

# Top Storage Considerations in VMware ESX Environments

**RTFM Education**

Beyond the Manual... with Mike Laverick

By Mike Laverick  
© RTFM Education

Sponsored by



Contact:  
[mikelaverick@rtfm-ed.co.uk](mailto:mikelaverick@rtfm-ed.co.uk)

# Contents

|  |    |
|--|----|
| Contents.....  | 2  |
| Introduction .....   | 3  |
| The Disk Partitioning of an ESX Host.....                  | 3  |
| VMware Snapshots and Creating/Locating Virtual Disks ..... | 7  |
| Monitoring Snapshot Files.....                             | 10 |
| Conclusions.....   | 11 |
| About the Sponsor .....                                    | 11 |

## Introduction:

The purpose of this whitepaper is to act as a companion to the recent webcast hosted by TechTarget.com and sponsored by Veeam Software. The whitepaper is not intend to restate the same points made in the webcast or be a "transcript" – but to offer a format within which the finer points of some of the technical issues raised in the webinar can be addressed.

You can view the full webcast at:

[http://www.bitpipe.com/detail/RES/1203544362\\_532.html?asrc=RSS\\_BP\\_KABPS\\_TORAGE](http://www.bitpipe.com/detail/RES/1203544362_532.html?asrc=RSS_BP_KABPS_TORAGE)

## The Disk Partitioning of an ESX Host

Like most operating systems, the reliability of the VMware build depends upon decisions made by the person behind the installation. It is critical to have a partition scheme that will protect itself from rapidly filling event log files and from users copying large files to the wrong location. Almost no one in the VMware community uses the "automatic" partition scheme called "Recommended" in the installer. Instead, nearly everyone in the VMware community uses his or her own manual partition scheme based on their own and others' experience.

There is a very long and interesting forum thread within which the VMware community has debated the advantages and disadvantages of various approaches. It was started by Steve Beaver, who is a very good friend of ours. If this interests you, then pop along and have a read. You'll see that there are as many partition schemes as there are people on the forum, and ours is just one example. The thread is called "Taking a poll of the manual partitions people are using for ESX 3.0" and the thread ID is 425022.

<http://www.vmware.com/community/thread.jspa?messageID=425022>

As ESX is based around the Linux/Unix world, the way these partitions are addressed is not with drive letters (C: D: E:) but with folders. You have been able to do something similar to this with Microsoft operating systems since Windows 2000 was released. As for the physical disks themselves, I usually recommend two 36GB or 72GB disks in a mirror.

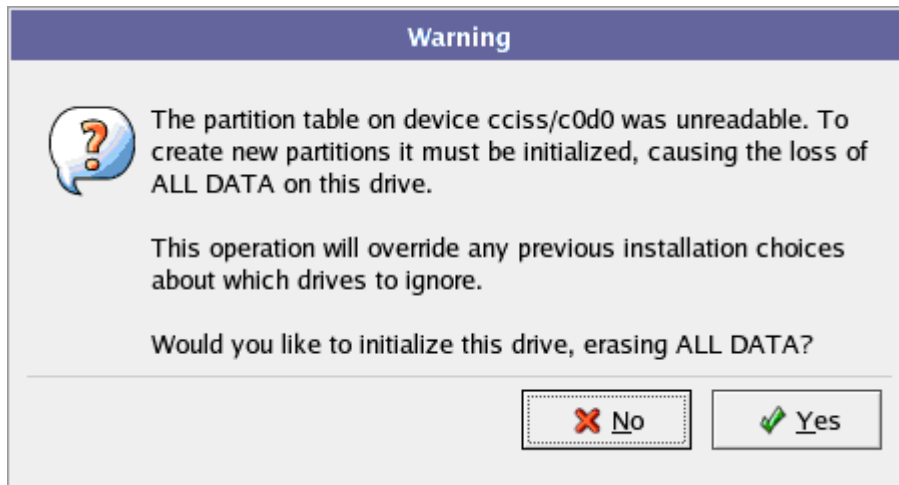
If you are installing ESX to the local storage, I would recommend disconnecting the SAN cables if you can. This prevents any chance of unintentionally installing ESX to SAN. It also reduces your chances of accidentally destroying terabytes of data on the SAN while installing ESX!

