APPLICATION NOTE

Using Shared Virtual Array® (SVA™) disk systems and SnapShot software with NetBackup for Windows 2000

APRIL 2004

Using Shared Virtual Array® (SVA™) disk systems and SnapShot software with NetBackup for Windows 2000

HARDWARE AND SOFTWARE APPLICATIONS USED

	Name	Vendor	Comments
ISV	NetBackup	VERITAS	Tested with VERITAS NetBackup Data Center 4.5, should work with earlier versions
Operation System	Windows 2000– SP 2	Microsoft	
SVA Software	SVAA 3.1.0 patch 10b (W2K) SVAC 3.1.0 patch 8 (W2K) SVA PATH patch 10b (W2K)	StorageTek	 Optional* Optional**
SVA Hardware	SVA 9200, 9500 and V960	StorageTek	Tested on SVA 9200
Backup Software	NetBackup Data Center 4.5	VERITAS	
Backup Hardware	Quantum DLT drive DLT tape	Quantum	

* SVAC is the graphical version of StorageTek® Shared Virtual Array® Administrator (SVAA) software.

** SVA Path is the failover product for SVA

1 ABSTRACT

StorageTek® Shared Virtual Array® (SVA™) has a feature called StorageTek® SnapShot software, which allows making instantaneous copies of any single virtual disk defined within the disk subsystem. In order to be able to take advantage of all the benefits of this feature, it is necessary to integrate the SnapShot operational steps with the common Relational Database Management System (RDBMS) used on the host systems. An RDBMS is a program that lets you create, update and administer a relational database.

Today the Open Systems market is leading the disk storage market and all the integration efforts remain to be done in this arena. This paper intends to present a simple integration of SnapShot with the product VERITAS NetBackup software. The tests performed here are based on a Windows environment. But, you can refer to the application note "Using SVA SnapShot with NetBackup" (June 2001) for more information about the same scenario, but based on a Solaris platform.

We assume the reader has a good knowledge of Windows disk administration, VERITAS NetBackup Software, SVA principles and SVA administration.

The last part of this document describes the different tests of backup with SVA and NetBackup.

All the scripts, prerequisites, constraints are also detailed in each test.

2 UNDERSTANDING SNAPSHOT

SnapShot software is a powerful data duplication product that harnesses the unique virtual disk architecture of the StorageTek SVA. SnapShot duplicates entire Open Systems Logical Units (LUNs) in seconds, using zero additional physical capacity.

SnapShot software's virtually instantaneous copying speed is accomplished by manipulating pointers to data within the virtual disk subsystem, instead of physically moving it. With no physical data movement required, SnapShot software creates copies of LUNs in approximately 2–5 seconds, no matter how much data is copied. SnapShot software also has the unique ability to create any number of copies needed or desired by the user. By eliminating the use of resources such as Central Processing Unit (CPU) cycles, channel utilization, I/O operations, and most importantly, time, SnapShot software creates a new paradigm in data duplication. Traditional data duplication products require physical data movement, which is expensive in terms of resources used and precious time.

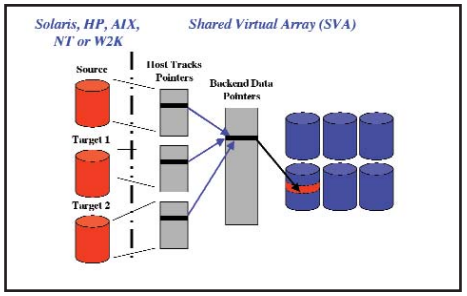


Figure 1. Snapshot mechanism.

How does it work?

SnapShot software simply updates pointers within the virtual disk subsystem mapping tables for the data views being duplicated at electronic memory speeds. This updated pointer references to the same disk array locations as the original source data—one physical copy of the data with multiple host "views" (**Figure 1.**).

SnapShot copies use no disk storage initially since only one physical copy of the data exists at the point in time when the snap is created. All copies and the original copy are totally independent, and can be updated changed or read without affecting each other. Only changes to either the source data or copies use additional physical storage, and any parts of the data that remain common are shared. The result of a SnapShot copy is the same as that of a traditional copy, the main difference is that the time required to create the copy with SnapShot software is measured in mere seconds versus traditional physical duplication products measured in minutes or hours.

3 ARCHITECTURE EXAMPLE

The aim of this section is to present the test environment for the SVA in order to use the SnapShot feature with VERITAS NetBackup software. Two main architectures are considered:

> Using a simple SAN architecture with only one server that is the production and the backup server
> Using an additional backup server to perform a so called "production-server-free backup."

3.1 SINGLE SERVER BACKUP

The figure below depicts the simple backup configuration. In this first configuration, the production server and the backup server are configured on the same machine. The aim of this section is to present an online/offline backup in a split mirror scenario with no additional backup server.

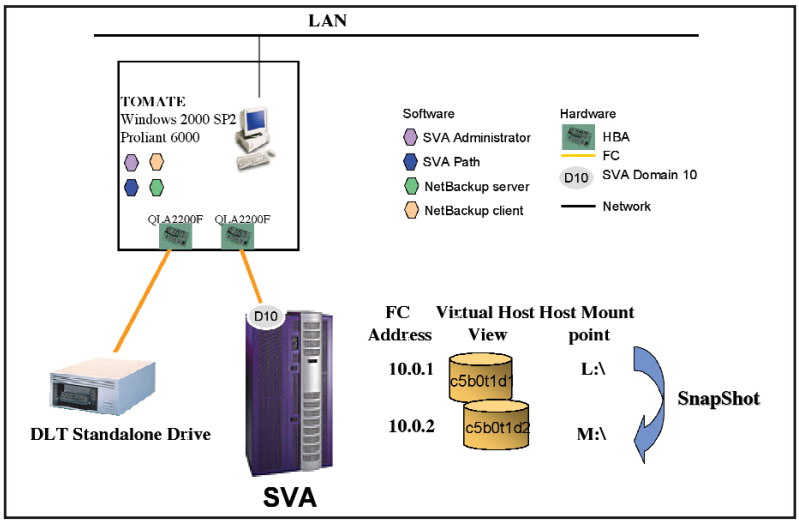


Figure 2. Single production/backup server configuration.

The production server (W2K TOMATE) uses one LUN of the SVA to store data files. Another LUN is used to get the SnapShot copy of the original data files. The server uses a Fibre Channel connection (QLogic QLA2200F) to communicate with the SVA subsystem on the domain 10. It is configured as a NetBackup client.

The same server acts as the backup server as well. It is connected to the tape drive via a Fibre Channel connection (QLogic QLA2200F). So, it is configured as the NetBackup server.

The tape drive used is a stand-alone one using a DLT tape.

3.2 PRODUCTION SERVERLESS BACKUP

The aim of this section is to present the second test environment for the SVA and the NetBackup software.

The figure below illustrates the basic architecture that intends to reproduce a typical schema with data production server and backup server:

- The production server (DELL NESTE) uses one LUN of the SVA to store the data and another one to receive the SnapShot. A Fibre Channel connection is used to connect the SUN to the SVA subsystem on domain 10 (Emulex LP8000).
- The backup server (W2K TOMATE) has a Fibre Channel connection to the SVA. Another HBA is used to make a connection to the stand-alone tape drive (via a SCSI bridge 3100). The backup server is configured as a NetBackup master server.
- Both servers are on the same LAN.

Note **All the software applications required to manage the SVA are installed on both servers.**

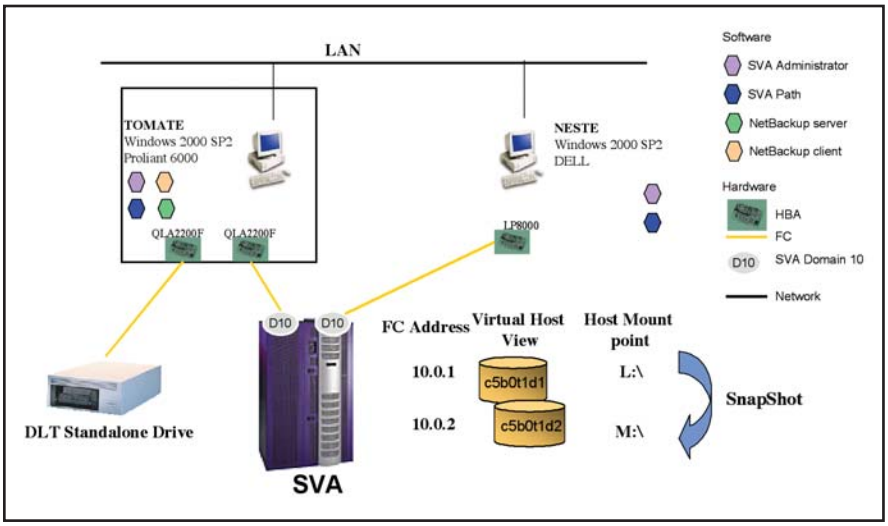


Figure 3. Dedicated backup server configuration.

