# Chapter 5

# Virtual Networking

## Solutions in this chapter:

- Virtual Switches
- Physical and Virtual NICs
- MAC Addresses
- Port Groups and VLANs
- Networking Tools

- ☑ Summary
- ☑ Solutions Fast Track
- ☑ Frequently Asked Questions

# Introduction

This chapter expands on the basics of virtual networking that were introduced in Chapters 1 and 2. We'll discuss virtual switches, physical and virtual NICs, bonded physical NICs, MAC addressing, Port Groups and VLANs, networking tools, and firewall possibilities. We'll go beyond the basics to discuss advanced network configurations and possible solutions. ESX provides remarkable flexibility when it comes to networking. This chapter is essential for the beginner because it offers the basic knowledge needed to set up ESX Server and your virtual machines. We'll build upon the basics to more advanced topics that both the beginner and possibly the advanced ESX administrator will find useful. Before moving on to the next chapter, though, we recommend you fully understand all of the information in this chapter because it will be referenced in later chapters, thus testing your understanding of the pertinent information detailed here.

After you've installed ESX Server on your Host, one of the first things you'll need to do when you log into the MUI is set up a virtual switch. You can think of a virtual switch as a software hub controlled by ESX that routes the traffic of your virtual machines both internally, between virtual machines on the same physical host, as well as externally, to your production network or the Internet.

Many possible networking configurations can enhance network communication, provide redundancy, and increase security for your production of virtual machines.

# Virtual Switches

Like a physical switch or hub, a virtual switch has a certain number of ports into which you can plug the virtual NIC of your virtual machines. A virtual switch has 32 logical ports; thus, it can support a maximum of 32 virtual machines per switch.

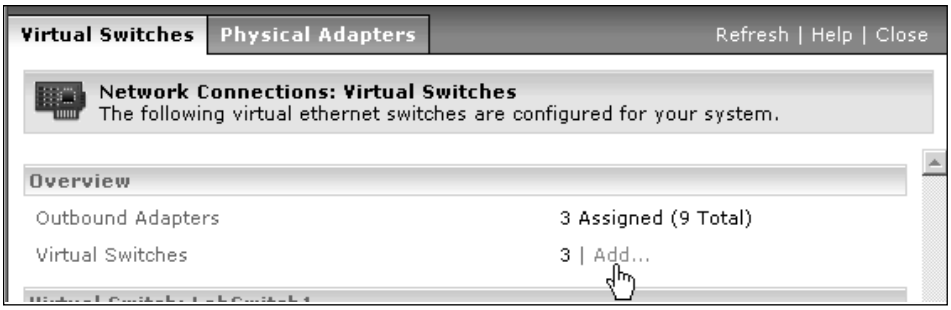
## VMnets

You may set up a virtual switch called a VMnet exclusively for routing high-speed network traffic between virtual machines on the same ESX Server (see Figure 5.1). VMnets can provide added security and isolation of network traffic and provide a more flexible and cost-effective network topology. VMnets can be used for advanced configurations such as parallelization (NLB) or redundant (MSCS) cluster solutions, as well as interesting firewall possibilities. Creating a VMnet is quite simple. Just follow these steps:

1. Open the MUI.
2. Click the **Options** tab.

3. Click **Network Connections…**
4. In the **Overview** section, click **Add**.

**Figure 5.1** Configuring a VMnet




5. In the **Network Label** property (see the details later in this section), type in a label such as VMnet0 or something more descriptive (perhaps heartbeat if this VMnet will be used for the heartbeat traffic of a cluster).
6. Do not check any of the network adapters to bind to this virtual switch. Click **Create Switch**.
7. Find the newly created virtual switch. It should look something like the image in Figure 5.2.

**Figure 5.2** A Virtual Switch


8. Notice the text: **No outbound adapters. Traffic will be routed locally**. Virtual machines connecting to this VMnet virtual switch will be able to route network traffic only within the ESX Server itself to those virtual machines that connect to it.

If you were to look at the configuration file of a virtual machine that was connected to a VMnet, the Ethernet section would look like the screenshot in Figure 5.3.

**Figure 5.3** The Configuration File Information of a Virtual Machine

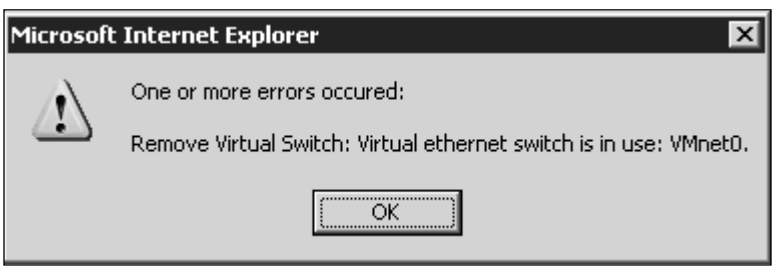
```
Ethernet1.present = "TRUE"
Ethernet1.connectionType = "monitor_dev"
Ethernet1.virtualDev = "vmxnet"
Ethernet1.devName = "vmnet_0"
Ethernet1.networkName = "VMnet0"
~
```

Notice in Figure 5.3 Ethernet1.devName ="vmnet_0," which is the VMnet. Since a VMnet is a virtual switch, it can be attached to by 32 guests on the same physical host for fast network traffic routing.

**NOTE**

If you attempt to delete a VMnet virtual switch that has virtual machines associated to it, you will receive an error message like the one in Figure 5.4.

**Figure 5.4** Attempting to Delete a VMnet Virtual Switch That Has Virtual Machines Bound to It




# VMnics

For external access (to your production network, other virtual machines or the Internet), your virtual switch can be bound to one or more of the physical NICs on your Host. You can bind multiple physical NICs to a virtual switch. for redundancy or load balancing purposes. Redundant physical NICs can be used in case one of the physical NICs fails or loses network connectivity. These physical NICs can also be combined, creating a bond, for load balancing or fault tolerance between the physical NICs.

Each virtual machine's virtual NIC logically plugs into a port in the virtual switch. Traffic from your virtual machine is passed out the physical NIC(s) that are associated with the virtual switch. See Figure 5.5.

**Figure 5.5** Passing Traffic through the Physical NICs on a Virtual Machine

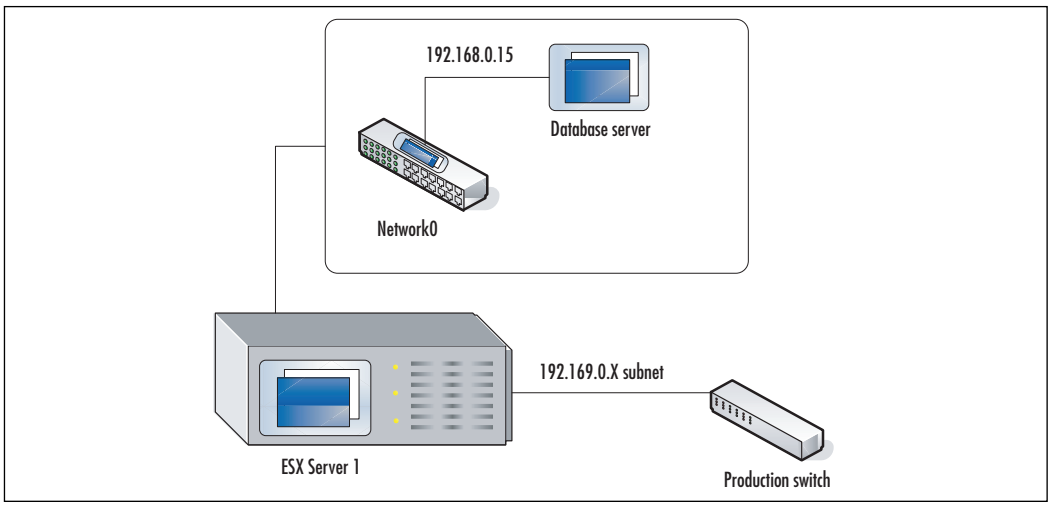



Figure 5.5 depicts a simple virtual switch configuration. ESX Server 1 has one virtual machine called Database Server with an IP address of 192.168.0.15. The Database Server virtual machine's NIC is plugged into the virtual switch labeled Network0, which is bound to the ESX Server's physical NIC. ESX Server 1's physical NIC is plugged into the Production Switch, which has access to the 192.168.0.x subnet. This is the simplest of configurations, but understanding the concepts presented in the figure is essential.

## Network Labels

In ESX Server, you can label your virtual switches. This label is important because it can serve as a "functional descriptor" for the switch. What this means is that you can label your virtual switch something like 192.168.0.x to indicate that all connections to this switch will be on this subnet. You can also label the virtual switch something like ESX Internal to represent that this particular switch is only for ESX internal networking between virtual machines (a VMnet) with no physical NICs of your ESX Server bound to it. Figure 5.6 depicts an ESX Server with two virtual switches.