After agreeing to the EULA, you may receive a warning about the `/dev/sdaN` being unreadable. This reference to `/dev/sdN` indicates the SCSI Disk (SD) which is found first (A,B,C). Also the dialog box should indicate on which controller this disk was found. In Figure 1 we can see that controller is actually a cciss. This is a Compaq Computer Smart Raid Array 6i on my HP Proliant DL385. `/c0d0` is the first controller (c0) and the first LUN (d0).

Since the partition table doesn't exist, the Anaconda installer will want to initialize the disk. This operation is potentially fatal if there is data on the drive. In this case, it is fine to accept "yes," as I know it is blank. If your ESX host is connected

to SAN, be very careful in accepting “yes,” as this could be very dangerous.

**Figure 1**



Below is a summary of a recommended partition scheme, followed by notes explaining their size and purpose. This partition scheme only uses about 15GB of space – so you could easily increase these values and add more partitions. Generally I create three primary partitions, and the rest are treated as logical drives in an extended partition. The disk management utility (called Disk Druid) automatically creates an extended partition for you using the remainder of the disk once you have created the primaries.

|       | File System | Fixed Size             | Size in MB | Force to Primary | Purpose  |
|-------|-------------|------------------------|------------|------------------|--|
| /boot | ext3        | X                      | 250        | X                | Core Boot (img) files  |
| n/a   | Swap        | X                      | 1600       | X                | Swap for Service Console   |
| /     | ext3        | X                      | 5120       | X                | Main OS location   |
| /var  | ext3        | X                      | 2048       |                  | Log files  |
| /tmp  | ext3        | X                      | 2048       |                  | Temporary Files  |
| /opt  | ext3        | X                      | 2048       |                  | VMware HA Logging  |
| /home | ext3        | X                      | 2048       |                  | Location of users' storage   |
| NA    | vmkcore     |                        | 100        |                  | VMkernel Panic Location  |
|       | Vmfs        | Fill to remaining disk |            |                  | Local storage is only required for VM Clustering – running clustering software within a VM |

**ext3**

EXT3 is the primary file system of Linux. Since the Service Console is based on Red Hat Linux, it is the one we will use. There are two other propriety file systems that are only accessible by the VMkernel. These are vmkcore and VMFS version 3.

**/boot**

This is where core boot files with an img extension are stored. After powering on ESX, the master boot record is located and the boot loader is run. In the case of ESX 3.x this is now the GRand Unified Bootloader (GRUB). GRUB then displays a menu that allows you to select what .img to execute. Img files are image files and are bootable. They are analogous to ISO files, which can also be bootable. I've doubled the VMware recommendation to err on the side of caution. Previous

VMware recommendations have made this partition too small. This gave people problems related to lack of disk space when upgrading from ESX 2.x to ESX 3.x.

### **/swap**

This is where the Service Console swaps files if memory is low. I've chosen to over-allocate this partition. The default amount of memory for the Service Console is 272. VMware usually takes this number and doubles it to calculate the swap partition size (544MB). The maximum amount of memory you can assign to the Service Console is 800MB. This is how I derived the 1600MB value. This means if we ever choose to change the default amount of memory assigned to the Service Console, we don't have to worry about resizing the swap partition. It doesn't have a mounting point, as no files from the administrator are copied there.

### **/ (referred to as the "root" partition)**

This is the main location where the ESX operating system and configuration files are copied. If you are from a Windows background, you can think of it as a bit like the C: partition and folders coming off that drive, such as C:\Windows or C:\Program directory. If this partition fills, you may experience performance and reliability issues with the Service Console, just like you would with Windows, or any other operating system, for that matter.

### **/var**

This is where log files are held. I generally give this a separate partition just to make sure that excessive logging does not fill the / file system. Log files are normally held in /var/log. But occasionally hardware vendors place their hardware management agent log files in /var.

### **/tmp**

In the past, VMware has recommend using a separate partition for /tmp – which I have always done in ESX 2.x as well. Since I have plenty of disk space, I have made this larger than it really needs to be.

