Paragon Drive Backup Enterprise Server Edition

Best Practices for MS Virtual Server

Contents

1 Introduction

This paper addresses some aspects of a Microsoft Virtual Server data protection by using Paragon Drive Backup Enterprise (DBE). It describes the concepts, limitations and best practices for Paragon Drive Backup Enterprise to protect operational solutions based on Microsoft Virtual Server. All mentioned recommendations are generic and not specific for a certain Virtual Server configuration.

1.1 About Drive Backup

Paragon Drive Backup Enterprise is a backup tool that implements the best of volume imaging techniques for reliable, fast and convenient data backup and restoration. The program includes end-user tools for building and automating recovery and replication procedures. It implements high-performance algorithms for intelligent data analysis and processing, provides an optimized manipulation for a large set of filesystems that covers all popular filesystems for Windows and Linux platforms and more features.

1.2 Backup Concepts

1.2.1 File-Level Backup and Volume Imaging

There are two concepts about backup subject. A *file-level backup* is oriented to store separate files. A *volume-level backup* or *imaging* is oriented to store whole filesystem of a volume.

A *file-level backup* naturally provides an intuitive and flexible way to select objects to store. File-oriented backup tools allow choosing any combination of both local and networking accessible files. A file-level data restoration allows to selectively restore only damaged files without affecting other ones. However, there are important file-related system objects which are not files and usually cannot be stored, restored and even accessed from a file level (operating system bootable code, for example).

A *volume imaging* can store any metadata associated with files including distribution information, security data, quotas, extended attributes, named streams, multiple hard and symbolic links and so on. Disk imaging tools generally provide higher backup performance because they do not involve filesystem drivers to the process. In addition, imaging tools can backup offline filesystems and even ones not being supported by a host operating system. A data restoration generally does not require a host operating system to run, so that volume-imaging technique is a perfect choice for system cloning and disaster recovery tools. Disadvantages of volume imaging are that it cannot be applied to remote resources and a general ineffectiveness of backup and restore of selective files within the volume-imaging framework.

1.2.2 Data Consistency

The fundamental requirement to backup is saving of *data consistency*. This means that if applications are stopped, and data are restored, and applications are restarted, they will run smoothly with restored data.

A *data consistency* is conditionally divided into a *physical* and *logical* consistency. A *physical consistency* means storing information about involved data files and file-related attributes in a state that is interpreted by applications as "integral", or "valid", or "repairable on-the-fly". A *logical* consistency means an application-level correctness of stored business data. An automatic correction of minor inconsistency of business data is usually provided by the *transactions mechanism*, which is the basis of modern technologies of information processing.

1.2.3 Offline and Online Backup

As regards to the data consistency concept, offline data are in consistent state (with the only condition that an application or a system was shut down correctly). Offline data archiving is referred to as *offline backup*. Its advantages are ensured data consistency, increased backup speed (due to absence of concurrent data access), resource saving and low impact to a system performance.

The disadvantage is that offline backup is not applicable for "24x7 availability" systems. For continuously working systems *online backup* methods should be applied.

1.2.4 Snapshots

The great challenge to online backup is to provide a coherent state of all open files and databases involved in a backup, under the condition that applications may continue writing to a disk.

The "volume snapshot" concept has met that challenge. A *snapshot* is a point-in-time copy of a volume involved to a backup process. It must be quickly created at a period when applications do not write to disk. Once snapshot has been created, a backup utility copies data from it while applications continue working with an original volume. Modern snapshot technologies reduce *backup window* down to few seconds.

There is hardware and software based snapshot solutions. *Hardware snapshots* such as the *Split-Mirror* technique instantly provide an independent copy of a whole data set; they make no impact to the system performance and can participate in off-host backup solutions. Disadvantages are that hardware snapshots are naturally hardware-dependent, double requirements to storage resources and some time is required in order to re-synchronize a hardware snapshot after it has been used, with the actual disk data.

Instead, *software snapshots* are hardware-independent and resource-saving solutions. The disadvantage is that they cannot make a copy of a whole data set instantly, so that they run and make additional computing load to a host system until a backup routine is ended and a snapshot is destroyed.

Unlike hardware snapshots, types of software snapshots, which are used in practice, are not independent from original data. In case an original volume fails, all of its snapshots fail as well. Software snapshots do not really protect data; they are used as supplementary mechanisms in backup solutions.

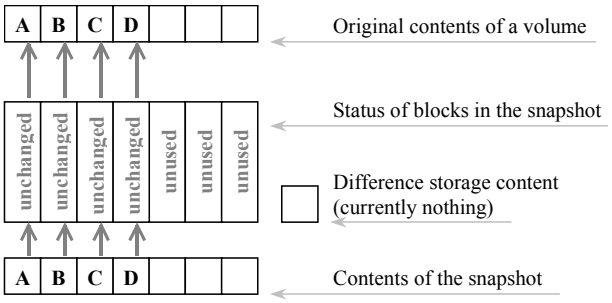
Paragon Drive Backup provides the original online backup technique (named *HotBackup*) and supports any snapshot technologies that can participate in the MS VSS framework.

1.2.5 Copy-On-Write

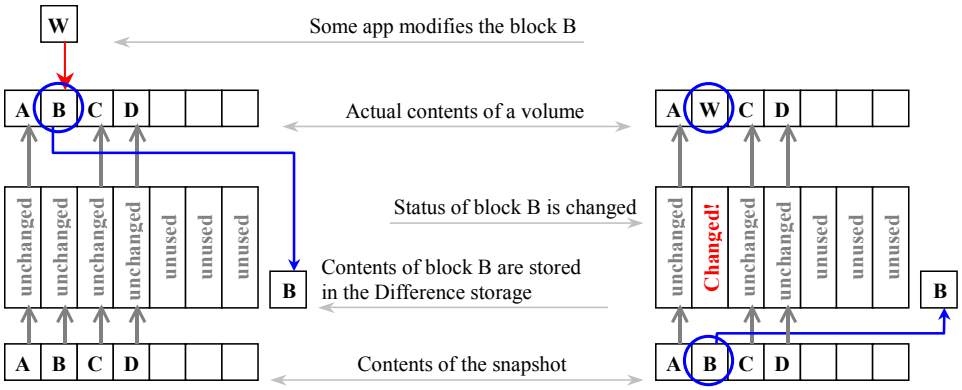
The *Copy-On-Write* (COW) method predominates among software snapshot techniques. It is based on the idea of preserving original contents of modified blocks before modifications are written to a disk, in special *difference storage*. COW maintains an original volume in the actual state and stores its original content at the moment of snapshot creation in the difference storage.

A system-level agent, which is responsible for snapshot maintaining, must keep track of all changes being made on a volume. On first attempt to overwrite a block, the snapshot agent copies original block's content to the difference storage and then allows modifying that block. When a backup utility acquires some block from the snapshot, the agent determines its status and takes a "changed" block from the difference storage while "unchanged" block is taken from the original volume. The following illustrates COW basics:

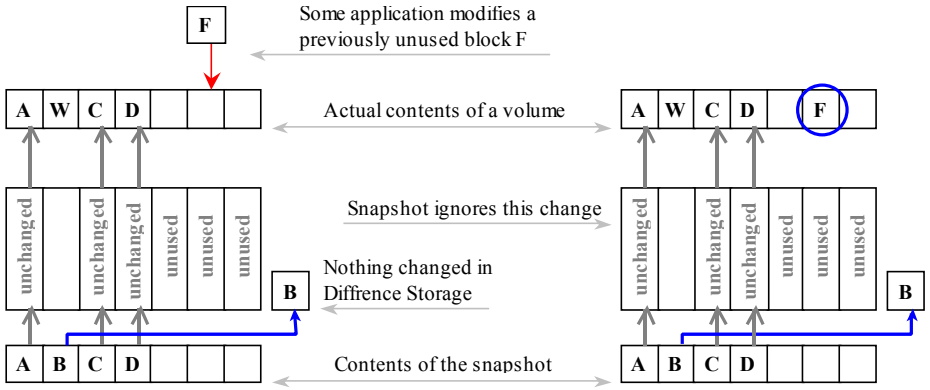
1. A volume was partially in use. Blocks A, B, C, and D contained data. A COW based snapshot was created for this volume. The snapshot provider watches only the used blocks of a volume. Initially they are marked as "unchanged".



2. Some application tried to modify the block B. Before modifications are made, the snapshot provider copies contents of the block B to the difference storage. The block is now marked as "changed". Then the block B is updated. The snapshot will redirect all queries to the block B to data stored in the difference storage. Queries to blocks A, C and D will be directed to appropriate blocks of the volume.



3. Some application tried to modify the block F, which was not originally in use. This block was not included in the snapshot. The snapshot provider does not take care of this change. Nothing is changed in the snapshot data.



1.2.6 Data Synchronization

Another problem for online backup functionality is that applications may temporarily hold open files in logically inconsistent state. The snapshot technique does not solve that problem as it concerns solely to business application operation. The true reason of the problem is that applications are unaware about a backup routine running and do not synchronize their data.

Microsoft has made a great attempt to solve the problem. The snapshot backup framework referred to as Volume Shadow Copy Service (VSS) has been built in latest versions of Windows (Windows XP, Windows Server 2003 and Windows Vista). VSS includes mechanisms for notifying applications about a backup, interchanging of related information between VSS participants and synchronizing execution of software components involved to the process.

VSS has several significant limitations:

- Only VSS compliant applications can benefit from VSS framework.

- VSS is a local solution that works within a single host.
  Remote applications and distributed data systems aren't controlled by VSS.
- VSS currently works to its full capacity on Windows 2003 only.

1.2.7 Write Inactivity Paradigm

There are other (partial) solutions for synchronization problem that are applicable for non-VSS compliant software. There is a popular solution based on the paradigm of *Write Inactivity Period* (WIP) that was introduced by St.Bernard Software Company in the middle of 1990-th.

It is supposed that business applications for intensive information processing use transactions mechanisms. A *transaction* collects I/O writes into a compact group in order to reduce a chance of incomplete transaction committing if a failure of any type occurred. Data are in consistent state between transactions, so that a period when application(s) do not write to a disk is the best moment for a snapshot capture. This period is referred to as WIP – *Write Inactivity Period*.

2 Technology Overview

2.1 Paragon HotBackup Description

Paragon HotBackup is an online backup technology for Win'NT+ family operating systems. It was developed in 2001 and integrated to all company's backup solutions in 2002-2003. Currently it supports all versions of Windows NT4, 2000, XP and 2003 (including x86, IA64 and AMD64 versions).

2.1.1 Hotbackup Concepts

HotBackup is not a snapshot technology. However its concepts appear to be similar to ones used in software snapshot technologies. In particular, a sort of WIP observation and COW scheme are implemented.

During an online backup, Drive Backup uses the kernel mode driver HOTCORE.SYS in order to monitor and control write activity of applications and an operating system. The driver intercepts disk I/O requests and implements the most time-critical part of COW scheme while the Drive Backup utility includes disk data analysis and archiving functions.

The HOTCORE driver is installed during the standard Drive Backup setup procedure, and it is the reason why the system restart is required in order to complete the setup procedure. HOTCORE does nothing until it is activated by the Drive Backup. In the idle mode the driver bypasses any calls, makes no impact to the disk subsystem performance and only takes few kilobytes of system memory.

2.1.2 How it Works

HOTCORE driver is activated only in case the online backup is performed. In an offline backup mode, the driver is not involved to the process.

- Drive Backup activates the HOTCORE driver in the beginning of the "physical" backup.
- HOTCORE waits for a pause between IO writes on the targeted volume.
- When the pause is observed, the driver takes a "snapshot".

Within the framework of HotBackup, a "snapshot" is a map of blocks to be protected by the COW scheme against loosing their original contents. The process requires the close cooperation between the driver and the utility. Finally, the "protection area" of the snapshot includes only used blocks with optional exception of *excluded files* (e.g. PAGEFILE.SYS and HIBERFIL.SYS). The embedded module for filesystem analysis allows reducing this step down to few seconds or less.

- During the snapshot capture, the driver watches the volume against I/O writes. If a write occurred before "snapshot" is created, the step is repeated.
- Upon successful snapshot creation, the driver applies the COW scheme to the "protection area".