4 CONFIGURATION STEPS

4.1 SVA CONFIGURATION

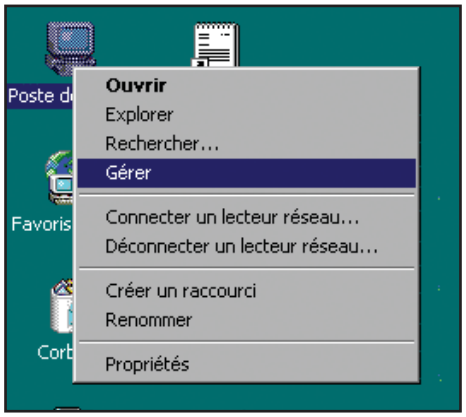
On the client server we must configure two devices on SVA, one for the customer data and one for SnapShot.

Here we suppose that the target is always defined on the SVA, but this device doesn't use physical capacity of the SVA. Before the backup, a release command is executed in order to clean the device before the SnapShot. After the backup, it is also possible to perform a release command on this to remove the capacity for data update after the SnapShot. Otherwise, the "image" of the data is stored on the target LUNs, just in case.

The SCSI partition release facility informs an SVA subsystem that a given SCSI partition no longer contains any useful data so the subsystem can release the space the partition is using. Thus, partition release enables you to take better advantage of an SVA's extended capacity to achieve lower overall storage costs and, at the same time, to securely erase unwanted data.

4.1.1 Make sure the SVA is correctly configured

Use the manager console to display the current general configuration.



Note **The different LUNs that will be defined must be configured from the production server. The file systems for the original data are always mounted on the production server.**

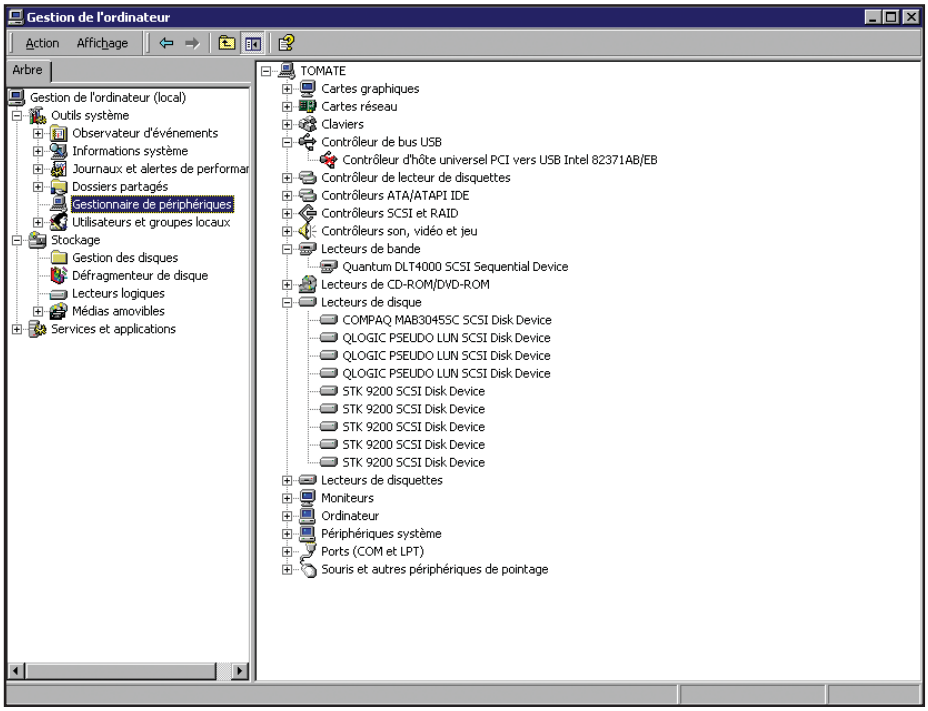


Figure 4. Device Manager Interface.

All the SVA LUNs available on the domain the server is connected to are displayed if everything is properly configured.

4.1.2 Define one functional SCSI LUN device with SVAA

> Declaration of SCSI LUN

```
SIB> defdev -subsys ECLIPSE -fdid 102 -scsiaddr 10.0.0 -name ECAM_NBU -fcapa 1g -dev-
typ SCSI
SIB9830I: Device (FDID 302) successfully defined with an exact size of 2.5 GB.
```

You must specify the first LUN as an ECAM device to be able to connect to the subsystem from the server used.

> Server and subsystem information

```
# sibadmin
Starting CLI...
SIB> dserver
SVAA Server: TOMATE           Date/Time: 07-09-2002 13:12:04 GMT+02:00

Version: 3.1
Description:
Trace Status: Disabled
Installed Features:
   SnapShot
Maintenance Level:
   0 PPFinfo PTF=L2P007G Patch=10b FIX=0 Issue=752695
Host:
   Name: TOMATE
   TCP/IP Port: 41248
Host OS Level:
     5
     0
SIB> qsubsys
SVAA Server: TOMATE           Date/Time: 07-09-2002 13:12:11 GMT+02:00

ECLIPSE
```

If there is no subsystem attached to the server, use the addsubsyspath command. For more information on the sibadmin commands, type help or help command.

4.1.3 Define two functional SCSI LUN devices with SVAC

Use the SVA Console to view and create LUNs.

Connect to the server that has a connection with the SVA and that has access to an ECAM device. This device enables the connection to a configured subsystem on the SVA.

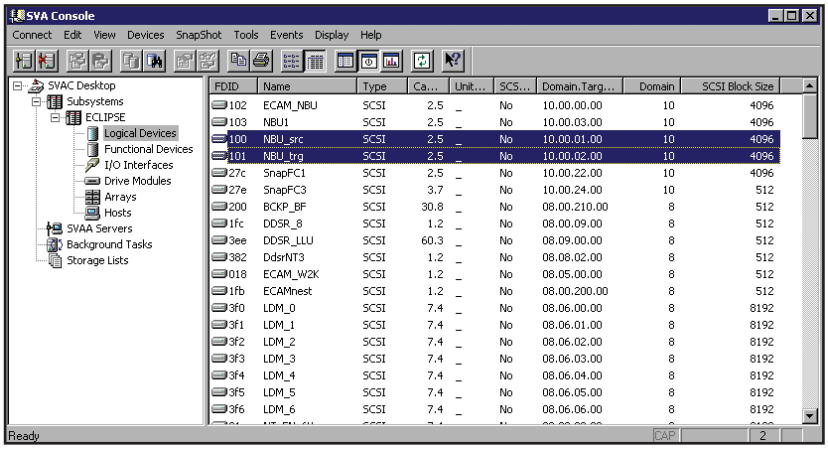


Figure 5. SVA Console Interface (connection with ECLIPSE).

To create a device, go to the menu Devices > Define New…and complete with the correct information.

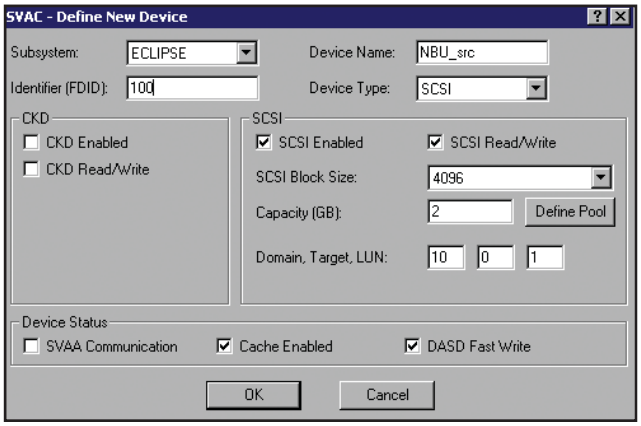


Figure 6. Definition of a New Device Interface.

> Create two new LUNs (one for the source and one for the target):

> Go to menu Devices > Define New …

> Complete with the correct information

> Specify the correct domain, target and LUN numbers

> To choose the identifier of the LUN, use the Define Pool button to display all the available FDID

> Choose OK to instantly create the LUN.

4.1.4 LUNs configuration

> Configure the SVA path

```
D:\>sppath –v
...
SPD=25 c5b0t1d1 dev=0,   5 type=2 SANID="STK 9200 0000000000300100"
SPD=26 c5b0t1d2 dev=0,   6 type=2 SANID="STK 9200 0000000000300101"
SPD=27 c5b0t1d0 dev=0,   4 type=2 SANID="STK 9200 0000000000300102"
```

> Generate kernel configuration files

```
D:\>setsp -g
D:\>setsp -S
D:\>setsp

================================================================================
 spd    Path/disk           Status    Pri Exc Buf Balance RtrCnt  RtrDly FailBack
================================================================================
  25    c5b0t1d1/5       Excluded      X        32     1       20      3000       1
 HardDisk 5                                      ID = "STK 9200 0000000000300100"
================================================================================
  26    c5b0t1d2/6       Excluded      X        32     1       20      3000       1
 HardDisk 6                                      ID = "STK 9200 0000000000300101"
================================================================================
  27    c5b0t1d0/4       Excluded      X        32     1       20      3000       1
 HardDisk 4    I:                                ID = "STK 9200 0000000000300102"
================================================================================
```

> Make LUNs available for the current server

```
D:\>setsp –e0 –l25
D:\>setsp –e0 –l26
D:\>setsp –e0 –l27
```

> Label and format the new devices

- Use the manager console to view the disks attached to the current server.
- Then, choose a basic disk rather than a dynamic one (dynamic disk does not support the SnapShot operation except using specific scripts detailed in the white paper named SnapShot with Windows 2000 Dynamic Disk).
- Create a partition using the wizard tool in the manager console (choose a label or use the quick format option).
- Choose the drive letter on which the device will be mounted (L: for the source LUN and M: for the target one).

