

VMware® Infrastructure 3

Advanced Technical Design Guide

~and~

Advanced Operations Guide

Two books in one!



Ron Oglesby
Scott Herold
Mike Laverick

Chapter 11: Backup and VMware Consolidated Backup

GOTCHA:

The most common error with VCB is operator error caused by mistyping directory paths or job names. In most cases, the errors will come from typing `d:\backup` rather than `d:\backups` or something similar.

Warning:

Please validate and confirm your backup and restore process works. The authors of this book do not accept any liability for data loss.

Methods of Backing Up

A VM is encapsulated in a few files. This introduces new ways of backing up beyond the conventional method of installing backup agents inside the guest operating system. Virtualization offers us the tempting possibility of backing up a VM without conventional backup agents. This means we can offload the network traffic generated by backup to external dedicated systems. Additionally, not installing backup agents into the guest operating system is more cost-effective. Before examining VMware's "Consolidated Backup" (VCB), it's worth comparing and contrasting the various backup choices at your disposal.

Backup Agents inside the VM

In this scenario, organizations continue installing backup agents inside the VM as we have done with our physical machines. This allows for normal, differential, and incremental backups without modifying existing procedures and methods. The powerful aspect of this approach is that we do not have to revalidate our disaster recovery or business continuity strategy to the same degree than if

we select a brand new method previously untried and untested within the organization.

The major downside is that we would not really be leveraging the power of virtualization to improve those backup and restore procedures. Additionally, you may be forced to adopt this approach within the guest operating system because VCB is not fully supported with your guest operating system. VCB is able to execute a “file-level” approach which allows for backing up the *individual* data files within a VMDK file. Unfortunately, this currently excludes Linux, Solaris, and Novell guests. In other words, “file-level” backup with VCB is a Windows guest operating system feature only. Ostensibly this is because VCB runs on physical Windows hosts (referred to as the VCB Proxy) and Windows natively only understands NTFS, FAT, and FAT32 file systems.

Backup Agents inside the Service Console

It is possible to install a Linux backup agent to the Service Console because it is a modified version of Linux Redhat. One advantage to this method is that you will need less backup agents – this is decidedly cheaper than installing backup agents for each VM.

The downside is that this method will only allow you to backup the VM’s virtual disks and associated files. It will not allow you to backup individual files within the VM. This is costly from a per-MB perspective and would take some time to setup and configure. Additionally, the backup would be throttled by the Service Console network and would put an unwanted CPU burden on the Service Console.

SAN LUN Replication and Network Replication

Another way of “backing up” is to use your SAN vendor’s tools to create “snapshots” LUNs within a SAN or between two SANs. This can be configured within a SAN, a server room, or greater distances if you have access to “Dark Fiber.” Dark fiber links are fiber channel connections within a city or between cities. If dark fiber is unavailable, you may have to resort to network-based replication using software such as DoubleTake. Dark fiber links will replicate data synchronously whereas network links will replicate asynchronously.

The difference is the former has little or no latency whereas the latter definitely will have latency. This can mean there is a difference of the data at the primary location from the backup location. This approach has some strong appeals; the SAN becomes the biggest single point of failure, not least because everything about our virtual environment is more or less stored on a SAN. Again, this approach is not without its downsides.

Firstly, dark fiber links are not universally available and can be very costly. Currently, the technology is only available to the organizations with the largest IT budgets such as banks, pharmaceuticals, oil companies, and governmental bodies. Secondly, the backup is not especially “granular.” If you wish to restore an individual file that’s been lost within VM, then SAN LUN Replication would not be the fastest approach to restoring it.

Third Party Backup Vendor

Another approach is a more blended one which may involve some or all of the options previously discussed, in part orchestrated by backup vendor. As virtualization grows in size the mature backup vendors are beginning to respond by developing their own custom solutions. This is worth investigating to see if your existing vendor has updated their software to be VM aware. At the very least your vendor should be supporting VCB as part of their software.

Additionally, there are a number of independent software solutions in the new third-party space surrounding virtualization. These companies are essentially “start-ups” who have sprung up in recent years or even months. Quite often they offer unique features because their main market is backing up VMs. They set themselves apart from the conventional backup vendors because they specialize in the field of virtualization. You will notice I’m not mentioning any product names here. This is because one of the authors of this very book works for one of these companies! We didn’t want to come across as partisan in this chapter. This is why we’ll concentrate on VMware’s VCB in the main. I will discuss VCB integration with third parties using another vendor’s backup software later.

Free Backup Solutions

There are a number of scripts which circulate the VMware forums and blogs. These quite often leverage the free tools that ship with ESX server. They are generally a Service Console approach and as a consequence usually backup the VM locally before copying it across the network to a secondary storage system. The merit of this approach is essentially a cost based one. It's attractive to people with modest size implementations where the volume of data to backup is similarly modest.

The downside of this approach is your support will be non-commercial and community based. However, if this approach fits your requirements, Alex Mittel of Oxford University, UK has written a very good backup script called "visbu."

Download Visbu:

<http://users.ox.ac.uk/~alexm/>

Community Forum Support:

<http://www.vmware.com/community/thread.jspa?threadID=70253>

Using VMware Consolidate Backup (VCB)

In contrast to the above methods, VCB backups the VM directly from the SAN. Strictly speaking, VCB is not a backup solution but a collection of APIs and drivers that enable third party vendors to backup a VM. With VCB there is no network hit on the ESX host during the backup process. The backup traffic is offloaded to a dedicated Windows 2003 server. SAN connectivity with the VCB software installed is referred to as the "VCB Proxy."

Two types of backup are supported with VCB; File-Level backups, and Full VM backups. With file-level backups the virtual disk of the VM is mounted to a folder on the VCB Proxy. The backup operator can select individual files inside the virtual disk. With a full VM backup the virtual disk is backed up alongside all the other files that make up a VM including the NVRAM and log files. As

you might expect, file-level backups are ideal for normal, differential, and incremental backups of data. In contrast, full VM backups are ideal for archiving the operating system state of a VM. With a full VM backup the virtual disks of a VM are “exported” into the 2GB sparse format to ensure that data as well. This export process ensures that free space is not included in the backup process.

Both approaches support a “hot backup” of the VM while the VM is powered on. A special “sync” driver is installed to the VM during the install of VMware Tools. This sync driver is able to reduce the activity generated in the file system within the VM. However, you will most likely still want to stop services inside the VM to ensure that files are unlocked in the file system during backup time.

As with other backup processes, there are downsides. Although the backup process is via the SAN, the restore process is across the network. This can slow the restore process significantly. Restoring from a “Full-VM” can be achieved quite neatly, but restoring an individual file means copying them from the VCB Proxy via Windows shares to the original destination. Additionally, VCB only works on a physical host with Fiber-Channel. SAN connectivity it is not supported with iSCSI SAN even when used with hardware initiators.

VCB can be used independently of a third party backup tool for “Full VM” backups, but most people will want to use it in conjunction with their third backup vendor. This is usually achieved with an integration module. Unfortunately, the help that is provided with the integration modules is not always very clear and in some cases inaccurate. The quality of the scripts can vary significantly too, with some people resorting to writing their own scripts that they can understand and troubleshoot rather than debugging or re-engineering scripts that have been written by others. If you decide to use the integration modules that are shipped free with VCB expect to spend some time on the VMware Forums reading posts and asking others for help. There are some supported backup vendors for which there are no integration scripts, and you may be forced to create your own.