- The utility starts the backup process. Archived blocks are immediately excluded from the protection area, so that the area is shrinking during the backup.
- If a foreign application tried to modify a "protected" block, the driver preserves its original contents in a buffer. Initially, blocks are stored in a memory buffer. When it becomes near full, the utility moves blocks to temporary files (named like "X:\hb_nn.tmp", where X: is a drive defined in the Settings).
- Normally Drive Backup performs the fastest streamlined backup of used blocks. However, if temporary files grew very fast or became very large, the utility pauses the streamlined mode and begins emergency backup of buffered blocks.

In online backup the following time-related restrictions are used:

- A snapshot must be created within 60 seconds.
- Buffered data must be processed within 10 seconds.

These restrictions are hard-coded and cannot be changed. If any of these restrictions were not satisfied, the online backup session fails. If a user aborted the backup operation or the utility failed, the driver automatically switches to the idle mode in 10 seconds.

2.2 MS VSS Basics

Volume Shadow Copy Service (VSS) is an open system-level framework for snapshot backup solutions. It was developed by Microsoft in close cooperation with leading vendors of backup solutions. VSS is included in Windows XP, Windows Server 2003 and Vista.

VSS provides mechanisms for notifying applications about a backup, interchanging of related information between VSS participants and synchronizing execution of software involved to the process. These mechanisms ensure consistent backup of online data for VSS compliant applications.

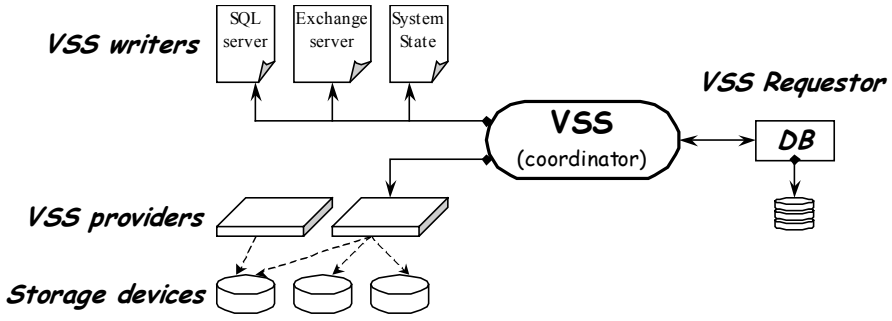
2.2.1 VSS Concepts

VSS is based on concepts of a snapshot and a volume shadow copy. Being invoked by a VSS aware backup utility, VSS creates snapshots for selected volumes and represents them as virtual read-only devices, which are referred to as *shadow copies of volumes*. Once shadow copies are created, a backup utility stores data from shadow copies while business applications continue writing to original volumes.

There are three kinds of software involved to the VSS framework:

- *Providers* (snapshot providers) – tools that create and maintain hardware or software snapshots.
- *Requestors* – utilities that acquire shadow copies, usually backup utilities.
- *Writers* – VSS compliant applications that hold open files on volumes, actual backup objectives.

Within the VSS framework, VSS writers are able to inform other VSS participants about files being in use, file grouping and restoration conditions. A group of files that constitute a whole entity in a business application and should be backed up together is named a *writer's component*. For example, all files that constitute a mailbox store in MS Exchange Server are represented as a single writer's component. VSS writer can be an application itself or a special agent, which provides VSS-to-application interaction.



VSS itself only coordinates activity of providers, writers and requestors. A standard Windows XP/2003 distribution includes the VSS coordinator, the universal software provider (VOLSNAP), several VSS writers for system components and the universal VSS writer for MS Desktop Engine (MSDEwriter), which is responsible for integration of MS SQL Server to VSS framework.

2.2.2 How it Works

Here is only a brief description of MS VSS operation. The comprehensive description of VSS can be found on appropriate Microsoft TechNet pages (e.g. see the descriptive topic "How Volume Shadow Copy Service Works", http://www.microsoft.com/technet/prodtechnol/windowsserver2003/library/TechRef/2b0d2457-b7d8-42c3-b6c9-59c145b7765f.mspx).

1. A VSS requestor (backup utility) invokes VSS to initialize the backup routine.
2. VSS acquires information from all VSS writers. As a result the Writers Metadata Document (WMD) is created. It contains distinctive names of applications and components, restoration parameters, list of files and other information.
3. A requestor receives WMD and makes a decision what to backup. It creates the Backup Components Document (BCD), which defines volumes, writers and components involved to the process and sends it to VSS. As an option, preferred VSS providers can be selected.
4. VSS commands to all involved VSS writers to finish running transactions and then "freeze". VSS waits until all involved writers complete that step.
5. Then VSS commands to appropriate VSS providers to create snapshots for selected volumes.
6. After snapshots are created, VSS allows writers to "thaw" and to write to disk. In addition, VSS asks writers if write operations were indeed suspended during the snapshot creation. If it was not the case, the whole backup procedure fails, snapshots are removed and the VSS requestor is notified.
7. Upon successful completion of above steps (within predefined time intervals), VSS creates shadow copies from snapshots and provides VSS requestor with appropriate references.
8. VSS requestor starts copying information from shadow copies…
9. Upon the process completion, the VSS requestor informs VSS about the backup completion status.
10. VSS informs all involved VSS writers about the backup completion status. VSS writers may use this notification to perform some specific actions (for example, Exchange truncates log files).

A shadow copy can be deleted immediately after the backup completes, or it can persist in the system. In the last case it can be mounted in same way as an ordinary on-disk volume (this feature is available in Windows Server 2003 only).

2.2.3 MS VSS Limitations

MS VSS has several significant limitations:

- Only VSS compliant applications can benefit from VSS framework.
- VSS is a local solution working within a single host. Remote applications aren't controlled by VSS.
- VSS currently works to its full capacity on Windows 2003 only.

2.3 How Drive Backup Integrates with MS VSS

Storage management frameworks obviously provide benefits and can simplify storage management activity. Now Paragon Drive Backup Enterprise is adapted to participate in VSS framework operation as a requestor.

2.3.1 Enabling Backup via VSS

To perform backup operations via VSS services, select the "Microsoft Volume Shadow Copy Service " option:

(menu:) General
↳ Settings…
↳ Hot processing options

↳ Hot processing technology

↳ Microsoft Volume Shadow Copy Service

Note that MS VSS service is available only in Windows XP, 2003 and Vista. In other operating systems, only Hotbackup technology is available for online backup.

Drive Backup provides a simplified VSS management, so that not all VSS options are controllable by a user. The program internally chooses only most reliable VSS operation modes.

2.3.2 How it Works

A VSS aware backup is performed in the following manner:

- A user chooses volume(s) to be backed up in the program's interface.
- When the "physical" backup operation begins, the program receives the compound WMD document from VSS (see #2 in the VSS description topic).
- Drive Backup determines VSS writers having components located on chosen volumes. These VSS writers are included to the BCD document (see #3). In other words, all VSS writers containing files on targeted volumes should participate in VSS operation at the snapshot creation (i.e. should be "frozen" and "thawed" by VSS).
- Drive Backup commands VSS to use the default order of VSS providers invocation: a hardware provider first (if available), a third-party software provider next (if available), the last is Microsoft's universal system provider VOLSNAP (always available).
- The program performs the volume-level backup of the created shadow copy set.
- After the operation completes, the shadow copy set is deleted.

Currently the program does not support restoration via VSS writers. In fact, VSS based restoration is generally just a file copying. Applications should be stopped, or appropriate components should be detached/dismounted manually in order to be restored.

2.3.3 Working with non VSS Compliant Software

As it was mentioned above, there are many popular software products that do not support VSS yet, and it is an obstacle for creating correct backup images of their data by using common purpose backup tools. However, some of these products provide management functions that allow imitating VSS-compliant behavior by quickly freezing an application activity before starting the backup process and resuming an application normal operation immediately after taking the snapshot of volumes. Examples: for Oracle, a special sequence of PL/SQL commands should be executed to freeze-thaw; for Microsoft Virtual Server, a special COM interface should be queried and particular methods should be invoked, and so on.

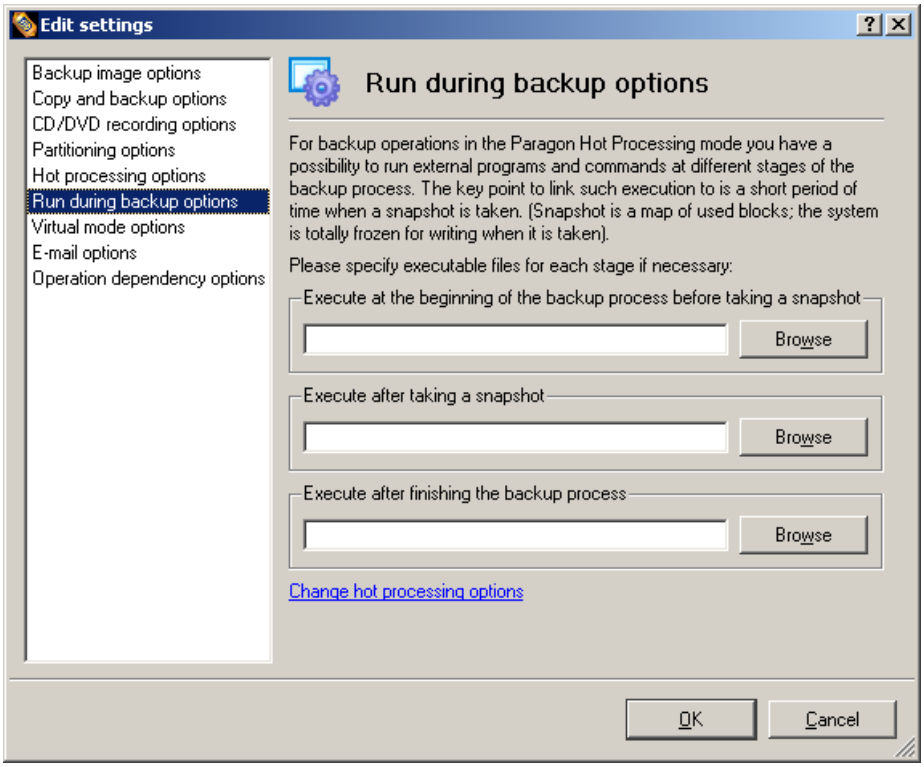
To support this functionality, Drive Backup provides the ability to execute user-defined commands at three crucial moments of the online backup routine, which are:

1. Just before starting snapshot(s) creation;
2. Immediately after snapshot(s) creation;
3. After the whole backup process completes.

It can be any valid commands for the Windows command interpreter (CMD.EXE):

- Executing any internal command of CMD.EXE
- Executing any batch file (*.BAT, *.CMD)
- Running any executable file with any arguments list
- Starting/stopping and pausing/resuming services
- Executing scripts of any type supported by Windows Scripting Host

These commands can be defined in the program's settings, on the "Run during backup options" page:



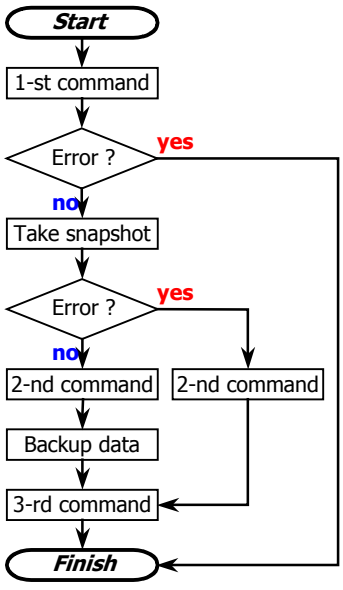
Drive Backup controls *exit codes* returned by these commands and can conditionally abort backup operations. Also the program passes operation-specific information to these commands by setting special *environment variables*. It includes the current status of backup operation, error flags, reference to hot backup technology being used and the list of volumes being backed up. The detailed description of "Run during backup" options is available in the Drive Backup online help.

The recommended purpose of commands:

1. "Execute before taking a snapshot" should try to "freeze" an application, which data are to be backed up.
2. "Execute after taking a snapshot" should "thaw" an application letting it to operate normally.
3. "Execute after finishing the backup" is usually unnecessary. It may be use for performing application-specific post-backup actions.