### **/opt**

Several Forum members have seen the /opt directory fill up very rapidly and then fill the / partition. This location is also *sometimes* used as a logging location for hardware agents. In VMware, HA has been seen to generate logging data here as well. So I create a separate partition for it, to make sure it doesn't fill the / partition.

### **/home (Optional)**

Technically, you don't need a separate partition. In the past, VMware recommended one for its ESX 2.x in production. This was due to the fact that VMs' configuration files, such as the vmx, nvram and log, were stored in /home. In ESX 3.x, all the files that make up a VM are more likely to be located on external storage. I still create it for consistency purposes – and if I have users on the local ESX server, those users are more likely to create files there than in a directory coming off the / partition.

### **vmkcore**

This is a special partition used only if the ESX VMkernel crashes, commonly referred to as a "Purple Screen of Death." If that happens, then ESX writes debugging information into this partition. After a successful boot, the system will automatically run a script to extract and compress the data to a "zip" file in /root. This file with tar.gz extension can be sent to VMware Support who will endeavour to identify the source of the problem. These PSODs are normally caused by failures of RAM or CPU. You can see a rogues' gallery of PSODs at

[http://www.rtfm-ed.co.uk/?page\\_id=246](http://www.rtfm-ed.co.uk/?page_id=246)

## vmfs

VMFS is VMware's ESX native file system that is used for storing all the files of the VM, ISO's and templates. Generally, we use external storage for this. The only case for using local storage for your VMs is when you do not have access to external storage. Here I am assuming you have access to external storage, and therefore you have no need for a local VMFS partition.

## GOTCHA:

There is one major qualification to this statement. If you want to run clustering software, such as Microsoft's Clustering Software, inside a VM - you will need local storage. VMware does not support storing virtual disks or Raw Device Mappings (RDMs) used in VM clustering scenarios on SAN or iSCSI based storage.

## /vmimages

In ESX 2.x we used to create a /vmimages partition, or mounting point, to a network location. This storage was used primarily for templates and ISOs. This partition is no longer required – as we now have more effective and easier ways of storing this data. Of course, if you are an ESX 2.x veteran who wants to keep using /vmimages for consistency purposes, then that's fine. It's just no longer required or recommended. Personally, I still like to have a portion of disk space given over to this partition location as a "working area" when I am dealing with large files. I've found my recent purchase of a SAN has made this something I use much less.

If you have done a clean install, even if you haven't created a /vmimages, you will still have directory called /vmimages – which contains files used internally by VMware ESX.

Click "new," and this will begin the process of creating partitions. Figure 2 shows the new dialog box. In this case I select /boot from the pull-down list (you can type in this area). I chose ext3 as the partition type – and typed 250MB as the size. I left the option as "Fixed" size, and indicated with the check mark that I want this partition to be a primary partition. If I left this unchecked, the system would begin creating logical drives in an extended partition

**Figure 2**

The screenshot shows a dialog box titled "Add Partition". It contains the following fields and options:

- Mount Point:** /boot
- File System Type:** ext3
- Drive:** cciss/c0d0: Compaq Smart Array - 69453 MB
- Size (MB):** 250
- Additional Size Options:**
  - Fixed size
  - Fill all space up to (MB): 250
  - Fill to maximum allowable size
- Force to be a primary partition

At the bottom of the dialog are two buttons: "Cancel" and "OK".

### **GOTCHA:**

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

## **VMware Snapshots and Creating/Locating Virtual Disks**

One of the main recommendations from VMware is that you should, where possible, locate the virtual disks that represent your boot disk, data disk and log files on different LUNs and on different datastores. This allows you to trade off fault tolerance against performance overheads of RAID1, RAID5 and RAID10. For many, this principle is merely an extension of best practices they have been following since running their guest operating system on physical machines. It's a good practice to separate data from OS/Applications for backup and recoverability. In addition, by locating the virtual disk on different LUNs you mitigate against saturating a LUN with excessive IOPS.