Installing VMware Consolidated Backup

Installing VCB is very easy; however, care must be taken with VCB Proxies connection to the SAN. Firstly, as you probably know, by default Windows 2003 will try to access hard disks and LUNs directly by writing a “disk signa-

ture” and mount them as part of its system. This is potentially damaging to the VMFS file structure. Ideally, you want to build this server without connectivity to the SAN until you have turned off this automount feature. There are a couple of ways of disabling the SAN connectivity. If you have physical access to the server, disconnect the SAN cables; alternatively, in the BIOS, temporarily disable the fiber-Channel host bus adapter. Once you have successfully installed Windows 2003 turn off automount using the “diskpart” utility.

Secondly, VCB currently does not support more than one HBA inside the VCB Proxy. The problem arises because Windows chooses by default to see two HBAs as presenting two LUNs, rather than multiple paths to the same LUN. Currently, the work-around is to disable the second HBA if you have one or disconnect it.

Turning off “Automount” on the VCB Proxy

1. Login as administrator to the Proxy.
2. Open a command-prompt with Start, Run, and cmd.
3. Type this command:

```
Diskpart
```

4. At this diskpart prompt type this command:

```
Automount
```

Note:

As automount is enabled by a default, this command should disable automount and give you the status information of “Automatic mounting of new volumes is disabled.”

5. Exit diskpart by typing this command:

```
Exit
```

Note:

You can now re-enable the SAN connectivity. If you open “Disk Management” console in Windows and are asked at any point to write a “disk signature” the safest approach is to cancel the dialog box prompt and wizard. Even safer still is not open Microsoft Disk Management tool at all. However, you may need to do this to configure other storage devices.

Installing and Configuring VCB

1. Download latest version of the VCB package from the VMware Website and Run the Installation Package.
2. Acknowledge the prompt to install the **VMware Virtual Volume Storage Bus Driver**.

Note:

This VCB LUN driver gives the VCB Proxy read-only access to the LUN and VMFS contained within; without this driver the Windows 2003 would not be able to access the files system to begin the backup process.

To use the command-line tools you will need open a command-prompt to the install directory of VCB. You may also wish to add this to the “path” entry to the environmental variables of Windows.

- a. Right-click **My Computer**.
- b. Choose **Properties**.
- c. Select the **Advanced** Tab.
- d. Click the **Environment Variables** button.
- e. Under **System Variables** select **path**.
- f. Click the **Edit** button.
- g. Add semi-colon to the end of the current path and type:
h.C:\Program Files\VMware\VMware Consolidated Backup Framework

Note:

Speech marks around “the path” are not required. You can confirm the path is known correctly by opening a command prompt and running one of the VCB utilities such as vcbMounter. If the path statement has been input correctly in Windows “Environment Variables” the utility should run without error messages such as “Bad command or file name.”

Licensing VCB

As with any product, VCB must be licensed correctly. VCB can be enabled for each ESX server by selecting the host in the Vi Client.

3. Select the Configuration Tab.
4. In the Software Pane, choose Licensing Features.
5. Select the second Edit... button.
6. Enable VMware Consolidate Backup in the dialog box.

VCB on the Command-Line

Most people who use VCB on a daily basis use integration scripts provided by their vendor or VMware. This means they do not need to know the VCB command-line utilities. However, because this is a book about VMware we want you to have a good knowledge of these command-line tools should you wish to write your own scripts. It might also assist you in troubleshooting and debugging scripts that have been provided by your backup vendor.

Some of the VCB commands are available on both the VCB Proxy and the Service Console – and some are only available on the VCB Proxy or the Service Console.

These utilities are on both the VCB Proxy and the Service Console:

- vcbVmName - Returns the identity of the VM
- vcbSnapshot - Applies a snapshot

-
- vcbMounter - Mounts virtual disks, and can trigger a Full VM backup
 - vcbExport - Exports a virtual disk in other formats

These utilities are ONLY available on the Service Console:

- vcbUtil - List resource pools and folder location
- vcbRestore - Restores a Full VM backup across from the VCB

This utility is ONLY available on the VCB Proxy:

- mountvm - Mounts virtual disks to a Windows folder

Warning:

When executed on the Service Console these commands *are* case-sensitive – but when executed on the VCB Proxy they are not. However, even in Windows some of the command-line switches are case-sensitive.

In this chapter, we won't be covering all the VCB commands and all the switches – only the most useful and commonly used which are vcbVmName, vcbMounter, vcbRestore, and mountvm.

Viewing VM Unique Identifiers

To backup a VM you need to be able to identify it in the VirtualCenter or ESX host inventory. A VM has 4 identifiers:

- The VM's name as displayed in VirtualCenter or ESX
- The VM's IP address
- The VM's UUID
- The VM's MOREF (Managed Object Reference)

Note:

The MOREF value is more commonly used by the VirtualCenter SDK (Software Development Kit). It is generated at the VM's first power on but after that it does not change.

While the VM's name and IP address are friendly and easy to determine they are also subject to change. The UUID and moref values are clearly less friendly but are guaranteed to be unique and unchanging. You can query VirtualCenter using the `vcbVmName` command from the VCB Proxy.

All the VCB commands have a similar syntax which uses these switches:

- `-h` to specify the name of an ESX host or VirtualCenter system
- `-u` to specify the username
- `-p` to specify the password

For example, to query VirtualCenter to find out the details of a VM called `mike01`, you carry out the following steps.

1. Login to the VCB Proxy as Administrator.
2. Open a command-prompt.
3. Type this command:

```
vcbvmname -h virtualcenter.vi3book.com -u lav-  
ericm -p password -s name:mike01
```

Note:

Here the account `lavericm` is a user who has privileges in VirtualCenter.

This would produce the output as follows:

```
[2007-07-12 14:15:09.884 'App' 3600 info] Current working directory: C:\
[2007-07-12 14:15:10.462 'BaseLibs' 3672 warning]
[Vmdb_Unset] unsetting unknown path: /vmomi/
Found VM:
moref:vm-810
name:mike01
uuid:5034aae8-35c6-51a0-8f18-4264a177ee65
ipaddr:192.168.3.14
```

Note:

As you can see the UUID value is quite long. If I am using it I will often use the Window “Command-Prompt” copy and paste facility to capture this text to the clipboard. I also use this function to complete the very long paths that are sometimes typed into dialog boxes and configuration tools.

As you’ve probably realized, you can use `vcbVmName` with `-s` search as follows.

```
-s moref:vm-810
```

```
-s uuid:5034aae8-35c6-51a0-8f18-4264a177ee65
```

```
-s ipaddr:192.168.3.14
```

Note:

The `ipaddr` can be specified as a raw IP address or FQDN/Hostname if you have name resolution configured on the VCB proxy.

The `vcbvmname` command can also be used to check the power state of all the VMs in the VirtualCenter environment as follows:

```
vcbvmname -h virtualcenter.vi3book.com -u lavericm -p  
password -s any:powerstate
```

Mounting and Unmounting VM's with vcbMounter

Another utility allows us to mount and unmount virtual machine's virtual disks from the command-line. This is normally done prior to a backup. The process mounts the VM's virtual disks to a folder on the VCB Proxy. These files then can be backed up as if they were local to the VCB Proxy, when in fact they are being remotely accessed using the VCB Framework Software through the HBA.