As concerns to the role of the "Run during backup" options, the backup routine is the following:

- Drive Backup runs the 1-st command trying to freeze an application. If the command failed, it should return a non-zero exit code.
- Upon error (non-zero exit code) Drive Backup aborts the whole backup routine.
- Otherwise (zero exit code) Drive Backup takes a snapshot for requested volumes.
- Upon error of snapshot creation, Drive Backup skips data backup and executes the 2-nd command (resume application) and the 3-rd command (post-backup actions) and then aborts the backup routine.
- Otherwise (snapshot is taken successfully) Drive Backup runs the 2-nd command (resume application).
- Regardless of the result of the previous step, Drive Backup starts backing up snapshot data.
- Drive Backup runs the 3-rd command (post-backup actions), with sending error information to it (if any).



The actions performed by the "Run during backup" commands may depend on results of previous steps of the backup routine. For this reason Drive Backup provides the "Run during backup" commands with information about the backup routine parameters and results of the previous steps execution. This information is stored in special *environment variables*, which are maintained by the Drive Backup at every command execution.

Environment variables:

Name	Values
PARAGON_VSSACTIVE	YES, NO
PARAGON_BACKUPLEVEL	BEFOREBACKUP, AFTERSNAPSHOT, AFTERBACKUP
PARAGON_BACKUPERROR	0..65536
VOLUME_NAME_LIST	List of volume names separated by space (for example "C:\ D:\ F:\")

These environment variables can be analyzed by a command, or "converted" to command's arguments.

Variable analysis example:

"Execute after taking a snapshot" command could be:

```
resume_app.bat
```

Fragment of the "**resume_app.bat**" batch file could be:

```
   .....
if %PARAGON_BACKUPERROR%=0 goto NoError
net send BackupSupervisor "MyApp backup failed!"

:NoError
net continue MyAppService
   .....
```

The above batch file sends the alert message to the "**BackupSupervisor**" user in case the backup routine failed (at the stage of the snapshot creation).

Conversion to arguments example:

"Execute before taking a snapshot" command could be:

```
stop_oracle.bat %PARAGON_VSSACTIVE%
```

Fragment of the "**stop_oracle.bat**" batch file could be:

```
   .....
if %1="YES" goto NoError
net send BackupSupervisor "VSS is not used in backup!"
exit 1

:NoError
   .....
```

The above batch file sends the alert message to the "**BackupSupervisor**" user in case VSS is not used for online backup (the synchronous snapshot of multiple volumes is needed, so that VSS is required); the error is signaled in the exit code – and it should lead to the backup routine aborting.

2.4 Choosing between Online and Offline Backup

The reasons to prefer offline backup are:

- Online backup via Hotbackup or VSS option is slower than offline backup.
- VSS initialization is long and generally unstable under high IO traffic on a targeted volume.
- A resulting image produced by Hotbackup is slightly larger than one produced in offline mode because of non-sequential image structure (see Hotbackup description).

- Neither of online backup options totally eliminates problems that are naturally inherent for online backup technologies in general.

VSS provide data consistency for VSS compliant applications only. Hotbackup does not guarantee 100% data consistency in any case, but only a very high probability of that. In fact, it provides perfect results for any applications that use transactions (such as MS Exchange and SQL servers).

Drive Backup provides some flexibility of choosing between offline and online backup mode. A user can choose between three modes: (a) "always offline backup", (b) "switch to online if a volume was in use" and (c) "always online backup". The differences between these modes are the following:

(a) Always offline mode:
- The program does not backup volumes being in use. It suggests rebooting the computer in order to complete the operation in the "Startup Bluescreen" mode.
- If a volume wasn't in use, the program switches to exclusive use of the volume. No applications can access any files on the volume until the backup procedure is finished.

(b) Switch to online if a volume was in use:
- If a volume was in use, the program performs the online backup. Other applications are allowed to access the volume during the operation.
- If a volume wasn't in use, the program performs the offline backup. As it was described above, no applications can access any files on the volume until the backup procedure is finished.

(c) "Always online backup":
- The program unconditionally performs the online backup. Other applications are allowed to access the volume during the operation.

3 Protecting Microsoft Virtual Server

Microsoft Virtual Server is a machine-level virtualization solution for Windows platform. Virtual Server creates multiple computational environments isolated from each other, by using virtualization abilities of underlying software and hardware. These environments are referred to as "virtual machines".

Virtual machines imitate behavior of physical computers. To use a virtual machine, one should install a guest operating system and populate it by necessary applications. Virtual machines and guest software are able to manage host hardware through the intermediation of Virtual Server and host operating system. Virtual machines and applications can be connected to each other, to a real local network or to Internet by using so-called virtual networks (that mechanism is supplied by Virtual Server).

Machine-level virtualization provides essential flexibility in choosing sets of required applications and appropriate operating systems for getting the best suit of necessary functionality. In addition, virtualization software (e.g. Virtual Server) provide guest operating systems and applications with abstracted hardware layer; for this reason virtual machines can be easily moved between different types of hardware without re-installation of HAL, kernel or peripheral equipment drivers (meanwhile the host operating system depends on underlying hardware).

The downside of virtualization is performance deceleration. Emulating a whole machine(s) provides essential CPU overhead. A complete duplication of instances of operating systems multiplies up requirements for memory and storage capacity.

However that may be, virtual machines become used more and more frequently for business solutions. The more business solutions depend on Virtual Server, the more important it is to protect it. Unfortunately Virtual Server does not include any tools that would allow protecting virtual machines and guest software from disaster.

The protection tasks can be completed by using Paragon Drive Backup Enterprise. The best practices presented in this guide are general principles, not guidelines for specific environments. Some considerations should be taken into account in order to provide reliable backup of virtual machines. This chapter discusses general requirements for Virtual Server configurations and backup and restore procedure details.

3.1 Overview of Virtual Machines Layout

Microsoft Virtual Server does not provide any specific tools for protecting virtual machines and their data and applications. Protection should be made by third party backup software. For this reason, understanding the structure of Virtual Server's virtual machines is essential for making optimal data protection scheme.

There are two general ways for protecting virtual machines and guest software:

1. An "inside way": a backup software is copied to the virtual machine and is used in same fashion as on physical computers.

Advantages:

- Natural and optimal use of a backup utility.

Disadvantages:

- Various operating systems and applications require different backup tools to be used
- Essential overhead of virtualization system (i.e. Virtual Server)
- Prolonged backup.

2. An "outside way": backup software is used on the host environment and backs up virtual machines like ordinary files (of course after implementing some preparation routines for Virtual Server).

Advantages:

- Common backup software can be used (not application-specific one)
- Independence from guest software; same backup utility can be used for any virtual machine
- Minimum overhead
- Fastest backup.

Disadvantages:

- Inability to perform differential or incremental backups of application data; only the "full backup" of the whole environment is available.

As concerns to Drive Backup, the "inside way" is just a generic used of the program, not specific to Virtual Server. Instead, the "outside way" requires performing some Virtual Server specific actions in order to bring virtual machines to a "backup-compatible" state. The following chapters will discuss only the "outside" usage of Drive Backup in regard to Virtual Server.

3.1.1 Virtual Machine Components

Virtual machines contain same set of components as physical computers; the difference is that virtual machines include virtual devices, not real equipment. Some virtual devices are mapped to physical devices of same function while others are emulated by Virtual Server.

For example, "virtual CPU" is mapped to the physical one; virtual machines are executed in the host operating system as specific processes. "Virtual RAM" is mapped into the virtual memory on the host operating system. "Virtual hard disks" are mapped to dedicated files. Virtual floppy and virtual DVD drives can be mapped to physical devices (in this case they allow to use physical floppy disks and DVDs within virtual machine) or to emulated devices, which use image files of appropriate formats as the substitution to real media. And so on.

3.1.2 Virtual Machine Files

In Microsoft Virtual Server, every virtual hard disk consists of one or two files: an obligatory primary file (*.**VHD**) and an optional undo file (*.**VUD**). The primary file holds basic revision of data on the virtual disk while the undo file contains changes being made to on-disk data, i.e. *difference* between the actual and the basic state of the virtual disk.

Undo file is automatically created if a virtual machine was specially configured for maintaining undo information. In this case, the primary VHD-file is not changed while the virtual machine is writing to a virtual hard disk; all

writes are stored to the undo file. When the virtual machine is turned off, a user is able to merge information from undo file to the primary one, or discard it.

When a guest operating system is running, Virtual Server creates a special file that is dedicated for dumping contents of "virtual RAM". This file is named "saved state file" and has VSV extension (*.**VSV**). Virtual Server can suspend any running virtual machine and dump virtual RAM and CPU state to the VSV file. It is named "saved state" because the instantaneous state of a virtual machine is retrievable from the VSV file.

As it has been mentioned in the previous section, virtual floppy drives and virtual DVD drives can be mapped to image files of appropriate formats (IMG for floppies and ISO for CD/DVD). Generally both formats presume that image file is byte-to-byte copy of a disk on sector level – no any compression or interpretation of disk contents.

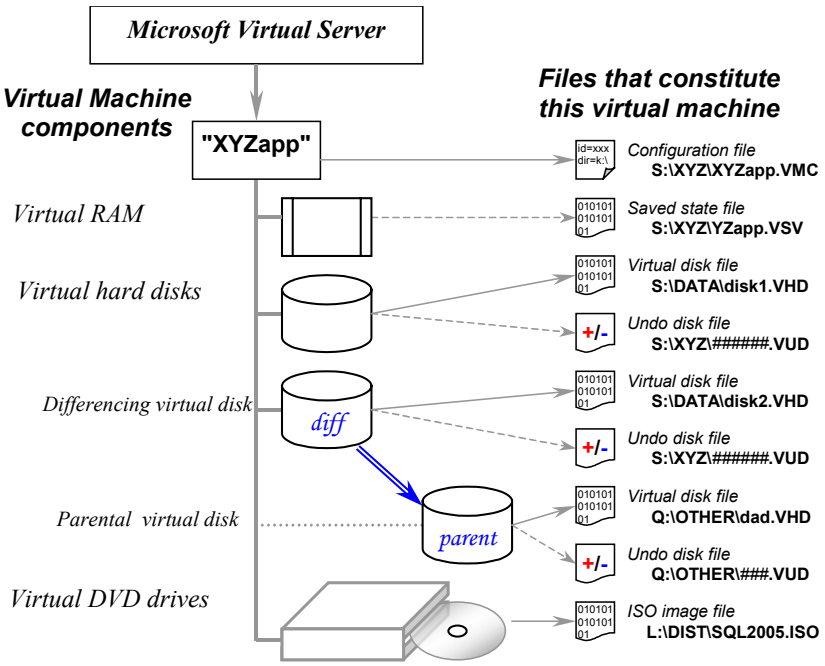
Finally, the layout of a virtual machine is stored in an appropriate configuration file (*.**VMC**).

Resuming the above, a virtual machine consists of following files:

Extension	Amount	Purpose
.VMC	1 per every machine	Virtual machine's configuration file.
.VSV	1 per running machine	"Saved state" file, stores memory dump.
.VHD	1 per virtual disk	Primary file of a virtual hard disk.
.VUD	0 or 1 per virtual disk	Undo file of a virtual hard disk.
.IMG / .IMA	0 – 4 per machine	Virtual floppy image file
.ISO	0 – 32 per machine	Virtual CD / DVD file

These files should not necessarily be placed all in same location. The virtual machine configuration file (*.VMC) and primary files of virtual hard disks (*.VHD) can be located anywhere on the computer as well as virtual floppy and virtual CD/DVD image files. Supplementary files such as saved-state file (*.VSV) and undo files (*.VUD) are placed in same directory with the virtual machine configuration file (*.VMC).

Below is the diagram illustrating relationship between virtual machine components and appropriate files:



Within the framework of Microsoft Virtual Server, there are three types of virtual hard disks, which can be used in virtual machines; these types are *fixed-size*, *dynamic-size* (or *expandable*) and *differencing* virtual disks. Fixed-size and dynamic-size virtual hard disks consist of exactly one primary file with one optional undo file.

Differencing virtual hard disks operate like undo files. A differencing virtual disk refers to a some *parent* virtual disk. Initially a differencing disk reflects contents of its parent disk; however when contents of a differencing

disk are modified, it stores changes in its primary and undo files. So that each differencing disk consists of the primary and undo files and files of its parent disk(s).