It's worth stating that for many people, their storage problems are to be found not inside the VM or in the cabling that connects their ESX host to their SAN/iSCSI/NAS system. Quite often, the performance problems reside at the source – the configuration of the array itself. This can be due to many factors, including:

- Wrong RAID level being selected
- Too many or too few disk spindles for a given RAID type
- Insufficient memory cache on the array

So far so good. Unfortunately, what VMware fails to tell you is that if you follow this practice of distributing your VMs across LUN it conflicts with a popular Vi3 feature called snapshots. Snapshots can be engaged manually, but are more commonly used during backup. Virtual machine files with a snapshot engaged are "unlocked" in the VMFS file system, which means that a VM can be backed up while it is running. The problem, which is, for want for a better word, a bug, hinges on VMware's adoption of an automatic file naming convention for virtual disks. This problem is a known issue, has been present since ESX 3.0.0, and unfortunately, does not appear to have been fixed in subsequent releases. You can find the KB article from VMware acknowledging this here:

<http://kb.vmware.com/kb/5096672>

Note: This problem has been resolved in the 3.5 release.

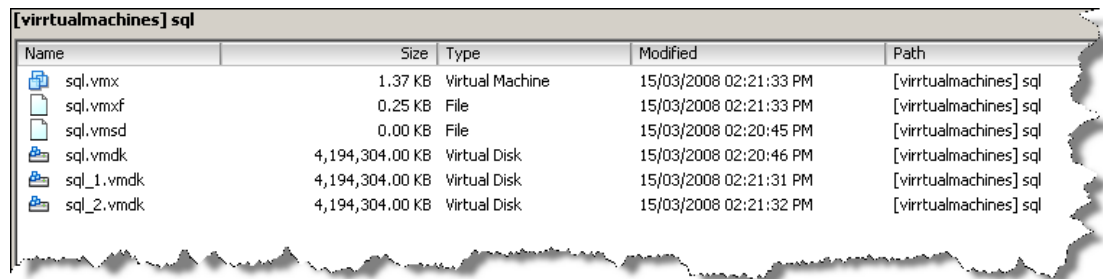
The bug and the work-around are perhaps best explained with an example. Let's say you create a VM with three virtual disks all stored in the same VMFS datastore. The Vi3 client does not allow you to name your own files. This was the case in VirtualCenter 1.x.x. Instead, it auto-generates the file names based on the VM names in the inventory. Suppose you create a VM called "sql" on a VMFS volume called /vmfs/volumes/virtualmachines. This would create a directory called /vmfs/volumes/virtualmachines/sql. After creating this VM, you then decide to edit its settings and add two additional disks. The result would be a number of

smaller files and our all-important virtual disks. The file names of these virtual disks would be:

```
/vmfs/volumes/virtualmachines/sql.vmdk  
/vmfs/volumes/virtualmachines/sql_1.vmdk  
/vmfs/volumes/virtualmachines/sql_2.vmdk
```

Figure 3 shows a screen grab from the Vi Client “Datastore Browser”

