

White Paper:  
VMware ESX REDO Demystified

Document Version 1.2

**RTFM Education**

Beyond the Manual... with Mike Laverick

By Mike Laverick  
© RTFM Education

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

**ESX Version:**

2.x

**Author:**

Mike Laverick 

**Audience:**

Enthusiast to Hotshot

**Style:**

Bit techy with some humour, not a po-faced article. I use unhappy faces ☹ to flag up events and experiences that are less than pleasant. I use a happy face 😊 to flag up something which is very advantageous.

**Objective:**

This article discusses the VMware's shrink feature and how utilising it can massively reduce the size of an exported disk – it then moves on to discuss how to automate this shrink process with SysInternal's "Sdelete" utility.

**Disclaimers & Acknowledgements:**

In this document I express some personal opinions – which may disagree with – this you right. But please don't "flame" me with your disagreements! On the other hand if you feel that there is technical error in this document – then I implore you to tell me so. I don't want to be responsible for any disseminating misinformation in any of my RTFM Guides or White Papers!

Lastly, I wrote this article pretty much off the top of my head. Then I picked up Ron & Scott's book just to check I'd got it down as I wanted to. So credit must go to them for making me feel less paranoid about getting things wrong!

If you want to read more on this topic I would heartily recommend reading VMware's own backup PDF guides. If you have Ron & Scott's book, Chapter 5: Designing the Guest Environment (pages 239-252) covers everything you'd want to know about VMDK/REDO's.

By the way... if you looked for 239 you might find your copy has a white page. Early copies of Ron & Scott's book has the printing error – the missing page can be download from [www.vmguru.com](http://www.vmguru.com)

# VMware ESX REDO Demystified

## How do REDO Files work?

Put very simply - when you engage a disk mode, a REDO file is created. Instead of changes within your Virtual Machine going to the Virtual Disks permanently, changes within the Virtual Machine are redirected to the REDO file instead.

Some debate centres around the terminology. What should the file be called? Is it a REDO "Log" file or should we simply call it the REDO file. A REDO file does not have .log extension, and does NOT contain "log" information in the traditional sense of the word (unlike vmware.log or vmkmessages). It is true that some dialog boxes call it the REDO Log file. Personally, I'm not a huge fan of calling it a REDO Log file - as I feel the term muddies the waters. Anyway, it's just a word - so let's not get too pedantic about shall we?

## ***Should I run on REDO all the time?***

VMware states very clearly that it is NOT recommended to run a production Virtual Machine in REDO mode all the time. So the use of this feature should be only for short-term purposes with the Virtual Machine running on the disk mode of "Persistent" most of the time.

There is a performance hit on accessing the REDO file. The vmKernel accesses the REDO every 16MB of changes. To get exclusive access to the file the vmKernel must apply a temporary "SCSI Reservation". This for a very brief period prevents access to ANY file on VMFS volume. Normally, these reservations only last briefly and are infrequent. Such as when you power-up or power-off a virtual machine. If REDO files the frequency and number of these SCSI reservations are much higher and affect performance.

The REDO file grows incrementally and as it does performance can deteriorate. Like any dynamically growing file you can get fragmentation and unlike the "monolithic format" used by the Virtual Disks - writing to a REDO file is slower.

As the REDO grows the time it takes to apply the changes from the REDO file to the Virtual Disks takes incrementally longer and longer. This will be a significant issue when we come on to look at hot-backups and the use of the REDO file.

If the REDO is allowed to grow incrementally and this is not monitored it could fill the partition and consume all the disk space in the VMFS volume where it resides. I've had this happen to me once or twice, because personally I don't have terabytes of SAN space that some of you enjoy - it is NOT a pleasant experience believe me. ☹

On the subject of REDO file size. VMware documentation has recommended this file should not be allowed to grow beyond 2GB. Later this recommendation was reduced even further to 1GB.

I have heard anecdotally that some users allow the REDO to grow much larger than this. I assume they have excellent physical disk infrastructures and lots of free space to counter-act the performance and storage challenges this must pose.

Like lots of vendor recommendations you can ignore VMware - but doing so is at your own peril. Being a cautious kinda of guy I err on the side of caution. Reading

this back this sounds like tautology. Have you ever heard of cautious guy being reckless?

Lastly, something that VMware doesn't flag up. If you use the "Undoable Disk Mode" you will have prompts to deal with. This can cause you problems if you choose to reboot the ESX Server. Most people will use ESX's Start-up options to automatically power-up and power-down the Virtual Machines when they reboot ESX itself.