3.1.3 Virtual Machine States

To backup a virtual machine correctly, one should save the whole set of its constituent files. However it is not the only condition for successful backup. Selected files should have synchronized data for the whole duration of the backup process or at least for the period of snapshot creation. In addition, all contents of the virtual machine should be stored in files, i.e. virtual RAM and CPU state must be dumped to a disk. This means that not every state of a virtual machine is appropriate for backup.

Within the framework of Microsoft Virtual Server, eleven different states are defined for virtual machines connected to the Virtual Server (the Microsoft Virtual Server SDK help, the "VMVMState" topic):

VM State	Description	Duration	Appropriate for VM backup?
Not connected	*The virtual machine is not connected to Virtual Server. Not described by Microsoft.*	Permanent	YES
Invalid	Invalid virtual machine state. If virtual machine exists, this state should not occur.	Permanent	? NO ?
TurnedOff	The virtual machine is off and the guest was not previously saved.	Long-term	YES
Saved	The virtual machine is off but the guest is saved.	Long-term	YES
TurningOn	The virtual machine is turning on or restoring.	Transient	NO
Restoring	The virtual machine is restoring the previous state of the virtual machine.	Transient	NO
Running	The virtual machine is restored and turned on, not suspended.	Long-term	NO
Paused	The virtual machine is turned on but has paused execution.	Long-term	NO
Saving	The virtual machine is saving the current state of the virtual machine.	Transient	NO
TurningOff	The virtual machine is turning off.	Transient	NO
MergingDrives	The virtual machine is merging undo drives.	Transient	NO
DeleteMachine	The virtual machine is being deleted.	Transient	NO

The diagram of virtual machine states with enabled transitions can be found in the Microsoft Virtual Server SDK help, the "Virtual Machine State Diagram" topic.

Briefly, only "not connected", "turned off" or "saved" virtual machines can be backed up successfully. Other states are not appropriate for backup because:

- "running" and "paused" states do not save virtual RAM contents on a disk,
- other states are actually transient (short-term) states between long-term ones.

The consequence of the above is that running virtual machines should be temporarily stopped to be backed up, at least for the time of snapshot creation; unfortunately it is a short blackout period for virtualized applications.

Switching a running virtual machine to the saved state takes just few seconds as well as restoring a virtual machine from the saved state; it is performed mush quicker than the correct turning off and turning on a virtual machine. Also it does not break open sessions and connections to virtualized applications in case the inactivity period is short enough.

3.1.4 Programmed Control of Virtual Machines

The Microsoft Virtual Server 2005 provides a rich COM interface, which gives the ability to monitor and control the virtual machine environment programmatically, from specially designed applications. This COM interface is integrated to the Windows Management Instrumentation Service (WMI). WMI provides the ability to use COM interfaces and system functions calls from Windows-based scripting languages (VBscript, Windows Scripting Host). The more details about WMI can be found on Microsoft WHDC Home page:

http://www.microsoft.com/whdc/system/pnppwr/wmi/WMI-intro.mspx

The Microsoft Virtual Server 2005 gives the brilliant illustration of power of WMI – all of the web-based user interfaces in Virtual Server use its COM interface via a Web-based scripting language.

The complete description of Microsoft Virtual Server 2005 COM interface can be found in MSDN as well as in the Microsoft Virtual Server SDK help distributed with Microsoft Virtual Server 2005.

As concerns to Drive Backup functionality, there is the ability to create VBS scripts for temporal stopping and quick restarting virtual machines. These scripts force the Virtual Server to behave in such a fashion as if it was VSS-compliant, without real re-development of Microsoft Virtual Server or Drive Backup.

3.2 Backing up Virtual Server Data with DBE

Some concerns should be taken into account in order to use Paragon Drive Backup Enterprise for successful backing up virtual machines managed by Microsoft Virtual Server 2005. The following circumstances should be considered:

- Are there virtual machine's files located on a mapped network drive(s)?
- Are these files spread over multiple volumes?
- Should virtualized application(s) work without interrupt, or short blackout is acceptable?
- Should other virtual machines to be backed up synchronously with this one?

The limitative factors for Drive Backup adaptability for application backup are the following:

- This version of Drive Backup does not backup network drives.
- Only backup via MS VSS option provides the ability to make coherent images of multiple volumes in online backup mode. VSS is available in Windows XP, Windows Server 2003 and Windows Vista only.

3.2.1 Recommended Backup Options

Virtual machines, which remain turned off or in saved state for whole duration of the backup operation, can be successfully backed up in offline mode, by using an offline backup as well as any online backup option. However it requires to hold virtual machine down for a long period.

In case prolonged blackouts are unacceptable, one should choose between online backup options only.

Microsoft Virtual Server 2005 requires Windows XP, Windows Server 2003 or Windows Vista as a host operating system. In these versions Windows, Drive Backup Enterprise supports two online backup option (HotBackup, MS VSS). The use of the "Microsoft Volume Shadow Copy" online backup option is preferred because MS VSS provides an ability to make a synchronous backup of multiple volumes.

3.2.2 Recommended Virtual Machine Layouts

General recommendations:

1. Place virtual machine files on local volumes.
2. Isolate virtual machines from an operating system.
3. Place virtual machines on dedicated volumes.
4. Avoid distributing virtual machines over multiple volumes whenever it is possible.
5. Use high-performance RAID sets for virtual machines location.

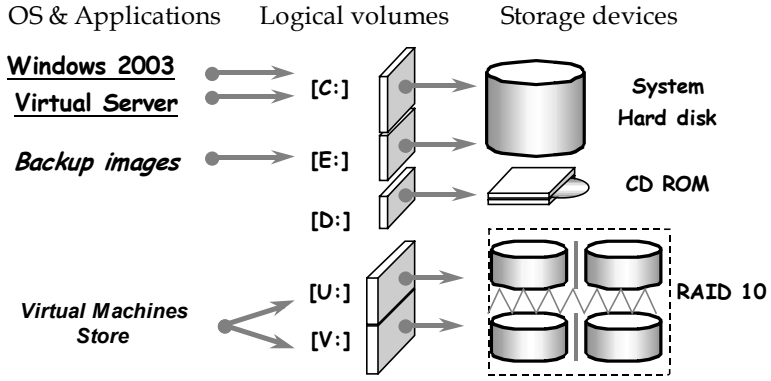
System-dependent recommendations:

6. Use preferably MS VSS option for online backup virtual machines.
   Choose the "Microsoft Volume Shadow Copy" item in the "Hot processing technology" pull-down list in the program settings for that purpose.
7. Virtual machines, which are turned off or saved for a long time, can be successfully backed up in offline mode. For these data, both online and offline backup modes provide successful results.

A preferred system configuration could be like this:

- Virtual machines should be placed on a dedicated fault-tolerant RAID set. For example, a RAID 10 provides both high reliability and performance levels.
- The operating system and Microsoft Virtual Server binaries are placed on the dedicated system hard disk. In case this disk is big enough, it can be partitioned in several volumes. OS and Virtual Server binaries should be placed on the dedicated system volume. Other volume(s) can be used for infrequently accessed data, e.g. for a temporal or a permanent store of backup images.

- In case the budget allows this, add an extra hard disk dedicated for storing backup images and recovery tool set.
- It is preferred to isolate virtual machines from each other.



3.2.3 Unsupported Virtual Machine Layouts

This version of Drive Backup has some limitations in relation to backing up Virtual Server data:

1. A virtual machine that is entirely located on a mapped network drive cannot be backed up by using Drive Backup, because this version of the program does not backup network drives.
2. A virtual machine that is partially located on a mapped network drive cannot be backed up by using Drive Backup, because not all database files can be stored.

Generally, a virtual machine partially located on a mapped network drive may be an obstacle for accurate data restoration for the whole volume. As Drive Backup is unable to backup completely such a virtual machine, it is unable to restore it correctly, too. To avoid possible problems caused by incomplete data restoration, do not apply a volume-level restoration of volumes containing distributed data. Instead, a file-level restoration should be applied for such volumes. Use the "Image Explorer" utility for the file-level restoration from Drive Backup's backup images.

3.3 Restoring Virtual Server Data with DBE

Microsoft Virtual Server 2005 does not support online restoration of virtual machines. Offline restoration is the only available method. In case some virtual machine became corrupt, it should be manually turned off, its constituent files must be restored from a backup image to their original location, and then a machine can be used again. In case some files are restored not to their original location, a machine requires manual intervention for the purpose of re-configuration.

Virtual machine associated data can be restored in two ways – as a part of the volume restoration procedure or by manual extraction of files from a volume's image and placing them to the original location. The difference between the volume-based and file-based restoration methods is the following:

- At *volume-level restoration*, all contents of a targeted volume are rolled back. File system metadata are also restored and put into consistent state. The volume-level restoration is very fast and is able to recover volumes from scratch. However, not only files of interest but any other files and objects, which are located on that volume, will be returned to their previous state.
- At *file-level restoration*, one can extract only required files from an image. It provides selective data retrieval. Other volume's contents are not affected. However the *file-level restoration* is rather slow and requires a targeted volume to be healthy. File-level restoration cannot be used for recovering damaged volumes.

A special precaution should be made for virtual machines deployment issues. The described backup and restore technologies are focused on operability maintenance of virtual machines only. For virtual machine deployment tasks, additional preparation routines may be required (manual use of SYSPREP). Please refer to the Virtual Server online help, the topic "Ways to deploy an operating system to a virtual machine" for more details.

3.4 Choosing between Hotbackup and VSS Online Backup Options

Drive Backup provides two options for snapshot-based online backup in Windows XP, 2003 and Vista. Virtual machines, which remain turned off or in saved state for whole duration of the backup operation, can be successfully backed up in offline mode, by using an offline backup as well as any online backup option. However it requires to hold virtual machine down for a long period.

To avoid prolonged virtual machine blackouts, online backup options should be used. Underlying technologies (Hotbackup and MS VSS) are different by their concepts and features. These options exhibit different levels of operation stability and resulting data consistency. Following considerations can be taken into account when choosing between options:

- VSS option allows creating coherent backup of multiple volumes. With Hotbackup option, Drive Backup always performs an asynchronous backup of multiple volumes.
  This feature can be determinative for backing up virtual machines distributed over multiple volumes.
- Hotbackup requires less memory and disk resources to operate. For example, latest updates of VSS (for Windows XP and 2003) need at least 300MB of disk space per every created shadow copy (at the moment of snapshot initialization), while Hotbackup will consume disk space only under noticeable disk I/O traffic. It will take 300MB only under high disk load.
- Both VSS and Hotbackup may fail to initialize or fail to maintain a created snapshot under conditions of high I/O traffic on a volume being backed up. VSS requires large amount of free space to maintain a snapshot, while Hotbackup mostly needs high backup performance compared to Exchange load. The backup performance depends on (a) throughput of archived volumes and backup storage and (b) compression throughput, which in turn depends mostly on CPU and memory speed.
- Hotbackup initialization is more stable in comparison to VSS. Hotbackup practically always initializes within predefined time intervals, while VSS may fail under high load.
- Only latest versions of VSS are stable enough. It is highly recommended to install the latest service pack available and also check for latest hot fixes for Windows XP/2003 related to VSS.

The tests have revealed a general instability of VSS initialization under a "stressed load" of the Virtual Server. To solve such a problem, either use the Hotbackup option or revise your maintenance plan in order to schedule the backup jobs at less stressed periods.

3.5 End-User Solutions for Virtual Server

As it was mentioned above, Microsoft Virtual Server does not provide any tools for making consistent backup of its virtual machines and their data. On the other hand, Virtual Server includes the program interface that allows controlling virtual machines execution.

Drive Backup provides end-user solutions for Virtual Server data backup in a form of the set of supplementary VBS scripts that allow to suspend temporarily and resume virtual machines. These scripts are intended to be referenced in the "Run during backup" options in order to manage Microsoft Virtual Server during the backup operation.

These scripts are available for authorized Drive Backup Enterprise customers from the Technical Support Website.