**Figure 3**

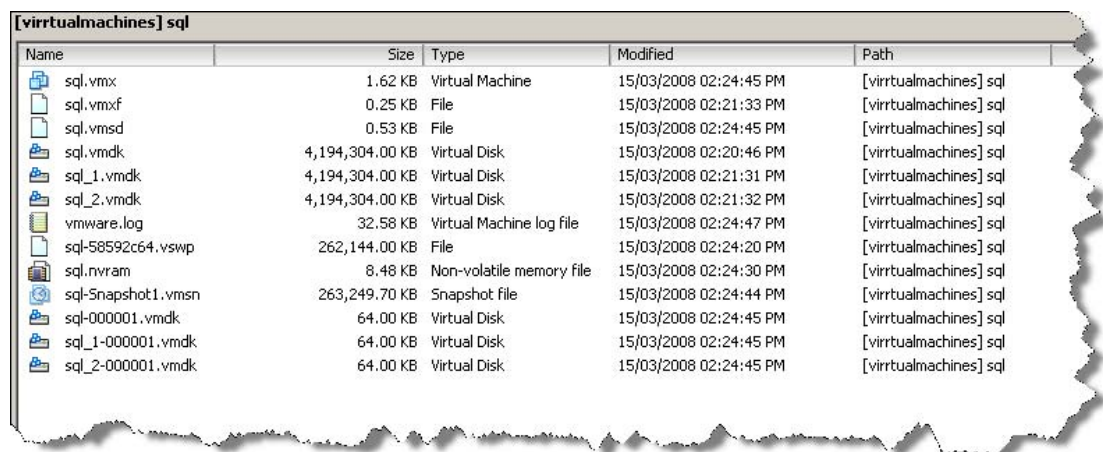


| Name       | Size            | Type            | Modified               | Path                  |
|------------|-----------------|-----------------|------------------------|-----------------------|
| sql.vmx    | 1.37 KB         | Virtual Machine | 15/03/2008 02:21:33 PM | [virtualmachines] sql |
| sql.vmx.f  | 0.25 KB         | File            | 15/03/2008 02:21:33 PM | [virtualmachines] sql |
| sql.vmsd   | 0.00 KB         | File            | 15/03/2008 02:20:45 PM | [virtualmachines] sql |
| sql.vmdk   | 4,194,304.00 KB | Virtual Disk    | 15/03/2008 02:20:46 PM | [virtualmachines] sql |
| sql_1.vmdk | 4,194,304.00 KB | Virtual Disk    | 15/03/2008 02:21:31 PM | [virtualmachines] sql |
| sql_2.vmdk | 4,194,304.00 KB | Virtual Disk    | 15/03/2008 02:21:32 PM | [virtualmachines] sql |

As you can see, VirtualCenter rather unpleasantly just serializes the filenames, which is not especially friendly. These filenames don't really tell you anything about what data is contained inside them. Putting that somewhat harsh observation to one side, how would snapshots work in this case? When you create a snapshot (either manually or automatically) the snapshot process generates a series of snapshot “delta” files at the location where the VMs VMX file is located. In this case, each virtual disk would receive its own delta file. There wouldn't be a problem, although many of us might become anxious about the amount of free space available. The names of the delta files would be based upon the virtual disk file name – these would be unique because all the file names of the virtual disks are unique.

Figure 4 shows the three delta files – sql-000001.vmdk, sql\_1-000002.vmdk and sql\_2-000001.vmdk

**Figure 1.4**



| Name               | Size            | Type                     | Modified               | Path                  |
|--------------------|-----------------|--------------------------|------------------------|-----------------------|
| sql.vmx            | 1.62 KB         | Virtual Machine          | 15/03/2008 02:24:45 PM | [virtualmachines] sql |
| sql.vmx.f          | 0.25 KB         | File                     | 15/03/2008 02:21:33 PM | [virtualmachines] sql |
| sql.vmsd           | 0.53 KB         | File                     | 15/03/2008 02:24:45 PM | [virtualmachines] sql |
| sql.vmdk           | 4,194,304.00 KB | Virtual Disk             | 15/03/2008 02:20:46 PM | [virtualmachines] sql |
| sql_1.vmdk         | 4,194,304.00 KB | Virtual Disk             | 15/03/2008 02:21:31 PM | [virtualmachines] sql |
| sql_2.vmdk         | 4,194,304.00 KB | Virtual Disk             | 15/03/2008 02:21:32 PM | [virtualmachines] sql |
| vmware.log         | 32.58 KB        | Virtual Machine log file | 15/03/2008 02:24:47 PM | [virtualmachines] sql |
| sql-58592c64.vswp  | 262,144.00 KB   | File                     | 15/03/2008 02:24:20 PM | [virtualmachines] sql |
| sql.nvram          | 8.48 KB         | Non-volatile memory file | 15/03/2008 02:24:30 PM | [virtualmachines] sql |
| sql-Snapshot1.vmsn | 263,249.70 KB   | Snapshot file            | 15/03/2008 02:24:44 PM | [virtualmachines] sql |
| sql-000001.vmdk    | 64.00 KB        | Virtual Disk             | 15/03/2008 02:24:45 PM | [virtualmachines] sql |
| sql_1-000001.vmdk  | 64.00 KB        | Virtual Disk             | 15/03/2008 02:24:45 PM | [virtualmachines] sql |
| sql_2-000001.vmdk  | 64.00 KB        | Virtual Disk             | 15/03/2008 02:24:45 PM | [virtualmachines] sql |

Things begin to unravel once the virtual disks are stored on three different VMFS datastores. In this case the naming convention is not imposed. So if I had VMFS volumes called LUN1, LUN2 and LUN3 – I would have three virtual disks created. They would be named