However, if the Virtual Machines are in an undoable mode you will have prompts asking how to handle the REDO file attached to the Virtual Machine. Without a human operator to answer these prompts the ESX will either not shutdown properly – or you will find only some of Virtual Machines will NOT powered back on after the ESX reboot. If these Virtual Machine run services which have dependencies you can find yourself having to do a lot of manual reboots in the right order. This is a rather unpleasant experience ☹

By default the REDO file is created with the same name as the Virtual Disk that you have set it on with the REDO extension. It is located in the same VMFS Volume that the Virtual Disk is located on.

### **Test 1: What happens if you accidentally have very Large REDO Files?**

Out of curiosity I created a 20GB Virtual Machine, and switched to undoable mode. I then proceeded to fill the virtual disk with data until Windows said the disk full. This created a 18GB REDO file. I proceeded to commit the REDO. This took about 45-60 minutes using very modest hardware (Dell 1650, 2xCPU PIII 1.41Ghz, 2GB, 10K SCA disks on Adaptec SCSI controller).

### **Conclusions:**

I was lucky it worked. I had plenty of free space in the VMFS when I did this (although I don't think this is a factor). The bigger worry is a REDO flushing the VMFS partition – but I don't think this inevitable – but it is a possibility – and I have experienced it in other situations.

Even on modest hardware committing of large REDO's does work. Other Virtual Machines in the same LUN could be affected by the vmKernel SCSI Reservation required to commit the REDO file to the virtual disk. If you have accidentally created a very large REDO. It might be a good idea to power down, but don't commit – backup the disk and the REDO – and give the commit process a try. If it fails you can go back to the backup and try again.

### **Test 2: Does the REDO get incrementally bigger and bigger?**

In this test I took a 10GB which was 2GB full, and began to fill it with dummy data. Once Windows said the virtual disk was full I looked at its size – the REDO was 6.1GB in size. Next I deleted data in the disk and emptied the Recycle Bin, The REDO remained the same size. So the REDO does not behave like a hard-drive – where if you delete information it then reduces in size.

I then began to fill the disk once again - this time performance seemed to decline – although I have worked with bigger REDO files in the past.

It appears that there is indeed an upper limit on the REDO which is related to the space available in Virtual Disk. At the end of this 2<sup>nd</sup> copy process REDO was 6.5GB in size. I noticed that the REDO file had increased – it has not done this at the same rate as the file copy. That is to say it did not double in size. This did seem to suggest that REDO isn't just a change log. Otherwise changes would grow and grow recording every file copy and file delete. This does seem to

suggest that the REDO cannot grow massively beyond the original virtual disk size.

The more you create and delete files in the Virtual Machine which goes to the REDO the slower the performance. This could be do process endemic in the REDO but also linked to fragmentation and truncation of the REDO file itself. This seems to suggest there is an "upper limit" on the size of REDO – and this upper limit is where performance is especially concern.

I kept on deleting and creating files – filling and emptying the disk. On the third attempt the REDO file seemed to stay the same size. It remained at 6.5 and didn't increase in size.

I then delete the data for the 3<sup>rd</sup> and last time. The REDO was 6.5GB. I applied this very big REDO which contains information that was mainly my deletes!

### **Conclusions:**

The maximum size of the REDO is related to the free space inside of the the virtual disk (notice how in both test the REDO was never greater than amount the of free space in the Virtual Disk). It does grow – but not indefinitely. It eventually reaches this point and does not go beyond it. So the chances of it the REDO flushing the physical disk is related to the size of the virtual disks, and amount of free disk space in the VMFS partition where they reside. When the REDO reaches its maximum size – performance declines significantly. This is why VMware recommends keeping the size of REDO as small as possible.

Anyway, now we have basic description of what REDO does – and some of the caveats surrounding their usage. Let take a look at what they can do for us.

