

VMware ESX 3.x Server  
and  
VirtualCenter 2.x  
(GA Build Eval)

Service Console Guide

Document Version 1.1

**RTFM Education**

Beyond the Manual... with Mike Laverick

By Mike Laverick  
© [RTFM Education](#)

For Errors/Corrections please contact:  
[mikelaverick@rtfm-ed.co.uk](mailto:mikelaverick@rtfm-ed.co.uk)

# Table of Contents

<b>Introduction .....</b>	<b>3</b>
<b>Module 1: Users and Rights .....</b>	<b>6</b>
Creating a New User .....	6
Elevating yourself to ROOT .....	6
Disabling Auditing on ROOT (Not Recommended) .....	6
Miscellaneous User Managements Tasks .....	7
<b>Module 2: Getting around &amp; Getting help .....</b>	<b>8</b>
<b>Module 3: File &amp; Folder Management.....</b>	<b>10</b>
<b>Module 4: Networking .....</b>	<b>15</b>
Viewing your Switches & Service Console Networking .....	15
Creating a vSwitch (Internal) .....	15
Creating a vSwitch (Single NIC) .....	16
Creating a vSwitch (Multiple NIC's).....	17
Deleting a Switch from vSwitch .....	18
Creating PortGroups for VLAN Networking NIC Team .....	19
Creating Vmkernel Switches .....	20
Changing your Service Console IP Settings.....	22
Setting the Speed and Duplex of NIC's.....	23
Recreating your vswif0 Interface .....	23
Removing a NIC from vSwitch.....	24
Deleting a PortGroup from vSwitch .....	24
Managing the ESX Firewall.....	24
Fixing ESX Networking after an Upgrade .....	27
Typical vSwitch Errors .....	29
<b>Module 5: Configuring Physical Storage .....</b>	<b>30</b>
Configuring SAN Storage .....	30
Configuring NAS Storage .....	30
Connecting to iSCSI Storage (Software) .....	31
Creating a VMFS Volume .....	33
Changing the Volume Label .....	35
Viewing VMFS Volumes/Partition Information .....	36
Viewing Available Physical Disk Space .....	37
Mounting and U-mounting VMFS's .....	38
Managing Storage Devices.....	38
Using USB storage Devices .....	39
<b>Module 6: Managing Virtual Machines .....</b>	<b>43</b>
Creating Virtual Disks .....	43
Creating and Managing RDM's .....	44
Exporting & Importing Virtual Disks .....	46
Copying Files from one ESX host to another.....	48
Renaming Virtual Disks .....	49
Resizing Virtual Disks.....	50
Deleting Virtual Disks.....	53
Using VMware-cmd .....	53
Changing Parameters within a VM with /proc .....	57
<b>Module 7: Performance Monitoring .....</b>	<b>58</b>
Using ESXTOP to Monitor Virtual Machines.....	58
Tools to Monitor the Service Console.....	64
<b>Module 8: Upgrading/Patching ESX.....</b>	<b>65</b>
Pre-Upgrade Checks .....	65
Custom Upgrading with a Tar-Ball .....	65
Upgrading to VMFS-3 from the Service Console.....	67

# Introduction

## Purpose of this guide

This guide is designed for people who already know ESX 3.x and VC 2.x quite well. Although it starts as a beginners guide initially, it pretty rapidly starts to assume very good knowledge of the system. I would recommend you get to grips with the GUI first, and feel comfortable with Vi-3 before attempting this guide.

That said – you might be wanting to carry out a discrete procedure from the command-line. So feel free to dip in – find what your looking for and then – dip out again! ☺

It is not a comprehensive guide to ALL the commands – just the primary ones. I hope to make this guide gradually more comprehensive, and cover all new commands that useful. I've deliberately not covered every single esxcfg command – because not all of them are terrifically useful...

There are some big topics that I have yet to add to this guide – this includes setting NTP and Active Directory authentication for the Service Console

Please email at the email address at the beginning of this document if spot any errors

Where possible – I don't use the VI Client. I only use the VI client if there is no other way – even if the VI client is easier. The reasoning behind this force the use of the command-line.

Perhaps I should state why using the command-line might be useful...

- Automation of tasks?
- Er, because everything else is broken – and login in at the ESX host at the Service Console is your only option
- Because sometimes its quicker (sometimes it's quicker using the mouse too...!)
- Because like Everest, it is there!

## But I prefer the mouse?

- Then this guide isn't for you...
- Choose File, and close... ☺
- But seriously, what are commands – only words that carry out instructions
- Who in Windows hasn't used the net use or net servicename stop/start command?
- I grew up on DOS/Windows3.x/Novell – a mix of GUI and commands. For which I am very grateful. I am happy in both environments.

Although I do find myself swearing when commands refuse to work, but that said I also swear at Windows just as much – perhaps like Pete from Big Brother have "Tourettes Syndrome". God knows when ever I watch Big Brother I feel uncontrollable the urge/need/desire to shout "wankers" at the top of my voice... Hopefully this guide will reduce the amount of swearing you do at computers. You must remember – they can't hear or understand your profanities – and talking to inanimate objects is usually the first stage of madness...

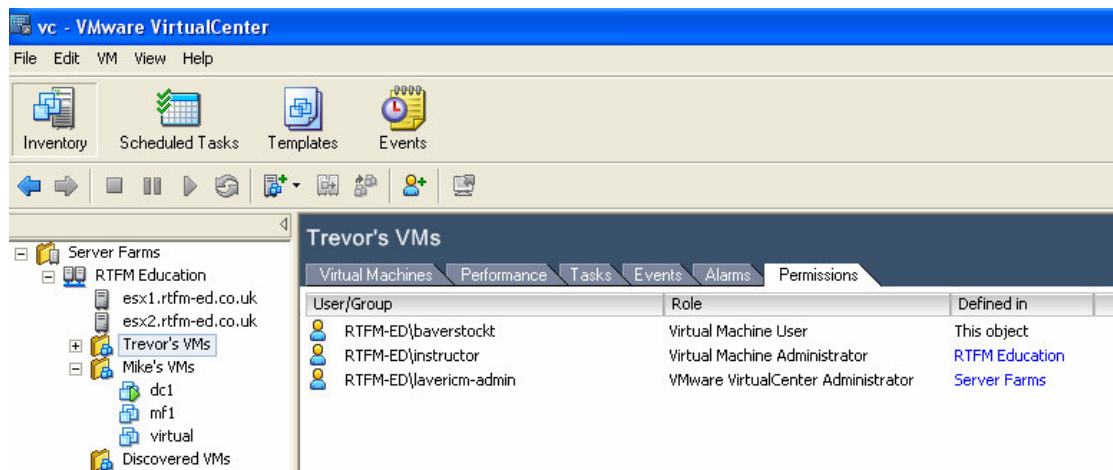
- Anyway, I digress - I've noticed as the GUI grew to dominate our environments – my command-line skills took a hit. If you don't use it, you lose it – as the saying goes.

I don't intend my skills to be eroded by a little mouse.☺

Of course they could be equally eroded by the keyboard. ☹

## Hardware

- I use a 192.168.2.x range on my network with 192.168.2.101 for esx1, 192.168.2.102 for esx2. My DNS server is 192.168.2.200. All IP's here are just example from my network range – replace with your own
- This guide assumes you have at least 2 servers with 2CPU's at 1.4mgz each
- 2GB of RAM
- 2x36GB Hard Drives (SCSI Internal)
- 4x Network Cards (production in a bond, vmotion and Service Console eth0)
- This is the specification of my server an old Dell 1650 PowerEdge using PIII processors!
- My hardware isn't officially supported by VMware anymore. Anyway it still runs. But at some point I am going to buy two DL380, and re-use this hardware as my VC box and a NAS box...
- I have VC set-up with a SQL Database. The layout of my VC looks like this before the upgrade:



Two Servers, Two Virtual Machine Groups (for me and my mate, Trevor)  
 The domain name is rtfm-ed.co.uk  
 lavericm-admin – is the VC Administrator set at Server Farms

instructor – is a Virtual Machine Admin from the RTFM Farm (used when I teach with my hardware)

baverstockt – is a Virtual Machine User with rights only for Trevor's VMs Group

## Software

- In my case VC Server 2.x and SQL 2000 (SP3) all runs on Windows 2003 with Service Pack 1. This was the most current release of Windows at the time of writing this document
- **Warning:** As ever my documents are released as is and without warranty

## Hardware & Software

- As we at the Service Console – you don't really need to have exactly what I have but it would help to have what I have or better

## Conventions in this Guide

- I use PuTTY to get Service Console sessions
- I use nano rather than vi to edit and save files – mainly because I'm no Linux guru, and I find nano more user friendly. Vi text editor is popular in the Linux community because it pretty much standard amongst every Linux distribution. Don't let anyone make you feel an idiot or small because you use nano. As long as you can successful manage the system that's all that counts in my book.

Linux people, do NOT flame me on the merits of using VI. I will absolute NOT respond! ☺

- ***Any Major titles marked with Red*** indicates this section is broken, released as is, and I intend to return to fix it. But got bored banging my head against a brick wall. ☺
- **IMPORTANT:**  
To have your command-line changes reflected in the VI Client you must restart the hostd service on the ESX host. You can do this by using

**service mgmt-vmware restart**

## Change Log from 1.1 to 1.2

- None

# Module 1: Users and Rights

## Note:

- In most case your rights will be governed by VirtualCenter and Active Directory
- However, at the Service Console the account database that's used is local to the Service Console
- After an upgrade of ESX from 2.x to 3.x – root does have rights to connect remotely using SSH.
- After a clean installation – the default is that root has no remote access via SSH. This to enforce traceability. User must logon as underprivileged users and elevate the rights to root. This change in rights is logged in /var/log/messages
- The easiest way to create a normal user for use with PuTTY would be the VI Client. You can point it at the ESX host and use the "User & Groups" tab to create a new user

## **Creating a New User**

1. **Logon at the Service Console as ROOT** (physical or ILO)
2. Type:

```
useradd lavericm-admin -p *****
```

## **Elevating yourself to ROOT**

1. **Logon at the Service Console** as your user account
2. Type

```
su -
```

### **Note:**

This Switches User and assumes root, unless otherwise specified. The - takes roots environmental settings (very important if you want to run any commands properly)

3. Type the password for root

### **Note:**

You can also use a tool called sudo. This allows you to logon as an underprivileged user, and run commands with the ID of a privileged user

## **Disabling Auditing on ROOT (Not Recommended)**

### Note:

- Some applications do not support levitation to a higher plain - for example WinSCP. Sure you could use WinSCP to gain access as an ordinary user, but then you might lack permission to copy the files you need. If you try to logon as root, WinSCP will give you access denied.
- If you wish to disable the restriction on ROOT not being allowed direct access using SSH then carry out the following task. I wouldn't recommend doing this as you will loose enforcement of your audit trail.

1. **nano -w /etc/ssh/sshd\_config**
2. Locate: **PermitRootLogin no**
3. Place a # in front of PermitRootLogin no like so: **#PermitRootLogin no**
4. **Exit Nano & Save the file**

5. **Restart sshd** with **service sshd restart**

## ***Miscellaneous User Managements Tasks***

- To change your password:

**passwd root**

- List connected users

**w**

### **Note:**

This gives a result like so:

```
11:20:21 up 3:20, 1 user, load average: 0.02, 0.08, 0.07
USER      TTY      FROM          LOGIN@      IDLE        JCPU        PCPU        WHAT
root      pts/0    192.168.2.254 8:30am     0.00s      0.04s      0.01s      w
```

# Module 2: Getting around & Getting help

**Note:**

- This module is for novices/newbies and people who have never used the Service Console before (sometimes referred to in the community as vmnix or the COS)
- The modules are NOT prescriptive – so if you would like skip and move to something more interesting please feel free to do so
- This section covers all the basics of using SSH and puTTY
- Oh, it occurs to me you might not know what these are... SSH is secure protocol (22) which allows you connect remote to an ESX host – giving a command-line interface. PuTTY is a free SSH Client which is very popular in the community. This saves you using an ILO to get to the command-line environment
- The command-line environment is actually something called the BASH (Bourne Again Shell)

Get help on a command	<b>man reboot</b> <i>or</i> <b>reboot -- help</b> <i>or</i> <b>reboot -?</b>	This works with both Linux & ESX commands. Q to quit Gives you short explanation. With   more this pauses the output <b>Warning:</b> Beware of -h this actually triggers a reboot of your ESX host!
View Last 10 commands	<b>history   tail</b>	History normally list your past commands, used with tail it will filter to show only the last 10
Clear Bash History	<b>history -c</b>	Bash has a history, not unlike doskey (except its in a file, not memory). Use this to clear your history of commands previously typed
Clear the screen	<b>clear</b>	Same as CLS in a cmd prompt
Switch to being root	<b>su -</b>	It is regarded as good practice to login with a VM Administrator account and then switch to being ROOT if required. The minus sign switches bash to ROOT and ROOT's environment. Without you would have the rights of root, but command might fail due to being in the incorrect path for the executable
Currently logged in user	<b>whoami</b> <i>or</i> <b>id</b>	I find it especially important to know who I am in all situations. Failure to know who you are can result in surprising results. Id shows your group membership as well.
Change the Date/Time	<b>date -u 0701180504</b>	The -u switch indicates the UTC format is being used. Numbers equate to mm/dd/hh/mm/yy



Display ESX Version Number	<b>vmware -v</b>	Should display information along this kind "VMware ESX Server 2.5.0 build-11343"
Search Man pages by index	<b>makewhatis man -k partition</b>	Creates an index of the man pages Searches manual pages for string of partition
Reboot ESX Server	<b>Reboot</b>	Reboots the server, does not ask are you sure... are you really sure?
Shutdown	<b>shutdown now "Server shutting down for maintenance"</b>	Shuts down the VM's and stops remote access – physical console is still accessible
Shutdown & Halt	<b>shutdown -h now "server shutting down for maintenance"</b>	Shuts down the VM's; Stops Remote Access – and does a fault halt of the system and volumes un-mounted

# Module 3: File & Folder Management

Purpose	Syntax Example/Sample	Notes
List files	<b>ls -l</b>	In PuTTY this colour codes the files, and directories and shows Type, Permissions, Group, User, Size (b) Date, and filename
List files with a pause	<b>ls -l   more</b>	The   is the pipe symbol commonly found where the \ is on a UK keyboard
List hidden files	<b>ls -a -l</b>	Shows files that are hidden – files are made hidden if prefixed with a period, such as ./install
Full Path location	<b>pwd</b>	Like what would see in a DOS command prompt
Return to the root	<b>cd /</b>	You have to put a space between d and /
Return one directory up	<b>cd ..</b>	Again, you need a space between cd and ..
Go to home directory	<b>cd</b>	CD on its own returns you to the home directory of the current users
Type the contents of a file	<b>cat instructor.vmx</b> <b>less instructor.vmx</b>	Same as the TYPE command in DOS Works better with longer files – use the keystroke [Q] to quit like man
Search for string inside a file	<b>grep lavericm /etc/passwd</b>	Here we are searching for a piece of text called lavericm in the file called passwd
Use a command together with grep	<b>ls /etc -l   grep vmware</b>	Here we are listing all the files in /etc that contain the string vmware
Edit the contents of a file	<b>nano -w /etc/fstab</b>	-w disables word wrap and stops unwanted carriage returns Control+X, [ENTER] [ENTER] exits nano and saves a file
Create/Delete a directory	<b>mkdir /root/scripts</b> <b>rmdir /root/scripts</b>	Without file path, directory made relative to your path
Delete a file	<b>rm filename</b>	If your deleting lots of files <b>rm -f *.txt</b> will delete all the txt files but will NOT prompt you
Delete many files without prompts	<b>rm test*.txt -f</b> <b>rm instructor*.vmdk</b>	The -f forces a removal off all the files that begin test and have the txt extension without a prompt. <b>DANGEROUS COMMAND!</b>  The second example would delete a virtual disk – both its metadata file and its “flat” file

Delete contents of directory and all files and subdirectories	<b>Rm /root/vm-support.3630 -r -f</b>	Here I am deleting the extracted contents of an untar'd vm-support script. -r does a recursive delete of all files and folders within vm-support.3630 without a prompt (-f) <b>DANGEROUS COMMAND!</b>
Mount a CD at the Service Console	<b>mount /dev/cdrom</b> <b>ls -l /mnt/cdrom</b>	Root only!
Mount an ISO at the Service Console	<b>mkdir /mnt/isocd</b> <b>mount -o loop -t iso9660 -r /vmimages/esx3.iso /mnt/isocd</b>	Create directory for the mount point -o loop means mount the device as a block device -t iso9660 means its using the iso file system (as opposed to say Joliet) -r means to mount read-only
Copy a file	<b>cp /vmimages/w2k3.iso /vmfs/volumes/nas_isos</b>	Be careful with the use of wildcards if you try to copy every file with *.* as you would with DOS/Windows – this would miss out files that DID NOT have extension. So w2k3.iso would be copied but w2k3 would not...
Securely copy a file from one ESX server to another	<b>scp /vmimages/w2kadvsrv-sp4.iso root@esx1.rtfm=ed.co.uk:/vmimages/</b>	This uses secure copy. You will prompted with some security warnings on the first copy for the first time. You will have manually type the password of the remote machine. <b>Warning:</b> You will need to enable firewall the SSH Client on the ESX where your files are located (source) but NOT where they are being copied to (destination)
Renaming a file/folder	<b>mv w2k3.iso cdw2k3.iso</b>	Move because effectively, any rename is move procedure.
Find a file	<b>whereis vmsnap.pl</b>	If you know the name of a file but can't remember where it is stored the whereis command is dead easy to use
Find a file	<b>find / -iname '*.conf'</b>	Find is much more powerful but can take longer based on your search criteria. Search here begins at the root / and is case-insensitive search by using -iname and '*.conf' would find every conf file – note ` ` are required – these are 'single quotes' not "double-quotes"
Find new files	<b>find / -mount -mtime -1 -print</b>	/ is the search point, -mount volumes mounted, -mtime is the duration and -print is the format for output to the screen
Find files of N size	<b>find / -mount -size +10240k</b>	This would find files off / taking up more than 10MG
Compress a single file	<b>gzip /vmfs/volumes/local-esx1-esx1/instructor/*.vmdk -best</b>	Works best with single files such a vmdk file Caution, automatically deletes original and adds a gz extension to

		the file name
Uncompress a single file	<b>gunzip</b> /vmfs/volumes/local-esx1-esx1/instructor/*.vmdk	As above but in reverse! There is also a utility called bzip2 and bunzip2 which use newer algorithms which offer better compression ratios and better performance
Compress Multiple files	<b>tar -czvf</b> /backup/allfiles.tgz /vmimages	This would backup all the files in the vmimages directory to a backup folder (c – create, z – compress, v- verbosely listed files being tar'd, use a file)
Uncompress Multiple files	<b>tar -xzvf</b> /backup/allfiles.tgz /vmimages	This would restore all the files in the vmimages directory to a backup (x – Extract, z – uncompress, v- verbosely listed files being tar'd, use a file)folder
Change Ownership	<b>chown</b> lavericm:lavericm instructor.vmdk	Change the Owner of instructor.dsk to be lavericm, also changes user group to be lavericm as well
Change Group Membership	<b>chgrp</b> lavericm instructor.vmdk	Change the Group rights to be the User Group, in this case lavericm

