



OpenVMS Backup Strategies In A SAN Environment

Ed Richard

Consultant
Computer Muscle

Intended Audience

- OpenVMS Backup Administrators that prefer to use the existing OpenVMS Backup Utility as their primary backup and restore strategy.
- OpenVMS Backup Administrators that are looking for OpenVMS tools that will assist them in more complex SAN environments.

Goals of This Session

- Familiarize OpenVMS Backup Administrators with existing HP tools, that allow DCL level interaction with their storage environment for backup purposes.
- Pass along ideas of how to manage storage devices (focusing on a SAN environment); to help achieve a **reliable** backup solution.

What We Will Not Be Discussing

- Limitations of storage (due to differences in OS versions and installed hardware base).
- GUI interfacing with storage.
- 3rd party backup solutions. This discussion will focus on OpenVMS solutions available from HP that support the OpenVMS Backup Utility. * One exception to this point will be a quick demo on a free Internet based backup calendar that could prove useful to your environment.

Audience Poll

- What Version of OpenVMS do you currently have installed?
- Who uses the Backup Utility exclusively, with no 3rd party backup software?
- Who uses a combination of OpenVMS Backup with some 3rd party backup software?
- Who is familiar with Command Scripter?
- Who is familiar with the Storage System Scripting Utility (SSSU)?
- Who is looking to move away from OpenVMS based backups to a 3rd party solution?

Topics In This Presentation

- HP StorageWorks Command Scripter
- HP StorageWorks Storage System Scripting Utility (SSSU) for Enterprise Array

SAN Compatible Versions of OpenVMS Discussed

- OpenVMS 7.1-x
- OpenVMS 7.2-x
- OpenVMS 7.3-x.

HP STORAGEWORKS COMMAND SCRIPTER

StorageWorks Command Scripter Overview

- HP defines Command Scripter as “a scripting application that provides command-level control of StorageWorks Systems”.
- Supported by OpenVMS 7.1 and later.
- Supported array controllers are:
 - HSG80, HSG60, HSZ70, HSZ80 and HSJ80.

NOTE: There are specific minimum supported levels of the ACS (Array Controller Software) required to utilize Command Scripter, see the online User Guide for more detail.

Command Scripter Syntax

- Command Breakdown

\$ cmdscript -f sys\$sysdevice

Device connected to controller

Switch to represent connect via device.

Call to Command Scripter executable.

- The following example will connect you to the controller, return the text to the screen, but leave you connected.

```
$ cmdscript -f sys$sysdevice  
HSG80> show storagesets
```

- The example below will connect you to the controller, return the text to the screen, then exit back to DCL.

```
$ cmdscript -f sys$sysdevice "show storagesets"
```

Using HP Command Scriptor To Ready A Device For Backup

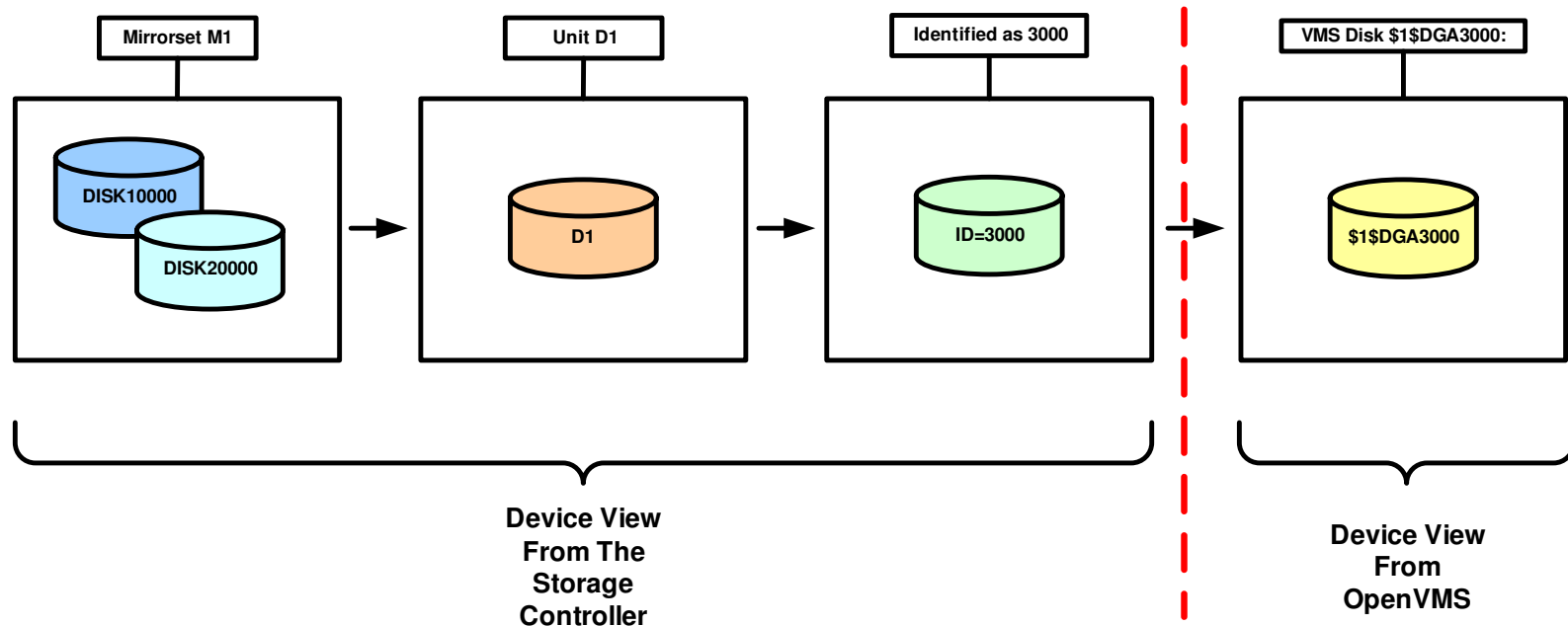
The illustrations to follow will:

- Assume Command Scriptor is installed with a system wide symbol pointing to the executable.
- All be executed in OpenVMS 7.3-1.
- Utilize HSG80 controller hardware.
- Use a single disk mirror (the VMS System Disk) as an example.

Note: The following examples will interoperate with different OS versions and controller hardware, etc. The examples shown are for illustration purposes only.

Single Disk Mirror View From The Storage Controller and OpenVMS

- The following image is to illustrate the relationship of a single mirrored device from the Storage controller and OpenVMS viewpoints.



Display the Device

- You'll first need to acquire the Device Name.

\$ show device sys\$sysdevice:

- This will display something similar to:

Device Name	Device Status	Error Count	Volume Label	Free Blocks	Trans Count	Mnt Cnt
\$1\$DGA3000:	Mounted	1	ALPHASYS	10871735	443	1

- Now that you have acquired the device name, you will be able to use this name to communicate directly with the storage controller via Command Scripter.

Translate the OpenVMS Device Name to the Controller Unit Number

- Use the DCL Pipe command to quickly sort through all devices to return the unit number for the device.

```
$ pipe cmdscript -f sys$sysdevice "show id" | search  
sys$input 3000
```

- This will return something similar to:

D1	3000
----	------

Use The Unit Number To Locate The Mirrorset

- Now that you have the Unit (D1) you will be able to find the mirrorset.

`$ cmdscript -f sys$sysdevice "show D1"`

- This will return something similar to:

LUN	Uses	Used by

D1	M1	
LUN ID: 6000-1FE1-0014-9240-0009-1310-0595-00B5		
IDENTIFIER = 3000		
Switches:		
RUN NOWRITE_PROTECT READ_CACHE		
READAHEAD_CACHE WRITEBACK_CACHE		
MAX_READ_CACHED_TRANSFER_SIZE = 32		
MAX_WRITE_CACHED_TRANSFER_SIZE = 32		
Access:		
HSG1,HSG2		
State:		
ONLINE to the other controller		
NOPREFERRED_PATH		
Size: 35556389 blocks		
Geometry (C/H/S): (7000 / 20 / 254)		
HSG80_1>		

Displaying The Mirrorset, Returns The Device Names

- To locate the disk membership just display the mirror.

\$ cmdscript -f sys\$sysdevice "show M1"

- This will display the following:

Name	Storage set	Uses	Used by
M1	mirrorset	DISK10000 DISK20000	D1

Switches:
POLICY (for replacement) = BEST_PERFORMANCE
COPY (priority) = NORMAL
READ_SOURCE = LEAST_BUSY
MEMBERSHIP = 2, 2 members present
State:
NORMAL
DISK10000 (member 0) is NORMAL
DISK20000 (member 1) is NORMAL
Size: 35556389 blocks
HSG80_1>

Split The Mirrorset

- Issue the following commands to split this examples' Mirrorset - M1.

```
$ cmdscript -f sys$sysdevice "clear cli"
```

```
$ cmdscript -f sys$sysdevice "set m1 nopolicy"
```

```
$ cmdscript -f sys$sysdevice "set m1 remove =  
DISK20000"
```

```
$ cmdscript -f sys$sysdevice "del failed DISK20000"
```

```
$ cmdscript -f sys$sysdevice "init DISK20000"
```