### ***Should the REDO file really be called the UNDO file?***

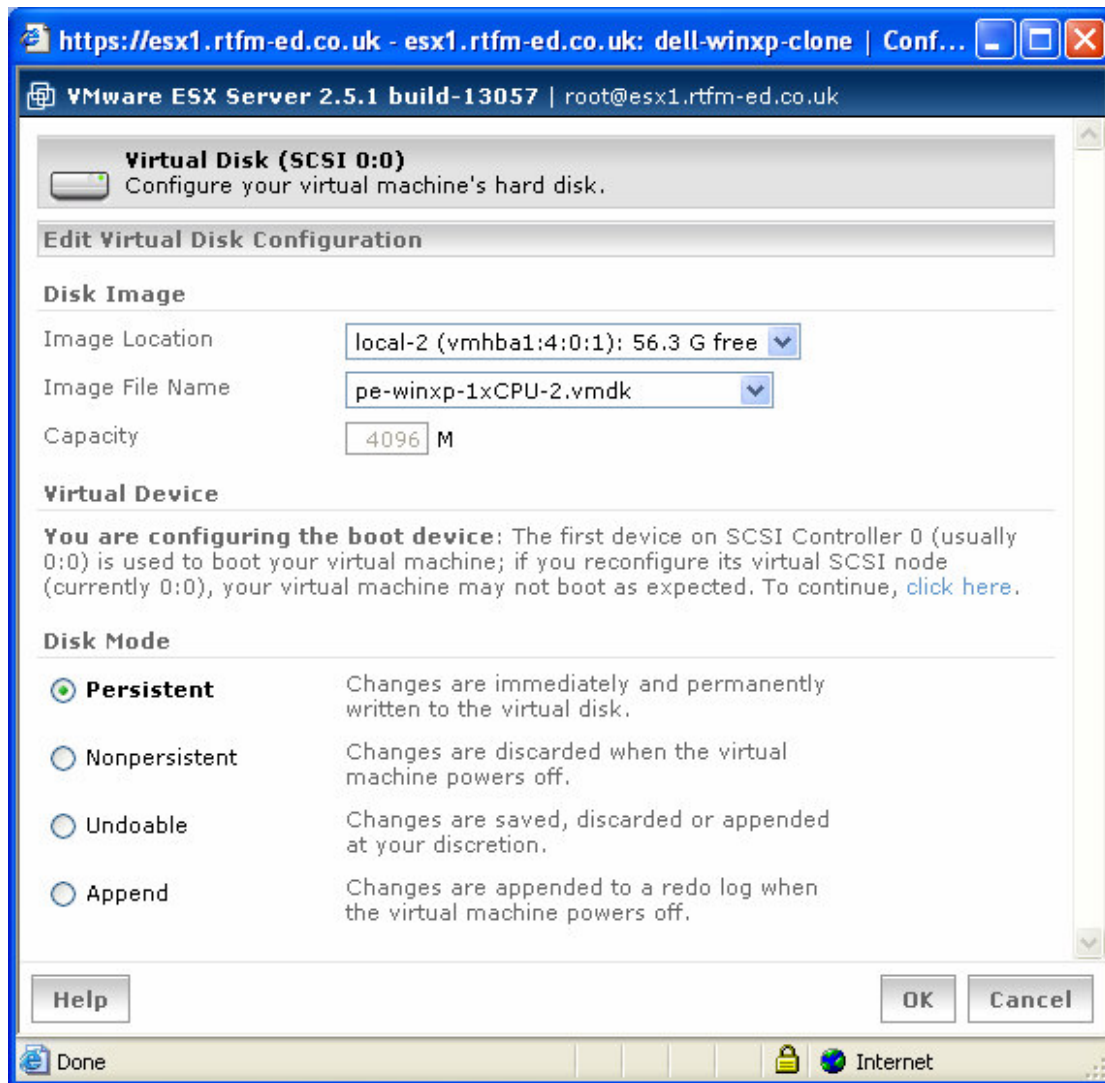
When I first began to use ESX I thought it was curious that the REDO file was called "REDO" – judging by many people first usage of ESX you sometime wonder if the file should have been called the UNDO file. One of my favourite modes is the "Undoable Disk Mode" – so initially I thought it odd that VMware should use the word "REDO"

I guess the reason VMware didn't is that would limit (in the users mind) the many wonderful usages the REDO can have.

Your first experiences of the REDO will probably come from experimenting with the "Disk Modes" offered in ESX MUI pages or the vCenter System. As you might recall there are four disk modes:

1. Persistent
2. Non-Persistent
3. Undoable
4. Append

**Figure 1:**  
**Disk Modes as displayed in ESX MUI**



First thing to state is that changing from these modes does require a power cycle of the Virtual Machine. Like a lot of the options behind the properties of Virtual Machine you have to power down the Virtual Machine to switch between one mode and the other. This obviously is a concern to people who must have the Virtual Machine available at all times like a physical server.

Second thing to state is that only the modes 2,3 and 4 use the REDO file. What makes them different is how they control the default "commitment" of the REDO file to the Virtual Disks. The term "commitment" merely means whether (or not) the contents of the REDO file are purged to the Virtual Disks or whether they are "Discarded" or "Kept" for a later date. I'm putting the words "commit" and "discard" in speech marks here because these are the terms used with in dialog boxes and CLI prompts.

In a simple analogy, you can think of the Virtual Disk like a Word document to which you have made changes that you have yet to save. When you shut down Word (shutdown the Virtual Machine) you asked if you want to save your changes (commit) or whether you want to not save your changes (discard). "Keep" would be like telling Word to save all your changes to another \*.doc file (REDO File) and

every time you open or close the document, Word would ask if you wanted to merge the contents of the other doc file to the original.

The only disk mode that DOES NOT use a REDO file is the persistent mode. As you might recall from the previous part of the document this is the default mode for Virtual Disks and the recommended mode for production Virtual Machines.