**Figure 5.6** An ESX Server with Two Virtual Switches

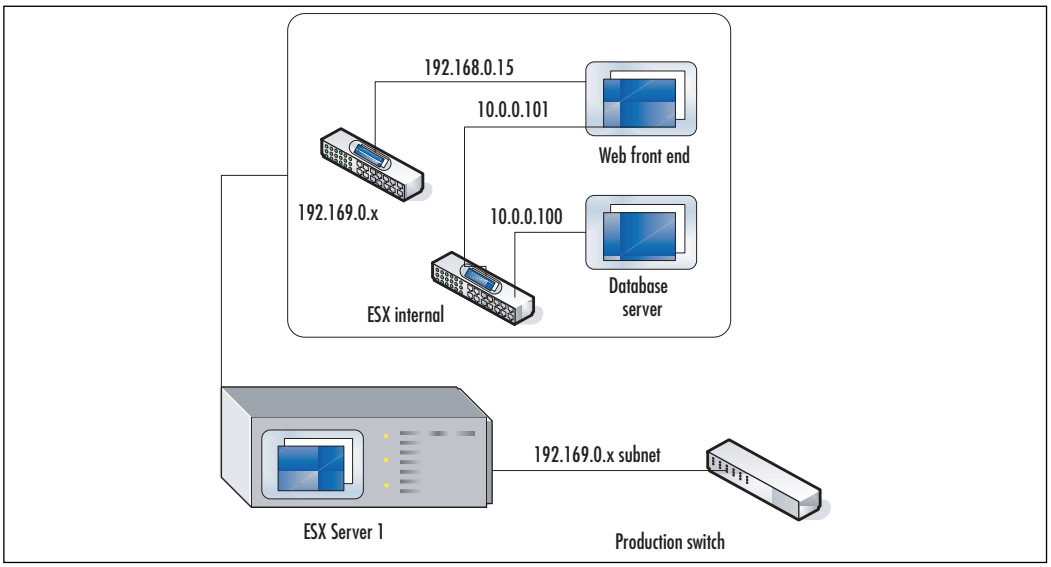



Figure 5.6 shows an ESX Server with two virtual switches and two virtual machines. One virtual switch is labeled 192.168.0.x and is bound to the ESX Server's physical NIC. The other virtual switch, a VMnet, is labeled ESX Internal. This VMnet virtual switch is not bound to any physical NIC and thus provides high-speed routing of network traffic only within the ESX Server itself. The virtual machine called Web Front End has two virtual NICs: one going into the production network via the virtual switch labeled 192.169.0.x and another virtual NIC talking to the Database Server via the ESX Internal VMnet virtual switch.

## NOTE

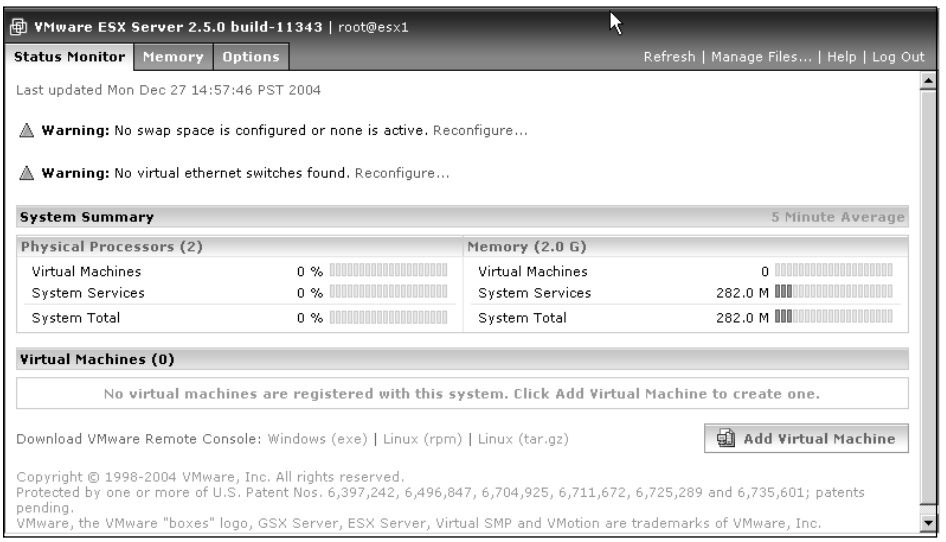
It is important to note that you can only change the label of a virtual switch if no virtual machines are associated with it or the virtual machines associated with it are turned off.

# Creating Virtual Switches

It is a fairly easy task to create a virtual switch. Just follow these steps.

1. Log in to the MUI as root. If it is your first time logging in to the MUI since you loaded ESX, you'll see a window that looks like the one in Figure 5.7. You will immediately notice that virtual Ethernet switches have been configured.

**Figure 5.7** Logging In to the MUI

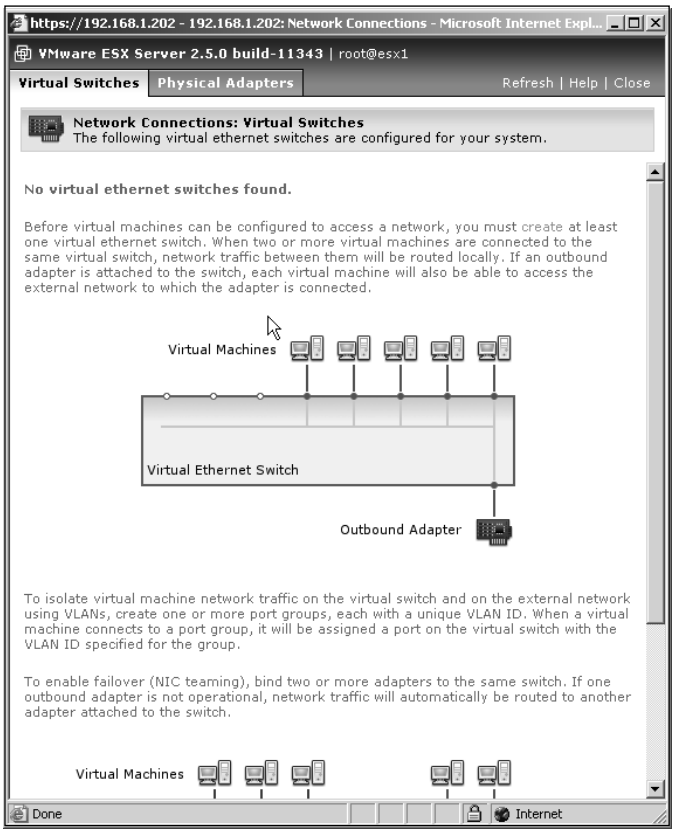



2. Click the **Reconfigure** link. You'll see a window like the one in Figure 5.8.

**Figure 5.8** Adding a Virtual Switch

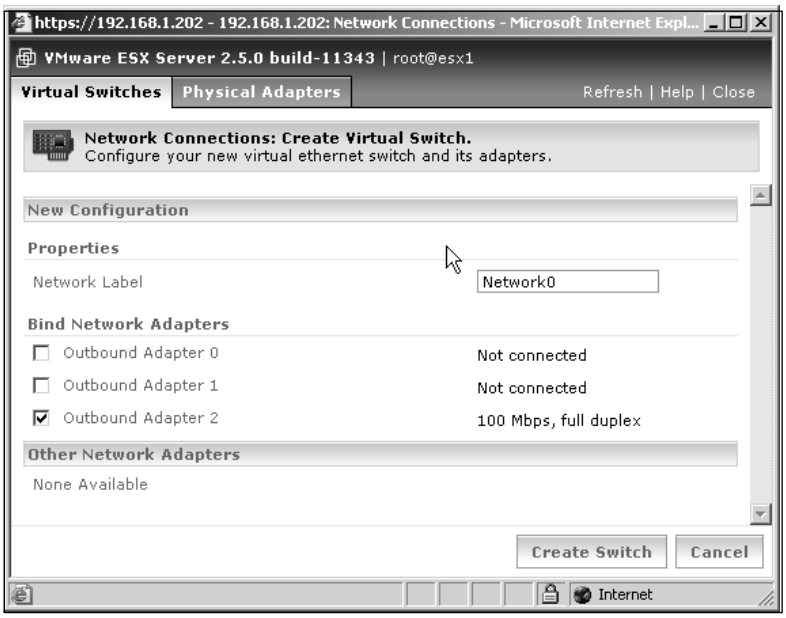
3. A window like the one in Figure 5.9 appears.

**Figure 5.9** Virtual Switch Window



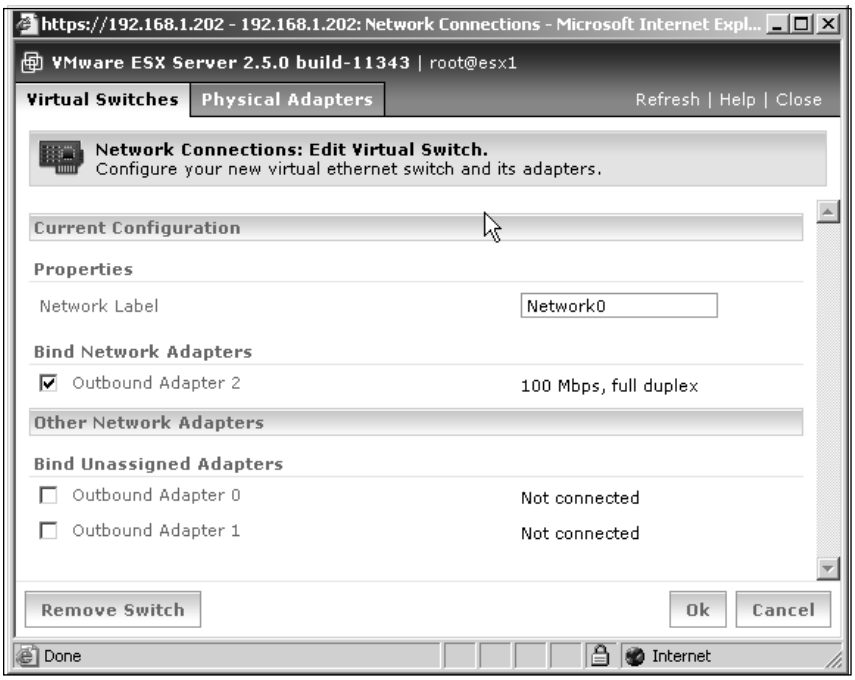

4. Read the contents of this window. It introduces virtual switches, NIC teaming, and port groups (all of which are discussed in greater detail in this chapter).
5. Click the **Create** link in the top paragraph. This takes you to the Create Virtual switch window (see Figure 5.10).

