selected as the boot LUN would be 10. This restriction was ended with the release of ESX 3.

Additionally, there are a number of physical hardware requirements for a SAN-based booting configuration:

- The Fibre Channel card used should sit highest in the PCI bus for performance reasons and should be plugged into an "active" storage processor on the SAN.
- Correct LUN masking and zoning should be implemented so only the ESX host can see the correct LUN. This has some implications when you replace a blade in the enclosure. The new blade might present a different World Wide Name (WWN) value, unless you have moved up to the modern blades, for which this issue has been resolved by virtualizing the HBA connection. The SAN will need reconfiguring to present this LUN to the new ESX host.
- Booting from the SAN is supported only in conjunction with a Fibre Channel switch. Direction conenction without a fibre-channel switch may not be supported with all storage vendors. I would consult the VMware SAN Configuration Guide and the VMware HCL if you are considering configuring this setup to confirm it is properly supported.
- IBM eServer BladeCenters that ship with IDE drives onboard need these drives disabled on each blade.
- If you intend to configure booting from a SAN using iSCSI, this is supported only with hardware iSCSI HBAs from vendors like QLogic and Emulex.

## Preparing for Installing a SAN-Based Booting Configuration

Before beginning the installation, I recommend that you set the main system BIOS to boot from the CD-ROM. In the main BIOS, you set the Fibre Channel storage controller to be above the internal RAID controller card. This stops the server from booting to local internal storage.

Once the main BIOS has been correctly configured, you will need to enter the configuration tool for your Fibre Channel card. The Fibre Channel device will need to be enabled as a boot device, and the LUN where installation files were copied selected as the boot LUN. Clearly, these settings are very hardware-specific. The instructions in this section are based on using HP ProLiant DL 385 with a QLogic 2340 Fibre Channel card. You should consult your vendor's procedures for enabling SAN-based booting on your specific hardware.

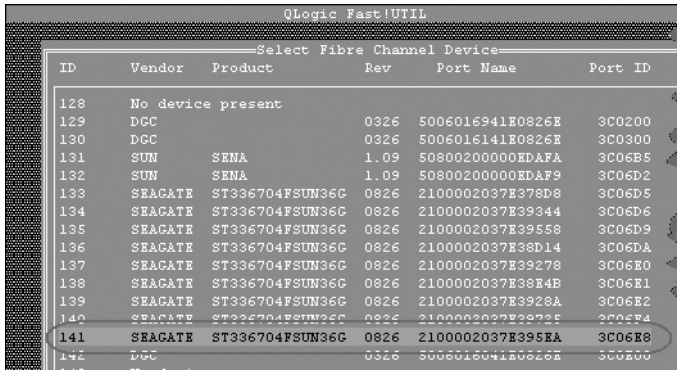
Finally, before you begin, I recommend either removing physical disks or using your RAID controller card to remove any LUNs. This will ensure that the only storage you see during the installation procedure is SAN-based.

## Configuring the Fibre Channel BIOS Options

Here are the steps for configuring the BIOS options:

1. Power on your server.
2. Press ALT+Q to enter the QLogic card's BIOS settings.
3. Choose Configuration Settings and press ENTER.
4. Choose Host Adapter Settings and press ENTER.
5. Set the Host Adapter BIOS option to Enabled.
6. Press ESC to exit the Host Adapter Settings menu.
7. In Configuration Settings, choose Selectable Boot Settings.
8. Set Selectable Boot Device option to Enabled.
9. Cursor down to select Current Boot Node Name and press ENTER.

The following diagram shows the disks displayed to my server.



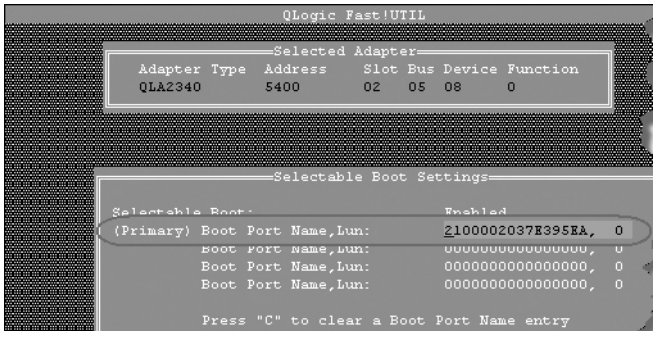

Here, my ESX host can see many LUNs/disks. You might find it much *safer and easier* to mask all LUNs away from the server except the SAN boot LUN. This will make selecting the correct LUN much clearer.

1. Select the appropriate LUN or disk from the list. I will use ID1, which has the NAA name that ends with 20nnnnnnnnnn95EA.