## ***Hey, Mike you loosing me here – Can you give me some examples?***

Am I? Opps, Sorry!

Perhaps it is that this stage we really need some scenarios to illustrate what these options allow...

### **Non-Persistent Scenario: The Developers**

You're a software developer and run lots of half-baked scripts that don't work properly and install software which makes changes of the registry – writing to HKLM rather than HKCU. Sound familiar? Well, anyway if you want a quick and easy way of running your troublesome scripts and resetting the Virtual Machine back to its clean state. Non-persistent is for you. After your scripts have screwed up the box, simply power-down and power-back up again – and give the scripts another try. Eventually they will work, won't they?

In this case the changes redirected to the REDO file are discarded without any prompts and never committed to the Virtual Disk.

ESX is an expensive solution to use in this scenario – why not use VMware Workstation? But if you have already purchased ESX and you have spare capacity you can give these developers powerful Virtual Machines – and a space within which to do their testing.

Another example is a Virtual Machine running as Internet Kiosk in Library. A dumb-terminal is used to RDP into a Windows XP+SP2 box to give access to the internet. Ordinary users cannot shutdown or restart the Virtual Machine (as this would break the RDP session!) But at the end of the day the librarians who can login and gain access to full system – power-down and power-up the Virtual Machine. The power-cycle discards all those lovely virus, spy-ware, and porn which have been downloaded by the users in library during the day.

Of course you could deliver such an environment using RDP to a server or by Citrix. But perhaps this cheaper and is a solution if you have software which cannot run acceptably in Terminal Services session.

### **Undoable Mode: Rolling out a Service Pack**

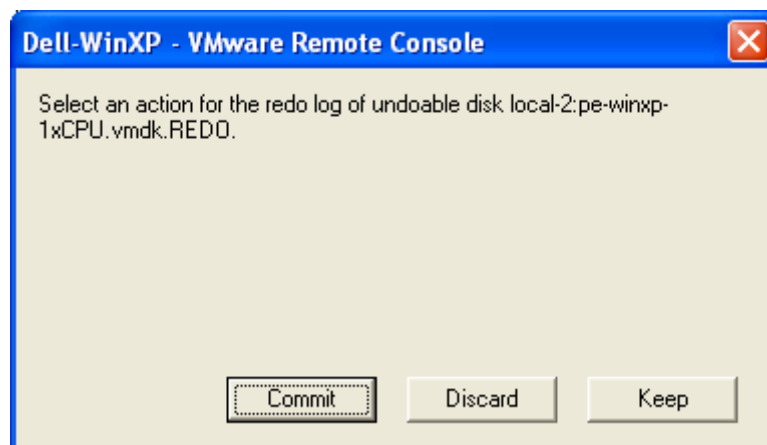
This is one my favourite modes. Imagine your virtual machine has big undo button like in Word – and you have completely screwed up the Virtual Machine and like to return to a good version of it.

For example, you want to roll-out one of those dependable and reliable service packs or security roll-ups. You know the kind that comes out one week and a couple of weeks later there's another that fixes the errors in the service pack? Anyway, you have Citrix server which is a very fragile animal which has been known to react very sensitively to service pack roll-outs.

So you set the Virtual Disk to be undoable. Power-up the Virtual Machine and apply the service pack. At the end of the install you ask the service pack to reboot the Virtual Machine. After that horrible and unpleasant things happen to you. You find on your Windows 2003 server with Citrix Presentation Server 3.0 or 4.0 that Internet Explorer is broken. You're pretty sure there is a hot-fix for this error but in the short-term you want to return the Virtual Machine back its good state.

You power down the virtual machine and choose Discard – you then power it up and go off to do some research.

**Figure 2:**  
**Remote Console dialog created when you power down a Virtual Disk in Undoable Mode**



When you do power up the Virtual Machine - because you discarded the changes in the REDO – you don't receive any dialog prompts.

You find on Citrix website that the problem you had is a "known issue". You kick yourself because you didn't do the research about this "known known...". It's a well documented problem and it's not an "unknown unknown..." to paraphrase the US Secretary of Defence.