Change Permissions	<b>chmod 774 /home/lavericm -R</b>  <b>chmod 754 /home/lavericm -R</b>	<p>Changing permissions on files – shouldn't be needed very often – as all files are created and owned by root (despite being created with VI Client with a Windows Logon)</p> <p>Change permissions on files using decimals to represent RWX. Used with the -R switch it sets these permission recursively. 77 is short for RWX for the user and group, and 4 for R for others.</p> <p>Although the Numbers are expressed in decimal - they are effectively binary where 1 is Execute, 2 is Write and 4 is R. They are represented not unlike jumper settings on the back of a SCSI device so</p> <table border="0"> <tr><td>000 would mean NOTHING</td><td>0</td></tr> <tr><td>001 would mean X</td><td>1</td></tr> <tr><td>010 would mean W</td><td>2</td></tr> <tr><td>011 would mean WX</td><td>3</td></tr> <tr><td>100 would mean R</td><td>4</td></tr> <tr><td>101 would mean RX</td><td>5</td></tr> <tr><td>110 would mean RW</td><td>6</td></tr> <tr><td>111 would mean RWX</td><td>7</td></tr> </table>	000 would mean NOTHING	0	001 would mean X	1	010 would mean W	2	011 would mean WX	3	100 would mean R	4	101 would mean RX	5	110 would mean RW	6	111 would mean RWX	7
000 would mean NOTHING	0																	
001 would mean X	1																	
010 would mean W	2																	
011 would mean WX	3																	
100 would mean R	4																	
101 would mean RX	5																	
110 would mean RW	6																	
111 would mean RWX	7																	
Check integrity of a file As Above	<b>sum /vmfs/volumes/local-esx1-esx1/instructor.vmdk</b> <b>md5sum /vmfs/local-esx1-esx1/instructor.vmdk</b>	<p>Sum is less good than – md5sum. You need to compare copied file to known good original – usually from a website</p> <p>Most frequently done to check integrity of downloads – checkout WinMD5Sum if you have download to Windows</p>																
Interrogating File format	<b>head /vmfs/volumes/local-esx1/instructor/instructor.vmdk   file</b> - or  <b>head /vmfs/volumes/local-esx1/instructor/instructor-flat.vmdk   file -</b>	<p>VMDK are just extensions. During import/export process vmkfstools -e or -i does not rename. Head examines the contents of the file a report true format</p> <p>VMDK: standard input: x86 boot sector (Monolithic)  VMDK: standard input: ASCII Text (Sparse or Metadata File)  For older VMware products look for COW in the printed string</p>																

To create a ISO file from mounted CD at the Service Console	<b>dd if=/dev/cdrom of=/vmimages/w2ksp4.iso bs=32k</b>	Not recommended, as there is no check done on the integrity of the ISO file <b>WinImage</b> - <a href="http://www.winimage.com">http://www.winimage.com</a>
To List the last ten lines of a log file	<b>tail /var/log/messages</b>	Tail can take parameters to show more or less information – with long files like logs it can be easier to handle than cat

## Module 4: Networking

### Note:

- Networking involves the use of command `esxcfg-vswitch`, `esxcfg-vswif0` and `esxcfg-vmknics`. These are quite involved commands – used in a particular order to achieve the results you are looking for. So I’ve decided to write more of a step-by-step guide than just a command-list with a brief explanation
- `esxcfg-vswitch` is the main command – and it has a mix of parameter in lower and upper-case. Lower-case parameter manipulate the switch, where as upper-case switches manipulate the portgroup. So to add a switch its `-a` and add a upper-case `-A` adds a portgroup
- In one way this is nice... but it’s incredibly easy to create a switch rather than portgroup attached to a switch... and I have done this a few times... ☹

### Viewing your Switches & Service Console Networking

1. To view your switches type the command:

**esxcfg-vswitch -l**

```
Switch Name      Num Ports  Used Ports  Configured Ports  Uplinks
vSwitch0        32         3           32                vmnic0

  PortGroup Name  Internal ID  VLAN ID  Used Ports  Uplinks
  Service Console portgroup0   0        1           vmnic0
```

### Note:

This shows me I have one vSwitch (vSwitch0) using one NIC (vmnic0) with one portgroup called "Service Console" which is not using VLAN.

2. To view your Service Console network settings

**esxcfg-vswif -l**

```
Name      Port Group  IP Address  Netmask  Broadcast  Enabled  DHCP
vswif0    Service Console  192.168.2.102  255.255.255.0  192.168.2.255  true    false
```

### Note:

Nothing to state here – but I think its interesting that it doesn’t show me my all important default gateway settings which would have been nice

3. To View your network card’s vmnic, pci (b:s:f), driver, link, speed, duplex and description:

**esxcfg-nics -l**

```
Name      PCI      Driver  Link Speed  Duplex  Description
vmnic0    01:02.00 e1000   Up          1000Mbps Full  Intel Corporation 82544EI Gigabit Ethernet Controller (Copper)
vmnic1    01:04.00 e1000   Up          1000Mbps Full  Intel Corporation 82544EI Gigabit Ethernet Controller (Copper)
vmnic2    02:08.00 e1000   Up          1000Mbps Full  Intel Corporation 8254NXX Gigabit Ethernet Controller
vmnic3    02:0a.00 e1000   Up          1000Mbps Full  Intel Corporation 82541PI Gigabit Ethernet Controller
```

### Creating a vSwitch (Internal)

1. To create a new switch type:

### **esxcfg-vswitch -a vSwitch1**

2. Then add a portgroup

### **esxcfg-vswitch -A internal vSwitch1**

#### **Note:**

Lower-case -a for adding a switch, and upper-case -A for adding a portgroup

3. If you run the command **esxcfg-vswitch -l** you will see this information

```
Switch Name      Num Ports   Used Ports   Configured Ports   Uplinks
vSwitch0        32          3           32                 vmnic0

  PortGroup Name  Internal ID  VLAN ID   Used Ports   Uplinks
  Service Console portgroup0  0         1           vmnic0

Switch Name      Num Ports   Used Ports   Configured Ports   Uplinks
vSwitch1        64          0           64                 64

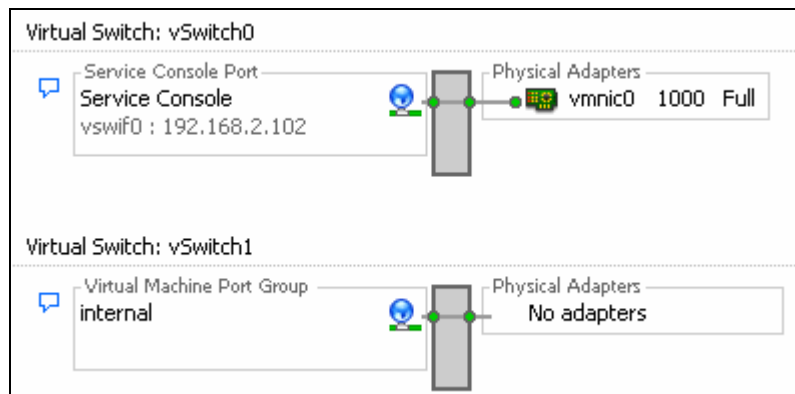
  PortGroup Name  Internal ID  VLAN ID   Used Ports   Uplinks
  internal        portgroup9  0         0
```

4. **If you wish to see this reflected in the VI Client** then type:

### **service mgmt-vmware restart**

#### **Note:**

This is how my switch configuration looks now



## ***Creating a vSwitch (Single NIC)***

1. In this case we will create a new switch and port group in one line with:

### **esxcfg-vswitch -a vSwitch2 -A production**

2. Next patch a NIC to the vSwitch with

### **esxcfg-vswitch -L vmnic1 vSwitch2**

#### **Note:**

Again this is a case-sensitive option. -l lists switches, whereas -L links



nic's to switches

3. If you run the command **esxcfg-vswitch -l** you will see this information

```
Switch Name      Num Ports  Used Ports  Configured Ports  Uplinks
vSwitch0        32         3          32               vmnic0

  PortGroup Name  Internal ID  VLAN ID  Used Ports  Uplinks
  Service Console portgroup0  0        1          vmnic0

Switch Name      Num Ports  Used Ports  Configured Ports  Uplinks
vSwitch1        64         0          64              

  PortGroup Name  Internal ID  VLAN ID  Used Ports  Uplinks
  internal        portgroup9  0        0          

Switch Name      Num Ports  Used Ports  Configured Ports  Uplinks
vSwitch2        64         2          64               vmnic1

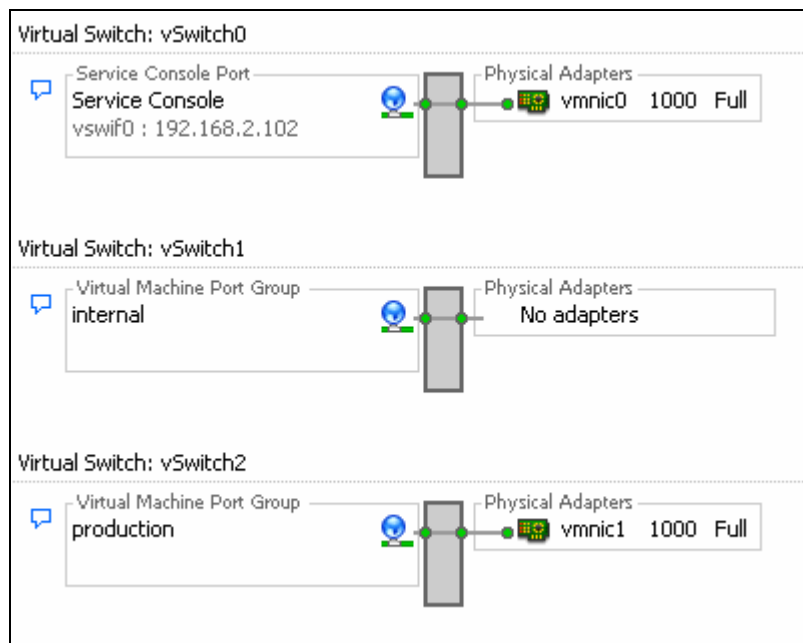
  PortGroup Name  Internal ID  VLAN ID  Used Ports  Uplinks
  production     portgroup10 0        0           vmnic1
```

4. If you wish to see this reflected in the VI Client then type:

**service mgmt-vmware restart**

**Note:**

This is how my switch configuration looks now



## ***Creating a vSwitch (Multiple NIC's)***

**Note:**

- It is very easy to create a NIC-Team just re-run the previous command, with a different NIC with

1. **esxcfg-vswitch -L vmnic2 vSwitch1**
2. If you run the command **esxcfg-vswitch -l** you will see this information

```

Switch Name      Num Ports  Used Ports  Configured Ports  Uplinks
vSwitch0        32         3           32                vmnic0

  PortGroup Name  Internal ID  VLAN ID  Used Ports  Uplinks
  Service Console portgroup0  0         1          vmnic0

Switch Name      Num Ports  Used Ports  Configured Ports  Uplinks
vSwitch1        64         0           64                vmnic0

  PortGroup Name  Internal ID  VLAN ID  Used Ports  Uplinks
  internal       portgroup9  0         0

Switch Name      Num Ports  Used Ports  Configured Ports  Uplinks
vSwitch2        64         3           64                vmnic2,vmnic1

  PortGroup Name  Internal ID  VLAN ID  Used Ports  Uplinks
  production     portgroup10 0         0          vmnic1,vmnic2

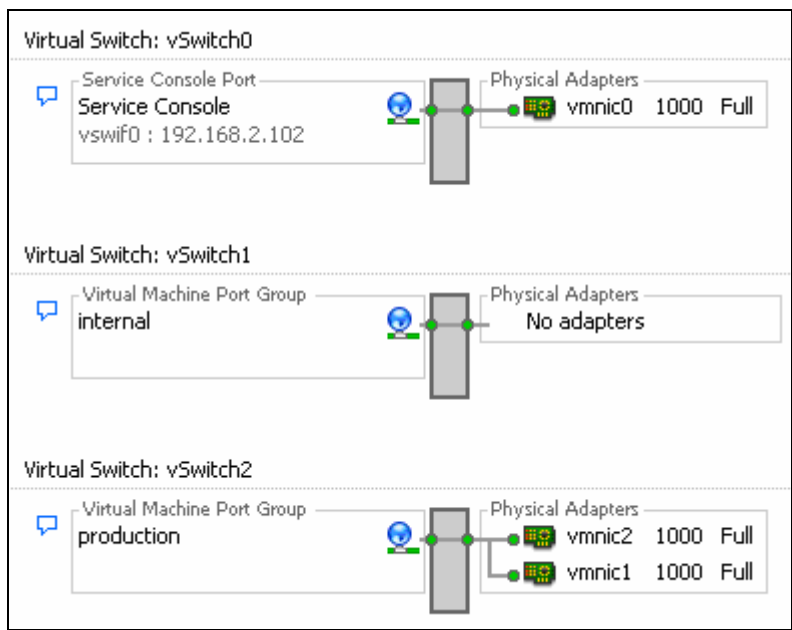
```

3. **If you wish to see this reflected in the VI Client** then type:

**service mgmt-vmware restart**

**Note:**

This is how my switch configuration looks now



### ***Deleting a Switch from vSwitch***

**Note:**

I'm now running out NICS for the next part of this guide. So I am going to blow away my vSwitch2 to free up my NIC's...

1. Type the command

**esxcfg-vswitch -d vswitch2**

**Note:**

Notice how it doesn't ask – are you sure, but then again neither does the VI client.

## **Creating PortGroups for VLAN Networking NIC Team**

1. We have to create a switch, a portgroup for each VLAN, allocate our NIC's and then set the VLAN ID

```
esxcfg-vswitch -a vSwitch2
esxcfg-vswitch -A accounts vSwitch2
esxcfg-vswitch -A rnd vSwitch2
esxcfg-vswitch -A sales vSwitch2
esxcfg-vswitch -L vmnic1 vSwitch2
esxcfg-vswitch -L vmnic2 vSwitch2
```

2. **The next part is to set the VLAN id** for each network (account, rnd and sales)

```
esxcfg-vswitch -v 10 -p accounts vSwitch2
esxcfg-vswitch -v 20 -p rnd vSwitch2
esxcfg-vswitch -v 30 -p sales vSwitch2
```

3. **If you run the command esxcfg-vswitch -l** you will see this information

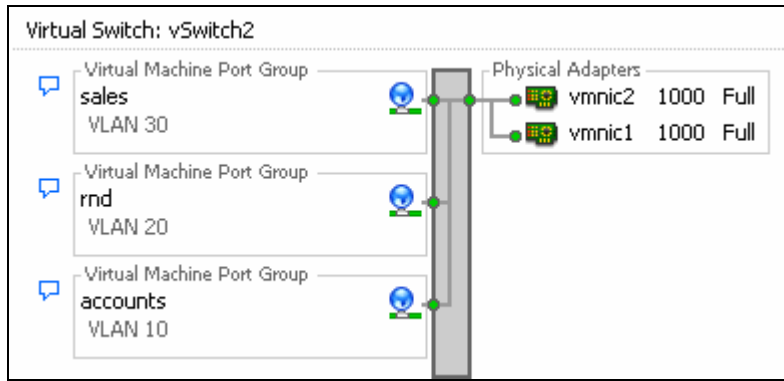
Switch Name	Num Ports	Used Ports	Configured Ports	Uplinks
vSwitch0	32	3	32	vmnic0
PortGroup Name	Internal ID	VLAN ID	Used Ports	Uplinks
Service Console	portgroup0	0	1	vmnic0
Switch Name	Num Ports	Used Ports	Configured Ports	Uplinks
vSwitch1	64	0	64	
PortGroup Name	Internal ID	VLAN ID	Used Ports	Uplinks
internal	portgroup9	0	0	
Switch Name	Num Ports	Used Ports	Configured Ports	Uplinks
vSwitch2	64	3	64	vmnic2,vmnic1
PortGroup Name	Internal ID	VLAN ID	Used Ports	Uplinks
sales	portgroup23	30	0	vmnic1,vmnic2
rnd	portgroup22	20	0	vmnic1,vmnic2
accounts	portgroup21	10	0	vmnic1,vmnic2

4. **If you wish to see this reflected in the VI Client** then type:

```
service mgmt-vmware restart
```

**Note:**

This is how my switch configuration looks now



## Creating Vmkernel Switches

### Note:

- This does seem a bit limited...
- I can't see away of enabling the VMKernel switch for Vmotion
- Nor can I see how I would set the default gateway for the switch either
- You need a good name for the portgroup – because the esxcfg-vswitch command just shows as any ordinary switch...
- You need to use **esxcfg-vmknic -l** to see it

Port Group	IP Address	Netmask	Broadcast	MAC Address	MTU	Enabled
VM Kernel	192.168.2.202	255.255.255.0	192.168.2.255	00:50:56:66:ea:3a	1514	true

1. First create the Switch, Portgroup and Assign a NIC

```
esxcfg-vswitch -a vSwitch3
esxcfg-vswitch -A "VM Kernel" vSwitch3
esxcfg-vswitch -L vmnic3 vSwitch3
```

2. Next use the **esxcfg-vmknic** command to add in a **VM Kernel NIC** and set the **IP** and **Subnet Mask**

```
esxcfg-vmknic -a "VM Kernel" -i 192.168.2.202 -n 255.255.255.0
```

3. Then set the **vmkernel default gateway** with

```
esxcfg-route 192.168.2.1
```

4. If you run the command **esxcfg-vswitch -l** you will see this information

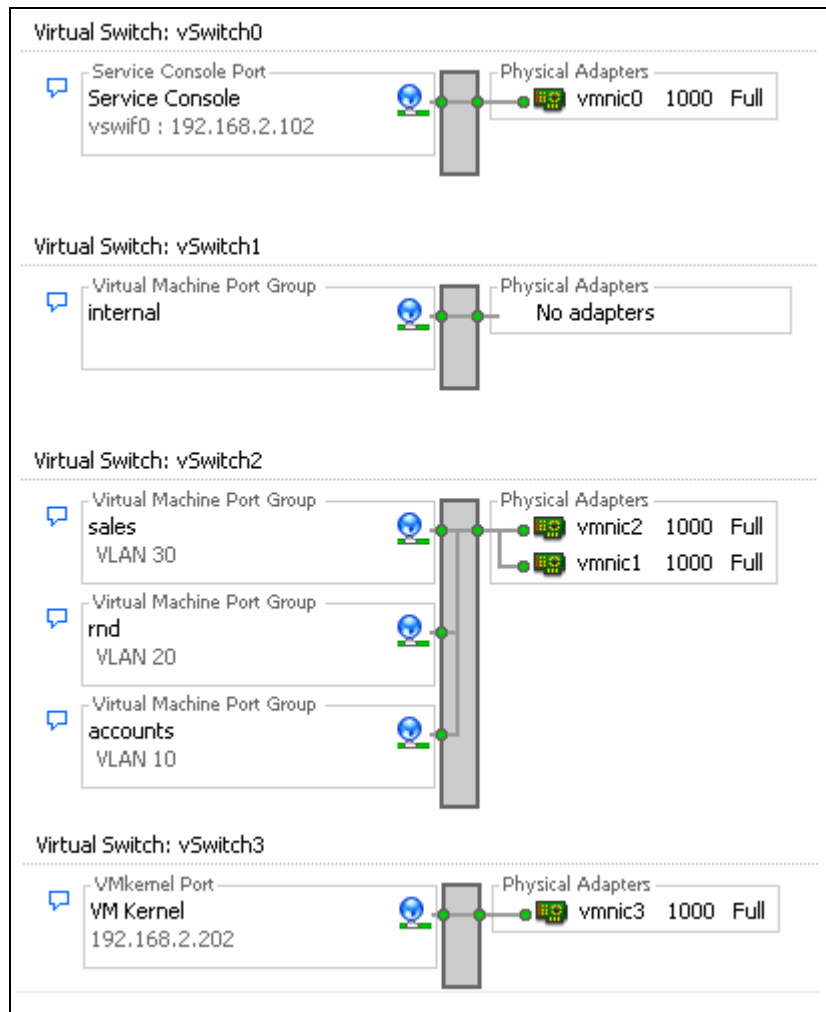
Switch Name	Num Ports	Used Ports	Configured Ports	Uplinks
vSwitch0	32	3	32	vmnic0
PortGroup Name	Internal ID	VLAN ID	Used Ports	Uplinks
Service Console	portgroup0	0	1	vmnic0
Switch Name	Num Ports	Used Ports	Configured Ports	Uplinks
vSwitch1	64	0	64	
PortGroup Name	Internal ID	VLAN ID	Used Ports	Uplinks
internal	portgroup9	0	0	
Switch Name	Num Ports	Used Ports	Configured Ports	Uplinks
vSwitch2	64	3	64	vmnic2,vmnic1
PortGroup Name	Internal ID	VLAN ID	Used Ports	Uplinks
sales	portgroup23	30	0	vmnic1,vmnic2
rnd	portgroup22	20	0	vmnic1,vmnic2
accounts	portgroup21	10	0	vmnic1,vmnic2
Switch Name	Num Ports	Used Ports	Configured Ports	Uplinks
vSwitch3	64	3	64	vmnic3
PortGroup Name	Internal ID	VLAN ID	Used Ports	Uplinks
VM Kernel	portgroup25	0	1	vmnic3

**If you wish to see this reflected in the VI Client then type:**

**service mgmt-vmware restart**

**Note:**

This is how my switch configuration looks now



## Changing your Service Console IP Settings

### Note:

- You can do this through a putty session but clearly unless you have more than one vswif interface (with a different IP address) you will get disconnected
- It is entirely possible – to have two vswif interface on two separate IP address – and use one connection to change the IP address of the other
- You do run the risk of losing SSH connectivity if you screw up – so perhaps doing this through your ILO is a safer-bet and less hard work
- You might also wish to include a new DNS entry for your ESX host before you make this change...