---

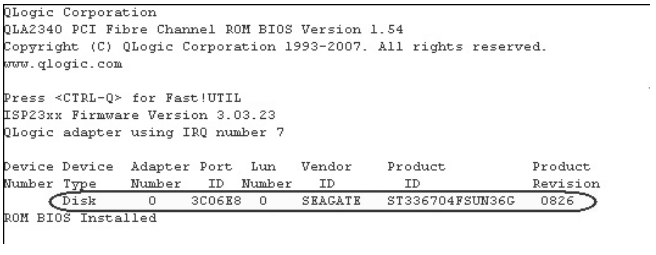
**NOTE** *A Network Address Authority (NAA) name is a special address that uniquely identifies a disk or a LUN in Fibre Channel- or iSCSI-based SANs. VMware still refers to disks by the vmhba syntax in the vSphere Client, but also shows the NAA values. You will learn more about these addressing schemes in Chapter 6.*

2. Press ENTER to select the target LUN, which will then be set as the Primary Boot Port Name and LUN.




3. Press ESC and choose to save your changes.
4. Finally, exit the utility and choose to reboot the system.

When you reboot, you will see that the QLogic card reports different settings, indicating that the BIOS is enabled and that the target LUN is selected.

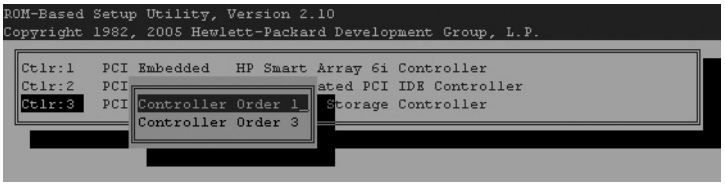



## Configuring the Main System BIOS

Different systems use different BIOS providers and different keyboard strokes to gain access to the main keyboard settings. For example, on most HP systems, you press F10 to enter the BIOS, but on Dell systems, it is normally F2.

The following is the procedure for configuring the BIOS on a HP ProLiant Server:

1. Press F10 at the prompt.
2. Choose the Setup Utility option and press ENTER.
3. Choose Boot Controller Order (IPL) and press ENTER.
4. Select the QLogic card and press ENTER. (On my system, this appears as Ctlr:3 PCI Slot 3 SCSI Mass Storage Controller.)
5. Choose Control Order 3. The following screenshot shows configuring the QLogic card to be the primary device for boot purposes.



***NOTE*** *If you were doing this in a Dell BIOS, you would choose the option called Hard Drive Sequence and use the +/– keys to move the QLogic card to the top of the list.*

6. Press ESC and then F10 to save your changes.

**Boot Order**

Normally, I have a personal preference for the boot order on my ESX hosts, which is as follows:

- Hard drive
- CD-ROM
- Floppy
- USB key drive
- PXE

This is to prevent me from accidentally leaving a CD, floppy disk, or USB key in a drive and finding a reboot boots to removable media. You might notice that the ESX CD does not ask you to "Press any key to boot from this CD," as some operating systems do.

When it comes to booting from SAN setups, I change this order to the following:

- CD-ROM
- Hard drive
- Floppy
- USB key drive
- PXE

This is to ensure that when I reboot the server, I get to the CD the first time, and the system doesn't try to boot from a blank LUN before I have had the opportunity to install ESX there.

## Installing ESX to a SAN LUN

Generally, the ESX installation procedure for a boot-from-SAN configuration differs little from an installation to local storage. The only real difference is that rather than selecting local storage when you partition the disk or set the location of the MBR, you select a LUN on the SAN.

To install ESX to a SAN LUN, insert the ESX CD, and then press the ENTER key to enter the graphical installation. During the installation, choose the correct LUN from the SAN when you partition the LUN, and remember to put the MBR on the same LUN.

## Summary

By now, I hope you are fully *au fait* with the ESX installation procedure. In time, you will probably do a few ESX installations, and all of this will become second nature to you. Eventually, you will get truly bored with the idea of performing manual installations from a CD-ROM. Manual installations take time and address only a small part of the overall configuration of the ESX host for use in a live environment. You'll learn more about that configuration in Chapters 4 and 6, which cover networking and storage.

You'll be pleased to hear that much of this configuration can be automated with deployment tools, some of which are free. However, before you consider automating the process, I advise reading the chapters that cover networking and storage if you are new to the product, so you are completely familiar with the postconfiguration of an ESX host. Then, if you're interested in this subject, you may wish to proceed to Chapter 17 entitled "Advanced Configuration Tools," which covers scripted installations with "kickstart" scripts, with the source code delivered across the network with a free PXE "virtual appliance."