3.5.1 General Overview

Script file name	Description
stop_all.vbs	All currently running virtual machines are stopped (switched to the "saved" state).
stop_vol.vbs	Stops running virtual machines that are fully/partially covered by the given volume-set.
stop_vm.vbs	Stops running virtual machines, which names are listed in the input arguments.
resume_all.vbs	Resumes all virtual machines being in the "saved" state.
resume_vm.vbs	Resumes "saved" virtual machines, which names are listed in the input arguments.
resume_in.vbs	Resumes "saved" virtual machines, which names are entered from the console input.
vm_layout.vbs	The informational script. Displays the constituent files of virtual machines and estimates relation between the given volume-set and VM's file-set.

These scripts are subdivided into three groups by their purpose:

- "stop_all.vbs", "stop_vol.vbs" and "stop_vm.vbs" scripts are purposed to temporarily stopping running virtual machines (i.e. turning to the "saved" state). These scripts are intended to be used in the "Execute before snapshot is taken" option in the "Run during backup options" section of the Drive Backup settings. Scripts differ from each other by criteria applied for selecting virtual machines to be stopped.
- "resume_all.vbs", "resume_vm.vbs" and "resume_in.vbs" scripts are purposed for resuming temporarily stopped virtual machines (i.e. restoring from the "saved" state). These scripts are intended to be used in the "Execute after snapshot is taken" option in the "Run during backup options" section of the Drive Backup settings. Scripts differ from each other by criteria applied for selecting virtual machines to be resumed.
- The "vm_layout.vbs" script is not intended to be used in the Drive Backup settings. The script allows to inspect virtual machine settings and layout. It is useful to determine correct parameters to be passed to scripts of two first groups.

3.5.2 Description of Function

stop_all.vbs

Routine:

The script scans all virtual machines connected to the Virtual Server and selects virtual machines being in active long-term states (*Running*, *Paused*) or in transient states that come to these ones (*TurningOn*, *Restoring*); these machines are almost synchronously ordered to switch to the *Saved* state. Then, the script waits until all virtual machines come to one of inactive long-term state (*TurnedOff*, *Saved*, *Invalid*). The wait-state duration is limited (5 minutes by default); this value can be changed by a user.

No analysis of virtual machines layout is performed.

Recommended usage:

The script is intended to be used in the "Execute before snapshot is taken" option in the "Run during backup options" section of the Drive Backup settings. It can be useful in case all virtual machines are connected to a same virtual network and their workability strongly depend on each other, or in case a user is not sure of virtual machine independency.

Examples of use:

```
cscript stop_all.vbs
cscript /nologo stop_all.vbs > out_stop.txt
```

stop_vm.vbs

Routine:

The script takes the list of virtual machines to be stopped from the command line arguments. It scans virtual machines from the list and selects only ones being in active long-term states (*Running*, *Paused*) or in transient states that come to these ones (*TurningOn*, *Restoring*); these machines are synchronously ordered to switch to the *Saved* state. Then, the script waits until all listed virtual machines come to one of inactive long-term state

(*TurnedOff*, *Saved*, *Invalid*). The wait-state duration is limited (5 minutes by default); this value can be changed by a user.

No analysis of virtual machines layout is performed.

Recommended usage:

The script is intended to be used in the "Execute before snapshot is taken" option in the "Run during backup options" section of the Drive Backup settings. It can be useful in case virtual machine dependencies are known and it is known, which virtual machines should be stopped in order to create a valid backup of virtual machines of unterest.

Examples of use:

```
cscript stop_vm.vbs VM_Exchange_Server VM_DomainController
cscript /nologo stop_vm.vbs VM_LAMP_Server
cscript stop_vm.vbs VM_PDC VM_DC > out_stop.txt
```

stop_vol.vbs

Routine:

The script takes the list of volumes being backed up from the command line arguments. It scans all virtual machines connected to the Virtual Server and selects only virtual machines, which are fully or partially located on these volumes. Those of selected virtual machines being in active long-term states (*Running*, *Paused*) or in transient states that come to these ones (*TurningOn*, *Restoring*) are synchronously ordered to switch to the *Saved* state. Then, the script waits until all listed virtual machines come to one of inactive long-term state (*TurnedOff*, *Saved*, *Invalid*). The wait-state duration is limited (5 minutes by default); this value can be changed by a user.

Recommended usage:

The script is intended to be used in the "Execute before snapshot is taken" option in the "Run during backup options" section of the Drive Backup settings. It can be useful in case it is required to stop only those virtual machines that can be successfully backed up together with the given set of volumes; all other virtual machines are leaving intact.

Examples of use:

```
cscript stop_vol.vbs C: F:
cscript stop_vol.vbs %VOLUME_NAME_LIST%
cscript /nologo stop_vol.vbs D: E: F: > out_stop.txt
```

resume_all.vbs

Routine:

The script scans all virtual machines connected to the Virtual Server and selects virtual machines being in *Saved* state; these machines are ordered to be restored (going to the *TurningOn* state). By default, the script quits immediately after initiating the virtual machines restoration.

Optionally it can wait until the restoration process completes. The wait-state duration is limited (5 minutes by default); this value can be changed by a user. This feature allows to reduce the disk subsystem loading induced by starting the backup process together with multiple bulk disk I/O operations. However, it may increase the blackout period of virtual machines by several seconds.

The script does not perform analysis of what (or who) stopped virtual machines. It may resume virtual machines not being stopped by some "`stop_XXX.vbs`" script previously called in this backup session.

Recommended usage:

The script is intended to be used in the "Execute after snapshot is taken" option in the "Run during backup options" section of the Drive Backup settings. It can be useful in conjunction with the "`stop_all.vbs`" script, or in case a user is not sure of virtual machine independency.

Examples of use:

```
cscript resume_all.vbs
```

resume_vm.vbs

Routine:

The script takes the list of virtual machines to be resumed from the command line arguments and selects virtual machines being in *Saved* state; these machines are ordered to be restored (going to the *TurningOn* state). By default, the script quits immediately after initiating the virtual machines restoration.

Optionally it can wait until the restoration process completes. The wait-state duration is limited (5 minutes by default); this value can be changed by a user. This feature allows to reduce the disk subsystem loading induced by starting the backup process together with multiple bulk disk I/O operations. However, it may increase the blackout period of virtual machines by several seconds.

The script does not perform analysis of what (or who) stopped virtual machines. It is the user's responcibility to provide the script with the correct list of virtual machine names.

Recommended usage:

The script is intended to be used in the "Execute after snapshot is taken" option in the "Run during backup options" section of the Drive Backup settings. It can be useful in conjunction with the "`stop_vm.vbs`" script. The user is responsible to provide the correct list of virtual machine names.

Examples of use:

```
cscript resume_vm.vbs VM_Exchange_Server VM_DomainController
cscript /nologo resume_vm.vbs VM_LAMP_Server VM_DC
```

resume_in.vbs

Routine:

The script expects getting the list of virtual machines to be resumed from the console input. The console input is filtered and virtual machines names are extracted from it. Then the script selects virtual machines being in *Saved* state; these machines are ordered to be restored (going to the *TurningOn* state). By default, the script quits immediately after initiating the virtual machines restoration.

Optionally it can wait until the restoration process completes. The wait-state duration is limited (5 minutes by default); this value can be changed by a user. This feature allows to reduce the disk subsystem loading induced by starting the backup process together with multiple bulk disk I/O operations. However, it may increase the blackout period of virtual machines by several seconds.

This script allows to organize a selective resuming of virtual machines being stopped by some "`stop_XXX.vbs`" script previously called in this backup session. The script can filter virtual machine names from console output produced by "`stop_XXX.vbs`" scripts. To organize the selective resuming of virtual machines, a user should redirect the output of a used "`stop_XXX.vbs`" script to a file and then redirect the console input of the "`resume_in.vbs`" from that file (see below).

Recommended usage:

The script is intended to be used in the "Execute after snapshot is taken" option in the "Run during backup options" section of the Drive Backup settings. It can be used in conjunction with any of "`stop_XXX.vbs`" scripts. The user is only responsible to set up correctly filenames in the Drive Backup settings.

The common use is the following:

1. Choose one of the "`stop_XXX.vbs`" scripts that will be used for stopping virtual machines. Let's choose the "`stop_vol.vbs`" script, for example.
2. *Optionally*, ensure that the "`AllowSupplementaryEchoes`" variable is set to the "`True`" value.

For that purpose, open the script in any text editor (e.g. NOTEPAD), and find the following text sowhere in the beginning of the script:

```
Dim AllowSupplementaryEchoes
AllowSupplementaryEchoes=True        (or AllowSupplementaryEchoes=False)
```

3. The "Execute before snapshot is taken" command in the "Run during backup options" section should be set to something like this:

```
cscript stop_vol.vbs D: E: F: > out_stop.txt
```

or

```
cscript /nologo stop_vol.vbs %VOLUME_NAME_LIST% > d:\temp\out_stop.txt
```

This command redirects the output of the "`stop_vol.vbs`" script to the "`out_stop.txt`". The filename and its path can be changed. The optional "**`/nologo`**" switch suppresses the script logo text, it only reduces the output file by three lines. The "**`VOLUME_NAME_LIST`**" is the environment variable automatically created by Drive Backup. It includes the list of volume names, which are being backup up by the current backup session. In the last example, contents of the "**`VOLUME_NAME_LIST`**" environment variable are passed as the command-line arguments to the script, so that the script selects the set of volumes that is strictly equal to the volumes being backed up.

4. The "Execute after snapshot is taken" command in the "Run during backup options" section should be set to something like this:

```
cscript resume_in.vbs < out_stop.txt
```

This command redirects the console input to the "`resume_in.vbs`" script from the "`out_stop.txt`" (as if its contents are entered to the script from the keyboard). The file name and its path should exactly correspond the file assigned for storing console output in the "Execute before snapshot is taken" command.

`vm_layout.vbs`

It the informational script not purposed to be used in Drive Backup settings. The script displays the virtual machine state and its constituent files: configuration and dump files, virtual hard disk files including parental virtual disks for differencing virtual disks as well as virtual CD/DVD and floppy images. It optionally estimates relation between the given set of volumes and the set of constituent files for every virtual machine.

Routine:

The script scans all virtual machines connected to the Virtual Server and displays files of the host computer that constitute each virtual machine. It includes the configuration file, the "saved state" file, primary and undo files for every virtual hard disk, files of parental virtual disks for differencing virtual disks and optionally image files of virtual floppy and virtual CD/DVD drives.

In case a list of volumes was passed in the command-line arguments, the script also estimates how the set of constituent files relates with the given set of volumes – is it fully covered, partially covered or not intersects with set of volumes. This information helps to make a correct decision about the set volumes to be backed up and about the list of virtual machines to be stopped.

Usage:

The script is not intended to be used in the Drive Backup settings. It can be used as a standalone script and should be run from the command line in the following fashion:

```
cscript vm_layout.vbs
```

3.5.3 Fine Tuning

The described end-user VBS scripts include the common set of internal parameters, which allow fine tuning of scripts behavior. These parameters control features that may need to be different in various instances of Virtual Server but are very rarely changed between events of use of scripts for every single instance. For this reason these parameters are defined in internal variables within scripts, so that a user may need to edit scripts source files once and avoid defining multiple values every time a script is used.

These parameters: allow/disable a verbose output mode of scripts, define the maximum value of wait-state when a script awaits virtual machines to come into *Saved* or into *Running* state, define criteria of selecting virtual machines to stop and so on.

`AllowSupplementaryEchoes`

Description:

Enables or disables the verbose output mode for scripts. In the silent mode, all scripts produce no output. The only exception is the "`vm_layout.vbs`", which in non-verbose mode suppresses information about constituent files of a virtual machine.

In the verbose mode, scripts display information about the selected set of virtual machines and given set of volumes, and display the list of virtual machines affected by the script; this list can be used by the "`resume_in.vbs`" script for selective resuming of virtual machines.

Note: the "`resume_in.vbs`" script will work correctly only in case an associated "`stop_XXX.vbs`" is run in the verbose mode.

Values:

`True` – the verbose mode is enabled. It is the default value.
`False` – the verbose mode is disabled.

Used in:

All scripts.

`MaxIterations`

Implicitly defines the wait-state period (in seconds) when a script awaits for virtual machines to come into *Saved* state (for "`stop_XXX.vbs`" scripts) or into *Running* state (for "`resume_XXX.vbs`" scripts). This period is measured in iterations of looking through the list of selected virtual machines, and iterations are repeated every second.