1. **To view your current IP and Netmask type:**

**esxcfg-vswif -l**

Name	Port Group	IP Address	Netmask	Broadcast	Enabled	DHCP
vswif0	Service Console	192.168.2.102	255.255.255.0	192.168.2.255	true	false

2. **To change your IP and Subnet Mask type:**

**esxcfg-vswif -i 192.168.2.203 -n 255.255.255.0 vswif0**

**Note:**

You change your default gateway by editing **nano -w /etc/sysconfig/network** and then restart your networking with **services network restart**

Your DNS settings are located in **nano -w /etc/resolv.conf**

## Setting the Speed and Duplex of NIC's

**Note:**

- We have always been able to change the speed & duplex of the vmkernel NIC's (ESX 2.x MUI) but in the past the only way to change the Service Console speed and duplex was by editing the /etc/modules.conf file
  - You can now change both the Service Console and the vmkernel nics through the GUI
  - There could be a chicken-egg/catch22 situation here though. If your Service Console NIC mis-neg's its speed/duplex you maybe unable to connect with the VI client to change it. That's when knowing the Service Console commands come in handy. 😊
  - This can also be useful to reassign a 100Mps card to a Service Console and then fix its speed/duplex
1. Workout which NIC has been assigned to the portgroup "Service Console" with

**esxcfg-vswitch -l**

```
Switch Name      Num Ports   Used Ports   Configured Ports   Uplinks
vSwitch0        32          3            32                 vmnic0

  PortGroup Name      Internal ID   VLAN ID   Used Ports   Uplinks
  Service Console    portgroup0   0         1            vmnic0
```

2. To view you current speed & duplex:

**esxcfg-nics -l**

```
vmnic0 01:02.00 e1000 Up 1000Mbps Full Intel Corporation 82544EI Gigabit Ethernet Controller (Copper)
vmnic1 01:04.00 e1000 Up 1000Mbps Full Intel Corporation 82544EI Gigabit Ethernet Controller (Copper)
vmnic2 02:08.00 e1000 Up 1000Mbps Full Intel Corporation 8254NXX Gigabit Ethernet Controller
vmnic3 02:0a.00 e1000 Up 1000Mbps Full Intel Corporation 82541PI Gigabit Ethernet Controller
```

3. To set the speed/duplex of vmnic0 to 100Mps/Half-Duplex type:

**esxcfg-nics -s 100 -d half vmnic0**

4. To reset to auto-neg

**esxcfg-nics -a vmnic0**

## Recreating your vswif0 Interface

**Note:**

- OK, suppose something horrible goes wrong and you loose your vswif0 interface – this is how you would recreate it...

1. **Logon locally to the ESX host or use your ILO card** (do you have choice!)
2. Create a new switch

**esxcfg-vswitch -a vSwitch0**

3. Create a new portgroup

**esxcfg-vswitch -p "Service Console" vSwitch0**

4. Assign a NIC

**esxcfg-vswitch -L vmnic0 vSwitch0**

5. Assign a vswif interface and set its ip/sn:

**esxcfg-vswif -a vswif0 -p "Service Console" i 192.168.2.102 -n 255.255.255.0**

### ***Removing a NIC from vSwitch***

- **esxcfg-vswitch -U vmnic vSwitch2**

### ***Deleting a PortGroup from vSwitch***

- **esxcfg-vswitch -D production vSwitch**

### ***Managing the ESX Firewall***

#### **Note:**

- Managing the firewall by the VI Client is real easy – it’s a tick of box style interface. If you use the GUI interface and then query with the command-line tool you get “friendly” information about what enabled. If you purely use the command-line tool you will just get TCP port numbers and directions (outgoing and incoming)
- They GUI tool also has some handy “built-in” friendly name for popular applications vendor specific agents like CommVault Dynamic/Static and many others
- By default 902 (VirtualCenter, uses TCP and UDP!), 80 (Web Access welcome page), 443 (Web Access Login page) and 22 (SSH) are enabled by default
- In additions there are some ports enabled for EMC AAM (Automated Availability Manager) and CIM (Common Information Model) developed by the DMTF, is a very broad approach to the management of systems and networks
- Other Ports worth knowing about:

903	Console sessions from Workstations to VM's
2049	NFS Storage (VM Kernel interfaces only)
2050-5000 8042-8045	HA and EMC Autostart Manager (Between ESX Hosts)
3260	iSCSI (VM Kernel interfaces only)
8000	VMotion Requests (VM Kernel Interfaces only)
2700	License Server communications



- A bit like in ESX 2.x it is possible to set 3-level of security (high, medium and low)
  - High incoming/outgoing blocked
  - Medium incoming blocked, outgoing not blocked
  - Low firewall off, no blocking of incoming/outgoing traffic

### Viewing your Firewall Settings

1. Type the command

```
esxcfg-firewall -q outgoing  
esxcfg-firewall -q incoming
```

**Note:**

You can also use esxcfg-firewall -q on its own. This gives you lots of stuff... most useful at the bottom:

```
Incoming and outgoing ports blocked by default.  
Enabled services: AAMClient CIMSLP LicenseClient sshServer CIMHttpsServer CIMHttpServer vpxHeartbeats  
Opened ports:
```

### Changing Your Security Level

Note:

If you wanted to weaken your security to medium you could use

1. Type the command

```
esxcfg-firewall --allowOutgoing -blockIncoming
```

**Note:**

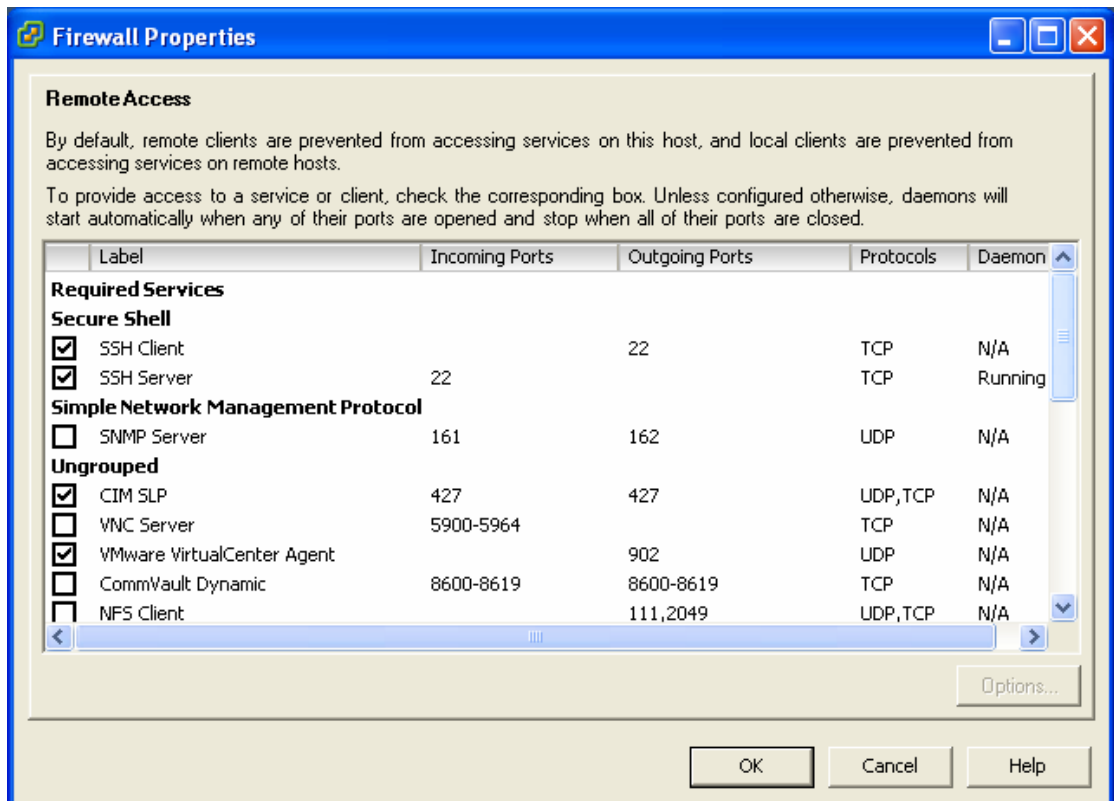
You should get warnings like so:

```
2006-07-06 14:39:53 (1965) WARN : Setting firewall default  
/firewall/blockOutgoing to 0  
2006-07-06 14:39:53 (1965) WARN : Setting firewall default  
/firewall/blockIncoming to 1
```

### Enabling a Single Service/Client/Agent

**Note:**

- If you want to SSH from an ESX host to another ESX host or SCP from an ESX host to another – you need to enable the SSH Client on port 22
- If you do this via the GUI and then do a esxcfg-firewall -q you will see friendly information like so:



Incoming and outgoing ports blocked by default.

Enabled services: AAMClient CIMSLP LicenseClient sshServer  
CIMHttpsServer CIMHttpServer **sshClient** vpxHeartbeats

Opened ports:

- To do the same from the command-line you would type the command:

**esxcfg-firewall -e sshClient**

**Note:**

To disable esxcfg-firewall -d sshClient

### Enabling non-Standard Ports

**Note:**

- Perhaps there is an application or service which is not listed for use with -e or -d
- Or the port numbers have been changed to non-standard ports
- It is possible to open a specific port by number, transport (udp/tcp) and direction (in/out)
- To enable port 22 outbound from the server type the command:

**esxcfg-firewall -o 22,tcp,out,ssh**

**Note:**

The ssh at the end is a friendly label. If I run esxcfg-firewall -q again at the bottom it states:

Incoming and outgoing ports blocked by default.

Enabled services: AAMClient CIMSLP LicenseClient sshServer

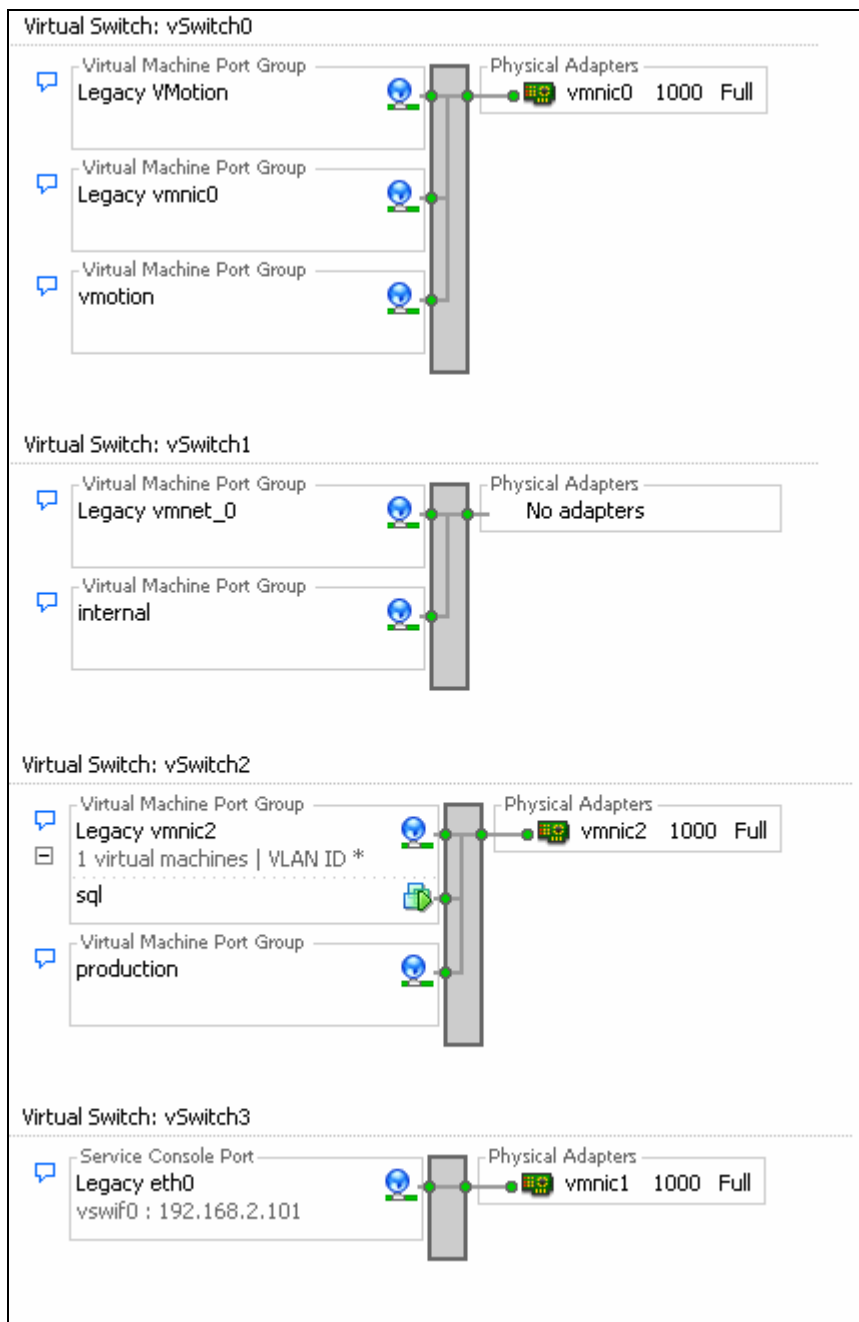
CIMHttpsServer CIMHttpServer vpxHeartbeats

Opened ports:  
ssh : port 22 tcp.out

## Fixing ESX Networking after an Upgrade

### Note:

- If you have read my upgrade guide to ESX 3.x you will know that networking looks significantly different after an upgrade – rather than a clean install
- The following shows how to “clean up” networking using the Service Console commands
- This is how my upgraded ESX server networking looks like



- So what am I going to do with this?

- What I am going to do is the following:
  - Remove the VMotion Switch0 altogether thus freeing up vmnic0
  - Remove the "Legacy vmnet\_0" portgroup from vSwitch1
  - Remove the "Legacy vmnic2" portgroup from vSwitch2
  - Remove vSwitch3
  - Create a new vswif interface on a new switch called vSwitch0 using vmnic0
  - Add vmnic1 to vSwitch2
  - Patch all my VM's to the Production on vSwitch2
  - Create with a VM Kernel Port (vSwitch3)

### Removing VMotion Switch0

- **esxcfg-vswitch -d vSwitch0**

### Remove "Legacy vmnet\_0" portgroup from vSwitch1

- **esxcfg-vswitch -D "Legacy vmnet\_0" vSwitch1**

### Remove "Legacy vmnic2" portgroup from vSwitch2

- **esxcfg-vswitch -D "Legacy vmnic2" vSwitch2**

### Remove Old vswif Interface and Remove vSwitch3

- Disable vswif0 interface, and remove from the system
- You must disable the vswif interface before you delete the switch it is attached to...
- Do this at the physical console via an ILO
- **esxcfg-vswif -d vswif0**
- **esxcfg-vswitch -d vSwitch3**

### Create a new vswif1 interface

- **esxcfg-vswitch -a vSwitch0**
- **esxcfg-vswitch -A "Service Console" vSwitch0**
- **esxcfg-vswitch -L vmnic0 vSwitch0**
- **esxcfg-vswif -a vswif0 -p "Service Console" -i 192.168.2.x -n 255.255.255.0**

#### Note:

You will get this message

```
[Date'Vnic' warning] Generated New MAC Address, 00:50:56:XX:XX:XX for
vswif0
Nothing to flush
```

#### Note:

I found my new vswif0 interface was already enabled. But if doesn't for you might like to try **esxcfg-vswif -d vswif0**

### Create a VM Kernel Port

- **esxcfg-vswitch -a vSwitch3**
- **esxcfg-vswitch -A "VM Kernel" vSwitch3**
- **esxcfg-vswitch -L vmnic3 vSwitch3**
- **esxcfg-vmknic -a "VM Kernel" -i 192.168.2.x -n 255.255.255.0**
- **esxcfg-route 192.168.2.1**

## ***Typical vSwitch Errors***

- **Getting your case-sensitivity muddled up!**

- -a add vSwitch
- -A add portgroup
- -l list vSwitches
- -L link vmnic
- -d delete vSwitch
- -D delete portgroup

- **Trying to change things that are in use**

"Failed to remove vswitch: vSwitch3, Error: PortGroup "Legacy eth0" on VirtualSwitch "vSwitch3" is still in use: 1 active ports, vswif0"

or

"Legacy vmnic2, Error: Unable to delete portgroup "Legacy vmnic2", for the following reasons: 1 active ports"