**Figure 5.10** Creating a Virtual Switch




6. By default, the first Network Label is Network0. You can leave this or give it a better functional description, such as a subnet it represents or whether it's an internal or external connection.
7. Under the Bind Network Adapters heading, check which physical adapters you want to bind to this virtual switch.
8. Click **Create Switch**. The window shown in Figure 5.11 appears.

**Figure 5.11** Virtual Switch Added




This is the Edit page for Virtual switch Network0. It is here where you can add or remove physical NICs from the virtual switch. Remember that if a virtual switch has VMs running and plugged into it, you won't be able to edit these settings until the VMs are shut down.

# Physical and Virtual NICS

Within a virtual environment physical NICs belong to your ESX server and can be bound to virtual switches or the service console. There are also virtual NICs which attaché virtual machines to virtual switches.

## Physical NICs

Your ESX Server should have a minimum of two physical NICs. During the ESX Server installation process, one of the physical NICs will be allocated exclusively to the ESX console. It is through this ESX Server Service Console NIC that all connections to the MUI are made, as well as SCP, SSH, or any other tool to access the ESX Server's file system. The other physical NIC will be dedicated to your virtual machines. Figure 5.12 depicts this simple configuration.

**Figure 5.12** The Configuration of an ESX Server with Two NICs

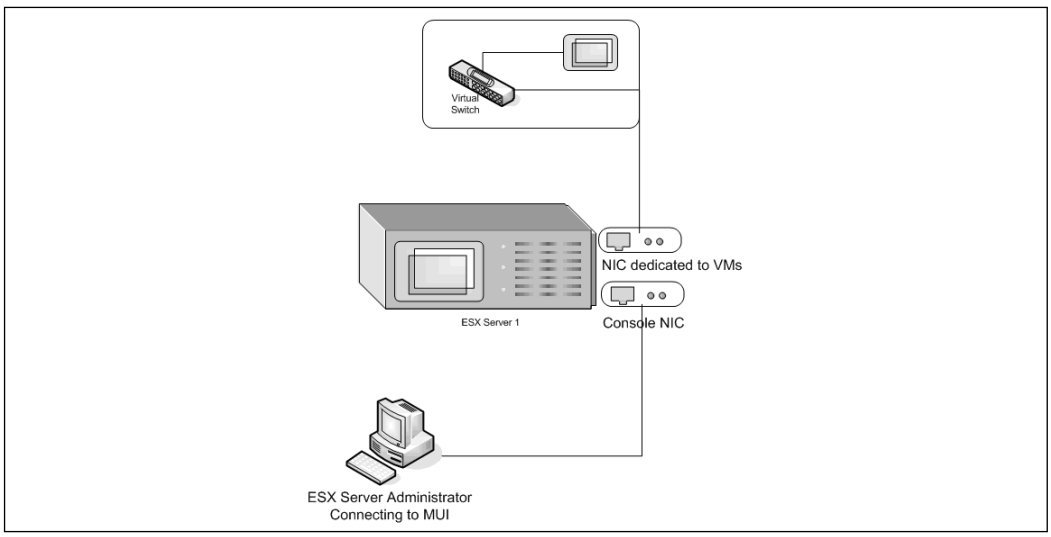



Figure 5.12 depicts an ESX server with two NICs: one dedicated to the ESX Server's Service Console and the other dedicated to the virtual machines on the ESX Server itself. How ESX Server sees and names these NICs from the operating system perspective is important to understand.

When you install ESX Server, the first NIC in the first PCI bus/slot that ESX detects will be the one it assigns to the ESX console. Physical NICs that are not allocated to the Service Console but are for use by the VMkernel and the virtual machines are labeled *vmnic0* and incremented from there as *vmnic1*, *vmnic2*, and so on. The name of the NIC will be derived from the PCI bus/slot location. If your ESX Server has different physical NICs, say Broadcom and Intel, the name of your physical NICs will depend on which NIC driver is loaded first. If you have a multi-function card, say a dual- or quad-port NIC, then the function is also numbered. For instance:

1 one port NIC might be named *bus1.slot.1 function1*

1 dual port NIC might be enumerated with the following names:

*bus1.slot1.function0*

*bus1.slot1.function1*

The multifunction card has the same bus and slot number and the function descriptor is incremented. To check out the naming of one of your physical NICs,

go to the command line of your ESX Server, and type in the following: **cat/etc/vmware/devnames.conf**.

The output may look something like Figure 5.13.

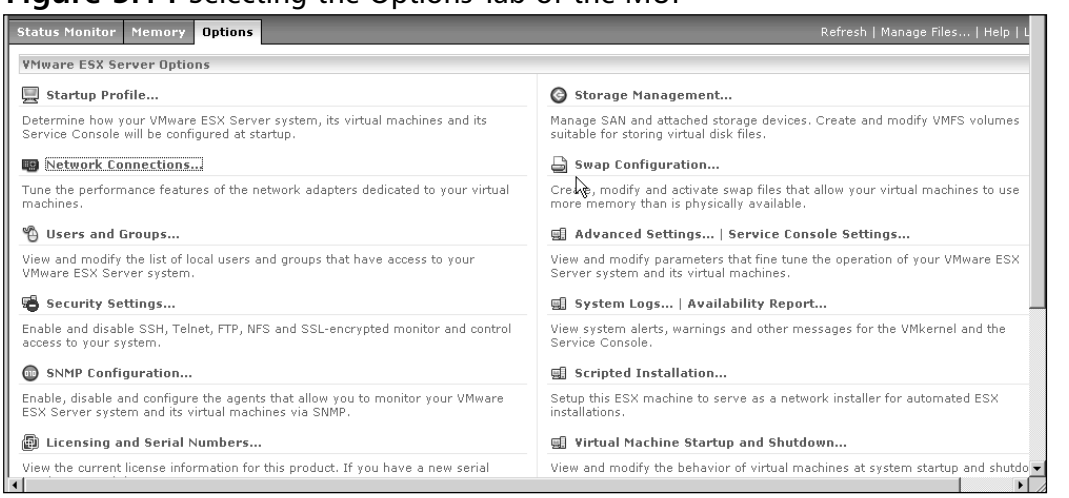
**Figure 5.13** Names of the Physical NICs

```
007:04.1        nic     vmnic0
007:06.0        nic     vmnic1
007:06.1        nic     vmnic2
010:01.0        nic     vmnic3
010:01.1        nic     vmnic4
011:04.0        nic     vmnic5
011:04.1        nic     vmnic6
011:06.0        nic     vmnic7
```

Listed in this file are the names of your physical NICs. In this example, the ESX Server has eight physical NICs.

Besides listing the names of your physical NICs, you can also adjust the speed at which they're set. To do this, log in to the MUI. Go to the Options tab and then double-click the Network Connections link (see Figure 5.14).

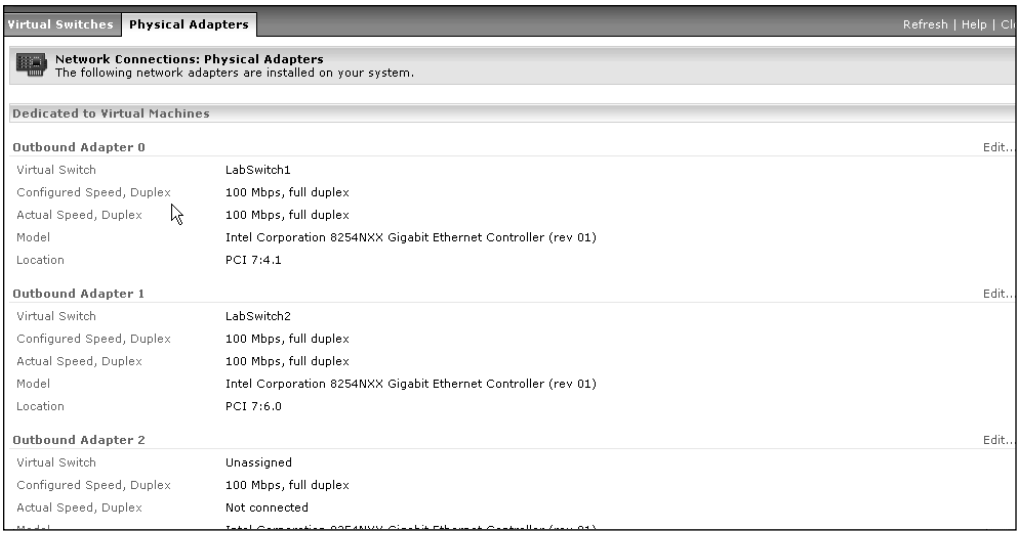
**Figure 5.14** Selecting the Options Tab of the MUI




Once the Network Connections page loads, click the **Physical Adapters** tab. You'll notice on the left of this window that the physical adapters are labeled Outbound Adapter 0, Outbound Adapter 1, and so on. Outbound Adapter 0 equals vmnic0, which equals 007:04.1. It can be a bit confusing.