Note **The target device is also mounted in order to get information on the physical disk name on which the system mounts the LUN. Use the command stkassign info M: to display the target disk name.**

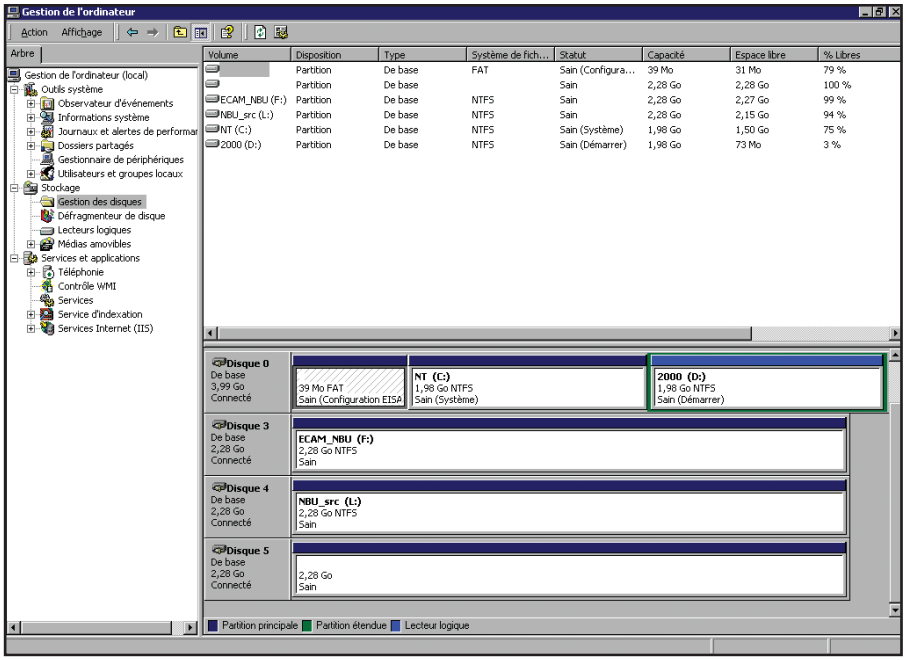


Figure 7. Manager Console Interface—Disk Management.

4.2 NETBACKUP CONFIGURATION

Use the VERITAS configuration tools to configure all the different volumes, the catalog backups, backup policies and storage devices. Go to Start > Programs > VERITAS NetBackup > NetBackup Administration Console.

Figure 8. The NetBackup Administration Console Interface.

The NetBackup master server is installed on the backup server. The master server is also a client. It is not necessary to install the NetBackup client on the production server because it is the backup server that performs the backup, never the data host.

Configuration Steps

The NetBackup administration console can be configured from the wizard interface or manually. Perform the following:

> Configure the storage devices (using a robot or a stand-alone drive). If the device is connected to the NetBackup server, it will be automatically detected.
> Configure the volumes (media) used.
> Specify the location of the catalog backup (cannot be the same media as the one used to do the data backups, a disk space can be specified).
> Define a storage unit group (a group of storage devices that can be used) and a volume pool (a pool of media available for this volume pool).
> Finally, define a backup policy, with a full backup schedule (to enable the manual backups), the clients to be backed up and the list of files to back up.

Note For the single server configuration, we use M:\ for the target device that will receive the snapped data. So the NBU_policy has to backup M:\. The target LUN cannot be mounted on the same device letter. Use whatever drive is free. For the dedicated backup server architecture, we use L:\ because the original device is not mounted on the backup server. You can use whatever drive is free.

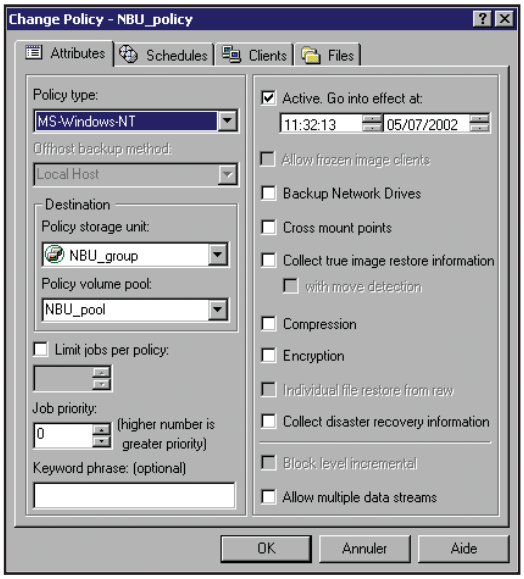


Figure 9. Backup Policy Interface.

Single Server

The backup policy is used to perform a manual backup of M:\ directory on the tomate client.

Additional Backup Server

The backup policy is used to perform a manual backup of L:\ directory on the client.

THE MAIN PROCESSES

The whole backup process is composed of three main processes:

> The release process: release the target LUNs for them to be cleaned for use
> The SnapShot process: snap the source to the target LUNs
> The backup process: backup from the SVA disks to the tapes.

The main script is in **Appendix A: Main Backup Script (No Additional Backup Server)** for the single production and backup server and in **Appendix B: Main Backup Script (Dedicated Backup Server)** for the configuration with the additional backup server.

The principle is the same while using a single production/backup server and using a dedicated backup server. The main difference with the second architecture is that all the operations are launched from the backup server.

All these processes are automated and use several scripts. Each process uses the lunconfig.lst file to get all the needed information about the source and target devices. Please refer to **Appendix C: LUNs Configuration File (No Additional Backup Server) or to Appendix D: LUNs Configuration Files (Dedicated Backup Server)**.

5.1 DEVELOPMENT OVERVIEW

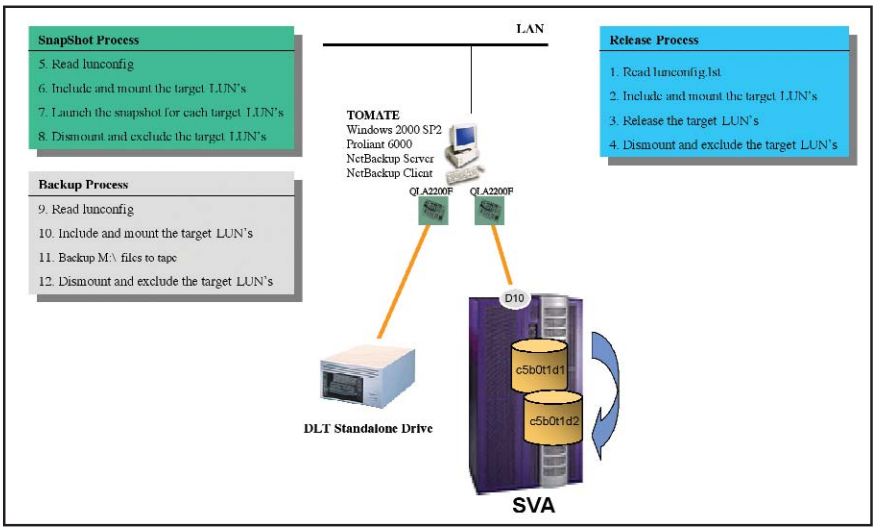


Figure 10. Processes development overview (single server).

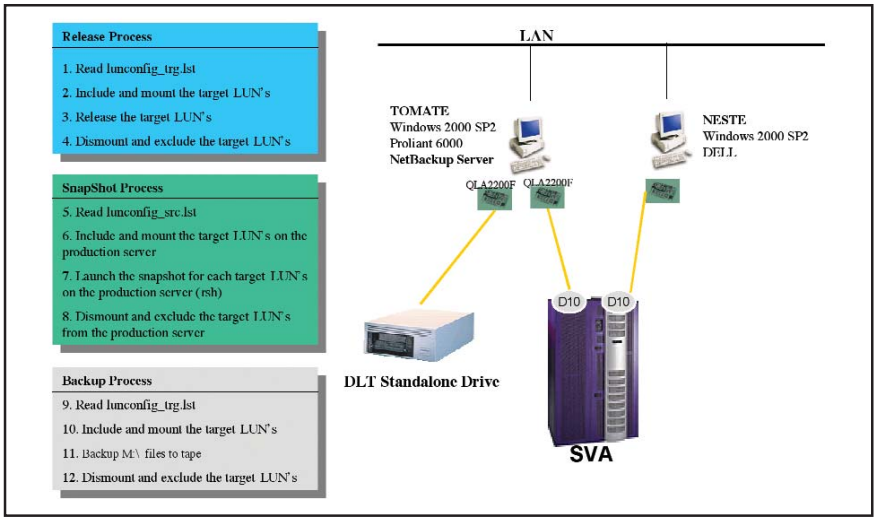


Figure 11. Processes development overview (dedicated backup server).