/vmfs/volumes/lun1/sql/sql.vmdk  
/vmfs/volumes/lun2/sql/sql.vmdk  
/vmfs/volumes/lun3/sql/sql.vmdk

Figure 5 shows that on each LUN/VMFS volume a separate sql directory is created with the file names each being sql.vmdk

**Figure 5**

The figure consists of three screenshots of file explorer windows, each showing the contents of a directory named 'sql' on a different LUN. The windows are titled '[LUN1] sql', '[LUN2] sql', and '[LUN3] sql'. Each window displays a table with the following columns: Name, Size, Type, Modified, and Path.

| Name      | Size            | Type            | Modified               | Path       |
|-----------|-----------------|-----------------|------------------------|------------|
| sql.vmx   | 1.47 KB         | Virtual Machine | 15/03/2008 02:37:49 PM | [LUN1] sql |
| sql.vmx.f | 0.25 KB         | File            | 15/03/2008 02:37:48 PM | [LUN1] sql |
| sql.vmsd  | 0.00 KB         | File            | 15/03/2008 02:36:37 PM | [LUN1] sql |
| sql.vmdk  | 4,194,304.00 KB | Virtual Disk    | 15/03/2008 02:36:40 PM | [LUN1] sql |

| Name     | Size            | Type         | Modified               | Path       |
|----------|-----------------|--------------|------------------------|------------|
| sql.vmdk | 4,194,304.00 KB | Virtual Disk | 15/03/2008 02:37:46 PM | [LUN2] sql |

| Name     | Size            | Type         | Modified               | Path       |
|----------|-----------------|--------------|------------------------|------------|
| sql.vmdk | 4,194,304.00 KB | Virtual Disk | 15/03/2008 02:37:48 PM | [LUN3] sql |

As you can see, the file names are all the same – and the serialization process is switched off. The follow-on effects on VMware snapshots are significant. Many customers would assume, in the case above, that the snapshot process will create three sets of deltas on LUN1, LUN2 and LUN3. This is not the case. What the system tries to do is create three deltas at the location of the VMX file. As the filenames are no longer unique, the snapshot fails to be created uniquely, as its file name is based upon the virtual disk filename. Although the snapshot files are created, they cannot be deleted, as acknowledged in the VMware KB 5096672.

The long-term fix for this issue lies firmly at the door of VMware. As mentioned before, it is a known issue and there is a KB article that explains the vulnerability and the work-around until the issue is patched. The simple solution to this issue is to rename the files associated with the VM so they are meaningful and unique. I prefer a naming convention like this:

sql-boot.vmdk  
sql-log.vmdk  
sql-data.vmdk

Virtual disks actually comprise of two files. The “metadata” file, which is called nameofvirtualmachine.vmdk, and data file itself, called nameofvirtualmachine-flat.vmdk. It is not good practice to rename these files with the mv command, as you normally would. Instead, use the command vmkfstools -E. This allows you rename both files simultaneously and keep them “linked” together. Just so you know, the metadata file is just a text file that “points” to the “flat” file. The metadata includes information such as drive geometries (cylinders, heads,

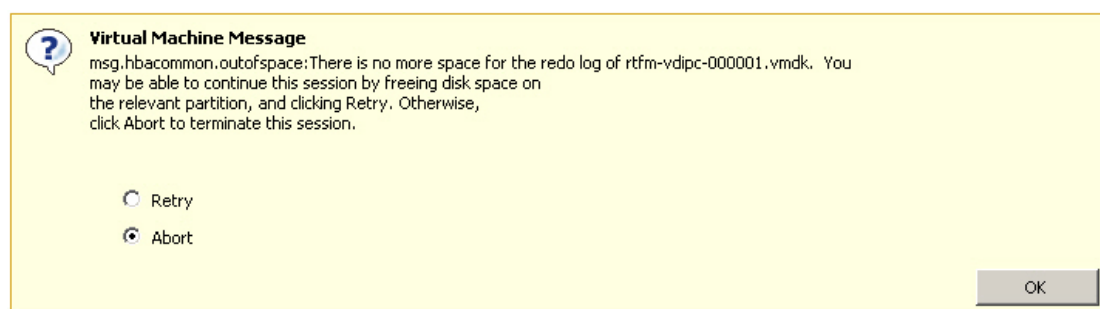
sectors) and other fascinating information such as what controller type (BusLogic or LSILogic) was used during the creation of the virtual disk.