Values:

Any positive integer value. The default value is 300, which corresponds to 5 minutes for switching off or resuming virtual machines.

Commonly it is required only few seconds to stop virtual machines.

Used in:

All "`stop_XXX.vbs`" and "`resume_XXX.vbs`" scripts.

Not used in the "`vm_layout.vbs`" script.

`RequireWaitResume`

Enables or disables the "`resume_XXX.vbs`" scripts to wait while resumed virtual machines come to the *Running* state. As it has been mentioned above, it is not obligatory required for the Drive Backup operation to wait while virtual machines complete the resume process, however it may reduce the chance fo the disk susbsystem overload and increase the overall system stability in the beginning of the backup process.

Values:

`True` – the wait mode is enabled. It is the default value.
`False` – the wait mode is disabled.

Used in:

All "`resume_XXX.vbs`" scripts.

Not used in the "`stop_XXX.vbs`" script.

`NeedControlVirtualDVD`

Enables or disables to take into account image files of virtual CD/DVD drives in virtual machines. When enabled, the script includes image files of virtual CD/DVD drives to the set of files that constitute a virtual machine; otherwise it is not true. This feature can affect on the kind of relation between the set of files and the given set of volumes and thus may be a definitive factor for the script whether to stop this virtual machine or not.

CD and DVD are removable media and operating systems usually can correctly manage sudden inaccessibility or disconnection of them. However is not true when a critical application or an operating system itself is run from DVD. The script is unable to estimate the importance of the DVD connection to the virtual machine normal operation. A user should decide it by self.

Values:

`True` – virtual CD/DVD images are treated as important files. It is the default value.
`False` – virtual CD/DVD images are ignored.

Used in:

Only "`stop_vol.vbs`" and "`vm_layout.vbs`" scripts.

`NeedControlVirtualFloppy`

Enables or disables to take into account image files of virtual floppy in virtual machines. When enabled, the script includes image files of virtual floppy to the set of files that constitute a virtual machine; otherwise it is not true. This feature can affect on the kind of relation between the set of files and the given set of volumes and thus may be a definitive factor for the script whether to stop this virtual machine or not.

Floppy discs are removable media and operating systems usually can correctly manage sudden inaccessibility or disconnection of them. However is not true when a critical application or an operating system itself is run from floppy. The script is unable to estimate the importance of the floppy connection to the virtual machine normal operation. A user should decide it by self.

Values:

`True` – virtual floppy images are treated as important files. It is the default value.
`False` – virtual floppy images are ignored.

Used in:

Only "`stop_vol.vbs`" and "`vm_layout.vbs`" scripts.

`UseValidBackupCriterion`

Is used for choosing the virtual machines selection criterion based on the relation of constituent files and the given set of volumes. It limits (or not limits) to select only virtual machines which are completely fit in the given volume set. This feature is the definitive factor for the script whether to stop this virtual machine or not.

Note: if constituent files of the virtual machine were completely fit in the set of volumes being backed up, the virtual machine is backed up correctly. It can be successfully restored from the backup image and restarted.

Instead, if constituent files of the virtual machine were partially covered by the set of volumes, the virtual machine is backed up incorrectly. It may be restored in inconsistent state. However, there are cases when this inconsistency does not affect on virtual machine operation. The script is unable to estimate its correctness. A user should decide it by self.

Values:

`True` – only virtual machines fully covered by the given volume set will be stopped. It is the default value.
`False` – virtual machines fully or partially covered by the given volume set will be stopped.

Used in:

Only "`stop_vol.vbs`" and "`vm_layout.vbs`" scripts.

Warning:

The listed above variables are purposed for change for customizing these scripts for better compliance to user's needs. Other variables as well as source code are not intended for any changes.

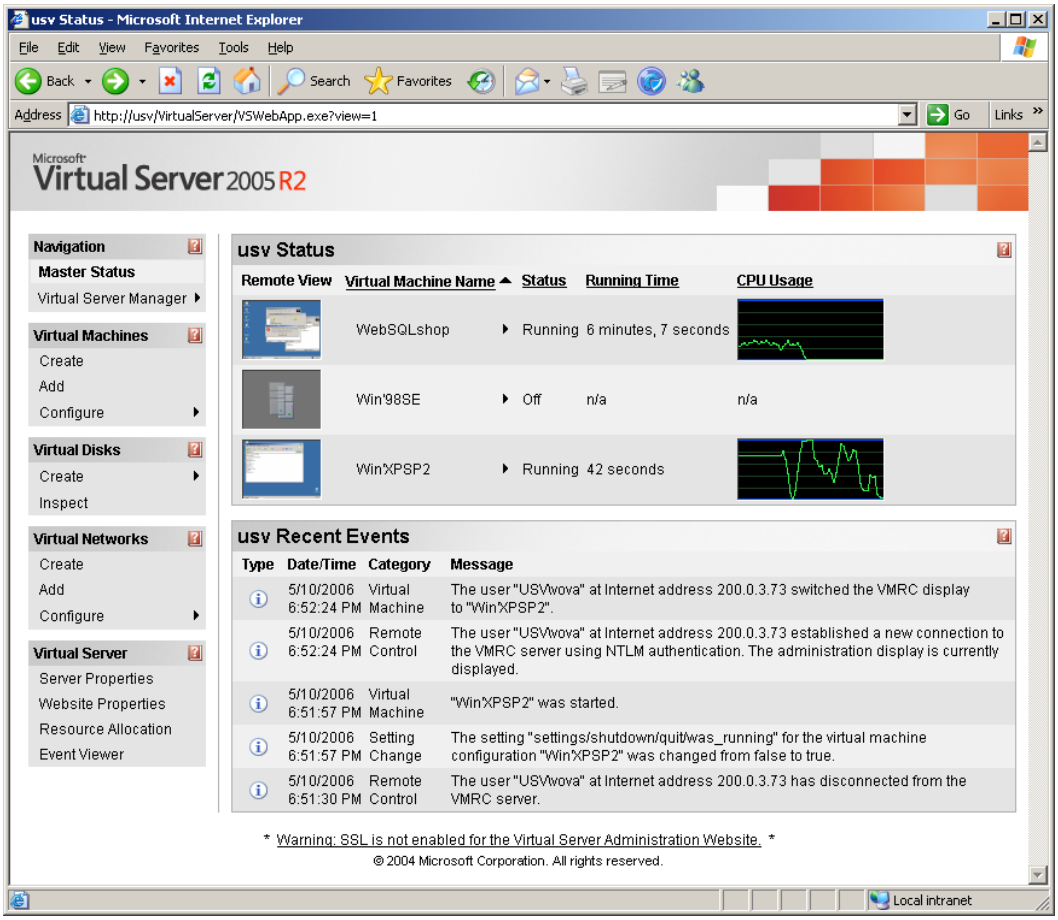
3.6 How to Use DBE for Protecting Virtual Server Data

3.6.1 How to Redistribute Virtual Machine Files

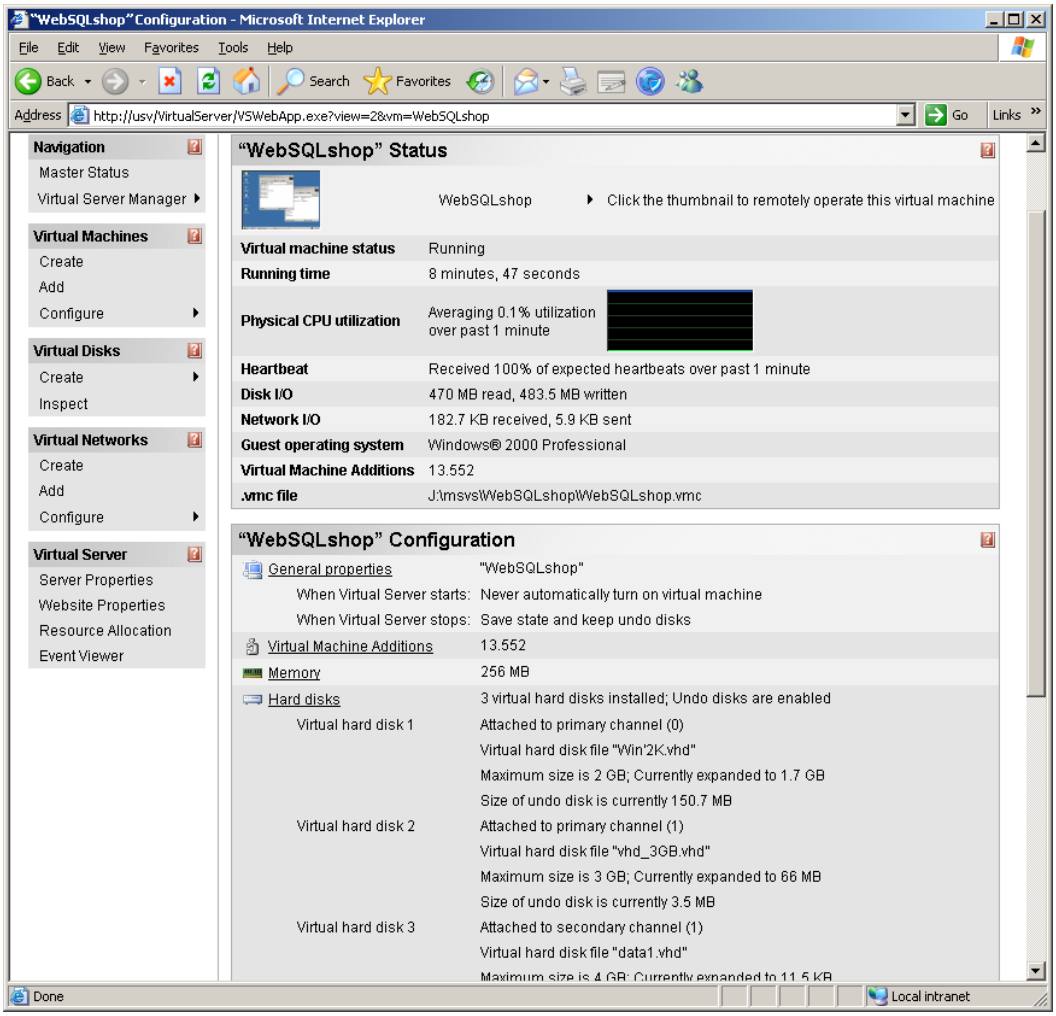
This section demonstrates how to change location of virtual machine files. Such procedure can be applied to a turned off virtual machine for getting better manageability.

Example 1: Moving virtual hard disk file to another location.

1. Run Virtual Server Administration website
   (Start→Programs→Microsoft Virtual Server→Virtual Server Administration Website)



2. Turn off the virtual machine of interest.
3. Commit Undo Disks for all virtual disks of this virtual machine.
4. Open virtual machine configuration page.



5. Remove the virtual hard disk of interest from virtual machine's hardware list.
6. Run Windows Explorer.
7. Move the appropriate .VHD-file to another location.
8. Open virtual machine configuration page.
9. Add the virtual hard disk to the virtual machine's hardware list by defining its new location.

3.6.2 How to Backup Virtual Machines

As it was mentioned above, Drive Backup allows only to backup volumes and is unaware of virtual machine structure. The functions of virtual machine management and structure recognition are completely moved to add-on scripts. This section demonstrates how to correctly use these scripts with Drive Backup.

Conditions: There is the "WebSQLshop" virtual machine located on [J:] and [X:] volumes. The virtual machine is running.

Purpose: Set up correct parameters for the task of the "WebSQLshop" virtual machine backup. Then create an automated scheduled task that will backup the "WebSQLshop" virtual machine.