5.2 BACKUP EXECUTION OVERVIEW

For all the mentioned scripts, please refer to **Appendix C, Appendix D, Appendix E, Appendix F,** and **Appendix G**.

5.2.1 Single server

For the single production/backup server, after all the configuration steps are completed, perform the following:

> Choose a directory where all the custom scripts will be located and initialize the STK_SCRIPTS_HOME environment variable.

> Copy all the scripts for this configuration into the directory you have chosen. In these scripts the environment variable STK_SCRIPTS_HOME is initialized to D:\STK_scripts.

- ***lunconfig.lst:*** file containing the LUNs configuration for SVA Path and Windows 2000 disk partitions.
- ***backup_script.cmd:*** main script that launches all the processes.
- ***release_luns.cmd and release.cmd:*** includes, mounts, releases, dismounts and excludes the target LUNs.
- ***snap_luns.cmd and snap.cmd:*** includes, mounts, snaps, dismounts and excludes the target LUNs.
- ***backup_luns.cmd***: includes, mounts, backs up dismounts and excludes the target LUNs.
- ***mount_luns.cmd and mount.cmd:*** includes and mounts the target LUNs specified in the lunconfig file.
- ***umount_luns and umount.cmd:*** dismounts and excludes the target LUNs listed in the lunconfig file.

> Change these scripts each time an absolute path is given to reflect your own organization.

> Change the lunconfig file to reflect your configuration.

> To perform a manual backup, launch the backup_script.cmd script. Please refer to **Appendix H: Example of Successful Backup (No Additional Backup Server)** for a detailed log of a successful backup.

Note **Do not forget to put the stkassign executable in the directory where all the scripts are copied. This command is used to mount, dismount, synchronize or get information on the disks.**

5.2.2 Dedicated backup server

For the architecture with the additional backup server, after all the configuration steps are completed, perform the following:

> Choose a directory where all the custom scripts will be located and initialize the STK_SCRIPTS_HOME environment variable either on the data host or on the backup host.

> Copy all the scripts for this configuration into the directory you have chosen. In these scripts the environment variable STK_SCRIPTS_HOME is initialized to D:\STK_scripts.

> On the production server:

- ***lunconfig_src.lst:*** file containing the LUNs configuration for SVA Path and Windows 2000 disk partitions to perform the SnapShot locally.
- ***msnap_luns.cmd and snap.cmd:*** includes, mounts, snaps, dismounts and excludes the target LUNs.

> On the backup server:

- ***lunconfig_trg.lst:*** file containing the LUNs configuration for SVA Path and Windows 2000 disk partitions to perform the SnapShot locally.
- ***backup_script.cmd:*** main script that launches all the processes.
- ***release_luns.cmd and release.cmd:*** includes, mounts, releases, dismounts and excludes the target LUNs.
- ***backup_luns.cmd:*** includes, mounts, backs up dismounts and excludes the target LUNs.
- ***mount_luns.cmd and mount.cmd:*** includes and mounts the target LUNs specified in the lunconfig file.
- ***umount_luns and umount.cmd:*** dismounts and excludes the target LUNs listed in the lunconfig file.

> Change these scripts each time an absolute path is given to reflect your own organization.

> Change the lunconfig file to reflect your configuration.

> To perform a manual backup, launch the backup_script.cmd script. Please refer to ***Appendix I: Example of Successful Backup (Dedicated Backup Server)*** for the detailed logs of a successful backup.

Note **Do not forget to put the stkassign executable in the directory where the scripts are duplicated both on the data and the backup hosts. This command is used to mount, dismount, synchronize or get information on the disks.**

5.3 HOW TO RESTORE

The procedure to restore files or directories is almost identical to using either the single server configuration or the dedicated backup server architecture. The main difference resides on the restore destination choice. To restore a file or directory, perform the following:

> Launch the NetBackup Administration Console

> Go to menu File > Backup, Archive, and Restore… The client interface appears.

> Click on the "Select for Restore" icon and choose the files or directories to be restored.

> Go to menu Actions > Start Restore of Marked Files… (**Figure 12.**)

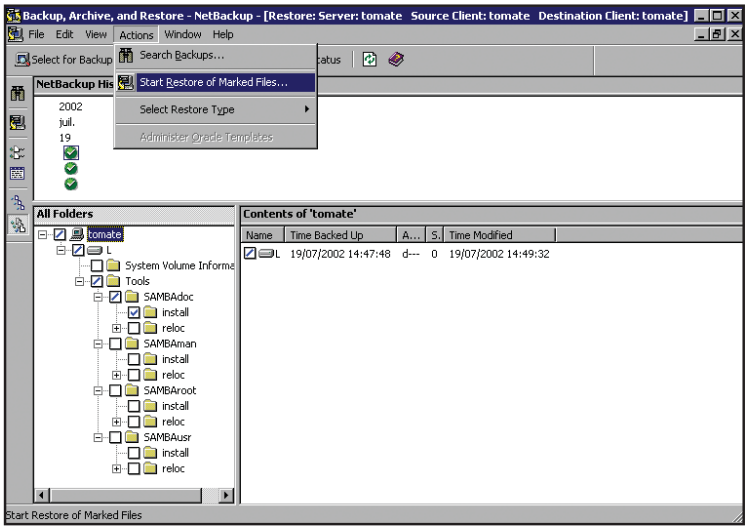


Figure 12. The Backup, Archive and Restore NetBackup Client Interface.

> Complete the restore options with the destination paths of the restore.

- For the single server configuration: As the backup is performed using the SnapShot target LUNs, the destination path of the restore is not the original location. So select the right path. Here, the L:\ directory was backed up using its SnapShot on M:\, so the source is M:\ and the destination L:\.
- For the dedicated backup server configuration: As the backup is performed on the backup server, the target partition is mounted on any available drive. Here, the L:\ directory is used on both servers. The restore destination path must be on the production server, which must be accessible via the network. Click on the Browse button and find the right path.

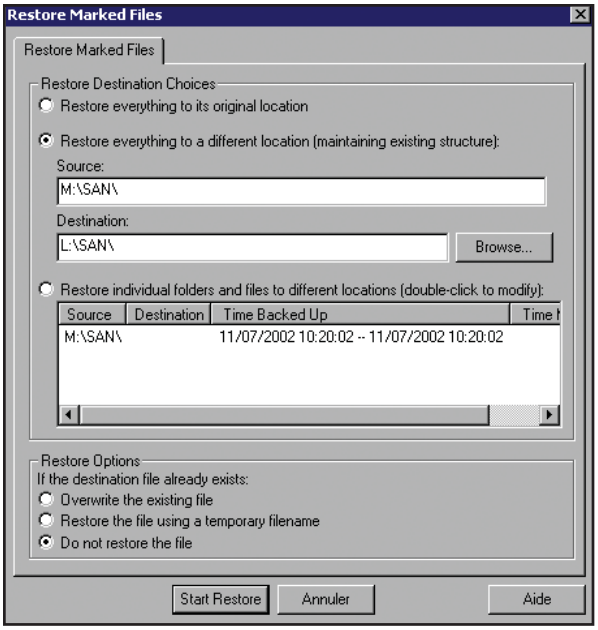


Figure 13. Restore Options.

> Then, choose whether an existing file must be overwritten or not.

> Initiate the restore by clicking on the "Start Restore" button.

> A message box appears to ask if you want to view the restore status. Choose Yes and the following window appears:

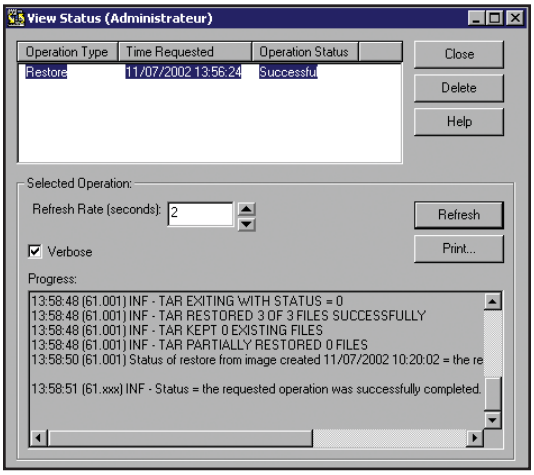


Figure 14. Restore Status Interface.

> If the operation is not successful, refer to the messages displayed to see what the problem is.

Note For the restore to be successful, the backup server must be able to access the production server via the network, and must have total control and rights to access the directory (read and write). If problems occur, please ask the administrator.

6 CRITICAL ANALYSIS

Advantages	Drawbacks
Simple: there is only one SVA command per file system.	Dummy data are sent from the client to the backup server during the execution of SnapShot and Release.
The data can be backed up to tape using another server that will mount the SVA SnapShot.	
All the other advantages of a cold backup remain.	