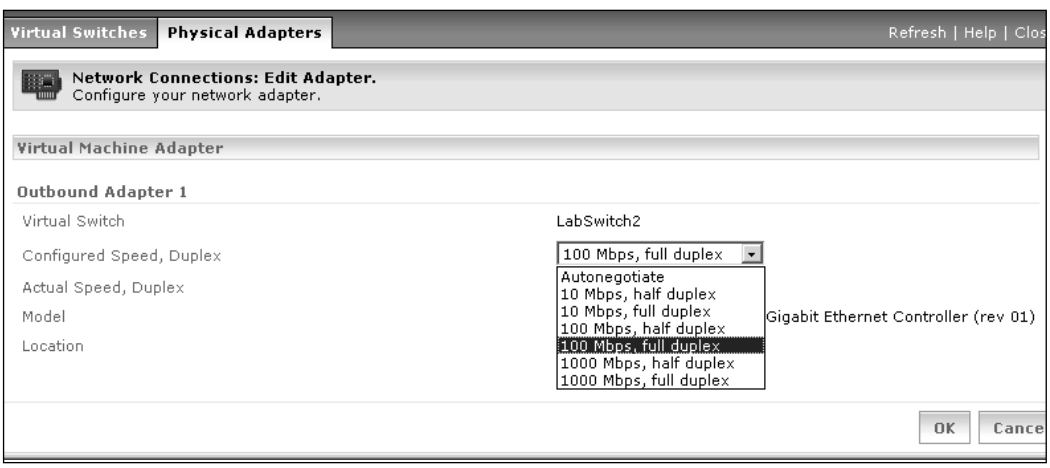
On the far right of the window, click **Edit** (see Figure 5.15).

**Figure 5.15** Loading the Network Connections Page




Once you click Edit on one of your physical adapters, the properties of that NIC appear in a new window. From here, you can change the speed of your NIC (see Figure 5.16).

**Figure 5.16** Changing the Speed of a NIC




Once you set the NIC to its desired speed, click OK to save the changes. According to VMware documentation, a reboot must occur for this change to take effect.

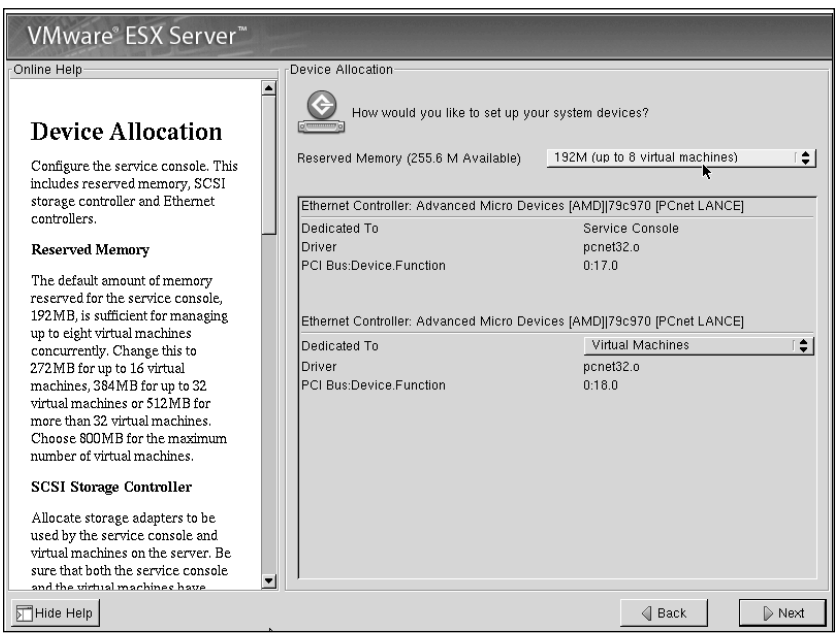
**NOTE**

It's always a best practice to hard-set the speed of your physical NICs. Make sure they're plugged into a managed switch and that both the switch port and the physical NIC are set to the same speed.

# The Service Console NIC

The service console NIC is the first NIC that ESX detects during the installation process. The detection process occurs initially during installation. You'll see a window similar to the one in Figure 5.17.

**Figure 5.17** Device Allocation




From this figure, you can see that the installation process detected two NICs, both AMD 79c970s. It assigned the PCI Bus:Device.Function 0:17.0 to the Service Console, which is the first NIC it found. It's highly recommended by the authors to always, we repeat always, allocate at least one physical NIC exclusively to the Service Console. It's through this NIC that you'll access the MUI, backup agents should run, and Virtual Center, if you use it (again, highly recommended), can manage your vir-

tual machines. There are ways to share NICs between the Service Console and your virtual machines, but we don't recommend it.

**NOTE**

ESX Server 2.5 supports up to 8-Gigabit Ethernet or 16 10/100 adapters.

At the time of this writing you could find out which NICs are supported at **www.vmware.com/pdf/esx_io_guide.pdf**.

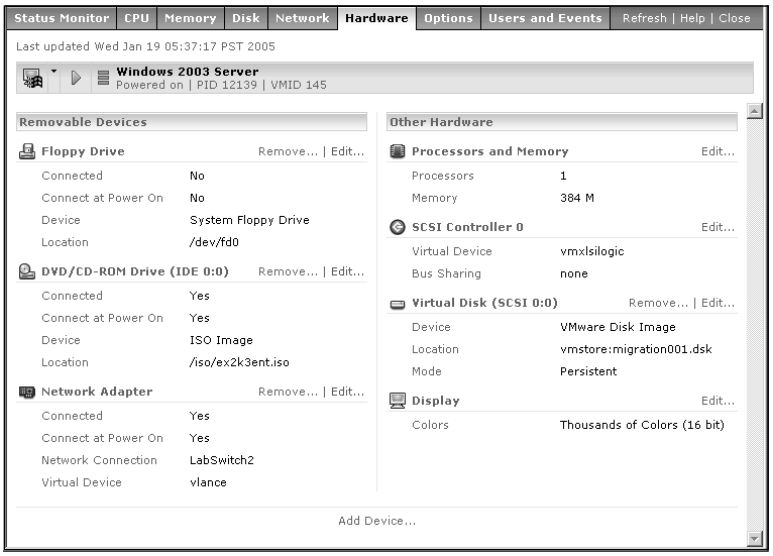
## Virtual NICs

Virtual NICs are what you assign to your virtual machines. Each virtual machine can have four virtual NICs. Like any physical NIC, a virtual NIC can have one or more IP addresses and a MAC address. Virtual NICs are assigned. Two NIC drivers are available:

- vlance
- vmxnet

vlance virtual NICs are what are given to your virtual machines by default when you create the configuration file (see Chapter 3). According to VMware documentation, these NICs require more processor overhead and provide less bandwidth. The vlance virtual NIC runs and its driver loaded by default. Figure 5.18 depicts a default configuration of a Windows 2003 virtual machine as you would see it in the MUI.

**Figure 5.18** A Default Configuration of a Windows 2003 Virtual Machine




Note the Network Adapter at the bottom left of Figure 5.18. Starting from the top, you can see that the virtual NIC is Connected and is set to Connect At Power On. Next, the Network Connection that the virtual NIC is plugged into is the virtual switch labeled LabSwitch2. Finally, the virtual device (virtual NIC) driver is the vlance driver.

The same setup would look like this if you were to open the configuration file for this virtual machine (see Figure 5.19).

**Figure 5.19** The Configuration File for a Virtual Server

```
#!/usr/bin/vmware
config.version = "6"
virtualHW.version = "3"
scsi0.present = "TRUE"
scsi0.virtualDev = "vmxlsilogic"
memsize = "384"
ide0:0.present = "TRUE"
ide0:0.fileName = "/iso/ex2k3ent.iso"
ide0:0.deviceType = "cdrom-image"
floppy0.startConnected = "FALSE"
floppy0.fileName = "/dev/fd0"
Ethernet0.present = "TRUE"
Ethernet0.connectionType = "monitor_dev"
Ethernet0.devName = "vmnic1"
Ethernet0.networkName = "LabSwitch2"
usb.present = "FALSE"
RemoteDisplay.depth = "16"
displayName = "Windows 2003 Server"
guestOS = "winNetStandard"
checkpoint.cptConfigName = "winNetStandard-775a212c"
priority.grabbed = "normal"
priority.ungrabbed = "normal"

scsi0:0.present = "TRUE"
scsi0:0.name = "vmstore:WindowsServer.dsk"
draw = "gdi"

Ethernet0.addressType = "generated"
uuid.location = "56 4d 15 db 56 86 7d b6-6b 85 26 4a 5b e7 11 d1"
uuid.bios = "56 4d 15 db 56 86 7d b6-6b 85 26 4a 5b e7 11 d1"
ethernet0.generatedAddress = "00:0c:29:e7:11:d1"
ethernet0.generatedAddressOffset = "0"
tools.syncTime = "FALSE"
```

Figure 5.19 shows what the same virtual server's configuration file would look like. Becoming familiar with the configuration files of your virtual machines will benefit you greatly in understanding and being able to troubleshoot virtual machine issues as well as create advanced virtual machine configurations.

The other virtual NIC driver is the vmxnet. The vmxnet virtual NIC is the more advanced network driver and is reported to have better bandwidth than the vlance, especially if they are used in conjunction with virtual switches bound to Gigabit Ethernet physical NIC (see Figure 5.20).