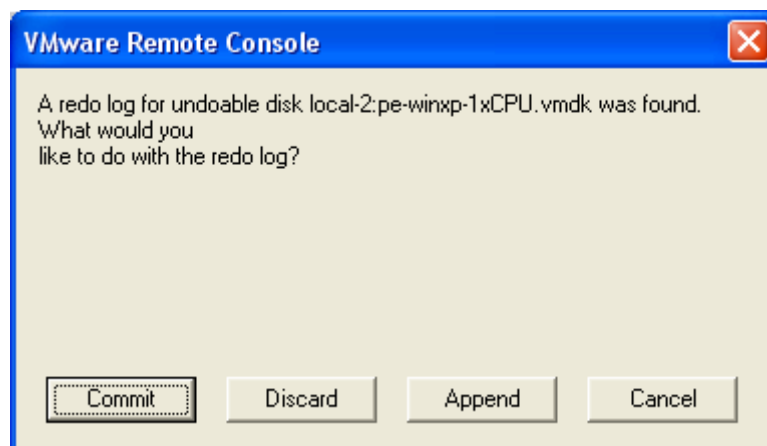
You apply the service pack again, and the relevant hot-fix from Citrix – and everything is as right as rain.

This time you decide you quite like to save this configuration to your Virtual Disk given all the grief and heart-ache it's given you. So you power-down the virtual machine and choose commit.

The last option in this dialog is "keep". Perhaps you will need a couple of days to test the reliability of this service pack – but within this time you find you need to power-down the Virtual Machine for whatever reason. You choose to power-down but choose "Keep" – this will maintain the REDO file indefinitely until you choose either "Commit" or "Discard".

Next time you power-up the Virtual Machine you will receive a dialog prompt. Choosing Append would maintain the current REDO file – with it getting incrementally larger until you eventually decided to commit or discard your changes to the REDO File. You decide to keep an eye on this REDO file because you don't want it get beyond the 2GB value that VMware recommends.

**Figure 3:**  
**Remote Console dialog from a power-up with Virtual Disk in a Undoable Mode**



### **Append: The Homer Simpson Scenario**

This has to be my least favourite mode. It works just like undoable BUT you don't receive any prompts. Changes keep accruing in the REDO indefinitely. There's a danger that if left unchecked the REDO gets bigger and bigger, performance takes a nose dive and the VMFS volume where the REDO file resides gets flushed.

To commit the changes from the REDO to the Virtual Disks - is NOT as easy as undoable. You have shutdown, switch from Append to Undoable or Persistent and choose to "commit" or "discard" your changes when you next power-up. ☹

Alternatively, you have to connect to the service console and use either vmkfstools or vmware-cmd to apply the REDO to the Virtual Disks. If you're a Windows person you might find the prospect of doing this a bit scary and intimidating. Quite why, I'm not sure - its just a command. Does it matter if it isn't a Windows CMD Prompt or in PuTTY?

One good thing about the Append option is it does protect you from a "DOH!!!" Homer Simpson moment. With undoable mode you are prompted commit, discard or keep. Suppose you have an "Homer Simpson moment" and choose the wrong option.

I've done it. ☹

Often it's not the end of the world for me as lots of my Virtual Machine are development ones- but it was a production virtual machine I might be looking for my backup tapes or have to repeat a lot of process lost to my accidentally discarded REDO. Opps ☹

As Append offers you no choice on how to handle the REDO file - there is no chance of a DOH! Moment. It is however, as a consequence less flexible.

This mention of backups leads us on to how REDO files can be used as part of backups.

### ***What are my Backup Options?***

So far we have looked at the REDO file from a non-backup perspective and handled the REDO file using a GUI interface. There is another very important and popular usage of the REDO from a backup perspective which utilises the CLI (command-line interface)

Primarily this allows backups to be scripted and automated – and more importantly to be done when the Virtual Machine is powered on! ☺

In the real world you are likely to take advantages of many technologies when it comes to DR and Backup. This referred to as CYA. As you might know this abbreviation stands for “Cover Your Ass”. It was one of the first things I learned when I started working in IT. It’s up there with RTFM.

I choose RTFM because cya.co.uk is already registered an organisation called the “Cumbria Youth Alliance”. I wonder if they are aware of the alternative meaning of their abbreviation. Quite possibly not, as rtfm.co.uk weren’t aware until I told them that RTFM stands for something other than “Real Time Financial Management”. Funnily, enough I never did get a response from my email.....

Anyway... I digress...

Why do we backup? Well to allow us to restore a good copy of a file when the original has become lost or corrupted. But we also backup because we hate crying in the server room, and that of feeling “fear and loathing” that over-comes one when you realise your going to get fired - and you have mortgage to pay, a wife/husband and children support and a credit bill the size of the GDP of small developing nation.

There are many backup solutions you can take advantage of – some may or may not – take advantage of the REDO file. Some are commercially available for a FEE and some are FREE scripts that generous people have written that backup your Virtual Machines to NFS/Windows/FTP services out there on your network.

Here’s a round up of you options:

- **These you have probably already have bought OR are free**
  - **Conventional Backup Agent running inside a Virtual Machine** as you would with a Physical Machine. This allows for standard normal, incremental or differential backups ideal for non-reproducible end-user data and give restore control down to the level of individual file.
  - **Conventional Backup Agent running in the Service Console.** In this case you install the Linux Backup Agent from your vendor into the Service Console – making the Agent away of the /vmfs mount point.