# Module 5: Configuring Physical Storage

## Note:

- This guide assumes you already have set-up NAS and iSCSI storage correctly
- **and** that you have created a VM Kernel vSwitch/Port Group outlined in the previous module
- ...and now you wish to connect to it
- This module repeats some of the vmkfstools command listed earlier – sorry about this repetition – but this gives more detail than simple explanation of the commands

## Configuring SAN Storage

### Note:

- I don't have a SAN ☹
- Someone give me an MSA1000 to play with? ;-)
- I promise to write some excellent documentation!!! ☺

## Configuring NAS Storage

### Note:

- Remember you must have a VM Kernel Switch to do this... and as authentication is driven through the Service Console NIC – there must be connectivity to the NAS via the Service Console NIC
- My NFS server is called nfs1.rtfm-ed.co.uk
- It has 3 exports on it /isos /templates/ /nas-vms an

### Mounting NAS Exports/Shares

1. Confirm you can ping your nfs1 by its name or ip address
2. Type the command:

```
esxcfg-nas -a isos -o nfs1.rtfm-ed.co.uk -s isos
```

### Note:

**The command should respond with:**

```
Connecting to NAS volume: isos  
isos created and connected.
```

### Note:

-a to add followed by the friendly label, -o to specify the nfs server, and -s to specify the export/share

I then continued to add in /template and nas-vms like so:

```
esxcfg-nas -a templates -o nfs1.rtfm-ed.co.uk -s templates  
esxcfg-nas -a nas-vm -o nfs1.rtfm-ed.co.uk -s nas-vm
```

### Listing NAS Exports/Shares

1. Type the command:

```
esxcfg-nas -l
```

### Note:

You should receive a response like so:

```
isos is isos from nfs1.rtfm-ed.co.uk mounted
nas-vms is nas-vms from nfs1.rtfm-ed.co.uk mounted
templates is templates from nfs1.rtfm-ed.co.uk mounted
```

## Restoring NAS Connections

### Note:

- If the NAS become unavailable you may have to force a restore the mount
- This sometimes happens to me if start-up my ESX hosts before the NAS is up

1. Type the command

```
esxcfg-nas -r
```

2. Followed by

```
esxcfg-nas -l
```

## Connecting to iSCSI Storage (Software)

### Note:

- My iSCSI box is Fedora Core 5 with iSCSI Enterprise Target installed and configured, its name is iscs1.rtfm-ed.co.uk
- iSCSI has 2 LUN's available, it supports dynamic discovery and its IP is 192.168.2.210.
- Use the iqn of iqn.2006-06.uk.co.rtfm-ed:storage.lvm on this iSCSI Target
- The esxcfg-swiscsi is more limited than the esxcfg-nas command. Although you can do many things – you cannot configure the IP address and port number of the iSCSI server
  - -e enable
  - -d disable
  - -q query if the adapter is enabled or disabled
  - -s force a scan
  - -k forcibly remove iscsi sw stack

## Enabling the iSCSI Adapter

1. Type the command:

```
esxcfg-swiscsi -e
```

```
Allowing software iSCSI traffic through firewall...
Enabling software iSCSI...
/usr/sbin/vmkload_mod /usr/lib/vmware/vmkmod/iscsi_mod.o
Using /usr/lib/vmware/vmkmod/iscsi_mod.o
Module load of iscsi_mod succeeded.
```

### Note:

This will enable the iSCSI Software Adapter like so:

Device	Type	SAN Identifier	
<b>iSCSI Software Adapter</b>			
vmhba40	iSCSI	iqn.1998-01.com.vmware:...	
<b>AIC-7899P U160/m</b>			
vmhba0	SCSI		
vmhba1	SCSI		

[Rescan...](#)

<b>Details</b>			
<b>vmhba40</b>			
Model:	iSCSI Software Adapter	IP Address:	<a href="#">Properties...</a>
iSCSI Name:	iqn.1998-01.com.vmware:esx1-51309f65	Discovery Methods:	Send Targets
iSCSI Alias:	esx1.rtfm-ed.co.uk	Targets:	0

## Setting the iSCSI Target IP with Discovery Mode

### Note:

Remember the Software iSCSI adapter does NOT support static discovery

1. Type the command:

```
vmkiscsi-tool -D -a 192.168.2.210 vmhba40
```

### Note:

-D sets the mode of Discovery, rather than static. -a is used to add an iSCSI Target, followed by IP address of iSCSI Device and the HBA that will be used. vmkiscsi-tool is a powerful little utility with lots of options – although documentation is a little bit thin on the ground. I intend to return to this section when I have more time to put some more flesh around it...

2. To list the targets configured use:

```
vmkiscsi-tool -l -T vmhba40
```

### Note:

This should report:

```
-----
NAME                               : iqn.2006-06.uk.co.rtfm-ed:storage.lvm
ALIAS                               :
DISCOVERY METHOD FLAGS              : 0
SEND TARGETS DISCOVERY SETTABLE   : 0
SEND TARGETS DISCOVERY ENABLED    : 0
Portal 0                           : 192.168.2.210:3260
-----
```

## Forcing a Rescan of the iSCSI Adapter

1. You might wish to list the current LUNs/Disk currently visible with

```
ls -l /vmfs/devices/disks
```

```
total 429569804
-rwx----- 1 root root          50 Jul 1 21:21 vmhba0:0:0:0 -> vml_01000000005546413350323330375442444d414e333336
-rwx----- 1 root root          50 Jul 1 21:21 vmhba0:1:0:0 -> vml_01000000005546413350323330375352544d414e333336
-rwx----- 1 root root          66 Jul 1 21:21 vmhba1:2:0:0 -> vml_0100000000334345303634504b202020203731303545464853535831373334
-rwx----- 1 root root          66 Jul 1 21:21 vmhba1:3:0:0 -> vml_01000000003343453035595041202020203731303638315430535831373334
-rwx----- 1 root root          66 Jul 1 21:21 vmhba1:4:0:0 -> vml_0100000000334345304446543230303030373132314e4c5a57535431373334
-rwx----- 1 root root          66 Jul 1 21:21 vmhba1:5:0:0 -> vml_0100000000334345304150415420202020373131384b51305a535831373334
-rwx----- 1 root root          66 Jul 1 21:21 vmhba1:6:0:0 -> vml_0100000000334345304244584e202020203731313835485231535831373334
-rwx----- 1 root root          73407865856 Jul 1 21:21 vml_01000000003343453035595041202020203731303638315430535831373334
-rwx----- 1 root root          73407865856 Jul 1 21:21 vml_0100000000334345303634504b202020203731303545464853535831373334
-rwx----- 1 root root          73407865856 Jul 1 21:21 vml_0100000000334345304150415420202020373131384b51305a535831373334
-rwx----- 1 root root          73407865856 Jul 1 21:21 vml_0100000000334345304244584e202020203731313835485231535831373334
-rwx----- 1 root root          73407865856 Jul 1 21:21 vml_0100000000334345304446543230303030373132314e4c5a57535431373334
-rwx----- 1 root root          36420075008 Jul 1 21:21 vml_01000000005546413350323330375352544d414e333336
-rwx----- 1 root root          36420075008 Jul 1 21:21 vml_01000000005546413350323330375442444d414e333336
```



**Note:**

I have two internal disks on an internal SCSI Controller (vmhba0:0 and vmhba0:1)

I also a have JBOD with six disks inside (vmhba1:1-6)

2. **Force the rescan** with

**esxcfg-swiscsi -s**

**Note:**

This should produce this result

```
Scanning vmhba40...
Rescanning vmhba40...done.
On scsi2, removing:
On scsi2, adding: 0:0 0:1.
```

3. If we run the ls command again we can see the 20GB LUN's from my iSCSI box

```
total 471512844
-kwx----- 1 root root 50 Jul 1 21:27 vmhba0:0:0:0 -> vml.01000000005546413350323330375442444d414e333336
-kwx----- 1 root root 50 Jul 1 21:27 vmhba0:1:0:0 -> vml.01000000005546413350323330375352544d414e333336
-kwx----- 1 root root 66 Jul 1 21:27 vmhba1:2:0:0 -> vml.0100000000334345303634504b202020203731303545464853535831373334
-kwx----- 1 root root 66 Jul 1 21:27 vmhba1:3:0:0 -> vml.01000000003343453035595041202020203731303638315430535831373334
-kwx----- 1 root root 66 Jul 1 21:27 vmhba1:4:0:0 -> vml.01000000003343453044465432303030373132314e4c5a57535431373334
-kwx----- 1 root root 66 Jul 1 21:27 vmhba1:5:0:0 -> vml.0100000000334345304150415420202020373131384b51305a535831373334
-kwx----- 1 root root 66 Jul 1 21:27 vmhba1:6:0:0 -> vml.0100000000334345304244584e202020203731313835485231535831373334
-kwx----- 1 root root 34 Jul 1 21:27 vmhba40:0:0:0 -> vml.010000000020202020564952545541
-kwx----- 1 root root 34 Jul 1 21:27 vmhba40:0:1:0 -> vml.010001000020202020564952545541
-kwx----- 1 root root 21474836480 Jul 1 21:27 vml.010000000020202020564952545541
-kwx----- 1 root root 73407865856 Jul 1 21:27 vml.01000000003343453035595041202020203731303638315430535831373334
-kwx----- 1 root root 73407865856 Jul 1 21:27 vml.0100000000334345303634504b202020203731303545464853535831373334
-kwx----- 1 root root 73407865856 Jul 1 21:27 vml.0100000000334345304150415420202020373131384b51305a535831373334
-kwx----- 1 root root 73407865856 Jul 1 21:27 vml.0100000000334345304244584e202020203731313835485231535831373334
-kwx----- 1 root root 73407865856 Jul 1 21:27 vml.01000000003343453044465432303030373132314e4c5a57535431373334
-kwx----- 1 root root 36420075008 Jul 1 21:27 vml.01000000005546413350323330375352544d414e333336
-kwx----- 1 root root 36420075008 Jul 1 21:27 vml.01000000005546413350323330375442444d414e333336
-kwx----- 1 root root 21474836480 Jul 1 21:27 vml.010001000020202020564952545541
```

## Creating a VMFS Volume

### Using FDISK to Create Partition

**Note:**

- Generally we create one VMFS partition per disk/LUN

1. **Logon to the Service Console as ROOT** type:

**fdisk /vmfs/device/disks/vmhba40:0:0:0**

**Note:**

Under ESX 2.x we would have need to know how Linux address this disk/LUN with /dev/sd? Syntax. This is no longer required.

2. Type **N**, to create a **new partition**
3. Type **P**, for a **Primary partition**
4. Choose **1** for the **Partition Number**
5. **Accept the defaults** for **First Cylinder** and **Last Cylinder**

**Note:**

This creates a partition which uses all of the free space on the disk adjust the last cylinder value to adjust size of partition - also fdisk defaults to being a Linux type which in this case is not good for our VMFS partition

6. Type **P**, to **print out the partition table** (make a mental note of the partition number here)
7. Type **T**, to change the **File System Type**
8. **Type the Hex Code of fb:** (the code for a VMFS partition)

**Note:**

This should give the result of

```
Changed system type of partition 1 to fb (Unknown)
```

Hex codes tell the system what file system partition will support – 07 NTFS, 82 Linux Swap, 83 Linux File System, FB for VMFS and FC for VMware Core Dump. At this stage the VI Client would identify the partition as VMFS unformatted.

9. Type **W**, to **write your partition table changes to the hard drive** – it will give you this status information:

**Note:**

```
The partition table has been altered!
```

```
Calling ioctl() to re-read partition table.
Syncing disks.
```

**Note:**

If you re-run `fdisk /vmfs/device/disks/vmhba40:0:0:0` and choose P to display the partition table

```
Disk /vmfs/devices/disks/vmhba40:0:0:0: 21.4 GB, 21474836480 bytes
255 heads, 63 sectors/track, 2610 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

           Device Boot      Start         End      Blocks   Id  System
/vmfs/devices/disks/vmhba40:0:0:0p1    1           2610     20964793+  fb  Unknown
```

### Format & Label the VMFS Partition

1. **Logon to the Service Console as ROOT**
2. To **format the new partition with the VMFS file system type:**

```
vmkfstools -C vmfs2 vmhbaA:T:L:V
```

In my case the hard disk partition is located on the Adapter 40, TARGET 0, LUN 0 on Volume/Partition 1.

So I would type:

```
vmkfstools -C vmfs3 -S iscsi-lun0
/vmfs/devices/disks/vmhba40:0:0:1
```

**Warning:**

Please note it is an UPPERCASE C you type here. A lowercase c creates a vmdk file. You can specify `-b` flag to set a block-size. You would need to do this if you thought any one of your virtual disks were going to be greater than 256GB in size.

From the PDF Guide "1MB, 2MB, 4MB, or 8MB. When entering a size, indicate the unit type by adding a suffix of m or M. The unit type is not case sensitive. vmkfstools interprets either m or M to mean megabytes"

2m allows 512GB max file size  
 4m allows 1024GB max file size  
 8m allows 2048GB max file size

**Note:**

This should give the result like so:

```
Creating file system on "vmhba40:0:0:1" with blockSize 1048576 and
volume label "none".
Successfully created new volume: 44a7bcf0-8b87cb86-9403-
00065bec0eb6
```

**Note:**

I repeated this for my other iSCSI LUN 1

**fdisk /vmfs/devices/disks/vmhba40:0:1:0**








**vmkfstools -C vmfs3 -S iscsi-lun1  
 /vmfs/devices/disks/vmhba40:0:1:1**

**Note:**

If we re-run ls -l /vmfs/volumes will see this information

```
total 4098
drwxrwxrwx 1 4294967294 4294967294      64 Jun 28 09:50 2cf78959-22cd65c7
drwxrwxrwt 1 root      root          1960 Jun 30 16:49 44a38c72-156b2590-be15-00065bec0eb7
drwxrwxrwt 1 root      root          1120 Jul  1 19:24 44a6c956-66056236-c671-00065bec0eb6
drwxrwxrwt 1 root      root           980 Jul  2 12:38 44a7be3d-ab15e6a9-4abe-00065bec0eb6
drwxrwxrwt 1 root      root           980 Jul  2 12:46 44a7c03d-159649b5-0e6a-00065bec0eb6
drwx----- 1 4294967294 4294967294      64 Jun 28 21:34 554f516c-ba779540
drwx----- 1 4294967294 4294967294      64 Jun 28 20:05 a857137a-a6f3c023
lrwxrwxrwx 1 root      root           35 Jul  2 12:49 iscsi-lun0 -> 44a7be3d-ab15e6a9-4abe-00065bec0eb6
lrwxrwxrwx 1 root      root           35 Jul  2 12:49 iscsi-lun1 -> 44a7c03d-159649b5-0e6a-00065bec0eb6
lrwxrwxrwx 1 root      root           17 Jul  2 12:49 isos -> 2cf78959-22cd65c7
lrwxrwxrwx 1 root      root           35 Jul  2 12:49 local-esx1 -> 44a38c72-156b2590-be15-00065bec0eb7
lrwxrwxrwx 1 root      root           35 Jul  2 12:49 local2-esx1 -> 44a6c956-66056236-c671-00065bec0eb6
lrwxrwxrwx 1 root      root           17 Jul  2 12:49 nas-vms -> 554f516c-ba779540
lrwxrwxrwx 1 root      root           17 Jul  2 12:49 templates -> a857137a-a6f3c023
```

and in the VI client...

Storage					Refresh
Identification	Device	Capacity	Free	Type	
 iscsi-lun0	vmhba40:0:0:1	19.75 GB	19.14 GB	vmfs3	
 iscsi-lun1	vmhba40:0:1:1	19.75 GB	19.14 GB	vmfs3	
 isos	nfs1.rtfm-ed.co.uk:isos	74.53 GB	19.60 GB	nfs	
 local2-esx1	vmhba1:4:0:1	68.25 GB	21.39 GB	vmfs3	
 local-esx1	vmhba0:1:0:1	33.75 GB	4.43 GB	vmfs3	
 nas-vms	nfs1.rtfm-ed.co.uk:nas-vms	74.53 GB	19.60 GB	nfs	
 templates	nfs1.rtfm-ed.co.uk:templates	74.53 GB	19.60 GB	nfs	

## Changing the Volume Label

**Note:**

To do this you need to know the UUID of your VMFS volume

1. Type the command

**ls -l /vmfs/volumes**

```

total 4098
drwxrwxrwx 1 4294967294 4294967294      64 Jun 28 09:50 2cf78959-22cd65c7
drwxrwxrwt 1 root      root      1960 Jun 30 16:49 44a38c72-156b2590-be15-00065bec0eb7
drwxrwxrwt 1 root      root      1120 Jul  1 19:24 44a6c956-66056236-c671-00065bec0eb6
drwxrwxrwt 1 root      root      980 Jul  2 12:38 44a7be3d-ab15e6a9-4abe-00065bec0eb6
drwxrwxrwt 1 root      root      980 Jul  2 12:46 44a7c03d-159649b5-0e6a-00065bec0eb6
drwx----- 1 4294967294 4294967294      64 Jun 28 21:34 554f516c-ba779540
drwx----- 1 4294967294 4294967294      64 Jun 28 20:05 a857137a-a6f3c023
lrwxrwxrwx 1 root      root      35 Jul  2 12:49 iscsi-lun0 -> 44a7be3d-ab15e6a9-4abe-00065bec0eb6
lrwxrwxrwx 1 root      root      35 Jul  2 12:49 iscsi-lun1 -> 44a7c03d-159649b5-0e6a-00065bec0eb6
lrwxrwxrwx 1 root      root      17 Jul  2 12:49 isos -> 2cf78959-22cd65c7
lrwxrwxrwx 1 root      root      35 Jul  2 12:49 local-esx1 -> 44a38c72-156b2590-be15-00065bec0eb7
lrwxrwxrwx 1 root      root      35 Jul  2 12:49 local2-esx1 -> 44a6c956-66056236-c671-00065bec0eb6
lrwxrwxrwx 1 root      root      17 Jul  2 12:49 nas-vmx -> 554f516c-ba779540
lrwxrwxrwx 1 root      root      17 Jul  2 12:49 templates -> a857137a-a6f3c023

```

**Note:**

The UUID is the value in blue. So the try and unchanging volume id is actually something like:

/vmfs/volumes/44a38c72-156b2590-be15-00065bec0eb7

2. Type the command:

**In -sf /vmfs/volumes/44a38c72-156b2590-be15-00065bec0eb7 /vmfs/volumes/esx1-local**

**Note:**

The LN makes symbolic links (a bit like shortcuts if you like) –s makes/change a “symbolic” link, -f over-writes the existing symbolic link

## Viewing VMFS Volumes/Partition Information

1. Type the command:

**vmkfstools -P /vmfs/volumes/local-esx1**

**Note:**

This returns information like so:

**VMFS-3.21 file system spanning 1 partitions.**

File system label (if any): local

Mode: public

Capacity 73282879488 (69888 file blocks \* 1048576), 41889562624 (39949 blocks) avail

UUID: 44a6c956-66056236-c671-00065bec0eb6

Partitions spanned:  
vmhba1:4:0:1

**Note:**

For NAS data store it would look like this:

**NFS-1.00 file system spanning 1 partitions.**

File system label (if any): templates

Mode: public

Capacity 80023715840 (19537040 file blocks \* 4096), 25076690944 (6122239 blocks) avail