**Figure 5.20** The vmxnet Virtual NIC Driver

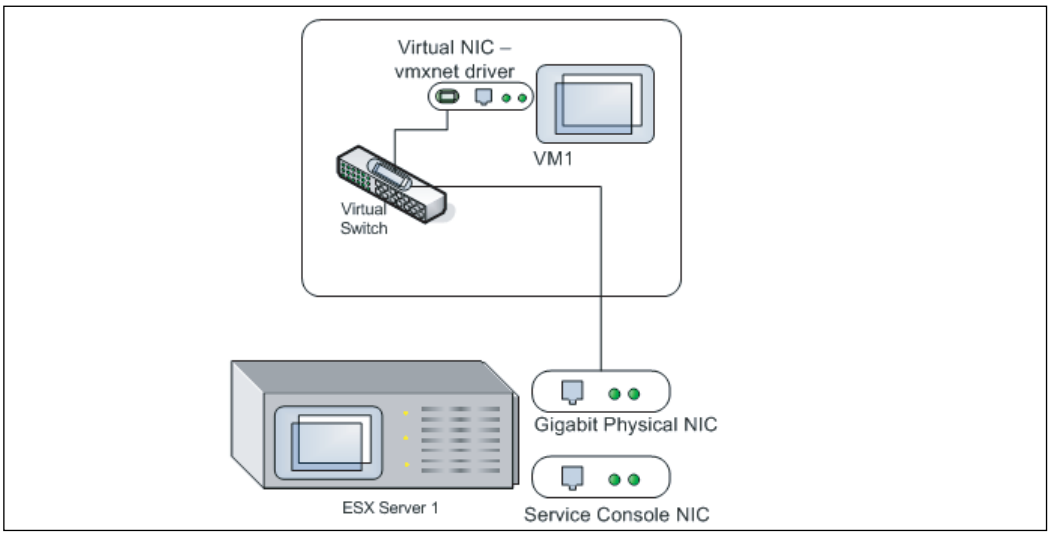



Figure 5.20 depicts a virtual machine with a single vmxnet virtual NIC. The virtual NIC is plugged into a virtual switch, which is bound to a physical Gigabit Ethernet adapter on ESX Server 1 itself.

vmxnet virtual NICs can be utilized only after you install the VMware tools onto your virtual machine.

**NOTE**

When you change a vlance virtual NIC to a vmxnet virtual NIC, it's like physically removing a network card from a computer and replacing it with another. You'll need to reapply all of your IP address properties if your virtual machine has static IP information set.

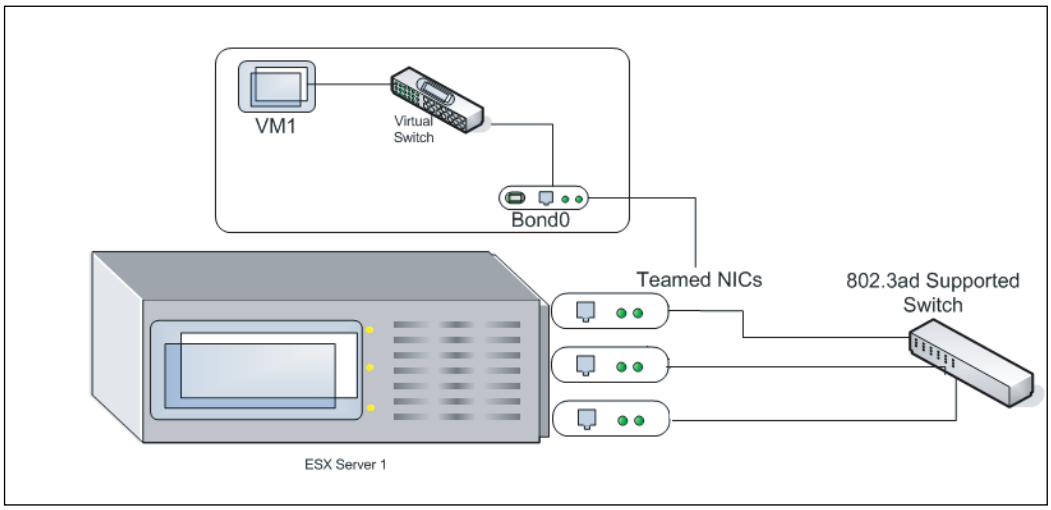
To add additional virtual NICs to your virtual machines (see Chapter 3).

# Bonded NICs

As mentioned earlier, you can bond two physical NICs together for fault tolerance or load balancing. This is very similar to "teaming" NICs. NIC teaming allows you to bind a minimum of two, or a maximum of 10 (10 10/100MB NICs since there's a maximum of eight Gigabit NIC ESX Server supports), physical NICs into a single

logical NIC. It's recommended that you bind identical physical NICs, although you can mix, say, one Gigabit NIC with one 100MB NIC in a bond. Figure 5.21 depicts a bond of three physical NICs.

**Figure 5.21** A Bond of Three Physical NICs




Notice in Figure 5.21 that Bond0 is now associated with the virtual switch. So traffic from VM1 knows nothing of the teamed NICs and thus requires no additional NIC teaming drivers as a physical server might. ESX Server handles the physical teaming and provides a virtual interface—in this example, Bond0—to your virtual machines.

A bond of teamed NICs can provide fault tolerance. If one of your physical NICs in the bond fails or is unplugged, this failure will be detected by ESX Server and network traffic will be redirected to the working NICs. This provides fault tolerance by preventing a single point of failure

A bond can also provide load balancing. ESX handles the load balancing of outgoing network traffic automatically. Incoming traffic is handled by your 802.3ad supported switches, which ESX server has no control over.

For more information on Bonds and NIC teaming, read the following VMware white paper at www.vmware.com/pdf/esx2_NIC_Teaming.pdf.

**NOTE**

For NIC teaming to work in your physical production network, you need to have physical switches that support IEEE 802.3ad static link aggregation.

# MAC Addresses

As you probably know, every physical NIC has its own unique MAC address. This is burned into the NIC by its manufacturer prior to shipping. What you may or may not know is that every virtual NIC has its own unique MAC address as well. A virtual NIC's MAC address can be changed manually if necessary.

## MAC Address Overview

The MAC address is a six-byte hexadecimal number that looks something like this: 00-09-6B-BF-6D-31.

The first three bytes are a unique Vendor ID for the manufacturer of the network adapter. This is called the organizationally unique identifier or OUI. In the MAC address example, the OUI would be 00-09-6B, which is IBM.

Although VMware does not manufacture a physical network adapter, it does provide virtual ones. Thus, VMware has its own OUI; in fact, it has two.

The first VMware OUI is for automatically generated MAC addresses, and in most cases these are the MAC addresses your virtual machines will use. The second VMware OUI is used for MAC addresses that are manually set.

If you were to vi (see Chapter 13) one of your virtual machine's configuration files, it might look something like the image in Figure 5.22.

**Figure 5.22** A vi of a Virtual Machine's Configuration Files

```
floppy0.fileType = "file"

ide0:0.startConnected = "TRUE"
Ethernet0.addressType = "generated"
uuid.location = "56 4d 08 50 15 ba b5 cb-98 52 ba 9d fa c8 7f 88"
uuid.bios = "56 4d 08 50 15 ba b5 cb-98 52 ba 9d fa c8 7f 88"
ethernet0.generatedAddress = "00:0c:29:c8:7f:88"
ethernet0.generatedAddressOffset = "0"
tools.syncTime = "FALSE"
```

The third line down, "Ethernet0.addressType =" shows that the MAC address, also known as the Ethernet address or physical address, has been automatically generated. The sixth line down, "ethernet0.generatedAddress = " shows that the MAC address in this example is 00:0c:29:c8:7f:88. The first three bytes, 00:0c:29, represent VMware's OUI for automatically generated MAC addresses. Every MAC address that is automatically generated in ESX (as well as GSX or Workstation) will have this OUI value. However, for the last three bytes, ESX Server uses a specific algorithm to generate MAC addresses.

Configuring & Implementing…

### VMware's Algorithm for MAC Address Generation

The following is taken from the ESX Server 2.5 Administration Guide and describes how VMware' s algorithm for MAC address generation works:

"We use the VMware UUID (Universally Unique Identifier) to generate MAC addresses. We then check for any conflicts. If there is a conflict, we add an offset and check again, until there is no conflict. (The VMware UUID is based on the path to the virtual machine and the host's SMBIOS UUID.)

Once the MAC address has been generated, it does not change, unless the virtual machine is moved to a different location—for example, a different path on the same server or a different ESX Server machine. We save the MAC address in the configuration file of the virtual machine.

ESX Server keeps track of all MAC addresses that have been assigned to the network adapters of running and suspended virtual machines on a given physical machine. ESX Server ensures that the virtual network adapters of all of these virtual machines have unique MAC addresses.

The MAC address of a powered-off virtual machine is not checked against running or suspended virtual machines. Therefore, it is possible, but unlikely, that when a virtual machine is powered on again, it can get a different MAC address. This is due to a conflict with a virtual machine that was powered on when this virtual machine was powered off."

## Manually Setting MAC Addresses

There may be times when you need to manually set the MAC address of a virtual NIC due to address conflict or if you just want to practice doing it (we recommend this be done only in your test environment). To manually set a MAC address, you

first need to know VMware's second OUI, which is 00:55:56. Next, you have to know what the range of the last three bits is.

Hexadecimal begins with 0 and ends with F. So, the range of any hexadecimal value is 0,1,2,3,4,5,6,7,8,9,a,b,c,d,e,f—thus, you have 16 possibilities instead of the normal ten we're used to.

We know that we must first start with the OUI value, which for a manually set MAC address is

00:55:56:xx:yy:zz

xx.yy.zz are the last three bits, and so the range for those values is 00.00.00 through 3f:ff:ff. Thus, the complete range of a manually set MAC address is

00:55:56:00:00:00 – 00:55:56:3f:ff:ff

For the first of the last three bits, or the xx portion, you cannot go above 3f. Ranges above 3f are reserved for VMware GSX and VMware Workstation. Here are some examples of possible MAC addresses that you can manually set:

00:55:56:01:00:ff

00:55:56:0a:bb:01

00:55:56:3e:ff:f9

A recommendation you might want to follow if you are going to set MAC addresses manually is to pick a range of MAC addresses, say, 00:55:56:01:a1:zz, and only increment the last bit until you run out of values, which in this example would be ff, thus allowing for 256 MAC addresses. Make sure you document your MAC address range and the virtual machines that receive them. This will help keep you from assigning the same MAC address to multiple virtual machines. If you have multiple ESX Servers, you may want to have multiple MAC address ranges, such as the following:

ESX Server 1 = 00:55:56:01:a1:zz

ESX Server 2 = 00:55:56:02:a2:zz

ESX Server 3 = 00:55:56:03:a3:zz

So, now that you understand how to manually set a MAC address, you need to know where to set it.