This allows for backups of the entire virtual disk probably more suited for “system” backups where you want to backup the boot disk of Virtual Machine. They could be used to backup end-user data but they may not give you the flexibility you required to restore just an individual file. The downside of doing this is that the agent consumes resources on the Service Console – and if you start adding other vendor’s agents (Insight Manager, IBM Director, Dell Open Manage) the Service Console will need more resources allocated to it.

It’s not a “bad” thing to do – just something that needs considering when “divvying” up the resources between the Service Console and VMkernel

- VMware’s VMsnap.pl and VMres.pl scripts

- [VMX Backup Script](#) (VMVBU – doesn't do virtual disks!)
- [Virtual Disk Backup Script](#) (vmbk.pl)
- [Virtual Disk Backup Script with Legatto](#) (save\_vm)
- **These you have pay for**
  - [ESX Ranger](#) from VizionCore
  - [Virtual Solution Box](#) (VOB) from Aexia

Now is not the time to discuss the merits of these various backup solutions or methods. This document is not about backup options and various advantages or disadvantages. This document explains how these providers might leverage VMware's REDO file during backups.

The best vendors use the REDO file system and automate the whole process for you. The worst require you to manually write the scripts to take advantage of the REDO file. Almost all of the providers listed above do this scripting for you. But if you want to use your existing vendor's backup solution you might want to chat to them and get their advice how best to proceed.

By far in the community the vmbk.pl method seems to be the most popular. Its cheap because it is free, comes with a manual – and because lots of people on the VMware Forum use it – you can help from these people if can't get it working properly.

## ***How to Backup Providers use the REDO***

Backup providers use the REDO when they are backing the Virtual Machine at Virtual Disk level.

They cannot backup the Virtual Disk is if it set to persistent without changing the power-state of the Virtual Machine. This because this file is locked by the file system (VMFS) and is in use by processes (the Virtual Machine). If a Virtual Disk is in a persistent state then the only way to back it up is to power-down the Virtual Machine and backup the file as you would any other. We could call this backup method a "cold backup". It is the simplest option if you can afford to have server down and unavailable during the backup period. It's relatively easy to script. But it may not be viable in today's 24/7/365 era.

If however, the Virtual Machine has a REDO file then we can do a "hot backup". That is to say a backup of the Virtual Disk while the Virtual Machine is powered on. The problem is we cannot use the conventional Virtual Disk Modes of Persistent, Undoable or Append as these modes are NOT recommended by VMware. So what we have is a bit of catch-22. To do a hot backup we need a REDO, but if we habitually use a REDO - this not recommended by VMware (for reason mentioned at the beginning of this article)

How do we get the best of both worlds?

This is nearly always achieved by some kind of scripting (even if the system you are using has a nice GUI front-end). What the script does is apply a REDO on-the-fly, backup the Virtual Disk and then commits the REDO while the Virtual Machine is still powered-on (normally not allowed using VMware's MUI or vCenter) and thus allowing the Virtual Machine to have persistent disk when the backup has completed.

Now, here comes the clever bit...

If we just had one Virtual Disk and only one REDO we could run a hot backup of the Virtual Disk. The problem is that changes would grow and grow in the REDO as we backed up the Virtual Disk. So imagine a 10GB disk which your backing up across the network, this might take sometime. During this time the REDO file get bigger and bigger. Once the backup has completed the REDO could have become very large in size – which then it might takes an exceedingly long time to commit. If we only have one REDO file the system must “freeze” the Virtual Machine while the REDO is being committed. This is not good ☹

So what must providers do is give the Virtual Machine two REDO files. Yes, you can have more than one. I didn't tell you that earlier did I. Sorry, about that – but I didn't want to confuse you!

How do the vendors switch on the REDO and then commit it. Well, lots of ways – one method is to use vmware-cmd or perl script from VMware that allows you to add a redo file and use vmware-cmd or perl script from VMware which can commit of REDO file without resort to the power button.

This is useful to know anyway – not just for backups. For when you want to add a REDO file on-the-fly and commit the REDO.

The very last thing to mention is backups of this kind are only “crash-consistent”. That is to say that the backup of this nature would be fine in the event of say power failure. BUT, because the Virtual Machine was changing while it was being backed up (changes going to the REDO file) then there is no cast-iron guarantee that if you were backing up a boot disk that it would be complete enough to be functional.

Therefore the ONLY way to get 100% guaranteed backups is to power off the server – backup the virtual disk files, and then power it up again.

Our goal however is to have hot-backups.

Some people react very negatively to being told that hot-backup are not 100% reliable. But the way I see it is that there is no 100% reliable method of backing up.