SnapShot works by creating different "views" of data rather than copying the data themselves. By eliminating the need to physically move data for duplication, SnapShot creates copies, almost instantly. With SnapShot's ultrafast duplication capability, backup operations are not on the critical path any more and could be run in parallel with other processes.

How to SnapShot data on the backup server?

In this section, we'll describe how to backup the 'backup server' without using SVAPATH. But in this architecture, the backup may be perform by:

> The production server: You need to install the media server and connect a storage pool to this server. This method consumes the CPU of the production server.
> The network: This is a standard configuration but has the most drawbacks of the backup scenarios because both CPU and network storage are consumed.

The only benefit of this solution is SnapShot. To execute the SnapShot we need to stop or suspend the application during approximately 2–5 seconds, no matter how much data is copied. You save a lot of time.

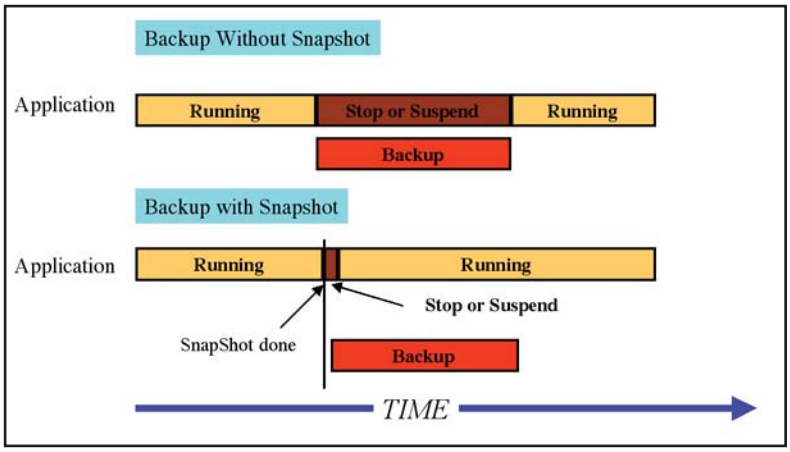


Figure 15. The real advantage of SnapShot.

7 APPENDIX

This section contains a copy of each script used in this project.

APPENDIX A: MAIN BACKUP SCRIPT (NO ADDITIONAL BACKUP SERVER)

backup_script.cmd

```
setlocal enableextensions
set echo off

@REM ************************************************************************
@REM This script is used to perform an automated backup using the SVA SnapShot fea-
ture
@REM and calling the VERITAS NetBackup utility to backup data on a tape (DLT drive)
@REM
@REM All the custom scripts used are located in directory D:\STK_scripts
@REM For further information on the detailed steps, please refer to file
@REM D:\STK_scripts\master.log
@REM ************************************************************************

@REM
@REM Releasing target LUNs
@REM ====================

@echo
@echo Releasing target LUNs.....................
@call D:\STK_scripts\release_luns.cmd D:\STK_scripts\lunconfig.lst

@REM
@REM Snapping LUNs from source to target
@REM ==================================

@echo
@echo Snapping LUNs from source to target..................
@call D:\STK_scripts\snap_luns.cmd D:\STK_scripts\lunconfig.lst

@REM
@REM Calling the backup policy
@REM ========================

@echo
@echo Calling the backup utility..................
@call D:\STK_scripts\backup_luns.cmd D:\STK_scripts\lunconfig.lst

ENDLOCAL

pause
```

APPENDIX B: MAIN BACKUP SCRIPT (DEDICATED BACKUP SERVER)

```
setlocal enableextensions
set echo off

@REM ************************************************************************************
@REM This script is used to perform an automated backup using the SVA SnapShot fea-
ture
@REM and calling the VERITAS NetBackup utility to backup data on a tape (DLT drive)
@REM
@REM All the custom scripts used are located in directory D:\STK_scripts
@REM For further information on the detailed steps, please refer to file
@REM D:\STK_scripts\master.log
@REM ************************************************************************************

@REM
@REM Releasing target LUNs
@REM ====================

@echo
@echo RELEASING TARGET LUNS
@call D:\STK_scripts\release_luns.cmd D:\STK_scripts\lunconfig_trg.lst

@REM
@REM Snapping LUNs from source to target
@REM ==================================

@echo
@echo SNAPPING LUNS FROM SOURCE TO TARGET
@rsh neste D:\STK_scripts\snap_luns.cmd D:\STK_scripts\lunconfig_src.lst

@REM
@REM Calling the backup policy
@REM ========================

@echo
@echo CALLING BACKUP UTILITY
@call D:\STK_scripts\backup_luns.cmd D:\STK_scripts\lunconfig_trg.lst

ENDLOCAL

pause
```

APPENDIX C: LUNS CONFIGURATION FILE (NO ADDITIONAL BACKUP SERVER)

lunconfig.lst

```
#
# This is the configuration file for the LUNs to Snap on the source server
#
# Fields are separated by tabs!
#
# Field Description:
#
# 1. The first field is the UNIQUE identifier of each line of this file
#    It is not used by the script at that time. Use SVALUN_<LUN Number>.
# 2. The second field is LUN number from SVA Path (use setsp to display the LUNs)
# 3. The third field is the SnapShot source partition letter
# 4. The fourth field is the SnapShot TARGET partition letter (set this parameter
carefully!!!)
# 5. The fifth field is the physical disk name for the target partition
#    Use “stkassign list” to display the currently mapped partitions
#
# examples are given below:
#
#              Target
# Target       SVAPath Src       Target   Target_Device
# SVA LUN      LUN#
SVALUN_2       26       L:       M:       \Device\HarddiskVolume6
```

APPENDIX D: LUNS CONFIGURATION FILES (DEDICATED BACKUP SERVER)

lunconfig_src.lst

```
#
# This is the configuration file for the LUNs to Snap on the source server
#
# Fields are separated by tabs!
#
# Field Description:
#
# 1. The first field is the UNIQUE identifier of each line of this file
#    It is not used by the script at that time. Use SVALUN_<LUN Number>.
# 2. The second field is LUN number from SVA Path (use setsp to display the LUNs)
# 3. The third field is the SnapShot source partition letter
# 4. The fourth field is the SnapShot TARGET partition letter (set this parameter
carefully!!!)
# 5. The fifth field is the physical disk name for the target partition
#    Use “stkassign list” to display the currently mapped partitions
#
# examples are given below:
#
#              Target
# Target       SVAPath Src       Target   Target_Device
# SVA LUN      LUN#
SVALUN_2       2        L:       M:       \Device\HarddiskVolume6
```

lunconfig_trg.lst

```
#
# This is the configuration file for the LUNs to Backup on the target server
#
# Fields are separated by tabs!
#
# Field Description:
#
# 1. The first field is the UNIQUE identifier of each line of this file
#    It is not used by the script at that time. Use SVALUN_<LUN Number>.
# 2. The second field is LUN number from SVA Path (use setsp to display the LUNs)
# 3. The third field is the SnapShot source partition letter
# 4. The fourth field is the SnapShot TARGET partition letter (set this parameter
carefully!!!)
# 5. The fifth field is the physical disk name for the target partition
#    Use “stkassign list” to display the currently mapped partitions
#
# examples are given below:
#
#              Target
# Target       SVAPath Target    Target_Device
# SVA LUN      LUN#
SVALUN_2       26       L:       \Device\HarddiskVolume5
```

APPENDIX E: RELEASE LUNS SCRIPTS

release_luns.cmd

```
@echo off
SETLOCAL ENABLEEXTENSIONS

REM
REM This script is used to perform a release on the target LUNs on the backup server
REM


REM
REM Update next line to reflect the absolute path of SVAA CLI Batch File
REM On your configuration
REM Use double quotes if spaces are part of the pathnames
REM ================================================================

set sibadmin="D:\Program Files\StorageTek\Shared Virtual Array Administrator
3.1.0\bin\sibadmin.bat"

REM
REM Check Usage
REM we need STK_SCRIPTS_HOME to be set properly
REM We need the LUN Configuration File as an input parameter
REM ===========

if [%1] EQU [] goto s_usage
if [%2] NEQ [] goto s_usage

if [%STK_SCRIPTS_HOME%] EQU [] goto s_usage
set local_run_path=%STK_SCRIPTS_HOME%

if NOT EXIST %local_run_path%\release.cmd goto s_usage
goto s_start

:s_usage

echo syntax error
echo Set STK_SCRIPTS_HOME environment variable properly.
echo usage: %0 LUN_configuration_file_name
goto s_end

:s_start

set lun_config_file=%1

REM
REM Setting default file names
REM =========================

set master_log=master.log

REM
REM Get Date and Time for Logging
REM ============================

for /F "tokens=1*" %%P in ('date /T') do (set d_day=%%P %%Q)
for /F "tokens=1*" %%P in ('time /T') do (set d_date=%d_day% %%P)
```

```
echo . >> %local_run_path%\%master_log%
echo . >> %local_run_path%\%master_log%
echo ============================================================ >>
%local_run_path%\%master_log%

echo ****** %0 starting on [%d_date%] >> %local_run_path%\%master_log%

echo. >> %local_run_path%\%master_log%

set lun_config_file=%1

REM
REM For Each LUN:
REM - Mount, Release, Check and Unmount
REM ==================================

REM [ delims is a tab in the following loop ]

FOR /F "tokens=1,2,3,4,5* delims= " %%G in ('findstr /V ^# %lun_config_file% 2^>nul')
do (call %local_run_path%\release.cmd %%H %%J %%K)

echo. >> %local_run_path%\%master_log%

echo ********************************************** >>
%local_run_path%\%master_log%

echo. >> %local_run_path%\%master_log%

:a_end

for /F "tokens=1*" %%P in ('date /T') do (set d_day=%%P %%Q)
for /F "tokens=1*" %%P in ('time /T') do (set d_date=%d_day% %%P)

echo. >> %local_run_path%\%master_log%
echo ****** %0 ending on  [%d_date%] >> %local_run_path%\%master_log%
ENDLOCAL

:s_end
```