If you vi a configuration file for your virtual machine and enter the Insert mode (Chapter 13), you need to edit two lines (see Figure 5.22).

The first line to edit in the previous example is the Ethernet0.addressType = "generated." You need to change the "generated" to "static." Next, in this example,

you need to change the ethernet0.generateAddress = "00:0c:29:c8:7f::88. Here, the MAC address has been changed from 00:0c:29:c8:7f:88 to 00:55:56:01:a1:01. Save the changes to your configuration file and then vi the file again and you should see the changes (see Figure 5.23).

**Figure 5.23** Changes Made to the Configuration File

```
ide0:0.startConnected = "TRUE"
Ethernet0.addressType = "static"
uuid.location = "56 4d 0e d8 35 1a a9 61-61 92 36 e9 21 38 cc ca"
uuid.bios = "56 4d 0e d8 35 1a a9 61-61 92 36 e9 21 38 cc ca"
ethernet0.generatedAddress = "00:55:56:01:a1:01"
ethernet0.generatedAddressOffset = "0"
tools.syncTime = "FALSE"
~
```

It's important that you follow this MAC address convention as ESX Server does not support any random MAC address scheme. Following this convention ensures you never run into a MAC address conflict within your environment.

# Port Groups and VLANs

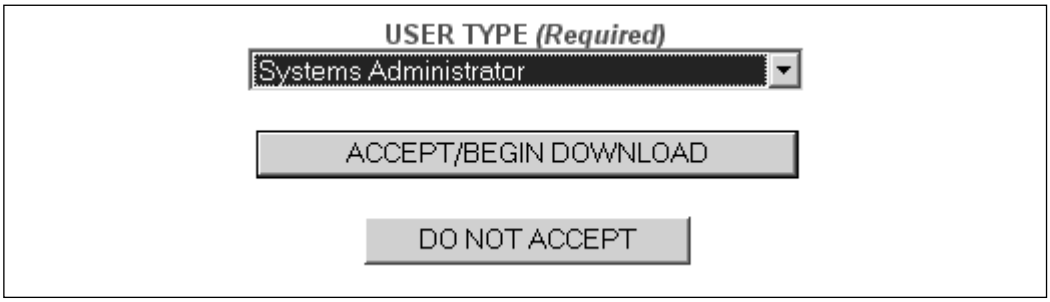
When you go to configure a virtual switch, you'll see in the same window the area in which you can configure a port group. So, what is a port group?

Port groups are VLANs basically. VLANs, or virtual local area networks, are defined by the IEEE 802.1Q standard.

A VLAN is a group of network devices on a LAN that is configured to communicate as if they were attached to the same LAN segment and were in the same broadcast domain. In reality, however, they can be geographically dispersed and connected to a number of different LAN segments. This is accomplished through VLAN tagging. VLAN tags, for example, can be switch port based, MAC address based, protocol based, or policy based. A VLAN tag is an extra four bytes that are inserted after the source and destination address in an Ethernet frame.

For more information on 802.1Q and VLAN tags, go to http://standards.ieee.org/getieee802/download/802.1Q-2003.pdf. Under the USER TYPE drop-down, we input Systems Administrator, clicked Accept/Begin Download, and got the download shown in Figure 5.24. See section 9 of this document for VLAN tag format information.

**Figure 5.24** User Type Drop-Down Menu

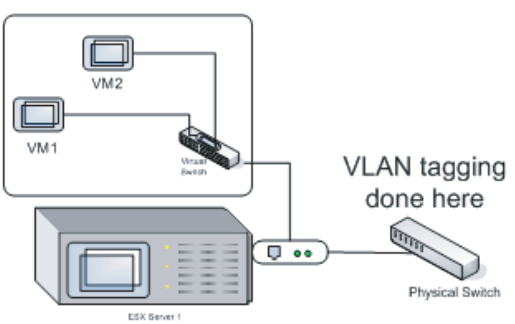



In ESX Server, there exist three options for VLANs:

- External Switch Tagging (EST)
- Virtual Switch Tagging (VST)
- Virtual Guest Tagging (VGT)

Exterior Switch Tagging (EST) mode relies exclusively on the tagging of network traffic by the physical switch (see Figure 5.25). All traffic goes through one of your ESX Server's physical NICs, which is attached to a physical switch port providing VLAN tagging on outgoing traffic and strips the tag for incoming traffic. The limitation to this method, of course, is that it's switch port based so the number of VLANs your ESX Server can support is limited to the number of physical NICs you have in your ESX Server.

**Figure 5.25** Exterior Switch Tagging



In Figure 5.26, there's one physical NIC dedicated to the virtual machines; thus, you could have only one VLAN. EST is supported by all versions of ESX.

Virtual Switch Tagging (VST) mode is the recommended configuration by VMware. This mode is used when you utilize port groups. You configure a port group to a virtual switch. You then assign one of your virtual machine's virtual network adapters to the port group itself. Figure 5.26 shows a basic VST configuration.

**Figure 5.26** A Basic VST Configuration

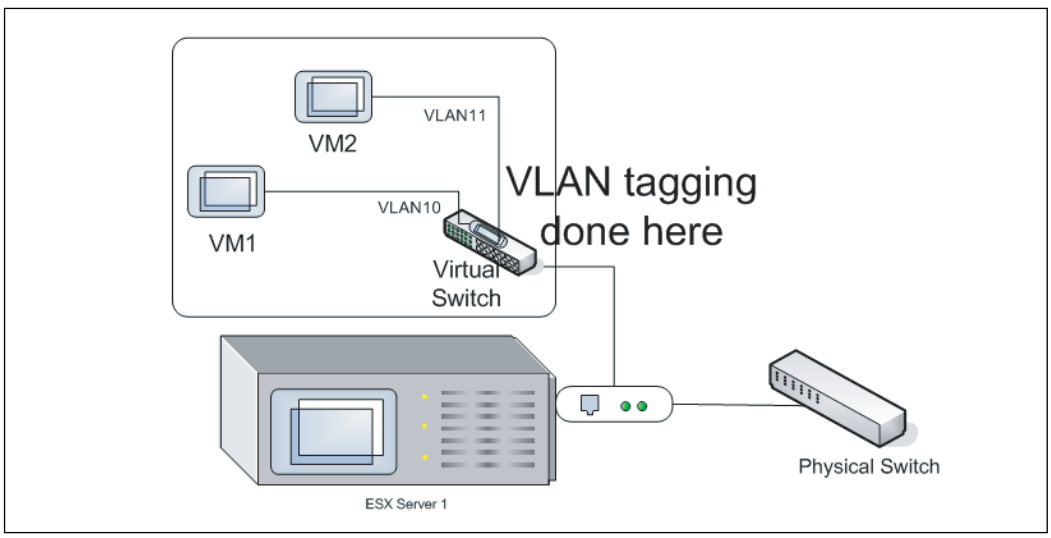
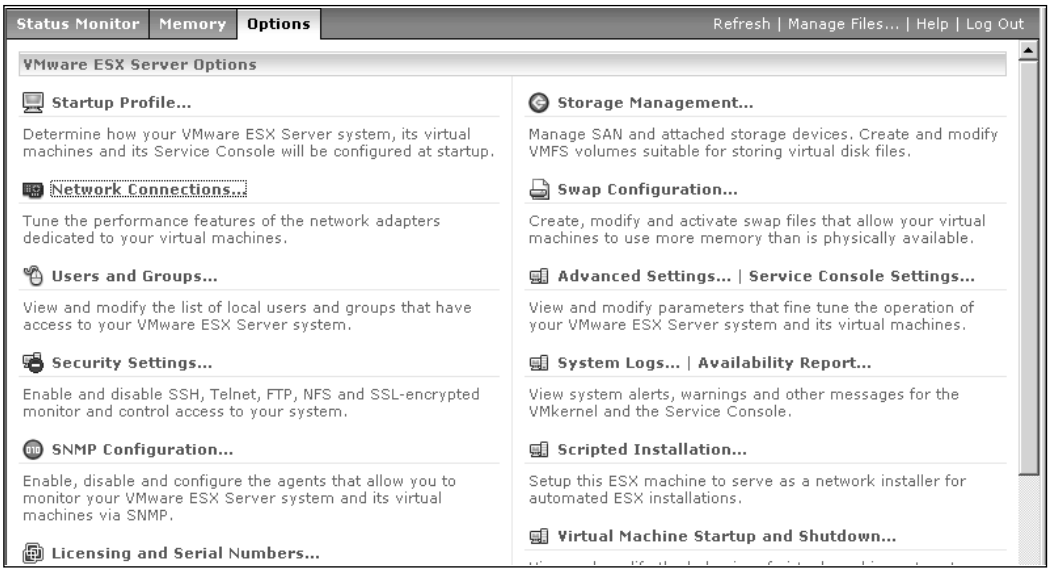



Figure 5.27 depicts two Port Groups: VLAN10 and VLAN11. Both port groups are assigned to the same virtual switch, which is bound to one physical NIC on the ESX Server. The physical NIC is plugged into a physical switch port. All necessary switch trunks and support for 802.1Q must be established within your production switches.