Once the virtual disks have meaningful and unique file names, the snapshot feature will work without error. It still, however, creates the snapshot in the location of the VMX file – not where the virtual disk is located. This functionality is not configurable, and it's somewhat disappointing that we cannot have complete control over the files of VM. Without control over the location of the snapshot, this LUN where the VMX resides will need temporary free space during the period that snapshots are in use. Additionally, it means that this particular LUN will take a great deal of I/O during the period that snapshots are engaged.

## Monitoring Snapshot Files

Unlike most files in VMware's ESX product, snapshots are growing files. They actually increment at a factor of 16MB blocks. Given that the snapshot delta files grow, this presents management challenges. The most critical of these is running out of free space in the VMFS volume where the deltas reside. If you do run out of space, horrible things will happen. The VM will stall and crash as it finds it cannot grow the delta file by another 16MB. Figure 6 shows this happening:

**Figure 6**



You might think I deliberately created this error dialog box. I'm sorry to admit the fact that this happened to me last week. You see an engaged a snapshot on one of my VMs (rtfm-vdipc). Of course, during the day I'm juggling lots of tasks simultaneously, and utterly forgot as I installed software into the VM that this snapshot was engaged. The result? The VM crashed – and I found I could not delete the snapshot and keep my changes. I was forced to revert the snapshot to release the VM from this state. As a consequence I lost all my changes. Fortunately, those changes weren't important, and were easily reproduced. While VirtualCenter does have alerts for running out of disk space, there exists no method of searching for VMs with snapshots, or receiving alerts when a snapshot is reaching the maximum recommended size (2GB). Solutions do exist for this issue, including using the Veeam Reporter tool, which can flag VMs in this state.

Additionally, some enterprising VMware Forum members have written Service Console scripts that search for the files that make up a snapshot, and then use text manipulation to generate a report to the CLI. You can find this script on the Xtravirtl.com website – it's called SnapHunter. Additionally, on the Yellow Brick website there is a free .sh script called "snapcheck." Finally, with the advent of the VMware Update Manager, it is possible to snapshot VMs prior to starting the patch management process. We will have to be careful to set a correct duration for the retention of the snapshots – too short and we wouldn't be able to roll back

after a bad patch has been applied, too long and we could find we run out of space or adversely affect performance.

There are some other storage considerations connected with snapshots in terms of performance. As mentioned earlier, it is not currently possible to select a location for the snapshot files. They are all created at the same location as the VMX file. For this reason, as you might expect, having many VMs with snapshots simultaneously could seriously affect performance.

## Conclusions

To summarize: First, a good partition table on an ESX host will prevent the root / volume filling up, and generally improve the stability and reliability of your ESX host. Second, while it is good practice to distribute the VM disks of across many LUNs – watch out for the bug in the VMware Snapshot feature that resides in older versions of ESX 3. Third, if you are experiencing disk performance problems, check if the VM in question has a snapshot. If not, you might need to look at features of the storage such as RAID levels, number of spindles and volume of memory cache on the array.

I hope you have found this addendum to the webcast useful – fleshing out the more technical aspects that could not be addressed in a webcast. Again, if you have any questions direct them to me at [mikelaverick@rtfm-ed.co.uk](mailto:mikelaverick@rtfm-ed.co.uk)

## ***About the Sponsor: Veeam Software***

Veeam Software, a VMware Technology Alliance Partner best known for its FastSCP product, provides practical, innovative solutions for managing virtual server environments. Veeam Software is led by Ratmir Timashev and Andrei Baronov, the founders of Aelita Software. Today the company offers products including Veeam Backup, the 2-in-1 virtual machine backup and replication solution; Veeam Reporter, an automated way to discover and document VMware ESX virtual server environments; Veeam Configurator, a simple way to configure ESX servers; and Veeam Monitor, for a bird's-eye view of key performance metrics across the entire VMware infrastructure. Learn more about Veeam Software by visiting [www.veeam.com](http://www.veeam.com).