Note **This script is used for the single server configuration. The one used for the production and backup configuration is almost identical. The only difference resides on the line with the calling release.cmd command. This difference is due to the different content of the lunconfig_trg.lst file.**

For the second configuration, replace that line by the following:

```
FOR /F "tokens=1,2,3,4* delims=   " %%G in ('findstr /V ^# %lun_config_file% 2^>nul')
do (call %local_run_path%\release.cmd %%H %%I %%J)
```

release.cmd

```
@echo off
setlocal enableextensions

REM
REM Check Usage - we need 4 parameters
REM - 1 - SVA Path Target LUN Number
REM - 2 - Target Partition Letter
REM - 3 - Target Physical Disk Name
REM ================================

if [%3] EQU [] goto s_usage
if [%4] NEQ [] goto s_usage
goto s_start

:s_usage

echo syntax error
echo usage: ^<%0 SVAPath_lun_number target_drive_letter target_physical_disk_name^>
goto s_end

:s_start
set svapath_target_LUN=%1
set target_letter=%2
set target_disk=%3

REM
REM Processing LUN
REM ==============

echo. >> %local_run_path%\%master_log%

echo Performing the release.................................
echo ****** Processing LUN: >> %local_run_path%\%master_log%
echo  SVAPath Target LUN:        %svapath_target_LUN% >>
%local_run_path%\%master_log%
echo  Target Paritition Letter:  %target_letter% >> %local_run_path%\%master_log%
echo  Target Disk Name:          %target_disk% >> %local_run_path%\%master_log%

REM
REM Include and mount Target LUNs
REM ============================

echo - Including Target LUN... >> %local_run_path%\%master_log% 2>&1
setsp -l %svapath_target_LUN% -e 0 >> %local_run_path%\%master_log% 2>&1
if %ERRORLEVEL% NEQ 0 goto s_error

echo - Mounting Target Partition... >> %local_run_path%\%master_log% 2>&1
%local_run_path%\stkassign mount %target_letter% %target_disk% >>
%local_run_path%\%master_log% 2>&1
if %ERRORLEVEL% NEQ 0 goto s_error

REM
REM Format Target LUNs - avoid catching side effects on NTFS
REM ========================================================

label %target_letter% NBU_trg >> %local_run_path%\%master_log% 2>&1
echo - Formatting LUNs...  >> %local_run_path%\%master_log% 2>&1
@(
```

```
echo NBU_trg
echo O
) | format %target_letter% /FS:NTFS /Q /V:NBU_trg >> %local_run_path%\%master_log%
2>&1
if %ERRORLEVEL% NEQ 0 goto s_error

REM
REM Unmount and exclude Target LUNs (Target Partition Letter becomes available to NT)
REM ================================================================================

echo - Syncing and Unmounting Target Partition... >> %local_run_path%\%master_log%
2>&1
%local_run_path%\stkassign sync %target_letter% >> %local_run_path%\%master_log% 2>&1
%local_run_path%\stkassign umount %target_letter% >> %local_run_path%\%master_log%
2>&1
if %ERRORLEVEL% NEQ 0 goto s_error

REM echo - Excluding Target LUN... >> %local_run_path%\%master_log% 2>&1
REM setsp -l %svapath_target_LUN% -e 1 >> %local_run_path%\%master_log% 2>&1
REM if %ERRORLEVEL% NEQ 0 goto s_error

goto s_end

:s_error

echo
echo ==== ERROR encountered!!! See messages above. ****** >> %local_run_path%\%mas-
ter_log%

:s_end

echo ****** Target LUNs Released. ****** >> %local_run_path%\%master_log%

ENDLOCAL
```

APPENDIX F: SNAPSHOT LUNS SCRIPTS

snap_luns.cmd

```
@echo off
SETLOCAL ENABLEEXTENSIONS

REM
REM Check Usage
REM we need STK_SCRIPTS_HOME to be set properly
REM We need the LUN Configuration File as an input parameter
REM ===========

if [%1] EQU [] goto s_usage
if [%2] NEQ [] goto s_usage
if [%STK_SCRIPTS_HOME%] EQU [] goto s_usage

set local_run_path=%STK_SCRIPTS_HOME%

if NOT EXIST %local_run_path%\snap.cmd goto s_usage
goto s_start

:s_usage

echo syntax error
echo Set STK_SCRIPTS_HOME environment variable properly.
echo usage: %0 _LUN_configuration_file_name >> %local_run_path%\%master_log%
goto s_end

:s_start

set lunconfig_file=%1

REM
REM Setting default file names
REM =========================

set master_log=master.log

REM
REM Get Date and Time for Logging
REM ============================

for /F “tokens=1*” %%P in (‘date /T’) do (set d_day=%%P %%Q)
for /F “tokens=1*” %%P in (‘time /T’) do (set d_date=%d_day% %%P)

echo . >> %local_run_path%\%master_log%
echo . >> %local_run_path%\%master_log%
echo ============================================================ >>
%local_run_path%\%master_log%

echo ****** %0 starting on [%d_date%] >> %local_run_path%\%master_log%

echo. >> %local_run_path%\%master_log%

REM
REM For Each LUN:
REM - Mount, Snap, Check and Unmount
REM ================================

REM [ delims is a tab in the following loop ]
```

```
FOR /F "tokens=1,2,3,4,5* delims= " %%G in ('findstr /V ^# %lunconfig_file% 2^>nul')
do (call %local_run_path%\snap.cmd %%H %%I %%J %%K)

echo. >> %local_run_path%\%master_log%

echo ********************************************** >>
%local_run_path%\%master_log%

echo. >> %local_run_path%\%master_log%

:a_end

for /F "tokens=1*" %%P in ('date /T') do (set d_day=%%P %%Q)
for /F "tokens=1*" %%P in ('time /T') do (set d_date=%d_day% %%P)

echo. >> %local_run_path%\%master_log%
echo ****** %0 ending on  [%d_date%] >> %local_run_path%\%master_log%
ENDLOCAL

:s_end
```

snap.cmd

```
@echo off
setlocal enableextensions

REM
REM Update next line to reflect the absolute path of SVAA CLI Batch File
REM Use double quotes if spaces are part of the pathnames
REM ================================================================
set sibadmin="D:\Program Files\StorageTek\Shared Virtual Array Administrator
3.1.0\bin\sibadmin.bat"

REM
REM Check Usage - we need 4 parameters
REM - 1 - SVA Path Target LUN Number
REM - 2 - Source Partition Letter
REM - 3 - Target Partition Letter
REM - 4 - Target Physical Disk Name
REM ================================

if [%4] EQU [] goto s_usage
if [%5] NEQ [] goto s_usage
if [%STK_SCRIPTS_HOME%] EQU [] goto s_usage
goto s_start

:s_usage

echo syntax error
echo Set STK_SCRIPTS_HOME environment variable properly.
echo usage: ^<%0 SVAPath_lun_number source_drive_letter target_drive_letter
target_physical_disk_name^>
goto s_end

:s_start

set svapath_target_LUN=%1
set source_letter=%2
set target_letter=%3
```

```
set target_disk=%4

REM
REM Processing LUN
REM ==============

echo. >> %local_run_path%\%master_log%

echo Performing the SnapShot.............................
echo ****** Processing LUN: >> %local_run_path%\%master_log%
echo  SVAPath Target LUN:        %svapath_target_LUN% >>
%local_run_path%\%master_log%
echo  Source Partition Letter:   %source_letter% >> %local_run_path%\%master_log%
echo  Target Paritition Letter:  %target_letter% >> %local_run_path%\%master_log%
echo  Target Disk Name:          %target_disk% >> %local_run_path%\%master_log%

REM
REM Include and mount Target LUNs
REM ==============================

echo - Including Target LUN... >> %local_run_path%\%master_log% 2>&1
setsp -l %svapath_target_LUN% -e 0 >> %local_run_path%\%master_log% 2>&1
if %ERRORLEVEL% NEQ 0 goto s_error

echo - Mounting Target Partition... >> %local_run_path%\%master_log% 2>&1
%local_run_path%\stkassign mount %target_letter% %target_disk% >>
%local_run_path%\%master_log% 2>&1
if %ERRORLEVEL% NEQ 0 goto s_error

REM
REM Snapping From Source to Target
REM ==============================

echo - Snapping LUNs... >> %local_run_path%\%master_log% 2>&1
call %sibadmin% snap -source %source_letter% -target %target_letter% >>
%local_run_path%\%master_log% 2>&1
if %ERRORLEVEL% NEQ 0 goto s_error

REM
REM Unmount and exclude Target LUNs (Target Partition Letter becomes available to NT)
REM ================================================================================

echo - Syncing and Unmounting Target Partition... >> %local_run_path%\%master_log%
2>&1
%local_run_path%\stkassign sync %target_letter% >> %local_run_path%\%master_log% 2>&1
%local_run_path%\stkassign umount %target_letter% >> %local_run_path%\%master_log%
2>&1
if %ERRORLEVEL% NEQ 0 goto s_error

echo - Excluding Target LUN... >> %local_run_path%\%master_log% 2>&1
setsp -l %svapath_target_LUN% -e 1 >> %local_run_path%\%master_log% 2>&1
if %ERRORLEVEL% NEQ 0 goto s_error

goto s_end

:s_error

echo ==== ERROR encountered!!! See messages above. ****** >> %local_run_path%\%mas-
ter_log%

:s_end
```

```
echo ****** Target LUNs Snapped. ****** >> %local_run_path%\%master_log%

ENDLOCAL
```