During the mounting process the vcbMounter applies a snapshot to the VM. This allows the virtual disks to be "unlocked" in the VMFS file system. The same principle works for RDM as well, but only if you have them set with "Virtual Compatibility" mode. When vcbMounter unmounts the VM's file, this snapshot and its deltas are merged into the virtual disk, and then deleted. You can see this process in action if you have the Vi Client open at the same time you run the vcbMounter command.

To mount a VM's files for a file-level backup by its name in VirtualCenter you would use the following:

```
vcbMounter -h virtualcenter.vi3book.com -u lavericm -p  
password -a name:mike01 -r d:\backups\mike01 -t file
```

As you can see, the -a switch allows you to set the attribute by which you identify the VM. The -r switch allows us to set the mounting point. If d:\backups didn't exist it would return an error; you must specify some kind of path for vcbMounter to work correctly. The -t switch allows us to set the type of backup. For Windows guests you can use both file and fullvm as types; with all other guest fullvm is the only option currently allowed.

The result of this command would look like this:

```
Opened disk: blkfst://snapshot-861[virtualmachines]  
mike01/mike01.vmdk@virtualcenter  xxxx/xxxx  
Proceeding to analyze volumes  
Done mounting  
Volume 1 mounted at d:\backups\mike01\digits\1  
(mbSize=4086 fsType=NTFS )  
Volume 1 also mounted on d:\backups\mike01\letters\C
```

You can see the VM is mounted twice for one virtual disk – once by drive letter (\letters\C) and another by a number representing the disk (\digits\1).

Figures 11.1 and 11.2 show the activity generated in the Vi Client by running the vcbMounter command.

Figure 11.1







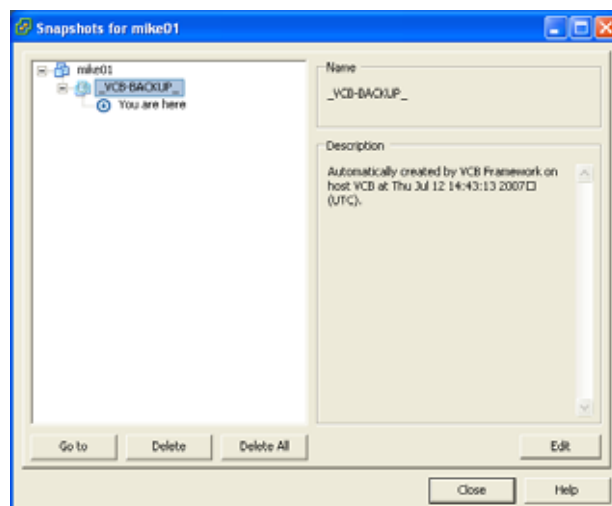
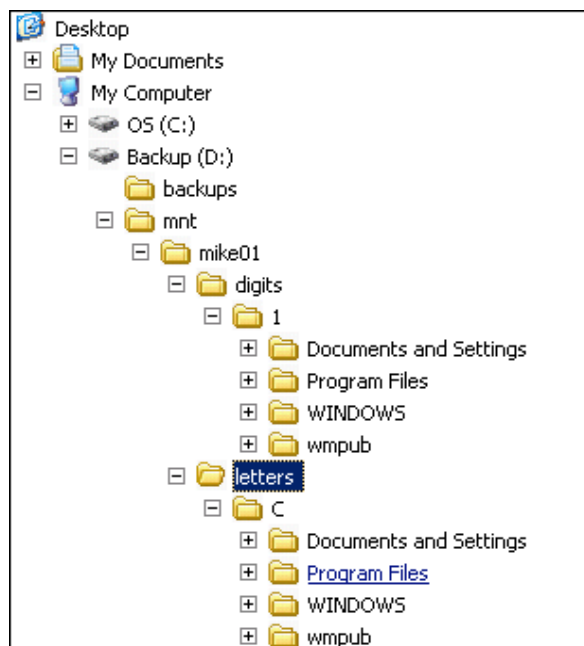
Recent Tasks		
Name	Target	Status
 Renew Disk Lease	 mike01	 Completed
 Create Virtual Machine Snapshot	 mike01	 Completed

Figure 11.2



Lastly Figure 11.3 shows the hard-drive of my VCB Proxy and shows the contents of mike01's virtual disk.

Figure 11.3



As you would expect, vcbMounter also allows you to unmount the VM's virtual disk. This creates a new disk re-lease and deletes the snapshot. You can unmount the virtual disk with this command:

```
vcbMounter -h virtualcenter.vi4book.com -u lavericm -p
password -U d:\backups\mike01
```







Note:

-U is a case-sensitive switch and directs vcbMounter to unmount the specified directory. The command-prompt would merely show the unmounting process with an output of this result:

```
Unmounted d:\backups\mike01\digits\1\ (formatted)
Deleted directory d:\backups\mike01\digits\1\
Deleted directory d:\backups\mike01\digits\
Deleted directory d:\backups\mike01\letters\C\
Deleted directory d:\backups\mike01\letters\
Deleted directory d:\backups\mike01
```

The Vi Client would show more information. Figure 11.4 shows the virtual disk's release and the snapshot's removal.

Figure 11.4

Recent Tasks		
Name	Target	Status
 Release Disk Lease	 mike01	 Completed
 Remove Snapshot	 mike01	 Completed

Completing a Full VM Backup

If you wish to do full VM backup the vcbMounter can achieve this for you. If you were attempting a full VM backup where you backup the VM's files then you would use fullvm as the type using -t switch. A full VM backup will backup all the files that make up a VM including the following:

- nvram
- vmx
- vmdk in the 2gbsparse virtual disk format
- log files

Two additional files are created; the catalog and unmount.dat file. The unmount file is used to automate the unmount process. The catalog file contains VirtualCenter metadata information so when the VM is restored the VM returns to the same ESX host, Resource Pool, and Virtual Machines and Templates location. A typical catalog file would look like this with mike01 being stored on VMFS volume called "virtualmachines":

```
version= esx-3.0state= poweredOn
display_name= "mike01"
uuid= "5034aae8-35c6-51a0-8f18-4264a177ee65"
disk.scsi0:0.filename= "scsi0-0-0-mike01.vmdk"
disk.scsi0:0.diskname="[virtualmachines]
mike01/mike01.vmdk"
config.vmx= "[virtualmachines] mike01/mike01.vmx"
host= esx3.rtfm-ed.co.uk
timestamp= "Thu Jul 12 16:04:24 2007"
```

```
config.suspenddir= "[virtualmachines] mike01/"
config.snapshotdir= "[virtualmachines] mike01/"
config.file0= "mike01.VM'sd"
config.file1= "mike01.vmx"
config.file2= "mike01.nvram"
config.logdir= "[virtualmachines] mike01/"
config.log0= "vmware-15.log"
config.log1= "vmware-16.log"
config.log2= "vmware-13.log"
config.log3= "vmware-14.log"
config.log4= "vmware-17.log"
config.log5= "vmware-12.log"
config.log6= "vmware.log"
folderpath= "/Datacenters/Lab DataCenter/vm/Mike's VM's"
resourcepool="/Datacenters/Lab DataCenter/host/Lab Cluster/Resources"
```

Warning:

Snapshot files created manually by an operator in the Vi Client are not backed up.