Stage 1: Gathering information:

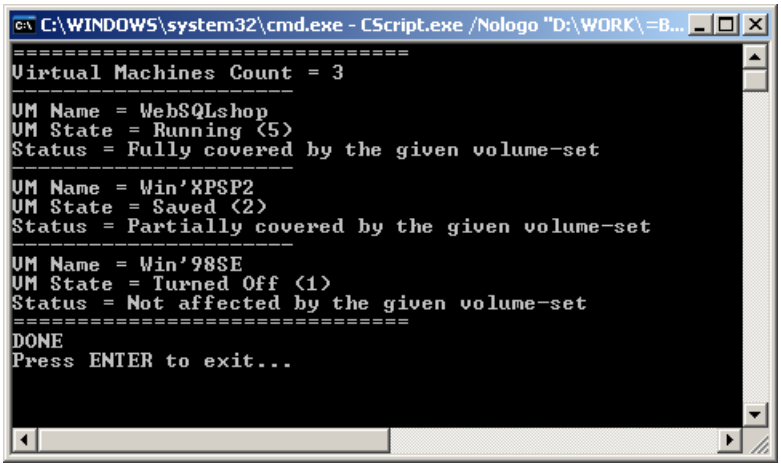
1. Run the "**vm_layout.vbs**" script from the command line:
   - Start→Run…
   - Enter the following command:

     cscript "*<path>***vm_layout.vbs**" *<volumes list>*

     where: *<path>* is the directory where downloaded VBS scripts were saved.
     <volumes list> is an optional set of volumes of interest, e.g. "J: K: X:"

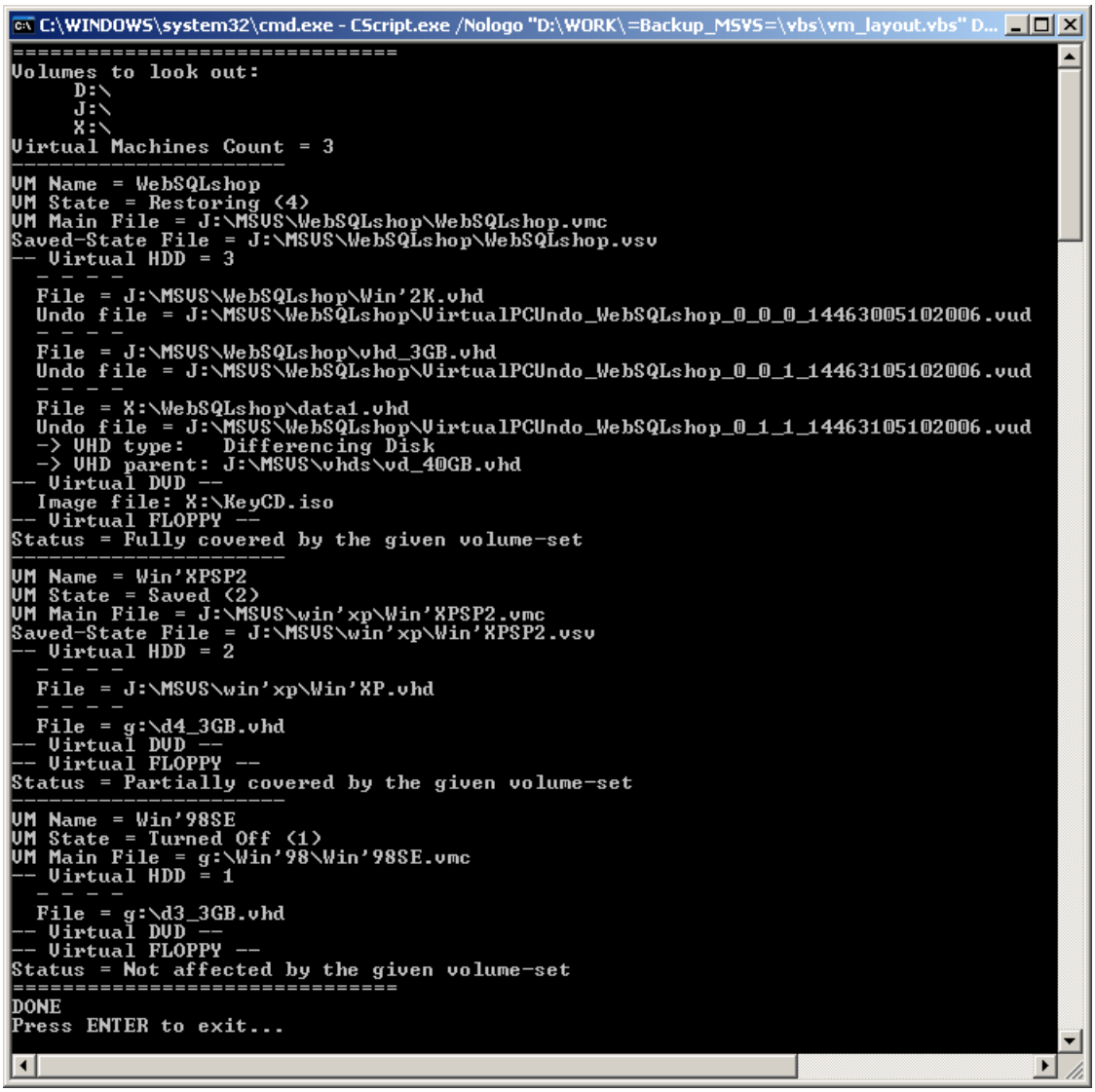
2. The "**vm_layout.vbs**" script scans all virtual machines connected to the virtual Server and displays all files that constitute every virtual machine.

(brief report):



3. In case the non-empty *<volumes list>* parameter was passed, the script also displays the relation between the given set of volumes and the set of constituent files for every virtual machine (partial, full or empty covering of file set).

(extended report):

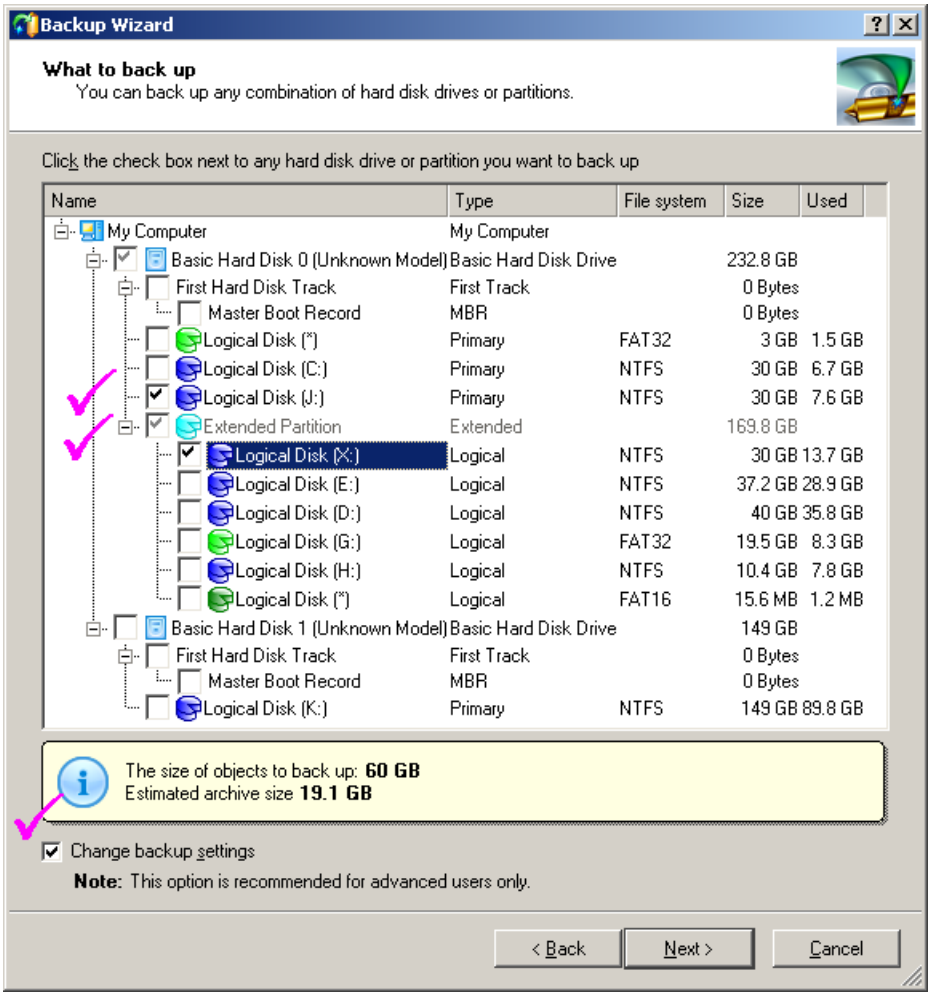


This information can be used for better planning backup task parameters. Remember volumes containing constituent files of the virtual machine of interest.

In this example, the "**`vm_layout.vbs`**" script will return display all files that constitute the "WebSQLshop" virtual machine, also it will signal that the file set is fully covered by the "J: K: X:" volume set. It means that when volumes [J:], [K:] and [X:] are backed up together, the "WebSQLshop" virtual machine is backed up correctly as well.

Stage 2: Setting up backup properties:

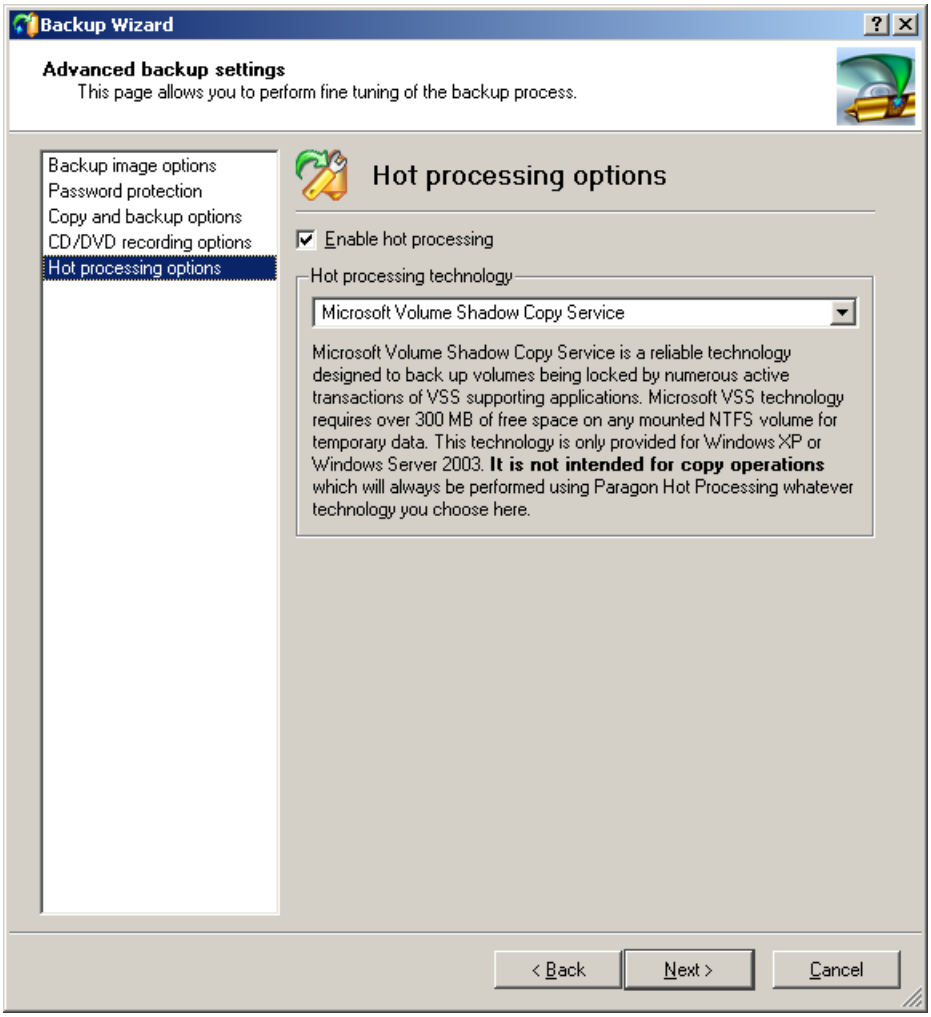
4. Run Drive Backup and invoke the Backup Wizard.
5. In the tree view of disks and volumes, select all volumes that were selected in the Stage 1.



6. *Important!* There is the "Change backup settings" checkbox on the bottom of the screen. Set this checkmark in order to control the backup settings for that task. Then press "Next" button.

In this example, the whole "WebSQLshop" virtual machine is located on the [E:] and [K:] volumes. So that at least these two volumes must be selected in the tree view.