UUID: ab184e34-6f68911d-0000-000000000000

Partitions spanned:  
nfs:templates

## Viewing Available Physical Disk Space

### Note:

1. Logon to the Service Console as ROOT
2. Type:

**df -h** (the linux method)

or

**vdf -h** (the VM Kernel Method)

```
Filesystem      Size  Used Avail Use% Mounted on
/dev/sda2       2.0G  1.3G  630M  67% /
/dev/sda1        46M   15M   29M  35% /boot
/dev/sda5       2.0G   33M  1.9G   2% /home
none            132M    0  132M   0% /dev/shm
/dev/sda7       494M   8.1M  461M   2% /tmp
/dev/sda6       992M  249M  693M  27% /var
/dev/sda9       28G   1.4G   25G   6% /vmimages
/vmfs/devices   591G    0  591G   0% /vmfs/devices
/vmfs/volumes/2cf78959-22cd65c7
                74G   54G   19G  73% /vmfs/volumes/isos
/vmfs/volumes/44a38c72-156b2590-be15-00065bec0eb7
                33G   28G   5.1G  84% /vmfs/volumes/local-esx1
/vmfs/volumes/44a6c956-66056236-c671-00065bec0eb6
                68G   46G   21G  68% /vmfs/volumes/local2-esx1
/vmfs/volumes/44a7be3d-ab15e6a9-4abe-00065bec0eb6
                19G  626M   19G   3% /vmfs/volumes/iscsi-lun0
/vmfs/volumes/44a7c03d-159649b5-0e6a-00065bec0eb6
                19G  626M   19G   3% /vmfs/volumes/iscsi-lun1
/vmfs/volumes/554f516c-ba779540
                74G   54G   19G  73% /vmfs/volumes/nas-vms
/vmfs/volumes/a857137a-a6f3c023
                74G   54G   19G  73% /vmfs/volumes/templates
```

### Note:

The partition table for /sda does NOT follow the recommendations for ESX 3.x – this partition table was taken for an ESX 2.x server which was upgraded

### Note:

If you wish to see how the vmhba syntax relates to /dev/sd? You will find that you can still use old ESX 2.x command "vmkpcidivv -q vmhba\_devs". However, it has been hugely deprecated – and you should instead

### esxcfg-vmhbadevs -q

```
vmhba0:0:0 /dev/sda
vmhba0:1:0 /dev/sdb
vmhba1:2:0 /dev/sdc
vmhba1:3:0 /dev/sdd
vmhba1:4:0 /dev/sde
vmhba1:5:0 /dev/sdf
vmhba1:6:0 /dev/sdg
vmhba40:0:0 /dev/sdh
vmhba40:0:1 /dev/sdi
```

**Note**

If you want to see all of the information together – vmhba ID, /dev name, LVM id and VMFS volume label you can use this perl script

```
#!/usr/bin/perl

@array = ` /usr/sbin/esxcfg-vmhbadevs -m `;

foreach (@array)
{
    ($vmk, $cos, $uuid) = split;
    ($tmp, $label) = split (/:/, ` /usr/sbin/vmkfstools -P
/vmfs/volumes/${uuid} | grep "File system label" `);
    print "$vmk \t $cos \t $uuid \t $label\n"; }

```

**This produces an output like so:**

```
vmhba0:1:0:1    /dev/sdb1      44a38c72-156b2590-be15-00065bec0eb7    local-esx1
vmhba1:4:0:1    /dev/sde1      44a6c956-66056236-c671-00065bec0eb6    esx3onlystorage

```

**Note:**

This only shows me my VMFS, not my NAS datastores

## ***Mounting and U-mounting VMFS's***

**Note:**

By default all VMFS volumes are mounted at start-up under the /VMFS mount point. Should you accidentally or deliberately unmount these partitions – there is a method for unmount and mounting them manually.

Remember there can be no running VM's on the VMFS because SCSI Reservation would stop access

1. **Logon to the Service Console as ROOT**
2. To **umount the VMFS's** type:

```
umount /vmfs
```

3. To **remount the vmfs's** type:

```
mount -t vmfs vmfs /vmfs
```

## ***Managing Storage Devices***

**Note:**

You can use a couple commands to handle storage devices. Some of these have already been mentioned like the command esxcfg-swiscsi -s to rescan the iSCSI Adapter(s). For those of you who come from ESX 2.x background it looks like cos-rescan.sh has been discontinued, its functionality has *probably* been integrated into the -s switch...

### **Forcing a rescan of Fibre Channel Device**

1. Type the command

```
vmkfstools -s vmhba1
```

**Note:**

You can also use **esxcfg-rescan vmhbaN** to be rescanned as well which give this kind of output:

```
Rescanning vmhba1...done.  
On scsi1, removing: 2:0 3:0 4:0 5:0 6:0.  
On scsi1, adding: 2:0 3:0 4:0 5:0 6:0.
```

## Managing SCSI Reservations of LUN's

### Note:

- Occasionally, things go wrong. Such is life. One example is when through a configuration error a SCSI reservation (lock) is put on a LUN, which subsequently doesn't get released. Until a LUN is released you won't be able manage the file system and other ESX host will not be very happy bunnies. Typically, these kind of reservation problems happen in clustering scenarios...
- This locking is controlled by the `-L` switch on `vmkfstools`...
- From the Server Configuration guide:

`-L reserve` .

Reserves the specified LUN. After the reservation, only the server that reserved that LUN can access it. If other servers attempt to access that LUN, they will get a reservation error!

`-L release` .

Releases the reservation on the specified LUN. Any other server can access the LUN again.

`-L lunreset`

Resets the specified LUN by clearing any reservation on the LUN and making the LUN available to all servers again. The reset does not affect any of the other LUNs on the device. If another LUN on the device is reserved, it remains reserved.

`-L targetreset` .

Resets the entire target. The reset clears any reservations on all the LUNs associated with that target and makes the LUNs available to all servers again.

`-L busreset` .

Resets all accessible targets on the bus. The reset clears any reservation on all the LUNs accessible through the bus and makes them available to all servers again

- This command use the device parameter... so uses the `/vmfs/devices/disks/vmhbaA:T:L:P` syntax

## Using USB storage Devices

### Note:

- By default USB device drivers are loaded automatically
- This can be disabled by modifying `/etc/modules.conf` and remarking out with `#` the line which begins `alias usb-controller usb-ohci`
- If someone has done this – then you can load the modules (drivers) manually and access the device as needed using the `insmod` command
- You can read but not write to a NTFS partition on a removable device

- FAT32 is supported for read/write. There are [some free EXT3 drivers](#) for Windows (the one I use currently only works on NT4, W2K and WXP, not W2K3)
- You might prefer to use EXT3 if you wish to export virtual disks in their monolithic format (on VMFS partition) into a the COW format (copy-on-write) which will take up less physical space on the disk
- In my case it was dead easy – I plugged the USB device into the server. It came up with a message – and said it allocated the id of /dev/sdc to the disk. So all I had to do was mount it with

```
mkdir /mnt/usbdisk
mount /dev/sdc1 /mnt/usbdisk
```

**Note:**

If this doesn't happen to you – then you MIGHT find the stuff below useful.

- Systems using the USB-UHCI device driver with the USB 2.0 interface can cause ESX Server to show a false warning message during the boot sequence. The error looks like this on boot-up

```
Mar 29 11:04:10 vmserver1 rc.sysinit: Initializing USB controller (usb-uhci):
succeeded
```

```
Mar 29 11:04:10 vmserver1 modprobe: Hint: insmod errors can be caused by
incorrect module parameters, including invalid IO or IRQ parameters
```

```
Mar 29 11:04:10 vmserver1 modprobe: /lib/modules/2.4.9-
vmnix2/kernel/drivers/usb/usb-ohci.o: init_module: No such device
```

```
Mar 29 11:04:10 vmserver1 modprobe: /lib/modules/2.4.9-
vmnix2/kernel/drivers/usb/usb-ohci.o: insmod /lib/modules/2.4.9-
mnix2/kernel/drivers/usb/usb-ohci.o failed
```

```
Mar 29 11:04:10 vmserver1 modprobe: /lib/modules/2.4.9-
vmnix2/kernel/drivers/usb/usb-ohci.o: insmod usb-ohci failed
```

```
Mar 29 11:04:10 vmserver1 rc.sysinit: Initializing USB controller (usb-ohci):
failed
```

**Note:**

This was taken from ESX 2.x server

It is safe to ignore this message, but if you want to configure your system so that this warning does not appear the next time you boot your ESX Server machine, follow these steps

**Link:** [Answer ID 1659](#)

You might find these other article related to USB and IRQ sharing which you might find interesting

[1651 - Using USB Storage Devices on the ESX Server Service Console](#)

[1290 - IRQ Sharing May Impact Performance](#)

[1326 - Using USB When ESX Server Disables It](#)

If you want to use a USB hard-drive to copy files from the ESX server to it – and you have these errors -then a general work around is to boot to a Knoppix Boot CD and do it that way. Remember if you are copying virtual disks – these will have to be exported to the COW format first.

If you have no errors like me then this how you can go about using USB



hard-drives and devices

- To list what type of USB host controller interface you have. There are 3 of USB host controller interface types (OHCI, UHCI and EHCI) You can use the command below to find out yours:

### **lspci -v | grep HCI**

#### **Note:**

My server returned:

```
00:0f.2 USB Controller: ServerWorks OSB4/CSB5 OHCI USB Controller
(rev 05) (prog-if 10 [OHCI])
Subsystem: ServerWorks OSB4/CSB5 OHCI USB Controller
```

so I know my server I know I need to load the OHCI driver, not the UHCI or EHCI driver

1. **Logon to the Service Console as ROOT**
2. [Optionally] Load up the USB Device Drivers

**insmod usbcore**

**insmod usb-storage**

**modprobe usb-ochi or modprobe usb-uhci or modprobe ehci**

3. Use the **dmesg** to print a out a list of active devices and scroll up to locate Initializing USB Mass Storage Driver....

```
Initializing USB Mass Storage driver...
usb.c: registered new driver usb-storage
scsi2 : SCSI emulation for USB Mass Storage devices
Vendor: IC25N040 Model: ATCS04-0 Rev: CA40
Type: Direct-Access ANSI SCSI revision: 02
VMWARE SCSI Id: Supported VPD pages for sdg : 0x1f
:VMWARE SCSI Id: Could not get disk id for sdg
:VMWARE: Unique Device attached as scsi disk sdg at scsi2, channel 0, id 0, lun 0
Attached scsi disk sdg at scsi2, channel 0, id 0, lun 0
scsi_register_host starting finish
SCSI device sdg: 78140160 512-byte hdwr sectors (38154 MB)
sdg: sdg1
scsi_register_host done with finish
WARNING: USB Mass Storage data integrity not assured
USB Mass Storage device found at 2
USB Mass Storage support registered.
```

#### **Note:**

The system assigns a SCSI device ID to the USB device (even though its like to be IDE Laptop Disk if it's a portable hard-drive). The critical bits are reference to the SDG. This tells me the USB device has been added to the end of all my other SCSI disks (sda, sdb, sdc). This will help me in the next stage which is creating a mount point and mounting the partition on the USB disk.

Adding additional hard-drives to the system can upset this allocation of *sdn*.

4. **Create a mount point** with

**mkdir /mnt/usb**

5. Mount the first partition on the disk with

**mount /dev/sdg1 /mnt/usb**

**Note:**

If you are unsure about the partition scheme on the disk you can use **fdisk -l /dev/sdg** to print to the console the partition table

6. **List Files** and Start using the disk with **ls -l /mnt/usb**
7. **Unmount the USB device when finished** with

```
cd /  
umount /mnt/usb
```

**Note:**

Even after unmounting the drive. If I should down the ESX server with the USB Storage attached I get this worrying message – which could be benign

```
scsi: device set offline - not ready or command retry failed after bus reset :  
host 2 channel 0 id 0 lun 0  
i/o error: dev 08:62, sector 4168  
i/o error: dev 08:62, sector 4168  
i/o error: dev 08:62, sector 0
```

# Module 6: Managing Virtual Machines

## IMPORTANT:

Even changes made inside a virtual machine whilst it is powered off, will require a restart of the hostd service with service mgmt-vmware restart

## Creating Virtual Disks

1. Power off your target virtual machine...
2. To create a virtual disk use the vmkfstools command:

```
vmkfstools -a lsilogic -c 10240m /vmfs/volumes/local-esx1/instructor/instructor_1.vmdk
```

### Note:

This is lower-case -c which creates a file, upper-case -C creates a vmfs file system. Here I am following the naming convention that VC would normally be applied. The first disk would instructor.vmdk, and subsequent disks would be serialised with instructor\_1.vmdk and instructor\_2.vmdk

This actually creates two files instructor\_1.vmdk (metadata, of a couple of KB) and an instructor\_1-flat.vmdk which is 10GB in size. I could have used 1g instead of 10240m on the command-line

The -a option allows users to indicate which device driver should be used to communicate with the virtual disk. Failure to set this could cause a question in the virtual machine on power-up depending on what SCSI Adapter controller is used in the VM. Personally, I always use LSILogic as the driver unless it is unsupported in the guest OS I am working with.

The default option when creating new virtual disks is "zeroedthick". A zeroed thick disk has all space allocated at creation time, and this space is wiped clean of any previous contents on the physical media. There are other options under the -d switch you may wish to investigate.

The only disk formats that can be used for NFS are thin and 2gbsparse. By default, files and virtual disks are stored in thin format on NFS servers, with blocks allocated on demand as needed. No other options, including RDMs, are supported on NFS.

3. **Next we need to add this into the virtual machine, we can do this by editing the VMX file**

```
nano -w /vmfs/volumes/local-esx1/instructor/instructor.vmx
```

and the lines:

```
scsi0:1.present = "true"  
scsi0:1.fileName = " instructor_1.vmdk"  
scsi0:1.deviceType = "scsi-hardDisk"
```

### Note:

scsi0:1 means it is the next disk after the boot disk, on the first virtual SCSI adapter. True means the device is connected, filename the virtual disk (metadata file only) and the device type indicates this virtual SCSI

disk, not a virtual IDE drive.

4. **Save** you VMX file **and Exit nano**
5. Power on your virtual machine with

**vmware-cmd -l**

**Note:**

That's a lower-case L for lima. This produces a list of registered VM's on the ESX host

**vmware-cmd /vmfs/volumes/44a38c72-156b2590-be15-00065bec0eb7/instructor/instructor.vmx start**

**Note:**

In the past we used to be able to use the friendly volume label. This is no longer supported as outline in [KB 2122](#)

## ***Creating and Managing RDM's***

**Note**

- There are two types of RDM – virtual compatibility mode and physical compatibility mode
- These are normally set by radio button option in the VI Client
- On the command-line the way these are specified are with two different switches
- Main uses of RDM's are for:

Accessing existing data on a SAN/iSCSI  
Required for physical to virtual clustering  
Required for clustering-across-boxes

### **RDM for Virtual Compatibility**

1. You may wish to check before you begin on which LUN's your ESX server has access to with:

**ls -l /vmfs/devices/disks**

2. Type the command:

**vmkfstools -a lsilogic -r /vmfs/devices/disks/vmhba40:0:0:0 /vmfs/volumes/local-esx1/instructor/instructor\_1.vmdk**

**Note:**

This creates a RDM metadata file in /vmfs/volumes/local-esx1 using my iSCSI Lun on vmhba40:0:0:0. It creates two files as in the -c example a metadata file called instructor\_2.vmdk and a media\_1-rdm.vmdk file. VMware automatically append -rdm to the end of the file for us!

**Note:**

The metadata file **MUST** be stored on a VMFS partition – it cannot reside on a NAS DataStore

**Note:**

You do have to specify the last 0 in vmhba40:0:0:0. This last 0 indicates you wish to use the entire LUN. You cannot create a RDM file to a specific

partition WITHIN a LUN

3. Next we need to add this into the virtual machine, we can do this by editing the VMX file

```
nano -w /vmfs/volumes/local-esx1/instructor/instructor.vmx
```

and the lines:

```
scsi0:2.present = "true"  
scsi0:2.fileName = "instructor_1.vmdk"
```

## RDM for Physical Compatibility

### Note:

Physical compatibility (required for physical to virtual clustering) is set up in a very similar way. With physical compatibility the SCSI Reservations/Filtering normally imposed by the vmkernel are "loosened" such that other systems that also want to impose SCSI reservations will work (such as clustering systems)

This is sometime referred to as a Pass-Through RAW Device Mapping

In this case the switch is -z

- **vmkfstools -a lsilogic -z /vmfs/devices/disks/vmhba40:0:1:0 /vmfs/volumes/local-esx1/instructor/instructor\_2.vmdk**

### Note:

In this case the system labels the corresponding file instructor\_2-rdmp.vmdk. So we know this is a RAW Disk Mapping file with physical compatibility

## Viewing RDM Information

### Note:

In the past it wasn't easy (impossible?) to view the contents of an RDM file (I guess you could have used vmkfstools -l to list files and see which were virtual disks and which were RDM's).

There is now a switch to query the metadata file which then reports information

1. **vmkfstools -q /vmfs/volumes/local-esx1/instructor\_1.vmdk**

```
Disk instructor_1.vmdk is a Non-passthrough Raw Device Mapping  
Disk Id: vml.010000000020202020564952545541  
Maps to: vmhba40:0:0:0
```

2. **vmkfstools -q /vmfs/volumes/local-esx1/instructor\_2.vmdk**

```
Disk instructor_2.vmdk is a Passthrough Raw Device Mapping  
Disk Id: vml.010001000020202020564952545541  
Maps to: vmhba40:0:1:0
```

### Note:

You can also cat the contents of the metadata file as well – but the VM MUST be powered off first!

```
cat /vmfs/volumes/local-esx1/instructor_2.vmdk
```

```

# Disk DescriptorFile
version=1
CID=0a8fee69
parentCID=ffffffff
createType="vmfsRawDeviceMap"

# Extent description
RW 8388608 VMFSRDM "instructor_2-rdm.vmdk"

# The Disk Data Base
#DDB
ddb.toolsVersion = "7172"
ddb.adapterType = "lsilogic"
ddb.geometry.sectors = "63"
ddb.geometry.heads = "255"
ddb.geometry.cylinders = "522"
ddb.virtualHWVersion = "4"

cat /vmfs/volumes/local-esx1/instructor_3vmdk

# Disk DescriptorFile
version=1
CID=3cfc3778
parentCID=ffffffff
createType="vmfsPassthroughRawDeviceMap"

# Extent description
RW 8388608 VMFSRDM "instructor_3-rdmp.vmdk"

# The Disk Data Base
#DDB
ddb.toolsVersion = "7172"
ddb.adapterType = "lsilogic"
ddb.geometry.sectors = "63"
ddb.geometry.heads = "255"
ddb.geometry.cylinders = "522"
ddb.virtualHWVersion = "4"

```

## ***Exporting & Importing Virtual Disks***

### **Note:**

- Those of you with some experience will know importing (into a VMFS volume) and exporting (out of a VMFS volume) virtual disks has two functions. First it serves as a safe method of converting disk into a format that can be taken to/from "foreign" file systems such as NTFS and ESX3. Historically, these file systems have tools that could corrupt files bigger than 2GB in size
  - These two formats have in the past been referred to as "Monolithic" and "COW" (copy-on-write). COW is no longer a favoured term as it is technically inaccurate – instead we refer to it as the "Sparse" format...
  - That is more or less still the same – however there is a switch (-d) that allow you control the format of the disk. The -d switch is not always required – it depends on if your exporting or importing.