APPENDIX G: BACKUP LUNS SCRIPTS

backup_luns.cmd

```
@echo off
SETLOCAL ENABLEEXTENSIONS

REM
REM Update next line to reflect the absolute path of NeTBackup bpbackup command
REM Use double quotes if spaces are part of the pathnames
REM ================================================================
set bpbackup="D:\Program Files\VERITAS\NetBackup\bin\bpbackup.exe"

if [%STK_SCRIPTS_HOME%] EQU [] goto s_usage
set local_run_path=%STK_SCRIPTS_HOME%

goto s_start

:s_usage

echo syntax error
echo Set STK_SCRIPTS_HOME environment variable properly.
echo usage: %0 LUN_configuration_file_name
goto s_end

:s_start

set lunconfig_file=%1

REM
REM Setting default file names
REM =========================

set master_log=master.log

REM
REM Get Date and Time for Logging
REM ============================

for /F "tokens=1*" %%P in ('date /T') do (set d_day=%%P %%Q)
for /F "tokens=1*" %%P in ('time /T') do (set d_date=%d_day% %%P)

echo . >> %local_run_path%\%master_log%
echo . >> %local_run_path%\%master_log%
echo ========================================================== >>
%local_run_path%\%master_log%

echo ****** %0 NBU backup starting on [%d_date%] >> %local_run_path%\%master_log%

echo. >> %local_run_path%\%master_log%


call %local_run_path%\mount_luns.cmd %lunconfig_file%
call %bpbackup% -p NBU_policy -i -w
call %local_run_path%\umount_luns.cmd %lunconfig_file%


for /F "tokens=1*" %%P in ('date /T') do (set d_day=%%P %%Q)
for /F "tokens=1*" %%P in ('time /T') do (set d_date=%d_day% %%P)

echo. >> %local_run_path%\%master_log%
```

```
echo ****** %0 ending on  [%d_date%] >> %local_run_path%\%master_log%
ENDLOCAL

:s_end
```

mount_luns.cmd

```
@echo off
SETLOCAL ENABLEEXTENSIONS

REM
REM This script is used to perform a mount the target LUNs on the backup server
REM

REM
REM Update next line to reflect the absolute path of SVAA CLI Batch File
REM On your configuration
REM Use double quotes if spaces are part of the pathnames
REM ================================================================

set sibadmin="D:\Program Files\StorageTek\Shared Virtual Array Administrator
3.1.0\bin\sibadmin.bat"

REM
REM Check Usage
REM we need STK_SCRIPTS_HOME to be set properly
REM We need the LUN Configuration File as an input parameter
REM ===========

if [%1] EQU [] goto s_usage
if [%2] NEQ [] goto s_usage

if [%STK_SCRIPTS_HOME%] EQU [] goto s_usage
set local_run_path=%STK_SCRIPTS_HOME%

if NOT EXIST %local_run_path%\umount.cmd goto s_usage
goto s_start

:s_usage

echo syntax error
echo Set STK_SCRIPTS_HOME environment variable properly.
echo usage: %0 LUN_configuration_file_name
goto s_end

:s_start

set lun_config_file=%1

REM
REM Setting default file names
REM =========================

set master_log=master.log
set lun_config_file=%1

REM
REM For Each LUN:
REM - Mount, Snap, Check and Unmount
REM ================================
```

```
REM [ delims is a tab in the following loop ]
FOR /F "tokens=1,2,3,4,5* delims= " %%G in ('findstr /V ^# %lun_config_file% 2^>nul')
do (call %local_run_path%\mount.cmd %%H %%J %%K)

echo. >> %local_run_path%\%master_log%

echo ********************************************** >>
%local_run_path%\%master_log%

echo. >> %local_run_path%\%master_log%

:a_end

ENDLOCAL

:s_end
```

Note This script is used for the single server configuration. The one used for the production and backup configuration is almost identical. The only difference resides on the line with the calling release.cmd command. This difference is due to the different content of the lunconfig_trg.lst file.

For the second configuration, replace that line by the following:

```
FOR /F "tokens=1,2,3,4* delims=   " %%G in ('findstr /V ^# %lun_config_file% 2^>nul')
do (call %local_run_path%\mount.cmd %%H %%I %%J)
```

mount.cmd

```
@echo off
setlocal enableextensions

REM
REM Update next line to reflect teh absolute path of SVAA CLI Batch File
REM Use double quotes if spaces are part of the pathnames
REM ================================================================
set sibadmin="D:\Program Files\StorageTek\Shared Virtual Array Administrator
3.1.0\bin\sibadmin.bat"

REM
REM Check Usage - we need 4 parameters
REM - 1 - SVA Path Target LUN Number
REM - 2 - Target Partition Letter
REM - 3 - Target Physical Disk Name
REM ================================

if [%3] EQU [] goto s_usage
if [%4] NEQ [] goto s_usage
goto s_start

:s_usage

echo syntax error
echo usage: ^<%0 SVAPath_lun_number target_drive_letter target_physical_disk_name^>
goto s_end
```

```
:s_start

set svapath_target_LUN=%1
set target_letter=%2
set target_disk=%3

REM
REM Include Target LUN (SVAPath)
REM ============================

echo - Including Target LUN... >> %local_run_path%\%master_log% 2>&1
setsp -l %svapath_target_LUN% -e 0 >> %local_run_path%\%master_log% 2>&1
if %ERRORLEVEL% NEQ 0 goto s_error

REM
REM Mount Target LUN (Target Partition Letter becomes available to NT)
REM ================================================================

echo - Mounting Target Partition... >> %local_run_path%\%master_log% 2>&1
%local_run_path%\stkassign mount %target_letter% %target_disk% >>
%local_run_path%\%master_log% 2>&1
if %ERRORLEVEL% NEQ 0 goto s_error


goto s_end

:s_error

echo ==== ERROR encountered!!! See messages above. ****** >> %local_run_path%\%mas-
ter_log%

:s_end

echo ****** Target LUNs mounted. ****** >> %local_run_path%\%master_log%

ENDLOCAL


umount_luns.cmd
@echo off
SETLOCAL ENABLEEXTENSIONS

REM
REM This script is used to perform a unmount the target LUNs on the backup server
REM


REM
REM Update next line to reflect the absolute path of SVAA CLI Batch File
REM On your configuration
REM Use double quotes if spaces are part of the pathnames
REM ================================================================

set sibadmin="D:\Program Files\StorageTek\Shared Virtual Array Administrator
3.1.0\bin\sibadmin.bat"

REM
REM Check Usage
REM we need STK_SCRIPTS_HOME to be set properly
REM We need the LUN Configuration File as an input parameter
```

```
REM ===========

if [%1] EQU [] goto s_usage
if [%2] NEQ [] goto s_usage

if [%STK_SCRIPTS_HOME%] EQU [] goto s_usage
set local_run_path=%STK_SCRIPTS_HOME%

if NOT EXIST %local_run_path%\umount.cmd goto s_usage
goto s_start

:s_usage

echo syntax error
echo Set STK_SCRIPTS_HOME environment variable properly.
echo usage: %0 LUN_configuration_file_name
goto s_end

:s_start

set lun_config_file=%1

REM
REM Setting default file names
REM ==========================

set master_log=master.log
set lun_config_file=%1

REM
REM For Each LUN:
REM - Check and Unmount
REM ===================

REM [ delims is a tab in the following loop ]
FOR /F "tokens=1,2,3,4,5* delims= " %%G in ('findstr /V ^# %lun_config_file% 2^>nul')
do (call %local_run_path%\umount.cmd %%H %%J)

echo. >> %local_run_path%\%master_log%

echo ********************************************** >>
%local_run_path%\%master_log%

echo. >> %local_run_path%\%master_log%

:a_end

ENDLOCAL

:s_end
```

Note This script is used for the single server configuration. The one used for the production and backup configuration is almost identical. The only difference resides on the line with the calling release.cmd command. This difference is due to the different content of the lunconfig_trg.lst file.