The syntax of vcbMounter command would be exactly the same as in the previous example except it would have -t fullvm as the type:

```
vcbMounter -h virtualcenter.vi3book.com -u lavericm -p
password -a name:mike01 -r d:\backups\mike01 -t fullvm
```

In this case, the d:\backups is both the mounting point and also the destination of the backup of the VM. You might notice when vcbMounter runs, the format used is "compact" by which VMware means the 2gb sparse format. Rather than taking the virtual disks name they convert the disk into a format that is more unique, "scsi0-0-0-mike01.vmdk." This information is used to identify the virtual disk on the SCSI Bus.

Completing a Full VM Restore

Clearly the restore process means copying back our backup to the ESX host, and restoring it from the 2gb sparse format to the "monolithic" or "thick" format. We could do this manually using WinSCP or Veeram's FastSCP utility and

then using the vcbRestore command at the Service Console. However, there is a more seamless way to complete this process.

Although Windows 2003 only does windows file sharing (SMB/CIFS) by default, we can install Microsoft's Services for Unix (SFU). This allows Windows to share out files using the Linux NFS protocol. The NFS protocol is natively understood by the VMkernel. Once SFU is installed and configured, we can then create VMkernel Port Group, perhaps on the Service Console's vSwitch0, and then NFS Mount point to our "backup" directory on the VCB Proxy. This allows us to use the vcbRestore command to pull the files down from the VCB Proxy and restore the VM.

The setup and configuration of SFU was covered in the Storage chapter. I've chosen to repeat those instructions again here with some slight modifications.

Windows does not allow unchallenged access to shares without authentication. As Windows and the ESX Host do *not* share a common end-user database, we need some method of "mapping" the users on ESX Host to Windows. The method I have chosen is a simple mapping of the accounts using the files present on any ESX Host.

Installing SFU

1. **Copy the passwd and group files** from any one of your ESX servers. These are both held in **/etc**. You can use free WinSCP tool to copy the files from the ESX host to your VCB Proxy.
2. **Extract the SFU package** and run the **MSI package** called **SfuSetup.msi**.
3. Choose a **Custom Installation** .
4. **Expand Authentication tools for NFS**, and select the **User Mapping Service**.
5. Select **Next** to the **setupid** and **case-sensitive options dialog box**.
6. Under **Local User Name Mapping Service**, select **Password and Group files**.

-
7. **Type name and path** for passwd/group files. For example,
c:\etc\passwd
c:\etc\group
 8. Select the **Windows Domain**.
 9. Select **Next, and accept the location for the install...**

Note:

Watch the status bar, check your email, make a cup of coffee, wonder how long you spend watching status bars... oh, and at the end of this - reboot your Windows/NFS Server.

Creating a User Mapping Between Administrator and Root

1. From the Start Menu, select Windows Services for UNIX.
2. Run the MMC, Services for UNIX Administration.
3. Select User Name Mapping node.
4. Choose Maps option, and under Advanced Maps click Show User Maps.
5. Click List Windows Users button - and select Administrator.
6. Click List UNIX Users button - and Select root.
7. Click the Add button.

Note:

8. When the warning box appears choose OK.
9. At the top of the console choose the Apply button.

Note:

A safer method would be to create a mapping of root to a lower-privileged Windows user with rights to the backup directories on the VCB Proxy.

Sharing out a Folder

1. On the **Windows Explorer**, **right-click the** folder, and choose **Share and Security**. In my case I shared the d:\backups directory.
2. Select the **NFS Sharing** tab.
3. Choose **Share this folder**.
4. Click the **Permissions** button, Select **Allow root access**.
5. Change the **Type of Access to Read-Only**.

Note:

Our ESX hosts only need read only rights to the Backups share just for purposes of restoring lost or deleted VMs.

6. Choose **OK** to exit the sharing dialogs.

GOTHCA:

Watch out foR CaseSensitivity on your sHaReNaMeS. Although Windows is not case-sensitive, it perfectly emulates NFS which *is case-sensitive*. If you want to remain, sane make them all in lower-case with no spaces.

Confirming the Windows/NFS Server is functioning

Note:

There are a number of tools we can use at the Windows/NFS server to see if things are working before adding in the NFS Share as IP Storage in the Vi Client.

rpcinfo -p (lists listening ports on the server, notice TCP, NFS v3, Port 2049)

program	version	protocol	port	
100000	2	udp	111	portmapper
100000	2	tcp	111	portmapper
351455	1	tcp	904	mapsVC
100005	1	udp	1048	mountd
100005	3	tcp	1048	mountd
100021	1	udp	1047	nlockmgr

100021	4	tcp	1047	nlockmgr
100024	1	udp	1039	status
100024	1	tcp	1039	status
100003	2	udp	2049	nfs
100003	3	udp	2049	nfs
100003	2	tcp	2049	nfs
100003	3	tcp	2049	nfs

showmount -e

Exports list on vcb.vi3book.com

/backups All Machines

Adding an NFS Mount Point

For this to work we will need a VMkernel Port Group with a valid IP address to connect to the VCB Proxy. I locate my VCB Proxy on network as the Service Console vSwitch so the restore process does not interfere with my VMs.

1. Select the Configuration Tab.
2. In the Hardware Pane, select Networking.
3. Click the Properties... of vSwitch0.
4. Click the Add button.
5. Choose VMkernel, and Click Next.
6. In the Port Groups Properties dialog, type a friendly name for this connection, such as vcb-backups.
7. Set an IP Address and Subnet mask.
8. Choose Next and Finish.

Note:

If you are more comfortable with the command-line you can add this vmkernel port group from the command-line with the following:

```
esxcfg-vswitch -A "vcb-backups" vSwitch0
```

```
esxcfg-vmknic -a "vcb-backups" -i 192.168.3.113  
-n 255.255.255.0
```

In the next step we will add in the backup mount point.

9. In the Hardware pane, select Storage (SCSI, SAN, NFS).
10. In the right-hand side of the Vi Client click Add Storage.
11. In the Wizard, choose Network File System.
12. In the Locate Network File System page... complete the dialog box as follows:

Server: Name of your NFS server, in my case vcb.vi3book.com

Folder: Name of mount you wish to access, in my case, /backups

DataStore Name: vcb-backups (or anything you deem suitable)

GOTCHA:

Remember NFS comes from Linux – even if you are using Microsoft SFU export/share names ARE case-sensitive!

Occasionally, the NAS maybe unavailable. If you want to force a reconnection to an NFS export you can do this through the Service Console with the following command:

```
esxcfg-nas -r
```

Note:

Now that you are more familiar with the command-line you could add the NAS connection from the command-line with the following:

```
esxcfg-nas -a backups -o vcb.vi3book.com -s /vcb-backups
```

Test Full VM Restore

Warning:

Do not start this process with the next step until you can verify you have a backup of the VM.

1. Locate you backed up VM in the Inventory.
2. Power Off the VM.
3. Right-click and choose Delete from Disk in the menu.
4. Open up a PuTTY session on the ESX host configured for NFS mount point serviced by SFU running on the VCB.
5. Ensure you have root level access. Then issue the vcbRestore command as follows:

```
vcbRestore -h virtualcenter.vi3book.com -u lav-  
ericm -p password -s /vmfs/volumes/vcb-  
backups/mike01/ -o on
```







Note:

-s specifies the subdirectory where the VM's files are located. The -o on switch tells vcbRestore to power on the restored VM when the process has completed.

Remember in the Service Console all commands are case-sensitive, so you must type vcbRestore with a capital R.

6. During the restore process the vcbRestore tool reads the catalog file and registers the VM with the ESX host, as shown in Figure 11.5.

Figure 11.5

Recent Tasks		
Name	Target	Status
 Reconfigure Virtual Machine	 mike01	 Completed
 Register Virtual Machine	 Mike's VMs	 Completed

At the Service Console the vcbRestore utility will give you a progress bar to indicate what is happening.

Converting

```
" /vmfs/volumes/virtualmachines/mike01/mike01.vmdk" (VMFS  
(flat)):
```


```
0%=====50%=====100%
```

```
*****
```

In this case the utility is also converting the 2gbsparse disks back into a “monolithic” or “thick” disk.

Towards the end of the restore process you might find that the friendly name of the VM is temporarily unfriendly UUID as the path to the VMX file. If you wait for vcbRestore to complete fully it does reconfigure the VM to show its friendly name. Figure 1.6 shows this unfriendly name.

Figure 11.6

```
 sanfs://vmfs_uuid:46811624-c6bf464d-40df-001560aa6f7c/mike01/mike01.vmx
```

Mounting a Virtual Disk to Retrieve an Individual File

The VCB Framework does come with a mountvm utility. This allows you to mount the backed up version of the virtual disks to a folder so that you can navigate the file and folder structure within it. It has a very simple syntax:

```
mountvm -d d:\backups\mike01\scsi0-0-0-mike01.vmdk  
-cycleId d:\mnt\mike01
```

The `-d` switch indicates a path to the virtual disk you wish to use. The `-cycleID` generates a unique ID for the mounted disk in Windows; it “cycles” it so it’s always unique. The last path statement is the mounting location.

If you navigate to the mounting location you would find the file system of the virtual disk there. It is possible to then copy files to the VM using conventional Windows shares.

Once you are finished with the virtual disk it can be unmounted with the `mountvm` command:

```
mountvm -u d:\mnt\mike01
```

Integration Modules and Commercial Backup Tools

Most third party vendors who support VCB will provide you with an integration module. Alternatively you may find them on VMware’s website. This usually takes the form of a zip file which you extract to the VCB directory. Once extracted you run an install batch file which allows you to configure the files for your product. The core file is the `config.js` file which is held in the `/config` directory of your VCB installation. This will typically include the path to your mount directory and authentication credentials are required to initiate the backup process. Additionally, there is normally a `readme.html` file which will walk you through the setup.

It’s perhaps worth saying that integration scripts and their README files vary in the quality, reliability, debugging, and error trapping. It is very much recommended to do research around the default scripts provided for how you can manually correct and improve them. This is beyond the scope of this book because there are simply too many vendors for us to test and verify. Additionally, scripts are quite frequently re-released and so anything in a published book regarding this aspect of the product is likely to become outdated quickly.

In this scenario, the VCB Proxy is both the backup server and backup client since the VCB Proxy is backing up files that are mounted locally. If your backup software doesn’t install the backup client then it should be installed manually.

As an example of a typical integration scripts I've decided to demonstrate this using Symantec BackupExec 9.1. This version of BackupExec is supported by VCB. The main reason I selected this vendor is because they are very popular, and I had an existing evaluation copy which I could install to my VCB server. This should not be regarded as an endorsement or recommendation of Symantec software – but as a typical demonstration of configuring integration scripts.

Instructions for Backup Exec 9.1

Setup the VCB Integration Module

1. After installing VCB and if required Microsoft SFU, install Backup Exec.

Note:

You will need complete a wizard after the installation when you first run the BackupExec software. I like to setup a directory on my backup partition called D:\BackupExec Files as a location for disk-to-disk backups.

2. Download the Integration Module for BackupExec to the VCB Proxy from VMware's website at this url:

http://www.vmware.com/download/vi/drivers_tools.html#backup

3. Extract the backupexec directory in the zip file to C:\Program Files\VMware\VMware Consolidated Backup Framework directory.
4. Open a command-prompt to the C:\Program Files\VMware\VMware Consolidated Backup Framework\backupexec directory.
5. Run the setup script by typing this:

`Install`
6. Press **[Enter]** to accept the path for the VCB Framework Software.
7. Press **[Y]** to confirm the configuration path.
8. Choose **[Y]** to review the VCB Framework Configuration File.

Note: Entries in the configuration file marked with two double-forward slashes // are comments. You must remove the // entries for the changes to take effect. If not the default options will be used.

In my case I un-remarked

// BACKUPROOT="C:\\mnt";

and replaced it with the disk partition and directory of personal preference. For example, I used the following:

BACKUPROOT="D:\\backups";

9. Next **modify the HOST entry specifying your VirtualCenter or ESX host**. In my case I modified:

HOST="bu02.eng.vmware.com";

and replaced it with the name of the VirtualCenter or ESX host system:

HOST="virtualcenter.vi3book.com";

10. **Lastly I modified the USERNAME and PASSWORD entry with an account that has rights to connect to VirtualCenter** for backup purposes.
11. **Then in Notepad**, choose **File and Save** and **File and Exit**.
12. Choose **[Y]** to review the **BackupExec Configuration file**.

Note:

This file allows you to alter the temporary location used to store your pre- and post-scripts during the backup process. The default setting is held in the /generic/config.js file. I choose not to make any changes here – but wanted to you see that could be changed if required.

13. **In Notepad**, choose **File and Save** and **File and Exit**.
14. Choose **[N]** view the readme-backupexec.html file.
15. **Reboot the VCB Proxy to allow BackupExec and Windows to be updated with the new environmental variables.**

Note:

You can just restart the BackupExec services and close and re-open your command-prompt. However, for peace of mind I decided to opt for a full reboot after running the install script.

Setting up a Backup Job for Full-VM backup

1. In the BackupExec Management Console select Job Setup.
2. Under Backup Tasks select New Job.
3. Under the Source/Selection window in the Selection List name type a friendly name for selection to be backed up like this:

Backup mike01.vi3book.com VM – Full VM Backup.

4. Under View Format click Text radio button.
5. Click the Insert button.
6. Type in the path to the mount point, using a FQDN to describe the directory name followed by -FullVM like this:

D:\Backups\mike01.vi3book.com-FullVM

Note:

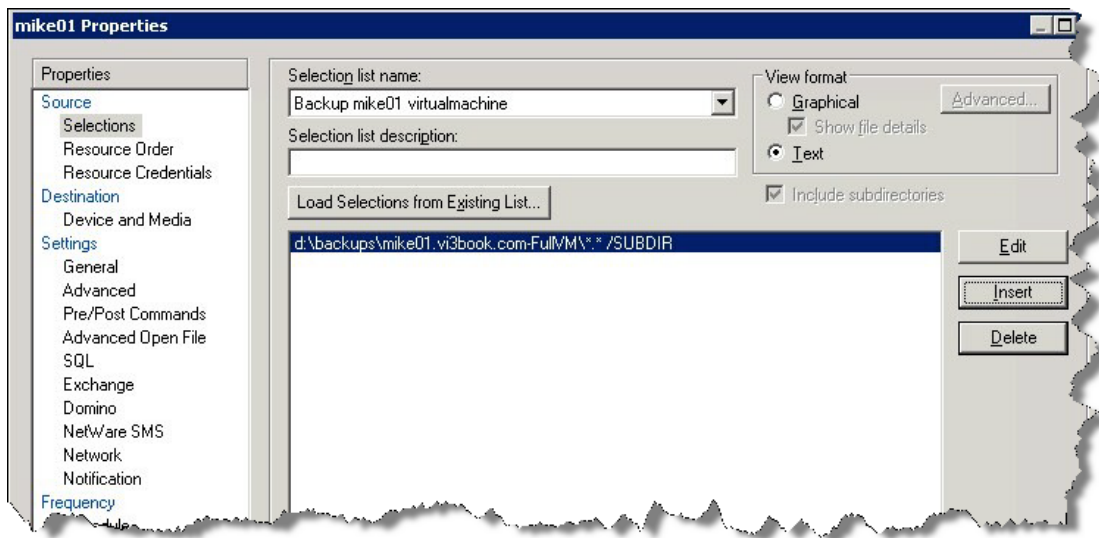
The integration script for BackupExec currently does not allow for multiple VMs to be backed up with one single job. At the moment this means that each VM requires its own backup job. The scripts could be modified to allow many VMs to be contained within one backup job. Examples of how to do this are available on the internet. One example is listed here:

http://searchservervirtualization.techtarget.com/tip/0,289483,sid94_gci1233940,00.html

7. Click OK to the Advanced Selections Dialog box.

Figures 11.7 and 11.8 show the respective BackupExec dialog boxes.

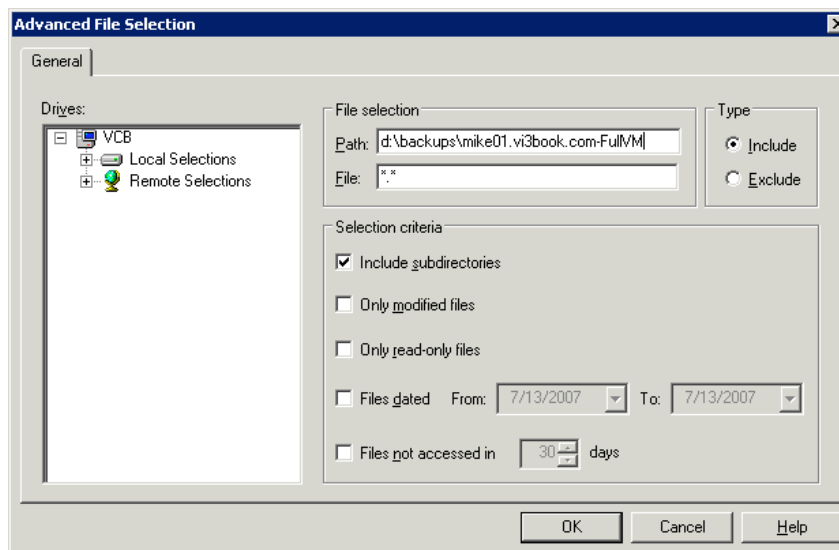
Figure 11.7



Note:

The reference to *.* /SUBDIR is added automatically by BackExec because the Advanced File Selection dialog has the option "Include Subdirectories" enabled by default.

Figure 11.8



8. In the Backup Job Properties Box, select Settings and General.
9. Specify a Job Name.

Note:

Type something memorable and short as you will need to supply this later on in other parts of the job. In my case I called the job mike01-FullVM.

10. Select your Backup Method – in my case I selected COPY – Back Up Files.

Note:

Not all backup methods are supported. Consult the readme.html file associated with your vendor's integration script. For example, BackupExec supports the following:

Copy - Back Up Files

Full - Allow incrementals and differentials using modified time.

Differential - Using modified time

Incremental - Using modified time

Daily - files that Changed Today

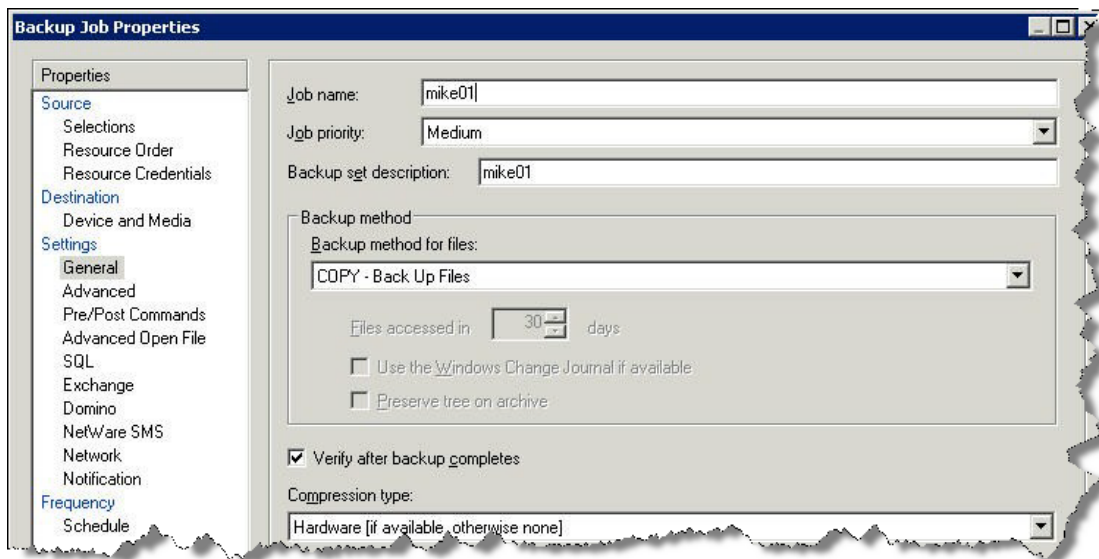
working set - All files last accessed in (x) days

Note:

Although the backup type specified above may allow it, do not enable the "Use the Windows Change Journal if available" option. It is not supported.

Figure 11.9 shows the options selected in Backup and Settings in the BackupExec Management console for Settings, General.

Figure 11.9



11. Next, in the Settings and Pre/Post Commands,
12. Add the following entries to the Pre and Post edit boxes respectively:

"C:\Program Files\VMware\VMware Consolidated Backup Framework\backupexec\pre-backup.bat" mike01-FullVM
mike01.vi3book.com-FullVM

"C:\Program Files\VMware\VMware Consolidated Backup Framework\backupexec\post-backup.bat" mike01-FullVM

Note:

Here mike01-FullVM is the name of the backup job and the FQDN-FullVM indicates the path to the mounting point to the following:

d:\backups\mike01.vi3book.com-FullVM

Warning:

You must supply the speech marks for the path to the scripts. Using spaces in the jobname and the FQDN are not supported. The reference to -FullVM is required and hard-coded so the pre-backup script knows which type of backup to trigger.

Note:

Essentially, these are variables that are passed to vcbMounter. It is vcbMounter that will create the snapshot, mount the VM's files to d:\backups\mike01.vi3book.com-FullVM, and then begin the export process. Once the export completes, vcbMounter will then merge and delete the deltas in the snapshot. From that point BackupExec will take over backing up the exported version of the VM held in the d:\backups directory. In respect, the pre- and post scripts in the integrator are automating the vcbMounter backup process.

GOTCHA:

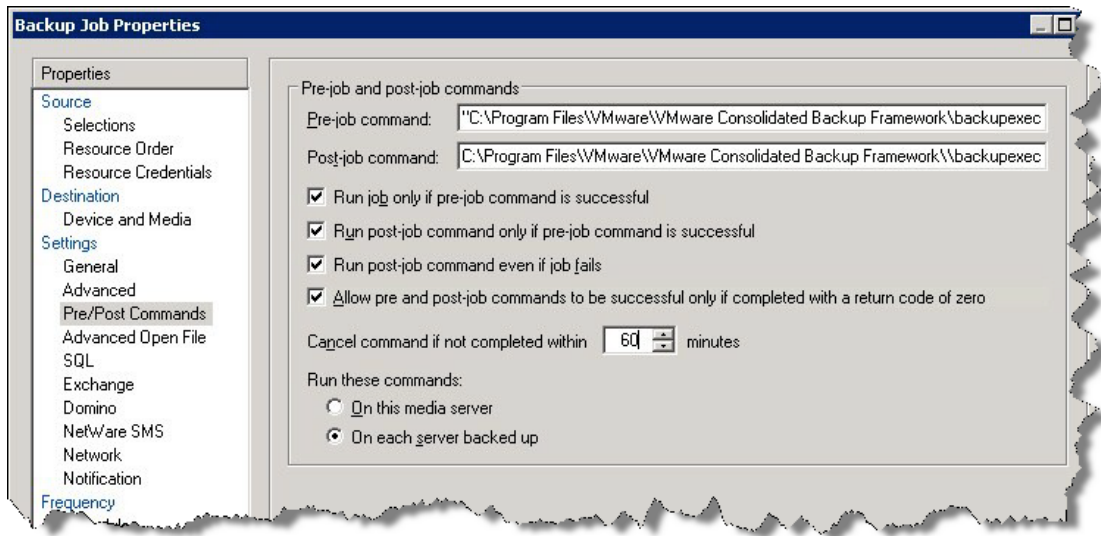
The README file that accompanies the backupexec integration modules incorrectly makes reference to a pre-command.bat and post-command.bat. This is incorrect and the correct file names are currently pre-backup.bat and post-backup.bat. Additionally, the README file does not warn you about using speech marks for long file names and directory paths with spaces.

13. Enable the run options including the option called "Allow pre and post-job commands to be successful only if completed with a return code of zero."
14. Increase the timeout for the cancelled command to be a larger value. In my case, I allow a 60 minute timeout.

Note:

This allows enough time for the post-script to trigger a merge and delete of the snapshot delta without the backup terminating the script due to a timeout. Figure 11.10 shows my selections and path statements.

Figure 11.10



15. Lastly, select a destination device or media for the BackupExec backup. I prefer to do disk-to-disk, followed by disk-to-tape backups for offsite usage. Select your destination using Destination and Device and Media.
16. Click the Run Now button.

GOTCHA:

Be careful not to revert or commit the VCB Snapshot during the backup process.

Note:

The process begins by applying a snapshot and exporting the VM to the d:\backups\ directory. Once the export has completed, the snapshot is merged and deleted – and the VM is fully operational at all times.

Next, BackupExec backups up the files in the d:\backups export directory. In my case this would be:

d:\backups\mike01.vi3book.com.

After the backup process is completed you will have two versions of your VM. The exported version is held in d:\backup and the BackupExec version with

the .bkf extension. You could delete the exported version held in d:\backups, as the BackupExec would restore the VM to that location anyway, ready for a Service Console vcbRestore.

Test Restore of a Full-VM Backup with BackupExec

First we will wipe all traces of the original VM from both VirtualCenter and from the exports directory on VCB Proxy.

1. Power on the Test VM and use the right-click Delete from Disk option.
2. Delete the Exported VM from the D:\Backups Directory.
3. Open the BackupExec Management Console.
4. Select the Restore Tab.
5. In the edit box under "Selection List" type a friendly name for the Restore Job like "Restore VM Mike01 FullVM."
6. Expand the + next to the name of the VCB Proxy and Select the Backup Job you wish to restore.
7. Click Run Now.

Note:

This should restore your VM back to the d:\backups directory, in my case D:\Backups\mike01.vi3book.com. Once the BackupExec restore has completed we can now restore it through the network using the NAS connection configured earlier.

8. Logon to the Service Console with root privileges.
9. Restore the VM with the vcbRestore command as follows:

```
vcbRestore -h virtualcenter.vi3book.com -u lavericm -p password -s /vmfs/volumes/vcb-backups/mike01/ -o on
```

Setting up a Backup Job for File-Level with BackupExec

A file-level backup allows you to navigate inside a VM's virtual disk and select individual directories or files. Rather than specifying the file selection manually

it is possible to enable a GUI-based browse of the VM's disks prior to running the backup job. You might find a graphical method of selecting files easier to manage than having to type long paths to files and directories. Once the files are selected in the selection list, the backup job can be run without the browse facility. In this example I have added a second disk to the mike01 VM and copied some sample files there.

1. Under Backup Tasks select New Job.
2. In the Backup Job Properties Box select Settings and General.
3. Specify a Job Name.

Note:

Type something memorable and short as you will need to supply this again later on in. In my case, I called the job mike01-File.

4. Select your Backup Method. In my case I selected COPY – Back Up Files.
5. In the BackupExec Management Console select Job Setup.
6. Keep BackupExec open.
7. Open a command-prompt and change into the C:\Program Files\VMware\VMware Consolidated Backup Framework\backupexec directory, the run:

browse-start mike01-file mike01.vi3book.com

Note:

This should produce the following output:

```
Opened disk: blkfst://snapshot-959[virtualmachines]
mike01/mike01.vmdk@virtualcenter.vi3book.com:902?xxxx/xx
xx
Proceeding to analyze volumes
Done mounting
Volume 1 mounted at
D:\backups\mike01.vi3book.com\digits\1 (mbSize=4086
fsType=NTFS )
Volume 2 mounted at
D:\backups\mike01.vi3book.com\digits\2 (mbSize=2039
fsType=NTFS )
```

Volume 1 also mounted on
D:\backups\mike01.vi3book.com\letters\C
Volume 2 also mounted on
D:\backups\mike01.vi3book.com\letters\D

8. Return to BackupExec.
9. Under the Source/Selection window in the Selection List name type a friendly name for the list of files to be backed up. Below is an example:

Backup mike01.vi3book.com VM – File Backup of D:

10. Ensure that under View Format that Graphical and Show file details is selected.

Note:

You should now be able to navigate the directory tree of d:\backups and find the files to be included in the file selection list.

Figure 11.11 shows my directory structure where I have selected everything in the D: drive of the VM except the IT directory.

Figure 11.11

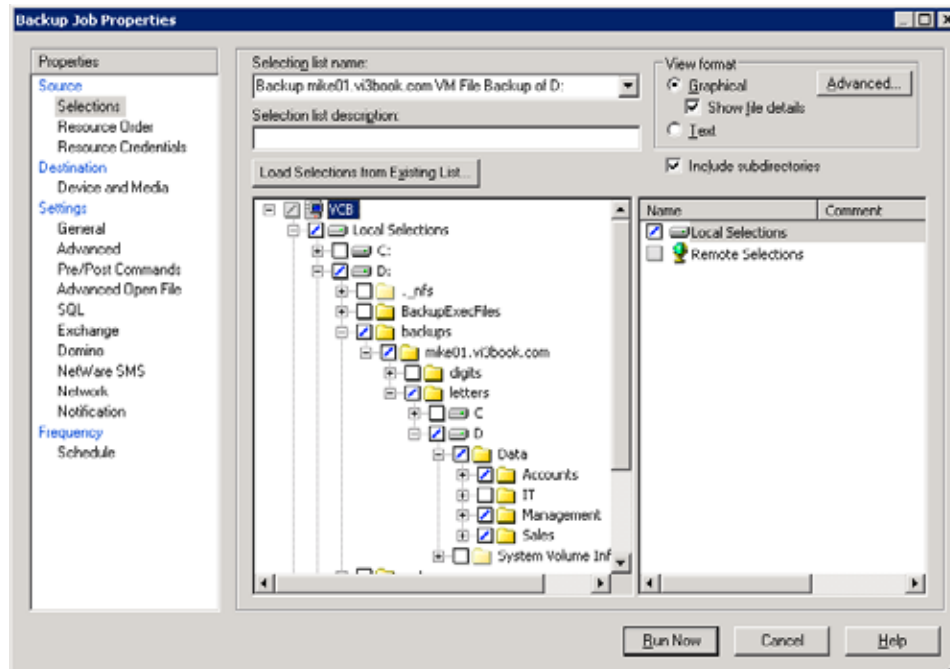
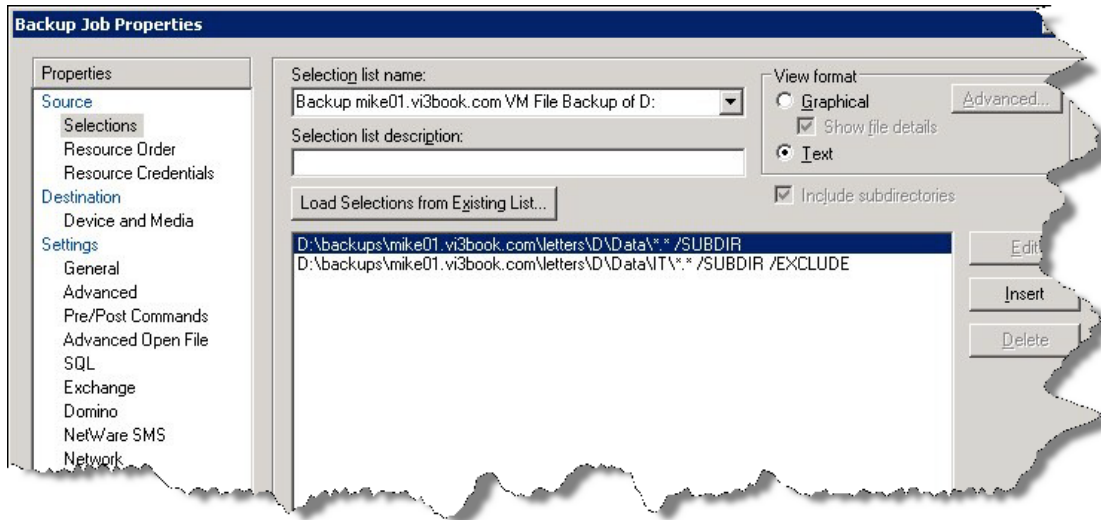


Figure 11.12 shows the View format which includes both the included and excluded directories.

Figure 11.12



Next in the Settings and Pre/Post Commands

11. Add the following entries to the Pre and Post edit boxes respectively:

```
"C:\Program Files\VMware\VMware Consolidated Backup Framework\backupexec\pre-backup.bat"
```

```
mike01-file mike01.vi3book.com
```

```
"C:\Program Files\VMware\VMware Consolidated Backup Framework\backupexec\post-backup.bat" mike01-file
```

Note:

Here mike01-file is the name of the backup job and the FQDN indicates the path to the VM mike01.vi3book.com.

12. Enable the run options including the option called "Allow pre and post-job commands to be successful only if completed with a return code of zero."
13. Increase the timeout for the cancelled command to be a larger value. In my case I allow a 60 minute timeout.

-
14. Select a destination device or media for the BackupExec backup. I prefer to do disk-to-disk followed by disk-to-tape backups for off-site usage. Select your destination using Destination and Device and Media.
 15. Before you run the backup, unmount the VM from the command-line using the following:

```
browse-end mike01-file
```

Warning:

You must unmount the VM's files before starting the backup after running the browse-start command. Otherwise the pre-backup script will attempt to apply a second snapshot and mount the VM for a second time. This will cause the backup to fail.

16. Click the Run Now button.

Note:

The process begins by applying a snapshot and mounting the VM to the d:\backups\ directory. Next, BackupExec backs up the files in the d:\backups mount directory.

17. After the backup process is completed the mount point is unmounted, and snapshot deltas are merged into the VM. This will leave you with backup files with .bkf extension.

Test Restore of a File-Level Backup with BackupExec

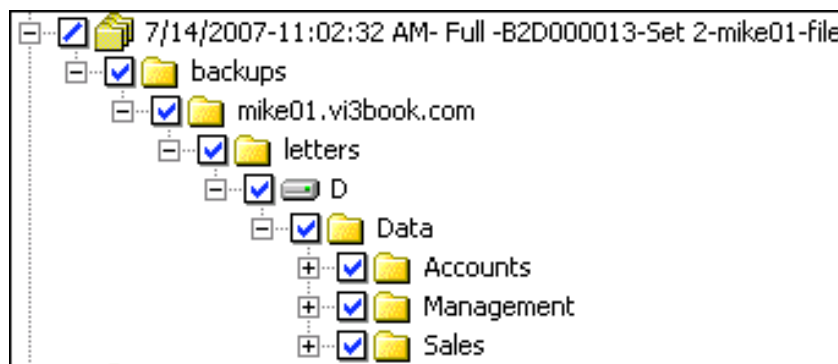
Restoring lost data files with VCB is a little less slick than a full backup. There isn't a handy vcbRestore-like utility to assist you in the process. The procedure is one of restoring the backup VCB proxy and then manually copying the lost files back to the VM. Another approach is to install a backup agent to the VM and restore the files that way. If you wish install a backup agent remotely to a VM after the main installation of BackupExec select the following in the menu.

1. Tools
2. Serial Numbers and Installation
3. Click Next in the Wizard.

-
4. Remove the tick next to Local Install.
 5. Enable the tick next to Remote Install.
 6. Right-click Windows Agent/Option Computers and Choose Add Computer.
 7. Locate your VM in the list, and Click OK.
 8. Enable Advanced Open File option and Remote Agent for Windows Servers.

However, I feel this in a way negates the whole point of tools like VCB, where we are trying to avoid the financial and network penalties of installing a backup agent. Fundamentally, the method is up to you. If you click the restore tab in BackupExec you should see a new catalog to select. Figure 11.13 shows the individual files I backed up from the D: drive of my test VM.

Figure 11.13



Conclusion

This concludes the chapter on backup and VMware Consolidated Backup. As you can see, new doors are opening in the world of backup triggered by virtualization. The problem is that door isn't quite fully open at the moment. The Holy Grail will be both backup and restore via the SAN offloaded to a backup proxy similar to VCB.