- This is what the MAN pages say about the `-d` switch:

**zeroedthick**

This is the default option when creating new virtual disks. A zeroed thick disk has all space allocated at creation time, and this space is wiped clean of any previous contents on the physical media.

**eagerzeroedthick**

An eager zeroed thick disk has all space allocated and zeroed out at creation time. Such disks may take longer time during creation compared to other disk formats.

**thick**

A thick disk has all space allocated at creation time. This space may contain stale data as it exists on the physical media.

**thin**

Space required for thin-provisioned virtual disk is allocated and zeroed on demand as opposed to upon creation. Another purpose is for backup – the sparse format holds the true data – minus any free space – so generally more efficient for backup purposes.

- **Related to this there is also the switch**

`-j`

Convert a thin virtual disk to thick with the additional guarantee that any data on thin disk is preserved and any blocks that were not allocated get allocated and zeroed out.

- Another purpose is from transporting virtual disks safely to/from other vPlatforms such as Workstation on Linux or Windows or VMware Server (nee GSX) on Linux or Windows
- Perhaps you don't have VC you still have a stand-alone ESX servers – and you still use templates and import/export method of ESX 2.x days...
- For those of you from a ESX 2.x background you should know the old `-e` switch is depreciated and the way do a `-e` is with `-i` and some additional switches in `vmkfstools`

**Exporting “Zeroedthick” Virtual Disks (nee Monolithic) into the “Sparse” (nee COW) Format**

**Note:**

In my case I am using `/vmimages` partition. But you could use any EXT3 location or even export directly to a mount NFS or SMB share

1. Type the command:

```
vmkfstools -i /vmfs/volumes/local-esx1/instructor/instructor.vmdk
-d 2gbsparse /vmimages/instructor.vmdk
```

**Note:**

You should get response like this

```
Destination disk format: sparse with 2GB maximum extent size
Cloning disk '/vmfs/volumes/local-esx1/instructor/instructor.vmdk'...
Clone: 3% done.
```

**Note:**

The disk is now in a portable format... you could tar it up and copy somewhere else for backup purposes – connect with WinSCP on your workstation and bring the files down to your PC. If you have VMware Workstation on your PC you could configure a virtual machine to use this disk

**Importing Virtual Disk from the Sparse Format****Note:**

For those you that a familiar with ESX 2.x this process hasn't changed much at all. What is different is because VMFS now stores directories, and the VM's configuration files are held in a directory – is where do you restore the disk too... The easiest way to do this I think is, to first create a VM without a virtual disk – this will set-up up the directory the configuration files. Then trigger the vmkfstools import to the vmfs volume and directory – and then edit the VMX file to tell the VM where its new disk is (as I showed just previously with -c)

1. Type the command:

```
vmkfstools -i /vmimages/workstationdisk.vmdk /vmfs/volumes/local-esx1/workstation1/workstation1.vmdk
```

```
Destination disk format: VMFS thick
Cloning disk '/vmimages/media.vmdk'...
Clone: 10% done.
```

**Copying Files from one ESX host to another****Note:**

SCP (Secure Copy) the utility to use. The security is not one of reliability of the copy – just stopping on the interception of the data through the network. This example copies a vmdk its sparse format. Remember to use ESX as SSH client this must be enabled in the ESX firewall first...

1. Type:

```
scp /vmimages/instructor*.vmdk root@esx2.rtfm-ed.co.uk:/vmimages
```

**Note:**

Don't forget the colon: after the name of your server

**Note:**

It will warn you that the first time you copy the destination host

```
" The authenticity of host 'esx1.rtfm-ed.co.uk' (192.168.2.102)' can't be
established.
RSA key fingerprint is 92:4b:ba:b5:ca:31:6f:e7:8c:2d:00:6e:cf:c6:b6:ea."
```

This is because the certificate of the corresponding machine is un-trusted and cannot be verified. To get rid of these first time messages you need to assign your own certificates or acquire them from a 3<sup>rd</sup> party.

2. Choose **Yes** to continue

**Note:**

The system will then warn you that:



*" Warning: Permanently added "esx1.rtfm-ed.co.uk", 192.168.2.102' (RSA) to the list of known hosts."*

3. At the prompt which reads "**root@esx1.rtfm-ed.co.uk password:**" type the **other servers root password**, in my case **\*\*\*\*\***

**Note:**

In this screen grab I copied the tar file for upgrade of esx 2.x to 3.x The system will give you a status bar like this...

```
vmware-esx-3.0.0-27701-upgrade-eval.tar.gz 68% 309MB 9.2MB/s 00:15 ETA
```

## Renaming Virtual Disks

**Note;**

Renaming virtual disks used to be dead easy. You would just the mv command, and edit your VMX file accordingly. Now virtual disks have their companion vmdk metadata file and the -flat.vmdk. We need a special command to rename both files and keep the data in synch. I guess you could manually edit the metadata file – but why bother when vmkfstools does it all for you.

1. **Before my rename the metadata in the instructor.vmdk** looked like this:

```
# Disk DescriptorFile
version=1
CID=a896726b
parentCID=ffffffff
createType="vmfs"

# Extent description
RW 4194304 VMFS "instructor-flat.vmdk"

# The Disk Data Base
#DDB
ddb.adapterType = "lsilogic"
ddb.geometry.sectors = "63"
ddb.geometry.heads = "255"
ddb.geometry.cylinders = "261"
ddb.thinProvisioned = "1"
ddb.virtualHWVersion = "4"
ddb.toolsVersion = "7172"
```

2. Type the command:

```
vmkfstools -E /vmfs/volumes/local-esx1/instructor/instructor.vmdk
instructorOS.vmdk
```

3. After the rename the metadata looks like this

```
# Disk DescriptorFile
version=1
CID=a896726b
parentCID=ffffffff
createType="vmfs"
```

```
# Extent description
RW 4194304 VMFS "instructorOS-flat.vmdk"
# The Disk Data Base
#DDB
ddb.adapterType = "lsilogic"
ddb.geometry.sectors = "63"
ddb.geometry.heads = "255"
ddb.geometry.cylinders = "261"
ddb.thinProvisioned = "1"
ddb.virtualHWVersion = "4"
ddb.toolsVersion = "7172"
```

**Note:**

Remember, you still have to manually edit the VMX file to reflect your changes

## ***Resizing Virtual Disks***

**Note:**

According to VMware employees on the forum, the `-x` switch is only supported for making virtual disks larger – not smaller. This has been the case in ESX 2.x. This surprised me – because in the past I have successfully used `-x` to make a virtual disk smaller. This however, was way back in the midst of time, circa ESX 2.0 (around 2004). So perhaps they withdrew it as function as is very easy to corrupt files when you do a shrink... If you want to shrink a virtual disk – perhaps you use ghost to clone a big virtual disks to a small one...

**Note:**

- There a couple of ways approaching this issue – while it is relatively easier to increase and decrease the VMDK file, Windows does not give you an easy way to reduce or increase the size of a partition.
- Therefore 3<sup>rd</sup> Party tools maybe required to complete the procedure – especially it the partition is the boot partition
- There are five main methods:
  - Partition Tools – you can use Power Quest Partition Magic on Client based OS, to resize the partition or Power Quests Volume Manager on Server based OS
  - These are commercial available tools – and they cost money – alternatively you can use the [Knoppix Boot CD](#) (version 3 and higher) and tool on the CD called `qparted` which does the same thing
  - If you have W2K3 (or W2K with the Resource Kit) you can go through a procedure using MS Diskpart utility. It's not a very friendly process and is quite convoluted. Its main advantage is that is cheap – as you do not need 3<sup>rd</sup> party tools achieve it. Its disadvantage is that as far as I am aware it's is not available in W2K or NT4. For how to do this more information about this visit [vmwareprofessional.com](#)
  - Ghost Method – Create a new disk of the appropriate size – and restore the data using a disk to disk restore method
  - DR method – Backup OS System State to another location – create a new OS disk file, and restore the data
- It's worth knowing the alternatives in case you cannot get your preferred method to work...

- Expect a number of reboots – for the one asked for by the Partitioning tool, and the one called for by Windows. This can be somewhat annoying!

**Caution:**

Copy/Backup/Gzip/Export the VMDK in case the next procedure fails

**Note:**

I noticed that Knoppix is susceptible to disconnections from the Remote Console especially when you are shutting down the VM after booting from the ISO

1. **Shutdown the VM**
2. Type the following command:

**vmkfstools -X 6144m -force /vmfs/volumes/local/instructor.vmdk**

**Note:**

If you follow this command with `ls -l -h` you will see it is bigger. Again I could have specified 6g instead of 6144m. It's -flat file that changes size – the metadata file stays the same but has new geometry information

3. **Start-up the VM, and check that it boots properly**

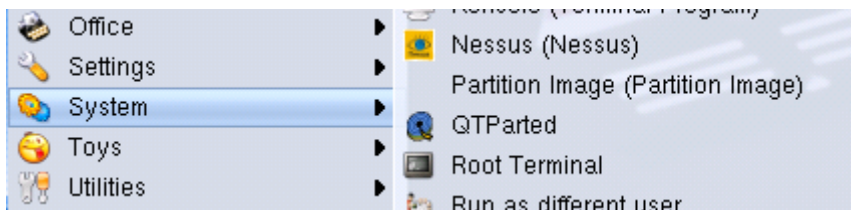
**Note:**

You may get “error loading operating system” messages if the vmdk file has become corrupted

4. **Connect to the ISO which contains the Knoppix Boot CD - Using Remote console attach to this ISO file making sure you enable X Connected at Power On**
5. Power on the VM
6. **Change the boot order on the VM to give the CD-ROM the highest priority – allow the system to boot to Knoppix CD or use the [ESC] key to select the CD-ROM as a the boot device**
7. **Click at the Knoppix KDE button**



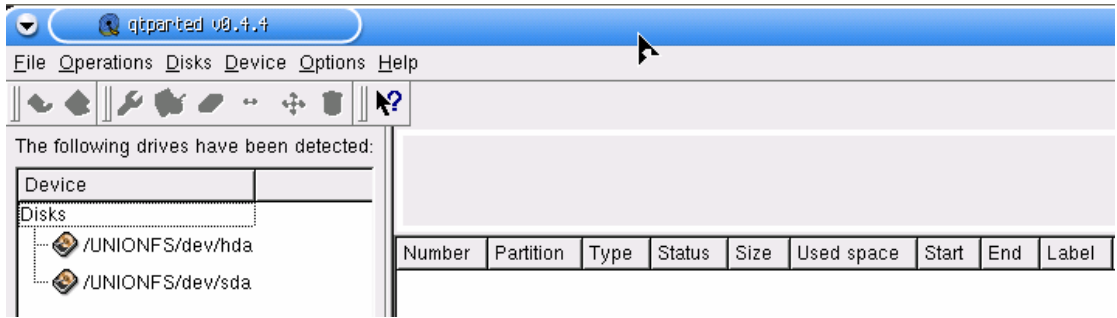
8. Choose **System Tools**



9. Choose **Qparted**

**Note:**

Qparted will display the partition/disk layout of the virtual machine.

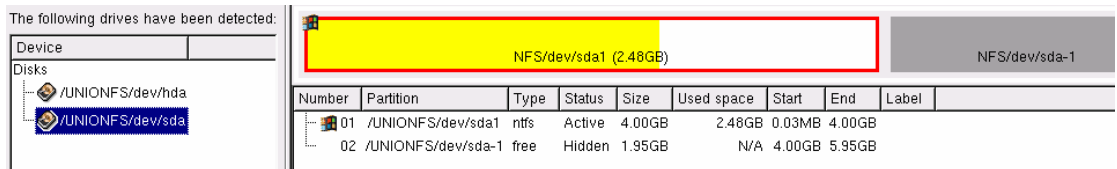


In this case the IDE CD-ROM and single SCSI boot disk (sda)

10. Select **/inionfs/dev/sda**

**Note:**

This then shows the partition table on that virtual disk

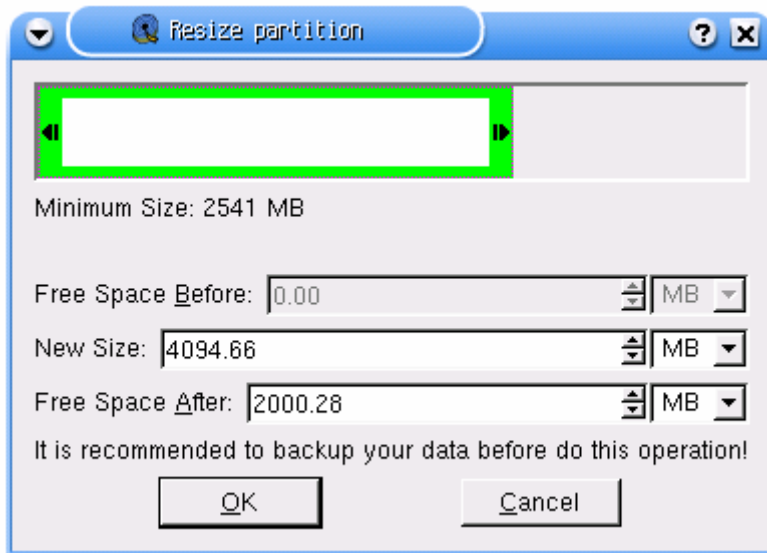


In this case a single C drive of 4GB with free space on the hard-drive of 2GB added by using vmkfstools -X 6096mg

11. Select **Partition No 01**

12. Right-click and choose **&Resize**

13. In the dialog box drag to increase or decrease the size of Partition 01



Using the  icon and Click OK

14. Click **File**, and choose **&Commit**

**Note:**

Confirm the warnings by click the Yes button – and watch the dialog boxes go by... and Confirm the Success dialog by clicking OK

#### 15. **Shutdown the Virtual Machine**

#### 16. **Disconnect the CD-Rom**

**Note:**

This will allow the VM to boot to the VMDK file rather than to the CD-Rom.

#### 17. **Reboot the Virtual Machine**

**Note:**

In Windows 2003 you will see check disk events – this appears to be normal. Additionally, you will find Windows 2003 will detect the hard-drive as if it was a new device and request a reboot.

## ***Deleting Virtual Disks***

**Note:**

I'm not really sure why there is a special command for deleting virtual disks – when we could easily use `rm namofvirtualdisk*.vmdk` – this would delete the metadata and also the `-flat.vmdk` file... but anyway – `vmkfstools` can do it too

1. Type the command

```
vmkfstools -U /vmfs/volumes/local/instructor.vmdk
```

**Note:**

It deletes both the metadata and the `-flat` file – and doesn't currently ask "Are you sure". Hee-hee... ☺

## ***Using VMware-cmd***

**Note:**

- It works by manipulating the VMX and sending instructions to the VM via the VMX file
- There are lots you can do with this command. Beyond what is feasible in this document. For further information look at the Scripting Guide for ESX
- It also returns 1 for positive results, and 0 for failures – so can use it for BASH shell scripts and if statements

### **To List Registered VMS**

1. Type:

```
vmware-cmd -l
```

**Note:**

List paths and names of VMX files. Useful for when you have type long paths to the VMX file. You can highlight an entry in the list and copy it to your command-line. Unfortunately, `vmware-cmd` no long supports volume names – instead you have to use the UUID path to the VMX file. ☹

I will give one example of `vmware-cmd` with the UUID, after that I will replace the string with `<UUID>` because it's so blummin long!

## To Register and Un-register a VM

1. Type:

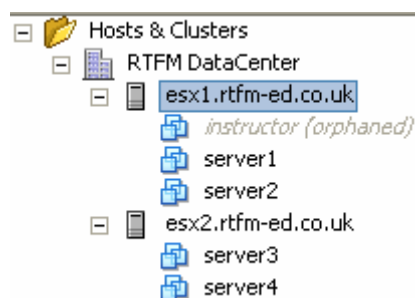
```
vmware-cmd -s unregister
```

```
/vmfs/volumes/44a38c72-156b2590-be15-00065bec0eb7/instructor/instructor.vmx
```

### **Note:**

Also includes register as a command. S stands for set. This un-registers a VM from an ESX host (stand-alone)

Unfortunately, it does NOT un-register it from VirtualCenter. Instead you are left with an "orphaned" VM that needs right-clicking and "Remove from the Inventory"



I think this happens because what vmware-cmd -s is manipulating is what used to be called the vm-list. It's not manipulating the VC database.

### **Note:**

Why is the command still useful? We say you had a server failure – which you had rebuild. You then need to "register" the VMX file with an ESX server to then be able to manage it. If you had 40 VMX you wouldn't want to do that by hand using the GUI.

## Power Options on a VM

### **Note**

- If you use trysoft, the Guest OS will be shutdown gracefully.
- The trysoft is a "mode" option – it tries to run the normal scripts but uses a hard shutdown/startup if the VM is not behaving properly.
- There are two other "modes" – soft which runs scripts but never does a hard start or stop or "hard" which powers off/on a VM as if you had hit the power switch. Sometimes people use trysoft first, and then if the VM refuses to power of, they follow it with hard power off

1. To  **a virtual machine** type:

```
vmware-cmd /vmfs/volumes/<UUID>/instructor1/instructor1.vmx start trysoft
```

2. To  **a virtual machine** type:

```
vmware-cmd /vmfs/volumes/<UUID>/instructor1/instructor1.vmx stop trysoft
```

3. To  a *virtual machine* type:

```
vmware-cmd /vmfs/volumes/<UUID>/instructor1/instructor1.vmx  
suspend
```

**Note:**

There isn't much in the way of status/progress here. The old-style "Remote Console" from ESX 2.x days used to give you dialog box with a status bar

4. To  a *virtual machine* type:

```
vmware-cmd /vmfs/volumes/<UUID>/instructor1/instructor1.vmx start  
trysoft
```

**Note:**

There is no resume switch on vmware-cmd – just a power-up which retrieves the suspend file, and resumes the machine

5. To  a *virtual machine* type

```
vmware-cmd /vmfs/volumes/<UUID>/instructor1/instructor1.vmx reset  
trysoft
```

**Note:**

This a soft reboot of the virtual machine

6. To find out the **Power Status** of a VM use:

```
vmware-cmd /vmfs/volumes/<UUID>/instructor1/instructor1.vmx  
getstate
```

**Note:**

It will return "On"; "Off"; "Suspend" and "Stuck" if the VM is waiting for interaction

### Finding the Heartbeat

1. **Type:**

```
vmware-cmd /vmfs/volumes/<UUID>/instructor1/instructor1.vmx  
getheartbeat
```

**Note:**

You should see a number. Repeat the command. If the number increments the machine is alive. It stays the same it is dead. It doesn't have a heart beat!

### Finding the Status Of Devices

**Note:**

The Vmware-cmd tools is able to find the configuration of the VMX file, and also change entries in the file. To find out what the status is of the CD-ROM you would use:

1. **vmware-cmd /vmfs/volumes/<UUID>/instructor1/instructor1.vmx getconfig ide0:0.deviceType**

**Note:**

The CD-ROM is emulated (but not Virtualized) IDE 0:0 channel. This is how the CD-ROM is addressed in a Windows Environment. A safe way of learning the VMX variables is just print the VMX file to the console with:

```
cat /vmfs/volumes/<UUID>/instructor1/instructor1.vmx | more
```

### Configuring & Connecting a Device

This used to work in the past – and none of these commands give an error message – but I don't get the desired result... They must have changed this from ESX 2.x days – and don't what the next step is...