VLAN10 and VLAN11 are the Port Group labels. You'll select from the list of Port Group labels when configuring your virtual machine. The following is a step-by-step process for creating a Port Group and assigning that Port roup to a virtual machine.

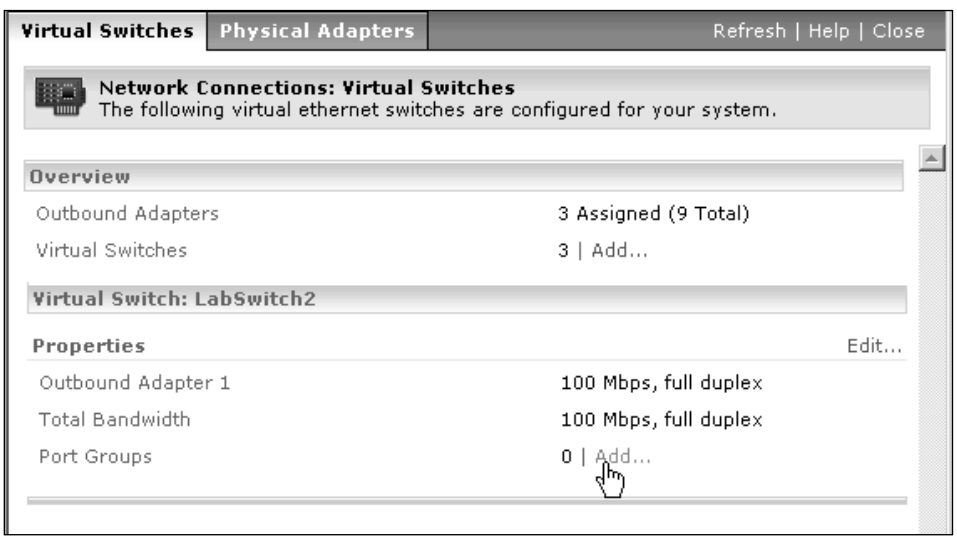
1. Open the MUI and click the **Options** tab (see Figure 5.27).

**Figure 5.27** Choosing Server Options



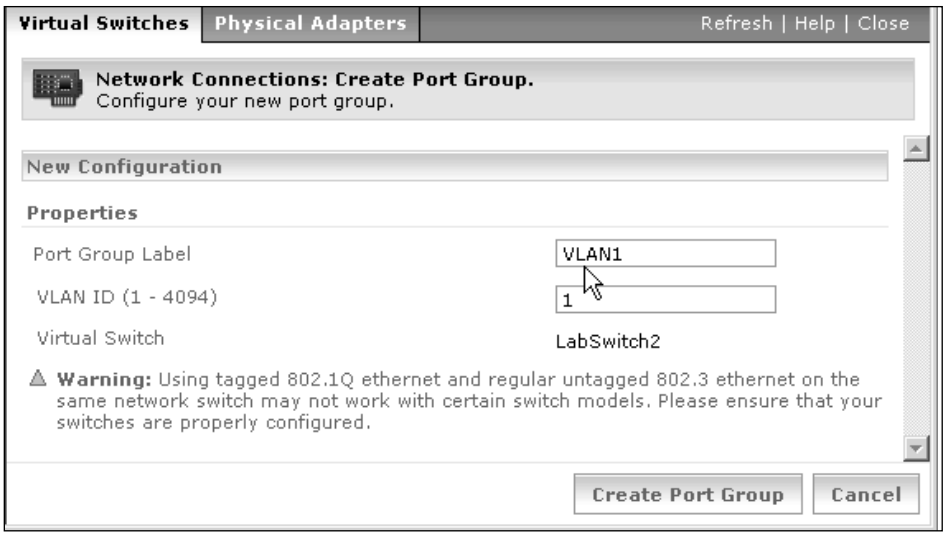

2. Click the **Network Connections** link.
3. From a virtual switch, click **Add** to the right of Port Groups (see Figure 5.28).

**Figure 5.28** Adding a Virtual Switch



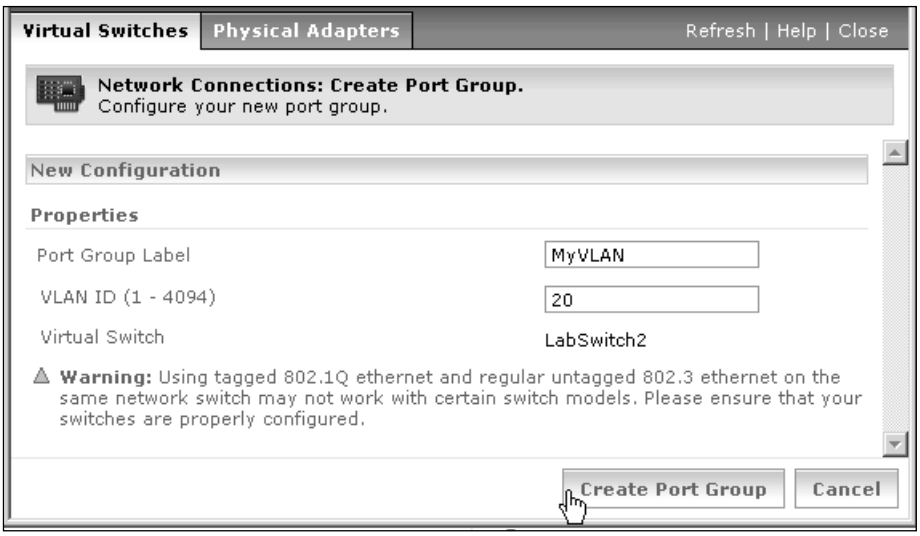
4. The **Create Port Group** window appears with default values. The first value is the Port Group Label (see Figure 5.29).

**Figure 5.29** Creating a Port Group



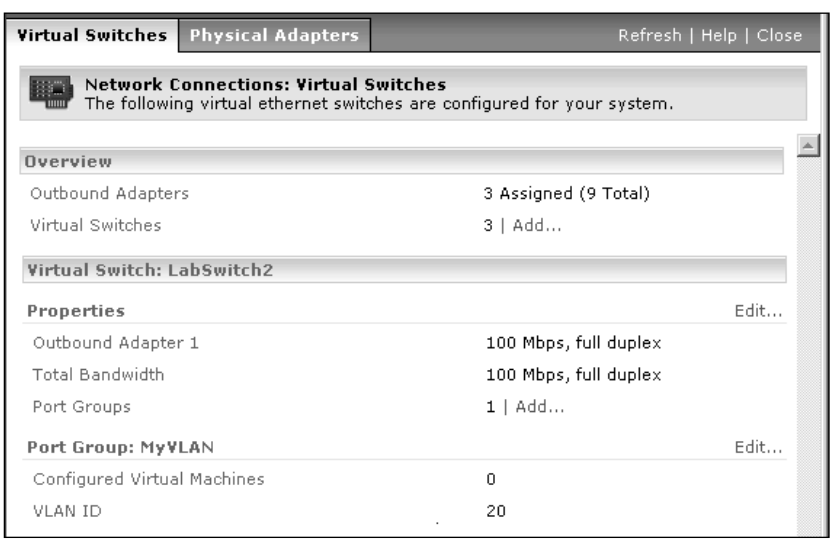

5. This is set by default to VLAN1, but you may change the label to anything you like. This label is ESX-specific and it's what you will assign to your virtual machines.
6. By default, VLAN ID is set to 1, which in the VLAN world is known as the "native VLAN" (it could also be 100) and should not be used if at all possible. Ask your network engineer what the native VLAN is and avoid using it. Change the VLAN ID.
7. Notice the warning at the bottom of Figure 5.30. Heed it.

**Figure 5.30** A Warning about Switch Configurations



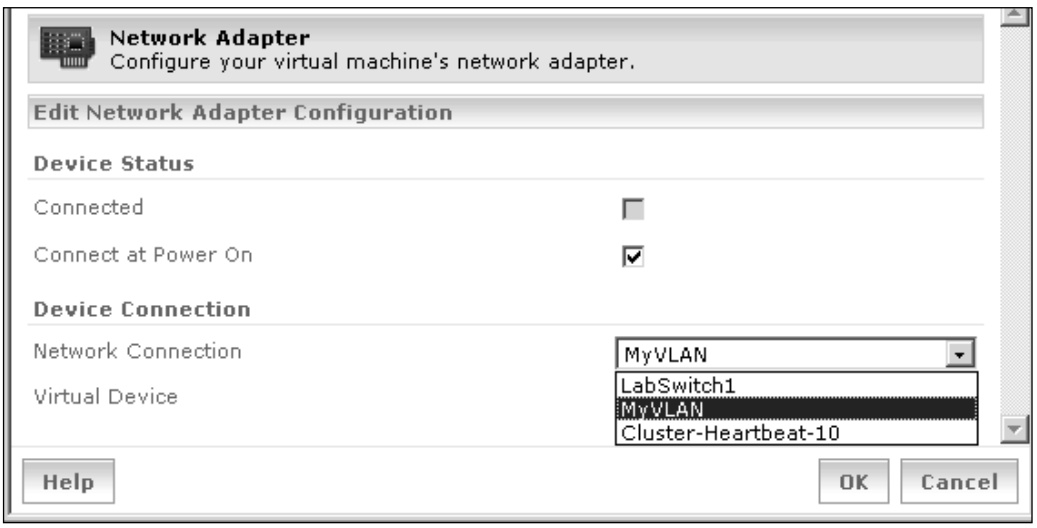

8. After you've made the changes to your Port Group click **Create Port Group**.
9. You should now be back in the Virtual Switches window. Under the virtual switch you created, you should see the Port Group you configured with the values you gave it (see Figure 5.31).