7. On the "Backup Settings" screen, set the backup parameters. For the absolute most of them default values are optimal. We will change only two of them in this example:
8. Go to the "Backup image options" page and set the "Normal" compression level. It usually provides good compression rate (3:1 – 6:1) with harmless speed deceleration.
9. Go to the "Hot processing options" page and select the "Microsoft Volume Shadow Copy Service" item in the "Hot processing technology" pull-down list.



10. Go to the "Run during backup options" page and set parameters as follows:
    - Set the "Execute before taking snapshot" parameter to:

      cscript *<path>***\stop_all.vbs**

      or

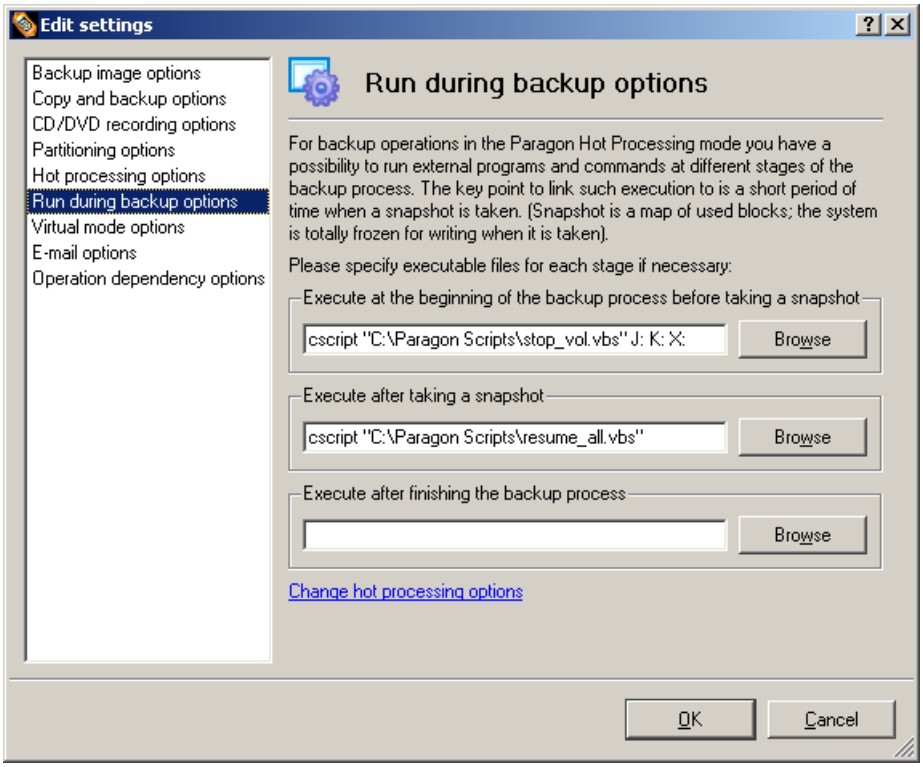
      cscript *<path>***\stop_vol.vbs J: K: X:**

      or

      cscript *<path>***\stop_vol.vbs %VOLUME_NAME_LIST%**

    - Set the "Execute after taking snapshot" parameter to:

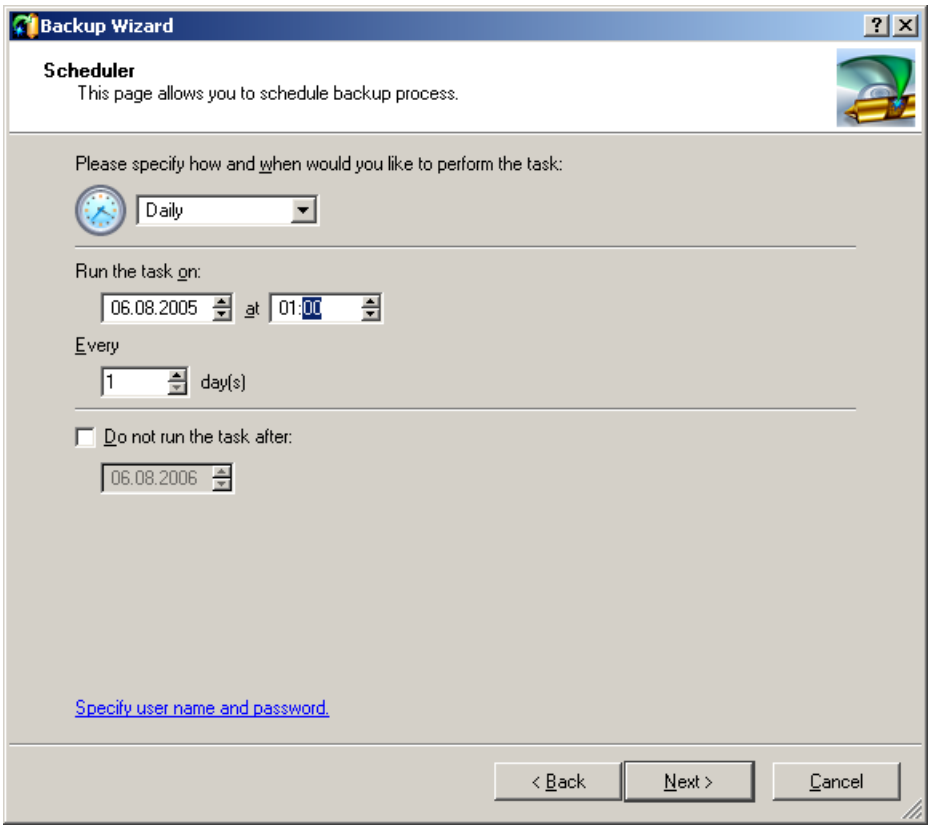
      cscript *<path>***\resume_all.vbs**



11. On the two next screens, choose the "Save data on the local/network disk" and select location for storing backup images.

Stage 3: Creating the scheduled task:

12. On the next screen, choose "schedule backup" to create a scheduled backup task.
13. Customize the schedule parameters.



At this moment, the program generates a PSL-script file containing appropriate commands for back up selected volumes and stores it in the "Scripts" subdirectory of the Drive Backup's installation directory. The PSL-script is written in Paragon Scripting Language intended for executing by the so-called Paragon Script Interpreter utility (SCRIPTS.EXE – for Windows environment) included to the Drive Backup distribution package.

Notes:

- The VBS-scripts mentioned above require the Windows Management Instrumentation service (WMI) to run. WMI service should be configured to run automatically and should be enabled each time these scripts run.
- The standard Scheduled Tasks service is used to run the Paragon Script Interpreter. Scheduled Tasks should be enabled in order to execute the automated backup task.
- For smoothly running a scheduled backup in a system where no one has been logged on, Drive Backup creates the special local user account, which is a member of the local Administrators group. This account should be created and enabled to log on locally otherwise scheduled backup tasks may fail to run.

The automatically generated script always uses the same filename(s) for storing a backup image. In other words, every time a scheduled backup task is run, it overwrites the previously created backup image. Either you must take care of regularly saving successfully created backup images or you should enhance the script.

Paragon Scripting Language allows to program sophisticated scripts with smart behavior. As concerns to this example, the so-called "Cyclic Backup" script is more convenient for this purpose. The "Cyclic Backup" script uses a limited-length stack of last backup images. As soon as the new image is created, the oldest image in the stack is deleted. The script guarantees availability of the last good backup images and implicitly controls the total capacity of backup images stored on the disk.

The "Cyclic Backup" script is available for authorized Drive Backup Enterprise customers from the Technical Support Website.

3.6.3 How to Restore Virtual Machines

This section demonstrates how to correctly restore a virtual machine from a previously created backup image.

Conditions: The "WebSQLshop" virtual machine was located on [E:] and [K:] volumes. It was backed up by using the MS VSS online backup option. Later, the virtual machine was corrupted by malware and now works incorrectly. The host operating system, Virtual Server are still working properly. The virtual machine is running.

Purpose: Restore the virtual machine from the full backup image and bring it to a workable state.

Stage 1: Disconnect the "WebSQLshop" virtual machine:

1. Run Virtual Server Administration website
   (Start→Programs→Microsoft Virtual Server→Virtual Server Administration Website)
2. Turn off the virtual machine of interest.
3. Delete the virtual machine from the Virtual Server.
4. (Optionally) delete the virtual machine's constituent files.

Stage 2: Extract the virtual machine's constituent files from the backup image:

5. Find the last backup image and place it on a local volume or on a shared network resource.
   It is the required step in case the image was stored on removable media.
6. Run Drive Backup and start the Image Explorer tool.
7. Open the backup image in Image Explorer:
   menu: File→Load Archive
8. Browse the image and find the virtual machine's constituent files.
9. Select all these files, invoke the popup menu and select the "Export" item and extract them to their original location. An export of multiple files and directories is supported.

To correctly export a directory from an image:

- Choose a directory in the backup image.
- Invoke the popup menu and select the "Export" item.
- In the "Select directory" dialog, choose the directory that is the parent for the exported one.

In case the virtual machine was the only object on backed up volumes, simply perform the volume-level restore of these volumes.

Stage 3: Connect the restored virtual machine back to the Virtual Server:

1. Run Virtual Server Administration website
   (Start→Programs→Microsoft Virtual Server→Virtual Server Administration Website)
2. Add the virtual machine to the Virtual Server:
   - Virtual machines→Add
   - Type in the fully qualified path and filename for the virtual machine's configuration file (in this example, it will be the "WebSQLshop.VMC" file).
3. Co to the virtual machine configuration page and inspect machine's status.
4. In case there are no errors displayed, resume the machine.

Note:

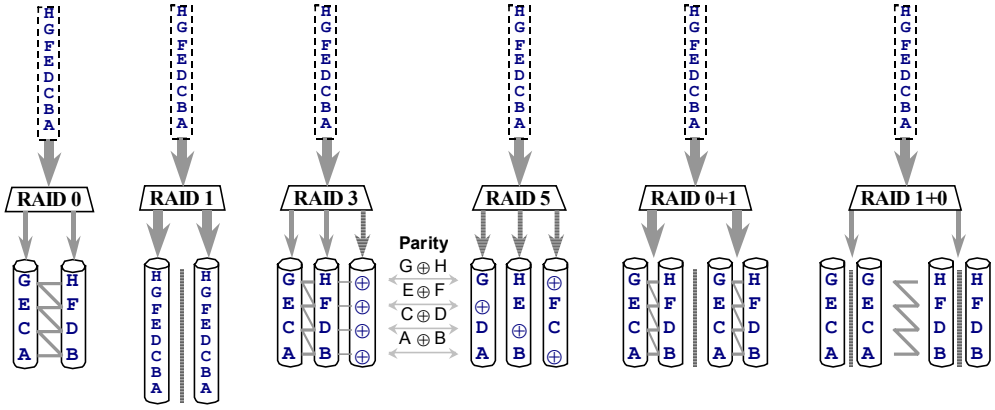
- If a virtual machine was in "turned off", "turning off" or "merging disks" states at the moment of the backup creation, it appears as "turned off" after the restoration.
- If a virtual machine was in other states (e.g. "running", "paused", "turning on" etc) at the moment of the backup creation, it appears as "saved state" after the restoration.

4 Appendix

4.1 RAID levels

Use of RAIDs is the most effective way to increase throughput and provide a hardware-level fault tolerance of a disk subsystem. Striped RAID sets provide highest read/write performance, which is found nearly a multiple of a single disk performance.

RAID level	Brief description	Amount of disks	Size factor	Performance factor	Fault tolerance
0	Striped Disk Array	N ≥ 2	× N	× N	No
1	Mirrored Disk Array	M ≥ 2	× 1	× 1	Yes
0+1	Mirroring of striped segments (RAID 1 over RAID 0)	M•N ≥ 4 M ≥ 2, N ≥ 2	× N	× N	Yes
3	Striped Disk Array with Isolated Parity	N+1 ≥ 3	× N	× N (for reads) × 1 (for writes)	Yes
5	Striped Disk Array with Distributed Parity	N+1 ≥ 3	× N	× N (for reads) × 1 (for writes)	Yes
10	Striping of mirrored segments (RAID 0 over RAID 1)	N•M ≥ 4 N ≥ 2, M ≥ 2	× N	× N	Yes



A more detailed characteristic of various RAID levels can be found in the Internet.