**Note:**

As long as you know the name of the variable in the VMX file you can change anything you like from the command line. For example there is a variable called `ide0:0.fileName` = which controls the path for the CD-ROM be it physical or an ISO image. There is a variable called `ide0:0.deviceType` = which controls if it is the physical or ISO image (`atapi-cdrom` for the physical and `cdrom-image` if it is an ISO file). In this example we will attach an ISO to the CD-ROM and Connect the CD-ROM to the VM. You may wish to have VC Console open whilst your doing this – to see visually the effect of your changes.

I assume that initially the CD-ROM is connected to the physical CD

1. First, **Disconnect the current device with:**

```
vmware-cmd /home/lavericm/vmware/instructor1/instructor1.vmx  
disconnectdevice ide0:0
```

2. Change to the system to use a ISO file instead of a physical CD with:

```
vmware-cmd /home/lavericm/vmware/instructor1/instructor1.vmx  
setconfig ide0:0.deviceType cdrom-image
```

Then set the image file to use with:

```
vmware-cmd /home/lavericm/vmware/instructor1/instructor1.vmx  
setconfig ide0:0.fileName /vmfs/volumes/isos/w2k3.iso
```

3. To **Refresh the OS - Reconnect the device** with:

```
vmware-cmd /home/lavericm/vmware/instructor1/instructor1.vmx  
connectdevice ide0:0
```

**Note:**

Now that the CD is set as an ISO file. You merely need to stages 2 and 3 to switch from one ISO file another. The whole process could be put in SH script like so:

```
vmware-cmd /home/lavericm/vmware/instructor1/instructor1.vmx  
disconnectdevice ide0:0
```

```
vmware-cmd /home/lavericm/vmware/instructor1/instructor1.vmx  
setconfig ide0:0.deviceType cdrom-image
```



```
vmware-cmd /home/lavericm/vmware/instructor1/instructor1.vmx  
setconfig ide0:0.fileName /vmimages/w2k3ent.iso
```

```
vmware-cmd /home/lavericm/vmware/instructor1/instructor1.vmx  
connectdevice ide0:0
```

**Notes: To Undo and go back to the physical CD-ROM**

```
vmware-cmd /home/lavericm/vmware/instructor1/instructor1.vmx  
disconnectdevice ide0:0
```

```
vmware-cmd /home/lavericm/vmware/instructor1/instructor1.vmx  
setconfig ide0:0.deviceType atapi-cdrom
```

```
vmware-cmd /home/lavericm/vmware/instructor1/instructor1.vmx  
setconfig ide0:0.fileName /dev/cdrom
```

```
vmware-cmd /home/lavericm/vmware/instructor1/instructor1.vmx  
connectdevice ide0:0
```

## ***Changing Parameters within a VM with /proc***

**Note:**

It seems like the info in /proc is quite limited which surprised me... so amount of stuff we can change is now limited as well...

## Module 7: Performance Monitoring

### Note:

- There are number of commands we can use to see how the Service Console is performing and how virtual machine are performing
- I will start with virtual machines as they are probably more likely to be source of the concern and problem
- Be very careful if you try to use the guest OS tools to do performance management/analysis – these were designed for OS running on physical hardware – our VM's are running on virtual hardware
- Good example of this is when Windows shows 100% CPU but outside of the VM you have capped the maximum CPU to 50%. The 100% shows Windows is using all of its 50% allocation – and does NOT reflect true CPU usage

### Using ESXTOP to Monitor Virtual Machines

#### Note:

- There's too much stuff here. Sorry! ☹ But one thing to look at if CPU is your constraining resource is the %ready values. The higher the %ready value – the more frequently the VM was "ready" to run but the CPU was too busy to service the request
- In this example of esxtop – we going to do something you do very often on courses – to demonstrate share values (the proportional system for relative resource management)
- To create contention I will "peg" the two VM's to the SAME CPU... you should never do this in the real world – in the world we strive (and often fail) to remove or reduce the effects of contention...
- What's contention when two or more VM's compete for the same resource...

#### Two VM's with cpubusy script – with the SAME share value

1. Power up JUST two VM's
2. **Create this VBS script** in each them

```
Dim goal
Dim before
Dim x
Dim y
Dim i

goal = 2181818

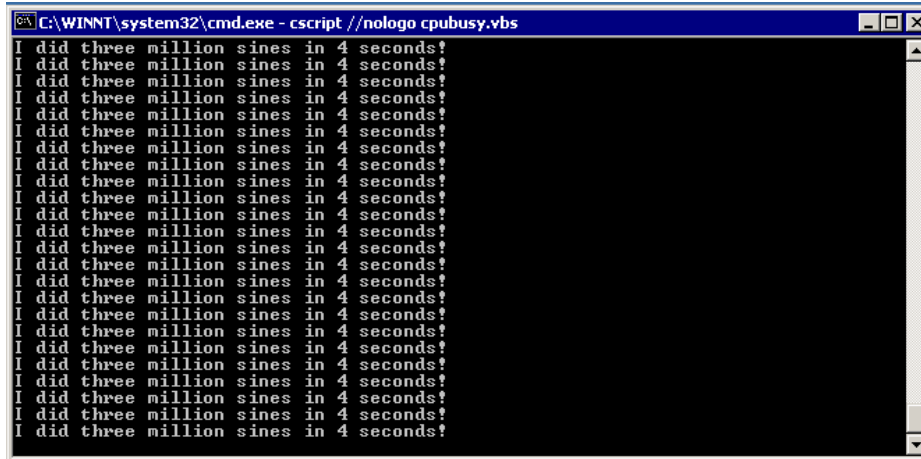
Do While True
    before = Timer
    For i = 0 to goal
        x = 0.000001
        y = sin(x)
        y = y + 0.00001
    Next
    y = y + 0.01
    WScript.Echo "I did three million sines in " & Int(Timer - before + 0.5) & " seconds!"
Loop
```

- Run the script on just one VM with:

**cscrip //nologo cpubusy.vbs**

**Note:**

This should produce lots of CPU cycles and this message in a cmd prompt:



**Run ESXTOP at the Service Console**

**Note:**

- esxtop is dynamic tool so every couple of second it updates... but here's a snapshot in time (you might want to zoom in here!)

```

4:52:27pm up 2:51, 46 worlds; CPU load average: 0.55, 0.55, 0.54
PCPU(%): 62.10, 40.50 ; used total: 51.30
CCPU(%): 0 us, 1 sy, 93 id, 6 wa ; cs/sec: 76
  
```

ID	GID NAME	NMEM	%USED	%SYS	%OVLP	%RUN	%WAIT	%BWAIT	%TWAIT	%CRUN	%CSTP	%IDLE	%RDY	%EXTRA	%MLMTD
1	1 idle	2	97.48	0.00	0.01	0.18	0.00	0.00	0.00	0.00	0.00	0.00	199.80	0.00	0.00
2	2 system	5	0.00	0.00	0.00	0.00	499.97	0.00	499.97	0.00	0.00	0.00	0.00	0.00	0.00
6	6 console	1	1.51	0.00	0.02	1.51	77.96	19.93	97.89	0.00	0.00	97.89	0.59	0.00	0.00
7	7 helper	13	0.01	0.00	0.00	0.01	1299.84	0.00	1299.84	0.00	0.00	0.00	0.09	0.00	0.00
8	8 drivers	13	0.01	0.01	0.00	0.01	1300.00	0.00	1300.00	0.00	0.00	0.00	0.00	0.00	0.00
9	9 vmotion	1	0.00	0.00	0.00	0.00	100.00	0.00	100.00	0.00	0.00	0.00	0.00	0.00	0.00
12	12 vmware-vmkauthd	1	0.00	0.00	0.00	0.00	100.00	0.00	100.00	0.00	0.00	0.00	0.00	0.00	0.00
13	13 instructor	5	1.70	0.00	0.03	1.70	427.41	69.60	497.01	0.00	0.00	97.82	1.27	0.00	0.00
14	14 server1	5	99.17	0.00	0.04	99.23	391.48	7.81	399.29	0.00	0.00	0.00	1.47	0.00	0.00

**Interpretation:**

Server1 is busy with %used at 99.17. The ESX server is doing it best to load the CPU request across my two physical CPU's (PCPU(%) is at 62.40 and 40.50. The %RDY value is quite low for VM called instructor and server1. So although server1 is busy, they are still performing well

**Run CPUbusy.vbs on the Second VM and Check out ESXtop**

```

4:56:45pm up 2:55, 46 worlds; CPU load average: 0.91, 0.61, 0.57
PCPU(%): 99.67, 99.61 ; used total: 99.64
CCPU(%): 1 us, 1 sy, 92 id, 6 wa ; cs/sec: 106
  
```

ID	GID NAME	NMEM	%USED	%SYS	%OVLP	%RUN	%WAIT	%BWAIT	%TWAIT	%CRUN	%CSTP	%IDLE	%RDY	%EXTRA	%MLMTD
1	1 idle	2	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	199.51	0.00	0.00
2	2 system	5	0.00	0.00	0.00	0.00	498.83	0.00	498.83	0.00	0.00	0.00	0.00	0.00	0.00
6	6 console	1	2.92	0.01	0.01	2.93	94.85	0.00	94.85	0.00	0.00	94.85	2.06	0.00	0.00
7	7 helper	13	0.01	0.00	0.00	0.01	1299.49	0.00	1299.49	0.00	0.00	0.00	0.08	0.00	0.00
8	8 drivers	13	0.01	0.00	0.00	0.01	1299.80	0.00	1299.80	0.00	0.00	0.00	0.00	0.00	0.00
9	9 vmotion	1	0.00	0.00	0.00	0.00	100.00	0.00	100.00	0.00	0.00	0.00	0.00	0.00	0.00
12	12 vmware-vmkauthd	1	0.00	0.00	0.00	0.00	100.00	0.00	100.00	0.00	0.00	0.00	0.00	0.00	0.00
13	13 instructor	5	98.56	0.00	0.04	98.56	397.78	0.00	397.78	0.00	0.00	0.00	3.66	0.00	0.00
14	14 server1	5	98.29	0.00	0.05	98.34	398.24	0.00	398.24	0.00	0.00	0.00	3.41	0.00	0.00

**Interpretation:**

With two VM's running the script my PCPU's are beginning to max out at 99.67 and 99.61. Now the VM called instructor is as busy as server1. The %RDY value has increased but not massively so. With just two VM's with a

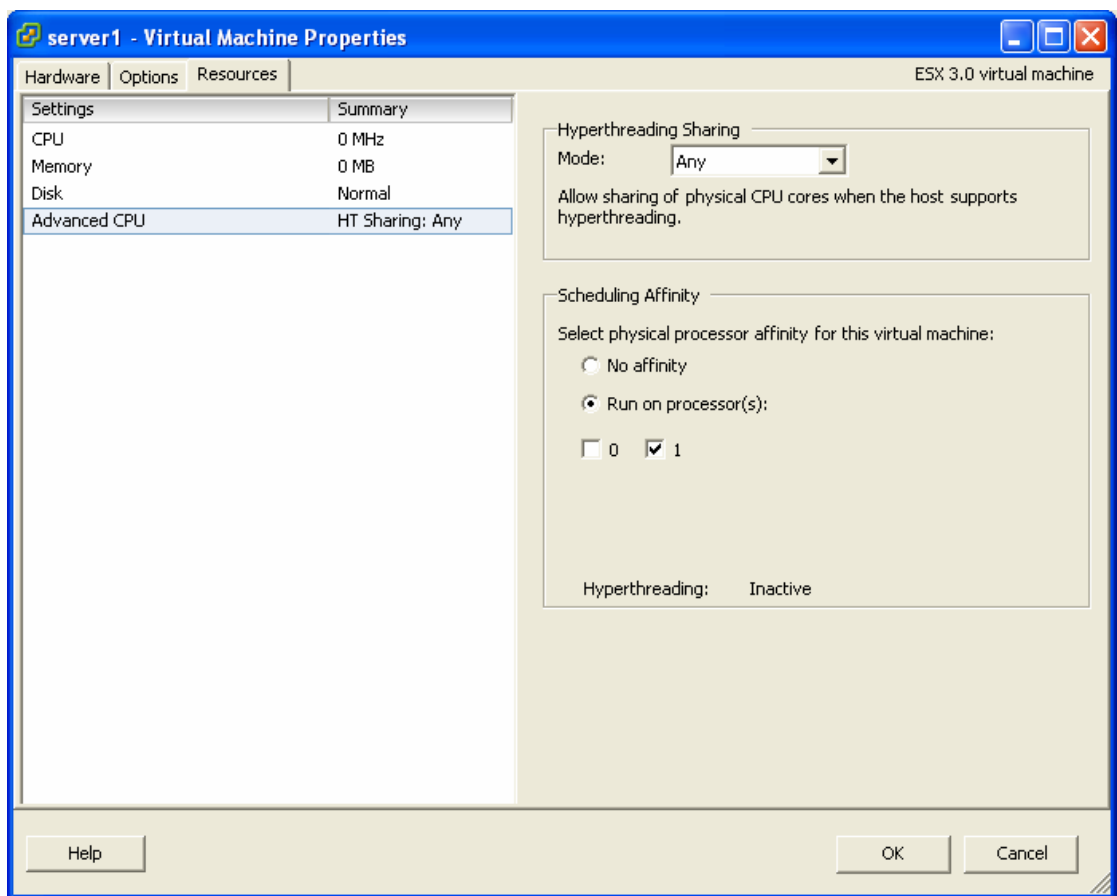
vCPU each the vmkernel is analyzing the cpu performance to make sure the two VM's don't fight against each other...

### Deliberate Mis-configuration – Creating contention

**Note:**

- By default a single vCPU VM runs on any CPU that has free CPU cycles...
- We can use "processor affinity" to cause contention – by pegging each VM to the same CPU

1. **Do this on BOTH VM's...**
2. **Properties of the Virtual Machine**
3. Select the **Resources Tab**
4. Choose **Advanced CPU**
5. Under **Scheduling Affinity**, choose **Run on processor(s)**
6. **Select just one CPU for the VM** like so:



7. Click **OK**

### Run ESXTOP at the Service Console

```

5:05:58pm up 3:05, 46 worlds; CPU load average: 1.24, 1.08, 0.89
PCPU(%):  2.88, 100.00 ;   used total:  51.44
CCPU(%):  0 us,  1 sy, 96 id,  2 wa ;   cs/sec:  114

```

ID	GID	NAME	NMEM	%USED	%SYS	%OVLAP	%RUN	%WAIT	%BWAIT	%TWAIT	%CRUN	%CSTP	%IDLE	%RDY	%EXTRA	%MLMTD
1	1	idle	2	97.13	0.00	0.00	1.41	0.00	0.00	0.00	0.00	0.00	0.00	198.57	0.00	0.00
2	2	system	5	0.00	0.00	0.00	0.00	499.99	0.00	499.99	0.00	0.00	0.00	0.00	0.00	0.00
6	6	console	1	2.80	0.00	0.04	2.80	1.43	95.76	97.18	0.00	0.00	97.16	0.01	0.00	0.00
7	7	helper	13	0.01	0.00	0.00	0.01	1299.95	0.00	1299.95	0.00	0.00	0.00	0.02	0.00	0.00
8	8	drivers	13	0.01	0.00	0.00	0.01	1299.97	0.00	1299.97	0.00	0.00	0.00	0.00	0.00	0.00
9	9	vmotion	1	0.00	0.00	0.00	0.00	100.00	0.00	100.00	0.00	0.00	0.00	0.00	0.00	0.00
12	12	vmware-vmkauthd	1	0.00	0.00	0.00	0.00	100.00	0.00	100.00	0.00	0.00	0.00	0.00	0.00	0.00
13	13	instructor	5	50.08	0.00	0.02	50.09	397.87	0.00	397.87	0.00	0.00	0.00	52.03	0.00	0.00
14	14	server1	5	49.89	0.00	0.02	49.91	397.72	0.00	397.72	0.00	0.00	0.00	52.36	0.00	0.00

### Interpretation:

Here my CPU0 is no longer busy, but CPU1 is max'd out at 100.00% Both VM's are getting an equal share of the CPU at 50.08% and 49.89%. The %RDY value has grown massively to 50%. This is contention and it's bad... VMware say anything over 5-10% needs dealing with.

### So what our options?

VMotion one of these VM's to less busy ESX host...