**Figure 5.31** Configuration of a Virtual Switch



10. The Port Group **MyVLAN** now displays with a VLAN ID of 20. However, there are no virtual machines configured for it yet. Close the window. You'll now configure a virtual machine to point to the Port Group.
11. Open the **Properties…** of a virtual machine and go to the **Hardware** tab (you may also select **Configure Hardware** from the virtual machine drop-down).
12. Either edit an existing virtual NIC or add a new one. When you're in the properties of the virtual adapter, you'll see MyVLAN (or whatever you've labeled it) in the drop-down menu for **Network Connection** (see Figure 5.32).

**Figure 5.32** Configuring a Virtual NIC




13. By selecting the Port Group Label and then clicking **OK**, you'll configure your virtual machine to the VLAN connection you configured earlier.
14. Close the virtual machine's properties page.

A quick recap: each port group represents one VLAN, and you can have multiple port groups per virtual switch—thus, you can have multiple VLANs per virtual switch. This eliminates the restriction of one VLAN per physical NIC tied to the one physical switch port we found in the EST mode. VLANs (aka, Port Groups) are handled by ESX Server itself for your virtual machines, and the virtual switch is responsible for VLAN tagging. VST is available only in ESX Server 2.1.0 and later releases.

The final mode for using VLANs within your ESX environment is called Virtual Guest Tagging (VGT). VGT requires that you run VLAN 802.1Q drivers in the virtual machine itself. This allows you to run any VLAN you need since the configuration is within the virtual machine itself and not tied to any switch or physical NIC. However, VLAN tagging takes extra CPU cycles away from your virtual machine and you need to ensure you use a driver that supports VLAN hardware acceleration. VGT requires ESX Server version 2.1.1 or later and, of course, your guest operating system needs to support the 802.1Q drivers.

For more information on Port Groups, VLANs, and VMware recommendations, read the well-written whitepaper at www.vmware.com/pdf/esx_vlan.pdf.

# Networking Tools

To assist with network troubleshooting and configuration, there are a number of tools, mostly run from the command line, that you can use.

## *service network restart*

Use the *service network restart* command if you're having problems with the Service Console NIC, perhaps regarding MUI responsiveness (service *servicename* restart can be used to restart any service, just substitute *servicename* for the actual service name). This restarts eth0, your service console NIC, as well as the lo, which is the local loopback. From the command line of your ESX Server, type **service network restart**.

The output should look something like the image in Figure 5.33.

**Figure 5.33** The Output of *service network restart*

```
[root@esx1 root]# service network restart
Shutting down interface eth0:                              [  OK  ]
Setting network parameters:                                [  OK  ]
Bringing up interface lo:                                  [  OK  ]
Bringing up interface eth0:                                [  OK  ]
[root@esx1 root]#
```

This command bounces the network service.

## findnic

The VMkernel labels the physical adapters it's given for virtual machines vmnic0, vmnic1, vmnic2, and so on. The lower the number, the lower the physical network adapter is in the PCI bus/slot hierarchy. If you have many physical NICs in your ESX Server, it may be difficult for you to map where physically, say, vmnic9 resides. This can

be especially true if you have dual- or quad-port network adapters. *findnic* can assist you in finding and labeling the physical adapter from the VMkernel nic name.

The *findnic* command allows you to bind an IP address to a vmnic—for instance, vmnic1—and then ping another IP address. From either the success of the ping or from the blinking lights on the physical adapter, you should be able to determine what physical NIC is vmnic1.

From the command line of your ESX Server and type **findnic –help**.

The following are your options:

Usage: findnic [-f --flood] [-i --interval seconds] nicname local_ip remote_ip.

Thus, an example would be **findnic vmnic1 192.168.1.203 192.168.1.202**.

The syntax in the previous example will bind IP address 192.168.1.203 to vmnic1 and then constantly ping 192.168.1.202.

The output should look like the image in Figure 5.34.

**Figure 5.34** The Output of *findnic vmnic1 192.168.1.203 192.168.1.202*

```
[root@esx1 root]#  findnic vmnic0 192.168.1.203 192.168.1.202
Warning: loading /tmp/vmxnet_nic_test will taint the kernel: no license
Checking eth1
PING 192.168.1.202 (192.168.1.202) from 192.168.1.202 : 56(84) bytes of data.
64 bytes from 192.168.1.202: icmp_seq=0 ttl=255 time=14 usec
64 bytes from 192.168.1.202: icmp_seq=1 ttl=255 time=19 usec
64 bytes from 192.168.1.202: icmp_seq=2 ttl=255 time=13 usec

--- 192.168.1.202 ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max/mdev = 0.013/0.015/0.019/0.004 ms
Done
```

You can look at the back of your ESX Server and see what NIC is constantly flashing. It should be vmnic1. Since in Figure 5.34 the pings are successful, you can determine that networking is successfully set up for vmnic1 and that fact can help you determine which physical network adapter vmnic1 is.

# ifconfig

Syntactically similar to ipconfig, ifconfig is a networking command-line tool you should familiarize yourself with. From the command line of your ESX Server, type **ifconfig –help**.

You should receive a number of different arguments (options). With ifconfig, you can review IP address information. You can also set IP address information.

If you type ifconfig with no argument, you'll receive the status of only the active interfaces. If you type ifconfig –a, you receive the status of all the interfaces, even those that are inactive or down.

## mii-tool

mii-tool can be used to modify the NIC settings of your virtual server. For example, if you want to change one of your NICs from autonegotiate to 100-Mbps Full, use this tool. Of course, you could always use the MUI as well.

An example of the command-line syntax to complete the change would be *mii-tool 100baseTx-FD eth0*. This sets the service console NIC to 100 Mbps and Full Duplex.

You could also do the same for the NICs reserved for your virtual machines by using *mii-tool 100baseTx-FD vmnic1*. This command would set vmnic1 to 100 Mbps and Full Duplex.

# Summary

VMware provides incredible flexibility when it comes to networking. VMnets provides network communication between virtual machines on the same ESX Server and when used with VMnics can provide a much more secure and cost-effective solution for network access to services. By allowing the administrator to easily control every aspect of network configurations down to customizing MAC addresses, virtual networking is another very cool component that comes from evolving into a virtual environment. Learn it well.

# Solutions Fast Track

## Virtual Switches

- ☑ Like a physical switch or hub, a virtual switch has a certain number of ports into which you can plug the virtual NIC of your virtual machines.
- ☑ Virtual switches allow you to create VMnets and VMnics, both of which allow your virtual machines to pass network traffic. VMnets pass this traffic only within the ESX Server itself, while VMnics are bound to at least one physical NIC, which provides the ability to pass network traffic to your corporate intranet, the Internet, and other virtual machines.
- ☑ Avirtual switch called a VMnet may be set up exclusively for routing high-speed network traffic between virtual machines on the same ESX Server.

## Physical and Virtual NICs

- ☑ You should have a minimum of two NICs for your ESX Server: one for the Service Console and the other dedicated to your virtual machines. It's through the physical NIC dedicated to your virtual machines that a virtual NIC can be attached to pass network traffic.
- ☑ When you install ESX Server, the first NIC in the first PCI bus/slot that ESX detects will be the one it assigns to the ESX console.
- ☑ Each virtual machine can have four virtual NICs.
- ☑ Like any physical NIC, a virtual NIC can have one or more IP addresses and a MAC address.

## MAC Addresses

- ☑ MAC addresses are, by default, automatically assigned; however. VMware lets you manually assign them as well.
- ☑ The MAC address is a six-byte hexadecimal number.
- ☑ A recommendation you might want to follow if you are going to set MAC addresses manually is to pick a range of MAC addresses, say, 00:55:56:01:a1:zz, and only increment the last bit until you run out of values

## Port Groups and VLANs

- ☑ VMware provides for three types of VLAN or Port Groups: External Switch Tagging (EST), Virtual Switch Tagging (VST), and Virtual Guest Tagging (VGT).
- ☑ Each port group represents one VLAN.

## Networking Tools

- ☑ A few command-line tools can assist you in managing the network infrastructure of your virtual environment, such as findnic, mii-tool, and others.
- ☑ Use the service network restart command if you're having problems with the Service Console NIC
- ☑ The findnic command allows you to bind an IP address to a vmnic.
- ☑ mii-tool can be used to modify the NIC settings of your virtual server

# Frequently Asked Questions

The following Frequently Asked Questions, answered by the authors of this book, are designed to both measure your understanding of the concepts presented in this chapter and to assist you with real-life implementation of these concepts. To have your questions about this chapter answered by the author, browse to **www.syngress.com/solutions** and click on the **"Ask the Author"** form. You will also gain access to thousands of other FAQs at ITFAQnet.com.

**Q:** Does VMware support NIC teaming?

**A:** Yes, and it's a very good idea to configure your ESX Server, especially if it's for production, with teamed NICs. This will provide hardware fault tolerance in case one NIC fails.

**Q:** How many physical NICs do I need on my ESX Server?

**A:** That depends on the number of virtual machines and the network traffic they produce. You should have a minimum of two: one for your Service Console and one dedicated to your virtual machines.

**Q:** If I'm building a cluster using Microsoft Clustering Service, what's the best configuration for my heartbeat NIC?

**A:** For a Cluster in a Box, create a VMnet that your nodes of your cluster can attach their heartbeat NICs.

**Q.** Can you attach virtual machines on any ESX Server to a VMnet?

**A.** No. Only virtual machines on the ESX Server that the VMnet resides can attach to it.