## ***How do we add a REDO to a running Virtual Machine?***

1. Open your web browser and copy the script from

<http://www.vmware.com/support/developer/scripting-API/doc/addredo.pl.txt>

2. **Logon to the Service Console**, as ROOT or VM Administrator
3. Run **nano -w addredo.pl**
4. **Cut and Paste**, the script above into the window
5. **Close nano, and save the file...**
6. **To use the script** – follow this example below...

```
perl addredo.pl esx1.rtfm-ed.co.uk root password  
/root/vmware/instructor1/instructor1.vmx scsi0:0
```

### **Note:**

Syntax requirements can be viewed by simply typing pl addredo.pl... To

use the script you must specify...

Name of ESX server...

Username and Password

Path to VMX file

Optionally, specify the scsi node (in case your VM has more than one virtual disks – scsi0:0 adds a REDO to the boot disk)

**Note:**

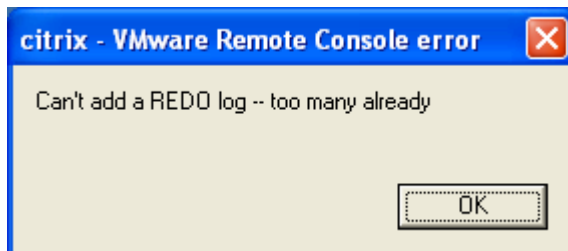
vmware-cmd also has a addredo command with very similar syntax. I think main advantage of the perl script method is that it can be done remotely – or that Linux backup agents (commonly installed to the Service Console) can be told to execute the addredo script *before* the backup begins, and when the backup is over, commit the REDO's *after* the backup is completed.

**Note:**

This can be repeated again to add a second REDO to the Virtual Machine. When the backup is complete the vendor can add a second REDO file, then commit the changes accrued in first REDO file to the virtual machine, followed by second REDO file

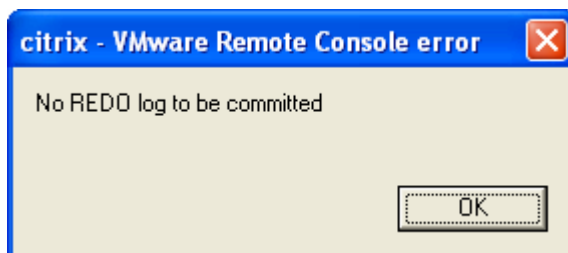
**Note:**

If you try and run addredo three times in the row. You will get this error message:



This is because you only allowed 2 REDO log files at anyone time

If you get your syntax incorrect or if a REDO file doesn't exist the Remote Console will give you this error message:



**Note:**

The addredo.pl file was created under ESX2.0/2.1/2.5.1. With ESX 2.5.2 two new scripts were added which replaced it called

vmAddRedo.pl

vmCommit.pl

## ***How do we use vmware-cmd to commit the changes from REDO?***

### **Commit First BIG REDO without a Freeze (Burrrrr!)**

#### **Note:**

vmware-cmd is just one way of committing a REDO to a virtual disk. If you are running on ESX2.5.1 (or possibly older) there is an error in vmware-cmd. So if you were going to use the command in anger. Look at this support article [Answer ID 1323](#)

Apparently this problem has been addressed in ESX 2.5.2

1. **vmware-cmd /home/lavericm/instructor1/instructor1.vmx commit scsi0:0 1 0 1**

#### **Note:**

This commits the REDO file of the machine to the first SCSI adapters first disk (in this case the boot disk of the OS). The last 3 parameters 1 0 1 control the Level, Freeze and Wait values.

All these options are important – but it's the Freeze option which is significant to us.

**Level** is used when you have one or more REDO file. It always you indicate which REDO commit. We have commit the first REDO first followed by the second REDO. If we committed the second REDO, it would apply the first followed by the second REDO

**Freeze** controls if you going to freeze the machine while the redo file is played against the Virtual Disk. 0 means no, and 1 means yes. It's quicker to freeze the machine, while applying the redo – but it means the server is offline during the committing of the REDO.

*If you use Level 0, then freeze option is ignored – and the system freezes the Virtual Machine while it is committing the REDO File.*

This can cause us a problem if the REDO file is large (as it could be with a long backup). So, the trick is to add a second REDO file which allows us to apply the larger REDO file *without* a freeze. Once it has been committed we can then commit the much smaller 2<sup>nd</sup> REDO File with the Freeze option and no wait option.

**Wait** 1 means that the system returns to operation as soon as commit begins, not waiting for it complete. 0 means wait for the completion of the commit before continuing

Got that?

If not read it again... and again...