For the second configuration, replace that line by the following:

```
FOR /F "tokens=1,2,3,4* delims=   " %%G in ('findstr /V ^# %lun_config_file% 2^>nul')
do (call %local_run_path%\umount.cmd %%H %%I %%J)
```

umount.cmd

```
@echo off
setlocal enableextensions

REM
REM Update next line to reflect the absolute path of SVAA CLI Batch File
REM Use double quotes if spaces are part of the pathnames
REM ================================================================
set sibadmin="D:\Program Files\StorageTek\Shared Virtual Array Administrator
3.1.0\bin\sibadmin.bat"

REM
REM Check Usage - we need 2 parameters
REM - 1 - SVAPath Target LUN
REM - 2 - Target Partition Letter
REM ================================

if [%2] EQU [] goto s_usage
if [%3] NEQ [] goto s_usage
goto s_start

:s_usage

echo syntax error
echo usage: ^<%0 SVAPath_lun_number target_drive_letter^>
goto s_end

:s_start

set svapath_target_LUN=%1
set target_letter=%2

REM
REM Un-Mount Target LUN (Target Partition Letter is no more available to NT)
REM ============================================================================

echo - Syncing and Unmounting Target Partition... >> %local_run_path%\%master_log%
2>&1
%local_run_path%\stkassign sync %target_letter% >> %local_run_path%\%master_log% 2>&1
%local_run_path%\stkassign umount %target_letter% >> %local_run_path%\%master_log%
2>&1
if %ERRORLEVEL% NEQ 0 goto s_error

REM
REM Exclude Target LUN (SVAPath)
REM ============================

echo - Excluding Target LUN... >> %local_run_path%\%master_log% 2>&1
setsp -l %svapath_target_LUN% -e 1 >> %local_run_path%\%master_log% 2>&1
if %ERRORLEVEL% NEQ 0 goto s_error

goto s_end
```

```
:s_error

echo ==== ERROR encountered!!! See messages above. ****** >> %local_run_path%\%mas-
ter_log%

:s_end

echo ****** Target LUN dismounted. ****** >> %local_run_path%\%master_log%

ENDLOCAL
```

APPENDIX H: EXAMPLE OF SUCCESSFUL BACKUP (NO ADDITIONAL BACKUP SERVER)

master.log

```
============================================================
****** D:\STK_scripts\release_luns.cmd starting on [jeu. 11/07/2002  10:19]


****** Processing LUN:
      SVAPath Target LUN:        26
      Target Paritition Letter:  M:
      Target Disk Name:          \Device\HarddiskVolume6
- Including Target LUN...
- Mounting Target Partition...
“M:” is now mapped to:
\Device\HarddiskVolume6
- Formatting LUNs...
Le type du syst?me de fichiers est NTFS.
Entrez le nom de volume en cours pour le lecteur M: NBU_trg

Attentionÿ: toutes les donn,es sur le lecteur de disque
non amovible M: seront perduesÿ!
Continuer le formatage (O/N)ÿ? O
Formatage rapide de 2337 Mo
Cr,ation des structures du syst?me de fichiers.
Formatage termin,.
   2393652 Ko d’espace disque au total.
   2379344 Ko sont disponibles.
- Syncing and Unmounting Target Partition...
Internal Disk Name = \\.\M:

Done
“M:” unmapped.
****** Target LUNs Released. ******

************************************************


****** D:\STK_scripts\release_luns.cmd ending on  [jeu. 11/07/2002  10:19]
.
.
============================================================
****** D:\STK_scripts\snap_luns.cmd starting on [jeu. 11/07/2002  10:19]


****** Processing LUN:
      SVAPath Target LUN:        26
      Source Partition Letter:   L:
      Target Paritition Letter:  M:
      Target Disk Name:          \Device\HarddiskVolume6
- Including Target LUN...
- Mounting Target Partition...
“M:” is now mapped to:
\Device\HarddiskVolume6
- Snapping LUNs...
Starting CLI...
SIB9863I: SnapShot is completed.
- Syncing and Unmounting Target Partition...
Internal Disk Name = \\.\M:
```

```
Done
"M:" unmapped.
- Excluding Target LUN...
****** Target LUNs Snapped. ******

************************************************


****** D:\STK_scripts\snap_luns.cmd ending on  [jeu. 11/07/2002  10:19]
.
.
============================================================
****** D:\STK_scripts\backup_luns.cmd NBU backup starting on [jeu. 11/07/2002  10:19]

- Including Target LUN...
- Mounting Target Partition...
"M:" is now mapped to:
\Device\HarddiskVolume6
****** Target LUNs mounted. ******

************************************************

- Syncing and Unmounting Target Partition...
Internal Disk Name = \\.\M:

Done
"M:" unmapped.
- Excluding Target LUN...
****** Target LUN dismounted. ******

************************************************


****** D:\STK_scripts\backup_luns.cmd ending on  [jeu. 11/07/2002  10:22]
```

APPENDIX I: EXAMPLE OF SUCCESSFUL BACKUP (DEDICATED BACKUP SERVER)

master.log (on the production server)

```
============================================================
****** D:\STK_scripts\snap_luns.cmd starting on [ven. 19/07/2002  14:40]


****** Processing LUN:
      SVAPath Target LUN:        2
      Source Partition Letter:   L:
      Target Paritition Letter:  M:
      Target Disk Name:          \Device\HarddiskVolume6
- Including Target LUN...
- Mounting Target Partition...
“M:” is now mapped to:
\Device\HarddiskVolume6
- Snapping LUNs...
Starting CLI...
SIB9863I: SnapShot is completed.
- Syncing and Unmounting Target Partition...
Internal Disk Name = \\.\M:

Done
“M:” unmapped.
- Excluding Target LUN...
****** Target LUNs Snapped. ******

************************************************


****** D:\STK_scripts\snap_luns.cmd ending on  [ven. 19/07/2002  14:40]
```

master.log (on the backup server)

```
.
.
============================================================
****** D:\STK_scripts\release_luns.cmd starting on [ven. 19/07/2002  14:40]


****** Processing LUN:
      SVAPath Target LUN:        26
      Target Paritition Letter:  L:
      Target Disk Name:          \Device\HarddiskVolume5
- Including Target LUN...
- Mounting Target Partition...
“L:” is now mapped to:
\Device\HarddiskVolume5
- Formatting LUNs...
Le type du syst?me de fichiers est NTFS.
Entrez le nom de volume en cours pour le lecteur L: NBU_trg

Attentionÿ: toutes les donn,es sur le lecteur de disque
non amovible L: seront perduesÿ!
Continuer le formatage (O/N)ÿ? O
Formatage rapide de 2337 Mo
Cr,ation des structures du syst?me de fichiers.
Formatage termin,.
   2393652 Ko d’espace disque au total.
```

```
   2379344 Ko sont disponibles.
- Syncing and Unmounting Target Partition...
Internal Disk Name = \\.\L:

Done
“L:” unmapped.
- Excluding Target LUN...
****** Target LUNs Released. ******

***********************************************


****** D:\STK_scripts\release_luns.cmd ending on  [ven. 19/07/2002  14:40]
.
.
============================================================
****** D:\STK_scripts\backup_luns.cmd NBU backup starting on [ven. 19/07/2002  14:40]

- Including Target LUN...
- Mounting Target Partition...
“L:” is now mapped to:
\Device\HarddiskVolume5
****** Target LUNs mounted. ******

***********************************************

Call backup utility with NBU_policy
- Syncing and Unmounting Target Partition...
Internal Disk Name = \\.\L:

Done
“L:” unmapped.
- Excluding Target LUN...
****** Target LUN dismounted. ******

***********************************************


****** D:\STK_scripts\backup_luns.cmd ending on  [ven. 19/07/2002  14:45]
```

After the release of the target LUNs, the SnapShot is performed on the data host via a rsh command.

ABOUT STORAGETEK®

Storage Technology Corporation (NYSE: STK), a $2 billion worldwide company with headquarters in Louisville, CO, has been delivering a broad range of storage management solutions designed for IT professionals for over 30 years. StorageTek offers solutions that are easy to manage, integrate well with existing infrastructures and allow universal access to data across servers, media types and storage networks. StorageTek's practical and safe storage solutions for tape automation, disk storage systems and storage integration, coupled with a global services network, provide IT professionals with confidence and know-how to manage their entire storage management ecosystem today and in the future.

StorageTek products are available through a worldwide network. For more information, visit www.storagetek.com, or call 1.800.275.4785 or 01.303.673.2800.

WORLD HEADQUARTERS

Storage Technology Corporation
One StorageTek Drive
Louisville, Colorado 80028 USA
1.800.877.9220 or 01.303.673.5151

MS 9156 B 04/04