Enable A New Temporary Disk for Backup

- Create a new device from the split mirrorset that will be seen by OpenVMS.

```
$ cmdscript -f sys$sysdevice "add unit D111  
DISK20000"
```

```
$ cmdscript -f sys$sysdevice "set D111 ID=111"
```

```
$ cmdscript -f sys$sysdevice "set D111 disable=all"
```

```
$ cmdscript -f sys$sysdevice "set d111 enable=(hsg1)"
```

- Run backup using \$1\$DGA111 as the OpenVMS source device.

```
$ mcr sysman io autoconfigure
```

```
$ mount/over=id $1$DGA111
```

Rejoin The Mirrorset

- After the backup is complete, perform the following to replace the mirrorset membership.

```
$ dismount $1$dga111
```

```
$ cmdscript -f sys$sysdevice "del D111"
```

```
$ cmdscript -f sys$sysdevice "set M1  
replace=DISK20000"
```

```
$ cmdscript -f sys$sysdevice "set M1 policy=best_p"
```

HP Command Scripter Conclusion

- Command Scripter is a powerful tool that allows seamless interaction with your storage, directly from DCL.
- Is fully supported by HP and more importantly, the OpenVMS Support Team.
- Can easily be implemented to automate a lights out OpenVMS backup solution.
- Not only will it automate your backups, it will also prove quite useful in the initial setup or continued maintenance of your storage environment.

StorageWorks Command Scripter Ordering Information

<u>Part Number</u>	<u>Quantity</u>
218206-B21	5 License/CD
218207-B21	10 License/CD
218208-B21	25 License/CD

See the following link for more information.

[http://www.hppssn.com/storage_software/command_scripter/
product.pdf](http://www.hppssn.com/storage_software/command_scripter/product.pdf)

STORAGE SYSTEM SCRIPTING UTILITY (SSSU)

Storage System Scripting Utility (SSSU) Overview

- HP Description, “The Storage System Scripting Utility (SSSU) for Enterprise Virtual Array is a command line application that allows you to configure and control EVA3000 and 5000 controllers.”
- OS version support is based on level of firmware.
- The SSSU is part of the EVA Platform kit. All customers who purchase an EVA should also receive the platform kit for their respective operating system. Therefore there is nothing extra to buy.

Smart Setup Of The Storage Environment From The Beginning Is Key

- To make setup a lot easier, use a virtual device naming scheme that is simple to recognize from the EVA as well as the operating system.
- For example, if you would like the system disk to be displayed as \$1\$DGA1000 create a Virtual Disk with the number 1000 somewhere in the name.

nodea_lun1_1000

- Your naming convention for the LUN is not as important for backups, since it is the OS Unit ID that is seen by OpenVMS.

- \$ SSSU

```
NoSystemSelected> select manager 90.0.0.1 user=username pass=password
```

-IP Address of SAN Appliance



HP WORLD 2004
Solutions and Technology Conference & Expo

SSSU Syntax

- SSSU syntax prompting for commands, continued.

NoSystemSelected> select system EVA1

Actual Name of controller pair

Command to select a specific controller pair to manage

EVA1> show VDISK

- Putting it all together to list all disks, without prompts.

```
$ sssu "select manager 90.0.0.1 username=username  
password=password" "select sys EVA1" "show vdisk"
```

Using SSSU To Automate Your OpenVMS Backup.

- The illustration to follow assumes:
 - The SSSU utility is installed and operational on your OpenVMS system.
 - You are familiar with your virtual storage environment; in areas such as root folders as well as usernames and passwords.
 - There is sufficient space in your environment to create snapshots. More specifically there is enough free space within the same Disk Group to create the snapshot.
 - *Even though the illustrations to follow will take you through the creation of a snapshot before demonstrating the removal of the snapshot; it is important to note that in a regular backup routine you will need to delete the previous snapshot before creating an identical one in the next backup rotation.