### **Commit the Second SMALLER REDO with Freeze**

#### **Note:**

At this stage the larger REDO file will have been removed. Leaving a smaller REDO file which changes its name from REDO.REDO to being REDO. This can be somewhat confusing. If you list the files you and look at the file sizes you can see which files is which

```
vmware-cmd /home/lavericm/instructor1/instructor1.vmx
commit scsi0:0 0 0 0
```

**Note:**

Notice how the wait option is set to 0 which means "don't wait"

***Erm, you kind of lost me there – can you simplify it a bit...***

OK, well here's the clever bit... With two REDO's we can commit the BIG redo *without freezing the Virtual Machine*, because during the commit process of the 1<sup>st</sup> REDO changes are going to the much smaller 2<sup>nd</sup> REDO. This means despite the REDO file being BIG the Virtual Machine continues running normally (because we didn't freeze or wait it)

Once the 1<sup>st</sup> REDO has been committed (it is deleted from the system after the commit process) – we can then commit the much smaller 2<sup>nd</sup> REDO file. When we apply the LAST REDO we can change the "wait" option to allow temporary freeze of the Virtual Machine... BUT because the 2<sup>nd</sup> REDO is much smaller – this doesn't take very long...

As the 1<sup>st</sup> REDO file no longer exist we can only specify level 0, the freeze option is not available now so we have no choice to 0 for freeze – as the REDO is very small and won't take to long to apply we can set wait to be 0

Below is a high-level view of what happens during many scripting solutions. In fact you might find this easier to understand than the above. In fact, you might find having read the higher-level view to go back and read the more detailed explanation above. It might make more sense to you.

1. Add 1<sup>st</sup> REDO
2. Start Backing Up VMDK
3. Once Backup has completed, add 2<sup>nd</sup> REDO
4. Begin applying the bigger 1<sup>st</sup> REDO without freeze (bigger because its been taking the changes while 3. was occurring)
5. When 1<sup>st</sup> REDO is completed – begin applying much smaller 2<sup>nd</sup> REDO – using the Freeze option
6. Process is over we back to single Persistent Virtual Disk

**Note:**

In an environment where there is significantly high number of read/writes you could find that even 2<sup>nd</sup> REDO is getting to large and takes too long to apply. After the 1<sup>st</sup> do has been applied, we could add another REDO and apply the 2<sup>nd</sup> REDO. Once it has applied we could apply the 3<sup>rd</sup> REDO file. Remember though we only can have 2 active REDO's that any one time.

***If this is how vendors achieve hot backups, what expected behaviour should I see?***

Now clearly every vendor has their own approach – and might use many different ways of creating the REDO's and committing them. However, in the principle of hot-backups are the same regards of the vendor. If believe you seeing unexpected behaviour it's to them should first go.

But simply put during the backup process you would expect to see at LEAST one REDO as it is required for a hot backup of the Virtual Disks.

Given what I've said about the usage of two REDO files you would hope to see REDO's for each Virtual Disk during backup. This shows the backup solution is a sophisticated one and recognises that many hot-backups will need two REDO files, not just one

At the end of the backup process you should not see a REDO file at all. Unless you are using non-persistent, undoable or append as a Virtual Disk Mode. Left behind REDO's created by your backup system *could* indicate the process is NOT completing properly and could indicate you are not getting a full and complete backup. This problem needs investigating to see if you have problem or not – and also what the resolution is. Notice my italics on the word *could* – left behind REDO files do not per se indicate a problem – but they *could* do...

## **Summary and Conclusion**

- REDO of "exciting" opportunities in terms of doing tests and backups. Obviously, not as exciting as skiing down the Alps. But in the context of I.T. this about as exciting as life for some can get!
- Don't use REDO file habitually and don't allow them to grow to impossibly large sizes
- If you handling the process manually using the ESX MUI some monitoring has to be done on the growth of the REDO File. How quickly the REDO file will grow is beyond the scope of any article for obvious reasons.
- The document has tried to give you an insight on how REDO files work – how you can use them for your everyday work and how they can be utilised by backup solutions.
- I wouldn't recommend giving yourself the brain-ache of writing your own backup scripts – there plenty of very nice people who have done the hard work for you. In fact the whole bought of the scripts or the bought back-up solutions, is that you don't have to think about this issue of REDO's, REDO levels, and Freeze and Wait. But it is nice to know roughly how these products achieve hot-backups.
- Hopefully this article gives you an insight on how hot-backups are done – and some ideas about identifying and troubleshooting them when the don't work!

## **For the Future?**

- Wouldn't it be nice to have many undo (or shall I say REDO) levels?
- You could take "snapshots" as you go along... and roll-back through the levels – you could even labelled these levels and have date stamps so you know what each snapshot or REDO file would roll you back too.
- I believe the workstation tool has this feature.
- But it would be nice if ESX did...