Increase shares to instructor because it is more important

Decrease shares to server1 because it is less important

Remove the contention – the CPU overlap we have just created

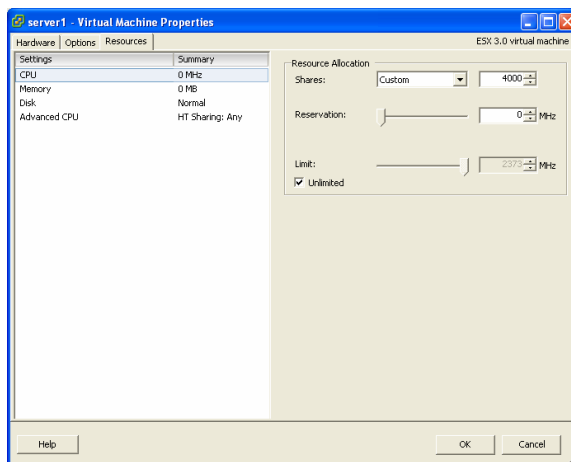
### Increasing Shares on VM

#### Note:

Shares are method of favouring on VM over others when contention takes place. No contention. The share values do nothing – and VM's scream along as fast as they can. But when contention occurs resources are divvied up according to the share value

In my case I am going to give the instructor a 3000 shares, and the other VM left at 1000. That's 4000 shares altogether so the instructor VM will receive  $\frac{3}{4}$  of CPU time, and server1 will get  $\frac{1}{4}$

1. Right-click one of your VM's
2. **Edit Settings**
3. Click the **Resources Tab**
4. Select **CPU** from the list of settings
5. Next to Shares, choose **Custom**
6. Increase to **4000**



### Run ESXtop at the Service Console

```

6:32:02pm up 4:31, 46 worlds; CPU load average: 1.08, 1.06, 0.69
PCPU(%): 7.30, 99.98 ; used total: 53.64
CCPU(%): 5 us, 2 sy, 92 id, 2 wa ; cs/sec: 77

```

ID	GID	NAME	NMEM	%USED	%SYS	%OVRLP	%RUN	%WAIT	%BWAIT	%TWAIT	%CRUN	%CSTP	%IDLE	%RDY	%EXTRA	%MLMTD
1	1	idle	2	92.71	0.00	0.00	1.47	0.00	0.00	0.00	0.00	0.00	0.00	198.54	0.00	0.00
2	2	system	5	0.00	0.00	0.00	0.00	500.00	0.00	500.00	0.00	0.00	0.00	0.00	0.00	0.00
6	6	console	1	7.22	0.00	0.05	7.22	1.49	91.28	92.77	0.00	0.00	92.75	0.01	0.00	0.00
7	7	helper	13	0.01	0.00	0.00	0.01	1299.95	0.00	1299.95	0.00	0.00	0.00	0.01	0.00	0.00
8	8	drivers	13	0.01	0.00	0.00	0.01	1299.95	0.00	1299.95	0.00	0.00	0.00	0.00	0.00	0.00
9	9	vmotion	1	0.00	0.00	0.00	0.00	99.99	0.00	99.99	0.00	0.00	0.00	0.00	0.00	0.00
12	12	vmware-vmkauthd	1	0.00	0.00	0.00	0.00	99.99	0.00	99.99	0.00	0.00	0.00	0.00	0.00	0.00
15	15	server1	5	20.01	0.00	0.01	20.02	395.77	0.00	395.77	0.00	0.00	0.00	84.17	0.00	0.00
16	16	instructor	5	80.14	0.00	0.03	79.98	397.51	0.00	397.51	0.00	0.00	0.00	22.47	0.00	0.00

### Interpretation:

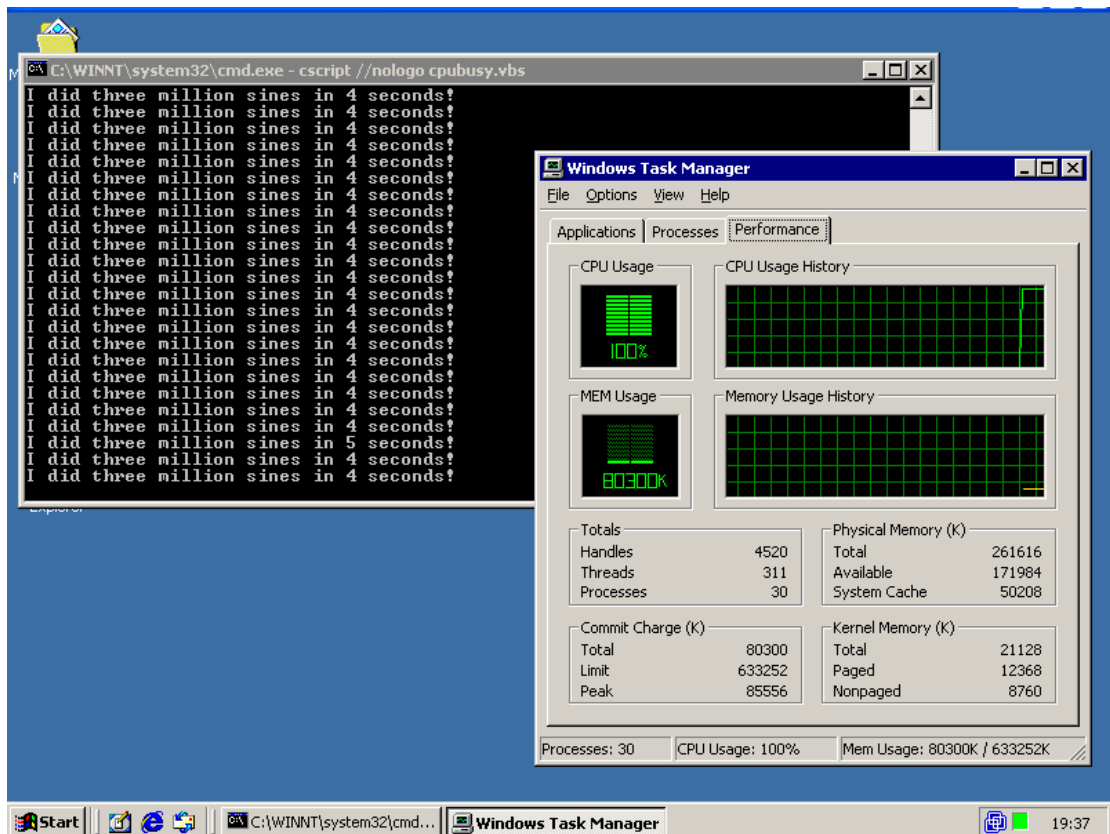
My physical CPU is just as busy. The Instructor vm is getting 80% of CPU ( $\frac{3}{4}$ ) and server1 is getting just 20% ( $\frac{1}{4}$ ). The %RDY value is reflecting an inverse proportion. So as Instructor is getting more CPU time is %RDY value is lower (22%) than server2, which is getting less CPU time (84%)

### Conclusions

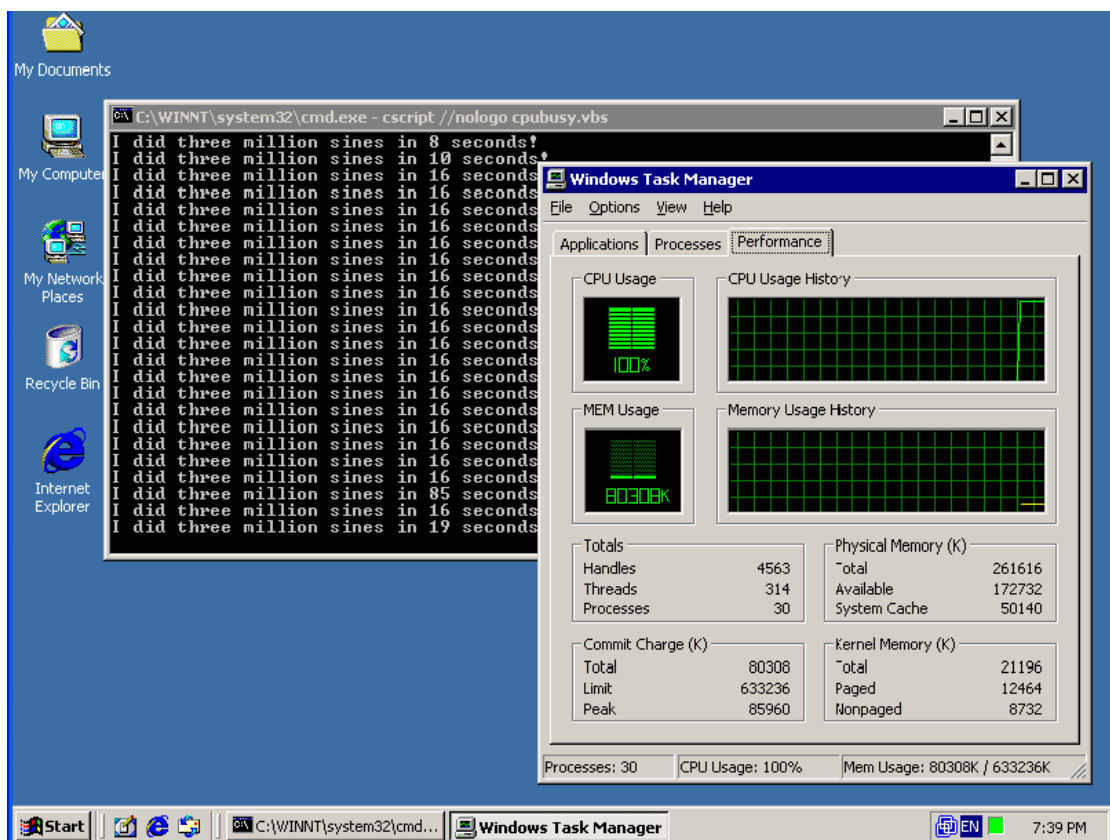
- High CPU values in esxtop in themselves are not a problem – in fact the higher the better. What’s the point of virtualisation if the server only runs at 5% of its capacity.

Or put another way...what’s the point of building an oil pipeline that only ships 5% of oil. The rest is spare and wasted capacity that could be used to increase output

- What matters is bottlenecks - %RDY values are good indicator of bottleneck in our pipeline
- Don’t trust what the guest operating system says. Both of my VM’s report 100% CPU usage – when actually they are using 80% ( $\frac{3}{4}$ ) and 20% ( $\frac{1}{4}$ ) of the processor
- See screen dumps on the next page...



Instructor with 4000 shares...



Server1 with 1000 shares

mmm... I'm looking at the time on the screen dumps here. I should call it day...

## ***Tools to Monitor the Service Console***

### **Note:**

Whereas a poor performing VM's will affect your users, a poor performing Service Console will not. It will however, annoy the pants of you ☺

### **Symptoms include:**

- Random disconnects from ESX and VirtualCenter
- Failure of ESX services to start or stay started
- Incomplete logging
- General sluggishness of management tools
- General annoyance and frustration – sometimes so bad you say out loud

“God I wish I bought Microsoft Virtual Server, then I would have good reason for knowing why things don't work”... ☺

### **CPU**

- **top** is a Linux tool which show you processes and process ID's
- You can use the pidid value or the command name to terminate a process if you think it has gone wrong
- **Killall -9 pidid** will kill a running process and it process tree
- **ps -ef | grep vmware** – would give you a full process list, filtering on the string vmware

### **Memory**

- **free -m** will list how much RAM has been allocated to the Service Console. Interesting on my upgrade ESX 2.x box the amount memory was left at 256MB. On a cleanly install ESX 3.x box its around the 500MB mark
- **vmstat 5 10** will give virtual memory/swap statistics – every 5secs in my case with 10 samples. Watch out for s/i and s/o any regular swap activity indicates that Service Console lacks RAM... just like in Windows

### **Disk**

- The likelihood of filling / is now much less but it could still happen
- Due to excessive logging in /var/log or agent logging in /opt. If these are directories rather partitions you might have problem
- Also if some miss uses vmkfstools -i they could be creating very big files in the wrong location
- You can use **df -h** or **vdf -h** to see how full your file systems are



# Module 8: Upgrading/Patching ESX

## Pre-Upgrade Checks

### Note:

If you are doing an in place upgrade of ESX 2.x check out this script. Of course, you could just re-install the COS, and keep your VMFS2 partition, and just upgrade the file system...

1. **Insert your ESX 3.x CD** into the ESX 2.x server
2. Type:

```
perl /mnt/cdrom/scripts/preupgrade.pl
```

This should give you a print out report of the status of the ESX server

## Custom Upgrading with a Tar-Ball

### Note:

- This is a straight cut & paste from my "upgrade" guide for ESX 3.x and VirtualCenter 2.x
- You can use a "tar ball" to either upgrade from ESX 2.x to 3.x OR... when VMware release a maintenance release of ESX 3.x – it might be a method of patching ESX 3.x
- Why do your upgrades this way? Well, it allows for unattended upgrades – without having to be physically present at the server or having to run around from one server to another with CD!
- This is a two-stage process – involving two scripts and reboot between script1 and 2
- I copied the Tar-Ball to area of free-space this happened to be my /vmimages location (I was concerned my /tmp partition might be too small having set the VMware recommended size of 512MB in ESX 2.x)
- Upgrading ESX in this way didn't seem to help with my virtual networking – I still had some machine which thought they were on the "production" switch – and others thought they were connected to a "Legacy Bond0" – even though I never had a bond.

### Run First Script & Set ESX Boot Option

1. **Transfer the Tar** file to the **ESX 2.x Host and** using WinSCP
2. **Logon to the Service Console with PuTTY, as ROOT**

3. **Untar the tar.gz** file with:

```
tar -xvzf esx-3.?.?-?????-upgrade.tar
```

### Note:

This will take some time!

4. **Change the boot option and Reboot ESX**

```
lilo -R linux-up  
reboot
```

### Note:

When you reconnect to the Service Console, confirm you have not loaded the Vmkernel. An easy check is to ls /vmfs. No Vmkernel, No VMFS... ☺

5. **cd** in to the **esx-3.?.?-?????-upgrade directory**
6. Run the Perl script called

**./upgrade.pl**

**Note:**

The system will verify files, and then ask you to read & accept an EULA!

7. Choose **Q**, to **quit reading the EULA**, and
8. Type **Yes** [Enter] to **accept the EULA**

**Note:**

You will receive status messages like so:

**Upgrading packages (this may take a while) ...  
.... done**

**Removing obsolete packages .... done**

**Upgrading system configuration ...**

**Running vmware-config.pl ... done**

**then lastly**

**\*\*\* You must now reboot the system into Service Console mode  
and run upgrade2.pl \*\*\***

**Reboot the server now?[y/n]**

9. At the **Reboot the Server now [y/n]** choose **N** and **[Enter]**
10. **Now edit the boot.conf file** to make sure the server boots to the Service Console with:

**nano -w /boot/grub/grub.conf**  
change default=0 to **default=1**

11. **Now do the reboot** with the command:

**reboot**

**Note:**

You might find you loose connectivity to the Service Console. I did on my first attempt. The second time I didn't get a script error – and my Service Console networking was intact. The interesting thing is the upgrade was done on an identical ESX 2.x host. I know it was identical – as it was the same physical machine cloned with ghost which contain an image of ESX. Perhaps these network errors were related to the boot errors?

### **Run Second Script & Revert to ESX Boot Option**

1. **Logon to the Service Console with PuTTY, as ROOT**

**Note:**

You will know you have boot to do the Service Console because of this message.

ALERT [Wed May 17 21:19:45 GMT 2006]:  
This VMware ESX Server machine is currently booted into troubleshooting mode.

2. **cd** in to the **esx-3.?.?-?????-upgrade directory**
3. **Run the Perl script** called

**./upgrade2.pl**

**Note:**

You will receive this status messages like so:

**This script will complete upgrading your ESX server to version 3.x.x**

**Verifying Files... don**

**Upgrading Packages (this may take a while)....**

**INIT: Version X.x.x reloading**

**...done**

**\*\*\* You must now reboot the system into the ESX Server mode to finish the upgrade \*\*\***

4. At the **Reboot the Server now [y/n]** choose **N** and **[Enter]**
5. **Now edit the boot.conf file** to make sure the server boots to the Service Console with:

**nano -w /boot/grub/grub.conf**  
change default=1 to **default=0**

6. **Now do the reboot** with the command:

**reboot**

**Note:**

This will boot the ESX 3.x Vmkernel for the first time. The remote upgrade is complete.

**Note:**

If you wish to automate the change to grub.conf, this script is from [IBM](#) is called bootcontrol.pl you can download the [zip file](#) directly here. The very first time you run this command it returns an error which states "Use of uninitialized value in numeric eq (==) at bootcontrol.pl line 99".

It does actually shift the boot option - and the error only occurs once. I've emailed the guys who wrote it - but I have a feeling its cause by our partially completed upgrade process. This command re-compiles the grub loader so that the Service Console Only (troubleshooting only) menu is chosen. It does that by changing the value in grub.conf called Default 0 to being Default 1

## ***Upgrading to VMFS-3 from the Service Console***

### **Acknowledgments:**

I would like to thank Mostafa Khalil, VCP (SE) for his "Troubleshooting" presentation at VMware TSX in Paris 2006.

### Gotchas

- You need free space to upgrade to VMFSv3
- Watch out for block sizes. VMFSv2 support a max block size of 16MB. Whereas VMFSv3 supports 8MB. You cannot do an upgrade of VMFS2 to VMFS3 if this is the case. You in backup and restore territory. Hopefully, this will effect a very few number of people

### Note:

- From man pages of vmkfstools.

"VMFS-2 to VMFS-3 file system upgrade is a two step process. Before file system upgrade can begin the vmfs2 and vmfs3 driver must be unloaded and the auxiliary file system driver, fsaux, should be loaded. The first step of upgrade uses the -T option. Once the first step completes, the auxiliary file system driver, fsaux, is unloaded and, vmfs2 and vmfs3 drivers are reloaded. The second step of file system upgrade makes use of the -u option."

#### **-T, --tovmfs3**

converts a VMFS-2 file system on the specified partition to VMFS-3 format, preserving all files on the file system. The conversion is in-place and the auxiliary file system driver (fsaux) module must be loaded. The ESX Server file system locking mechanisms will try to ensure that no local process or remote ESX Server is currently accessing the VMFS file system to be converted. The conversion is a one-way operation and once the VMFS-2 file system is converted to VMFS-3 file system, it cannot be rolled back to VMFS-2.

#### **-u, --upgradefinish /vmfs/volumes/<label/UUID>/**

once the first step of file system upgrade has completed (using -T), the vmfs2 and vmfs3 modules are reloaded and the -u option is used to complete the upgrade

1. **Logon to the Service Console as ROOT**
2. **Unload the vmfs2 driver** with:

```
vmkload_mod -u vmfs2
```

3. **Unload the vmfs3 driver** with:

```
vmkload_mod -u vmfs3
```

4. **Load the FS Auxiliary Driver** with the upgrade function

```
vmkload_mod fsaux fsauxFunction=upgrade
```

5. Run the first stage of the upgrade with

```
vmkfstools -T /vmfs/volumes/local -x zeroedthick
```

### Note:

-x zeroedthick (default) . Retains the properties of VMFS-2 thick files. With

the zeroedthick file format, disk space is allocated to the files for future use and the unused data blocks are not zeroed out.

-x eagerzeroedthick . Zeroes out unused data blocks in thick files during conversion. If you use this sub-option, the upgrade process might take much longer than with the other options.

-x thin . Converts the VMFS-2 thick files into thin-provisioned VMFS-3 files. As opposed to thick file format, the thin-provisioned format doesn't allow files to have extra space allocated for their future use, but instead provides the space on demand. During this conversion, unused blocks of the thick files are discarded.

**Note:**

This will give you the following message

```
I got /vmfs/volumes/44a38c72-156b2590-be15-00065bec0eb7
```

```
VMware ESX Server Question:
```

```
Please make sure that the VMFS-2 volume /vmfs/volumes/44a38c72-156b2590-be15-00065bec0eb7 is not in use by any local process or any remote ESX server.
```

```
We do recommend the following:
```

1. Back up data on your volume as a safety measure.
2. Take precautions to make sure multiple servers aren't accessing this volume.
3. Please make sure you have approximately 1200MB of free space on your volume. Note that the number is an upper bound, as the actual requirements depend on physical layout of files.

```
Continue converting VMFS-2 to VMFS-3?
```

```
0) Yes
```

```
1) No
```

6. Type **0**, and press [**Enter**]

**Note:**

Currently, you get no status information while this is proceeding. It should complete with this message:

```
Filesystem upgrade step one completed. Step two must be completed after the vmfs2 and vmfs3 modules are reloaded. When ready, run 'vmkfstools -u /vmfs/volumes/local' to complete the upgrade.
```

7. Once this part has completed confirm that your format is VMFS-3

```
vmkfstools -P /vmfs/volumes/local
```

Note:

This should report something like this:

```
VMFS-3.21 file system spanning 1 partitions.
```

```
File system label (if any): local
```

```
Mode: public
```

```
Capacity 36238786560 (34560 file blocks * 1048576), 5511315456 (5256
```

```
blocks) avail
UUID: 44a38c72-156b2590-be15-00065bec0eb7
Partitions spanned:
vmhba0:1:0:1
```

8. Confirm you files haven't disappeared in the process with:

```
ls -l /vmfs/volumes/local
```

9. **Next, unload the Auxiliary Files System Driver and reload you VMFS2 and VMFS3 driver** with:

```
vmkload_mod -u fsaux
vmkload_mod vmfs2
vmkload_mod vmfs3
```

**Note:**

You should get "Module load of vmfs2succeeded" and "Module load of vmfs3succeeded"

10. Restart the hostd service for these changes to be reflected in the VI client with:

```
service mgmt-vmware restart
```

**Note:**

The guide finishes here – if you were doing an upgrade I would progress to Module 4 of my upgrade guide