Example Of A Typical SSSU Scripted Backup

- Create a listing of all disks that meet your criteria.
- Open the disk listing and extract the OS LUN number from the disk name.

```
$ show device dga/mount/out=DEV.lis
```

```
$ open DISK_LISTING DEV.lis
```

```
$ read DISK_LISTING LINE/end_of_file = EXIT
```

```
$ OS_UNIT = f$extract(6,4,""f$element(0,":",LINE)"")
```

Using SSSU To Return The Virtual Disk Name

- Locate the Virtual Disk that is Associated with Numeric LUN.

```
$ pipe SSSU "sel man 90.0.0.1 user=username  
pass=password" "sel sys EVA1" "show vdisk" |  
search sys$input 1000\out=TEMP_LIST1.lis
```

- The above command will yield the following in TEMP_LIST1.lis

```
\Virtual Disks\NODEA\nodea_lun1_1000\ACTIVE
```

- Special Note: The VDISK used both the LUN and OS Unit ID in it's naming convention.

Using SSSU To Take A Snapshot Of A Disk Then Assign An OS Unit ID

- Take a snapshot of the Virtual Disk.
\$ SSSU "sel man 90.0.0.1 user=username
pass=password" "sel sys EVA1" "add snapshot 1000_2
VDISK=""\Virtual
Disks\NODEA\nodea_lun1_1000_2\ACTIVE"
ALLOCATION_POLICY=DEMAND"
- Assign a new OS Unit ID to the snapshot.
\$ SSSU "sel man 90.0.0.1 user=username
pass=password" "sel sys EVA1" "set VDISK=""\Virtual
Disks\NODEA\nodea_lun1_1000\1000_2"
OS_UNIT_ID=200"

Using SSSU To Add A New LUN From The Snapshot

- Enable the new snapshot to be seen by OpenVMS.

```
$ SSSU "sel man 90.0.0.1 user=username  
pass=password" "sel sys EVA1" "add LUN 200  
VDISK=""\Virtual  
Disks\NODEA\nodea_lun1_1000\1000_2""  
OS_UNIT_ID=200"
```

- When the above command completes, enable the snapshot in OpenVMS by keying:

mcr sysman IO AUTOCONFIGURE

After which there will be a new disk device called \$1\$DGA200, which will be an identical copy of \$1\$DGA1000.

Using SSSU To Remove The Snapshots

- After the snapshot has been created and backed up, you are now faced with it's removal. The removal process is the creation process in reverse.
- Again, the following is an example for illustration purposes only. It is realized that the DCL is not always complete in the examples. It assumes your temporary snapshot LUNs are all numbered in the 200 series and that your naming convention of the temporary snapshot VDISKS are appended with an “_2” (as shown in the previous examples).

Using SSSU To Remove The Snapshot LUN

- Create a listing of all snapshot disks that meet your criteria. Open the disk listing and extract the OS LUN number from the disk name.

```
$ pipe SSSU "sel man 90.0.0.1 user=username  
pass=password" "sel sys EVA1" "show lun" |  
search sys$input "NODEA"/out=LUN.lis
```

```
$ open LUN_LISTING LUN.lis
```

```
$ read LUN_LISTING LINE/end_of_file = EXIT
```

```
$ LUN_UNIT = f$edit(LINE,"trim")
```

```
$ SSSU "sel man 90.0.0.1 user=username  
pass=password" "sel sys EVA1" "del lun  
"" 'LUN_UNIT' ""
```

Using SSSU To Remove The Snapshot VDISK

- Create a listing of all VDISKS that meet your criteria. Open the VDISK listing and extract the Snapshot Virtual Disk name.

```
$ pipe SSSU "sel man 90.0.0.1 user=username  
pass=password" "sel sys EVA1" "show vdisk" |  
search sys$input "_2"/out=VDISK.lis
```

```
$ open VDISK_LISTING LUN.lis
```

```
$ read VDISK_LISTING LINE/end_of_file = EXIT
```

```
$ VDISK_UNIT = f$edit(LINE,"trim")
```

```
$ SSSU "sel man 90.0.0.1 user=username  
pass=password" "sel sys EVA1" "del vdisk  
"" 'VDISK_UNIT' ""
```

SSSU Summary

- SSSU is part of the EVA platform kit, so no special licensing to use the product is required.
- SSSU can be used to automate the creation of snapshots in an OpenVMS/EVA environment.
- The snapshot can be mounted to OpenVMS as a legitimate volume, then backed up to tape or disk.
- Remember, in order to fully automate this procedure, it is a good idea to remove snapshots from the previous backup run *prior* to creating new snapshots.
- Naming conventions are key!

Keeping All Backups Organized

- There is a free tool sponsored by one of HP's business partners (Quantum) that enables you to easily set up a backup media strategy. The product is called Color Protect Back-Up Calendar.
- Back-Up Calendar is easily accessed via your browser at www.colorprotect.com

QUESTIONS...

HP WORLD 2004

Solutions and Technology Conference & Expo

Co-produced by:

