

BCVs	PdevName (c#t#d#s2)	SymDevName (eg. 03C)	Attribute	Status
Device 1		6197		
Device 2		0198		

4

Create a device group, add standard devices and associate BCVs

devices. The device group type should be of type REGULAR.

Your device group name: SunTel

a) Create a regular device group with the name stdg<SGN>.

```
# symdg -type REGULAR create stdg<SGN>
```

b) Add the two standard devices to the device group. Add the first device using its Physical Device Name (c#t#d#s2), and the second device using its Symmetric Device Name (eg. 088).

```
# symld -g stdg<SGN> add pd PdevName
# symld -g stdg<SGN> add dev SymDevName
```

c) Verify that the device group was added to the SYMAPI database

```
# symdg list
```

d) Show detailed information about your device group and record the Logical Device Names (LdevName) given to your two Standard Volumes.

```
# symdg show stdg<SGN> | more
```

Std. Volumes	PdevName	SymDevName	LdevName
Device 1		0229A	Dev sd
Device 2		0328J	Dev 028

5	Query the BCV devices in your device group. Because you set the environment variable SYMCLI_DG, it is not necessary to specify the device group with the -g flag. # symcli query Did the command run successfully? <u>Yes</u>												
	(g) Set and export the variable that holds the value of your device group name. If this variable is set, the device group name need not be specified in the command line for subsequent steps. Note: Variables are case sensitive. # export SYMCLI_DG=stdg<SGN>												
	<table border="1"> <thead> <tr> <th>BCVs</th> <th>PdevName</th> <th>SymDevName</th> <th>LdevName</th> </tr> </thead> <tbody> <tr> <td>Device 1</td> <td></td> <td></td> <td>0197</td> </tr> <tr> <td>Device 2</td> <td></td> <td></td> <td>0198</td> </tr> </tbody> </table>	BCVs	PdevName	SymDevName	LdevName	Device 1			0197	Device 2			0198
BCVs	PdevName	SymDevName	LdevName										
Device 1			0197										
Device 2			0198										
	e) Associate the BCVs you identified, with the device group stdg<SGN>. # symbcv -g stdg<SGN> associate dev SymDevName # symbcv -g stdg<SGN> associate dev SymDevName f) Verify association and record the Logical Device name given to your BCVs. # symdg show stdg<SGN> more												

6	Establish the pairing between the BCVs and the Standard Volumes. This synchronizes the contents of the BCVs with those of the Standard Volumes. If this is the first ever establish operation between the BCVs and the Standard Volumes, the “-full” option may have to be specified. This is because, the default is an incremental establish. c) Establish a BCV with each standard device in your device group. Use the -full and -opt options. <pre># symmir establish -full -opt</pre> d) Query the status of the BCV devices. Obtain the synchronization rate, estimated time to completion. The command below repeats the query every 5 seconds, 10 times. <pre># symmir query -i 5 -c 10</pre> Synchronization rate: _____ Estimated time to completion: _____ BCV Device Status: _____
---	--

7	Split the BCV. A BCV must be fully synchronized before it can be split. e) Before synchronization is complete, attempt to split the BCV pairs. <pre># symm1r split</pre> Did the split action succeed? _____ f) Run the "query" again and when the synchronization completes, attempt the split again. <pre># symm1r query -i 5</pre> <pre># symm1r split</pre> <pre># symm1r query</pre> You now have a point-in-time copy of your "data"! g) Reestablish your BCV you split above and again query your devices. <pre># symm1r establish</pre> <pre># symm1r query</pre> Why did this establish complete so quickly? _____
8	TimeFinder Lab 2 will explore the use of a logical volume manager to present BCV volumes to the same host. Before proceeding to the next lab, ensure that the Standard-BCV pairs are split. Delete the SYMCLI device group "stdg<SGN>" that you created. <pre># symdg -force delete stdg<SGN></pre> This concludes TimeFinder Lab 1.

Lab Exercise 2: TimeFinder Integration Lab.

Purpose:	In this lab exercise students will learn to manipulate the BCV Disk Group/Volumes/File systems, and the Standard Volume Disk Group/Volumes and File systems..
-----------------	---

Objectives:	After completion of this exercise, the students will be able to ▪ Establish the BCVs with the Standard Volumes and verify that they are fully synchronized. ▪ Split the BCVs and the Standard Volumes. ▪ Perform Business Continance Operations. ▪ Re-establish the BCVs with the Standard Volumes. ▪ Restore from the BCVs to the Standard Volumes after performing Business Continance Operations.
--------------------	--

References:	EMC Solutions Enabler Symmetrix Timefinder CLI - Version 6.0 PRODUCT GUIDE P/N - 300-000-876 - REV A06
--------------------	--

Lab 2 Part 1: TimeFinder – LVM (Veritas) Integration

STEP	ACTION
------	--------

1	Verify that VERITAS is installed and licensed. <pre># pkginfo grep -y VRTS (Look for VRTSvxvm packages) # vxlicense -p (to preview the licenses)</pre>
---	---

2	Create a Veritas Disk Group and a filesystem on the Standard Volumes. Identify the Symmetrix Logical Volumes available to your host. <pre># syminq more</pre> Find the two Standard Volumes assigned to you and complete the tables. Ensure that the Volumes you have chosen are not already a part of a Veritas Disk Group. <pre># vxdisk list</pre>
---	--

Note: Standard Volumes are those which do not have any description under the "Type" column in the output of the <code>syminq</code> command.	
Standard Volumes	PdevName (c#t#d#s2)
Device 1	
Device 2	
	SymDevName (eg. 031)

Create a Veritas Disk Group. The following command creates a Veritas Disk Group named `stdg<SGN>` and adds your two STD Volumes to the Disk Group. The STD Volumes are given the VM Disk Names `mystdd01` and `mystdd02` respectively.

```
# /etc/vx/bin/vxdisksetup -i c#t#d# (on both disks)
# vxdg init stdg<SGN> mystdd01=c#t#d#s2
mystdd02=c#t#d#s2
```

Create Subdisks. The commands below create two subdisks, each of length 100MB. The subdisk named `mystdd01-01` is created on the VM Disk `mystdd01` and the subdisk named `mystdd02-01` is created on the VM Disk `mystdd02`.

```
# vmake -g stdg<SGN> sd mystdd01-01 mystdd01 len=100m
# vmake -g stdg<SGN> sd mystdd02-01 mystdd02 len=100m
```

Create a striped Plex. A striped plex named `myplex01` with a stripe width of 64KB is created using the two subdisks created with the previous commands.

```
# vmake -g stdg<SGN> plex myplex01 layout=stripe
ncolumn=2 stwidth=64k sd=mystdd01-01,mystdd02-01
```

Create a Volume. Plex must be attached to a Volume. The command below creates Volume named `myvol01` and attaches the plex `myplex01` with it.

```
# vmake -g stdg<SGN> -u fs gen vol myvol01 plex=myplex01
```

Start the Volume.

```
# vxvol -g stdg<SGN> start myvol01
```

Create a file system on the Volume. Note that the "raw" disk access to the Volume must be used for creating a file system.

```
# newfs /dev/vx/rdisk/stdg<SGN>/myvol01
```

Create a mount point directory and mount the file system. The "block" disk access should be used for mounting the file system. <pre># mkdir /students/<SGN>/LAB2STD # mount /dev/vx/dsk/stdg<SGN>/myvol01 /students/<SGN>/LAB2STD</pre>	3 Add data to the Standard Volumes. The script create 3000 files. <pre># /students/scripts/CreateFiles.sh /students/<SGN>/LAB2STD 3000 -nop &</pre>	4 Identify a pair of BCVs accessible by the Secondary host. Record the information about your assigned BCVs in the table below. # symbcv list pd <table border="1"> <thead> <tr> <th>BCVs</th> <th>PdevName (c##t##s2)</th> <th>SymDevName (eg. 031)</th> </tr> </thead> <tbody> <tr> <td>Device 1</td> <td></td> <td></td> </tr> <tr> <td>Device 2</td> <td></td> <td></td> </tr> </tbody> </table>	BCVs	PdevName (c##t##s2)	SymDevName (eg. 031)	Device 1			Device 2			5 Create the device group, add the standard volume, and associate BCV pairs. <pre># symdg create stdg<SGN> -type regular # symld -g stdg<SGN> add dev STDsymDevName1 # symld -g stdg<SGN> add dev STDsymDevName2 # symbcv -g stdg<SGN> associate dev BCVsymDevName1 # symbcv -g stdg<SGN> associate dev BCVsymDevName2</pre>
BCVs	PdevName (c##t##s2)	SymDevName (eg. 031)										
Device 1												
Device 2												

6	Confirm creation of the device group. <pre># symmir -g stdbg<SGN> query</pre> What is the "State" of your device pairs? Are there any invalid tracks?	
7	Establishing the BCVs. This synchronizes the contents of the BCVs with that of the Standard Volumes. <pre># symmir -g stdbg<SGN> est -full -exact -nop</pre>	
8	Query the status of the devices. Verify synchronization. <pre># symmir -g stdbg<SGN> verify -synched</pre>	
9	Split the BCVs and the Standard Volumes. This split uses PowerPath to briefly hold the I/O to create a consistent image on the BCV volumes <pre># symmir -g stdbg<SGN> split -ppath STDDEVS -instant -nop</pre>	
10	On the secondary host, Import the disk group and mount the file systems <pre># vxkg -C import stdbg<SGN> # vxvol -g stdbg<SGN> startall # mkdir /students/<SGN>/LAB2BCV # mount /dev/vx/dsk/stdbg<SGN>/myvol101 /students/<SGN>/LAB2BCV</pre> Were you able to mount the file system? How many files are in the /students/<SGN>/LAB2BCV directory?	

11	On the secondary host: Remove all testfiles from the BCV volumes. <pre># cd /students/<SGN>/LAB2BCV # rm testfile.*</pre> Unmount the file system and deport the disk group <pre># umount /students/<SGN>/LAB2BCV # vxddg deport stddg<SGN></pre>	
12	Re-establish the pairing between the BCVs and the Standard Volumes. This operation copies the data that has changed on the Standard Volumes since the split, to the BCVs (incremental copy). The changes made to the BCVs will be over written. Verify that the changes made to the BCVs are lost and that the changes to the Standard Volumes are copied over. <pre># symmir -g stddg<SGN> est -nop</pre>	
13	Split the BCV, import the disk group, mount the file system and verify that all 3000 testfiles are back on the BCV volumes On the primary host: <pre># symmir -g stddg<SGN> split -ppath STDDEVs -instant -nop # umount /students/<SGN>/LAB2STD</pre> On the secondary host: <pre># vxddg -C import stddg<SGN> # vxvol -g stddg<SGN> startall # mount /dev/vx/dsk/stddg<SGN>/myvol01 /students/<SGN>/LAB2BCV # cd /students/<SGN>/LAB2BCV # ls -al testfile.* wc -l</pre> How many files are there?	

14

On the secondary host, make changes to the Volume Manager configuration on the BCV Disk Group and add new data to the BCVs.

Create a Subdisk, Plex, and a Volume in the BCV Disk Group. Make a file system on the new Volume, create a mount point directory and mount the new file system. In the commands below `bcv_VM_Disk` is the VM Disk name assigned to your BCV device. This name can be determined from the output of the `vxdisk list` command. The second column of the output gives the VM Disk name. Typically this will be something like `bcv04C` (ie. The string "bcv" prefixed to the Symmetrix Device Name of your BCV device). For example the first command below could be `vxmake -g stddg<SGN> sd bcv04C-02 bcv04C,204800,204800`. This command creates a subdisk of length 100MB, offset by 100MB from the start of the disk. This is because, the first 100MB is already allocated to our first Volume.

```
# vxmake -g stddg<SGN> sd bcv_VM_Disk-02
bcv_VM_Disk,204800,204800
```

```
# vxmake -g stddg<SGN> plex myvol02-01 sd=bcv_VM_Disk-02
```

```
# vxmake -g stddg<SGN> -u fs gen vol myvol02 plex=myvol02-01
```

```
# vxvol -g stddg<SGN> start myvol02
```

```
# newfs /dev/vx/rdisk/stddg<SGN>/myvol02
```

```
# mkdir /students/<SGN>/BCVLAB2FS
```

```
# mount /dev/vx/dsk/stddg<SGN>/myvol02
/students/<SGN>/BCVLAB2FS
```

Add 500 files to the new file system on the BCV Disk Group using the following script.

```
# /students/scripts/CreateFiles.sh
/students/<SGN>/BCVLAB2FS 500 -nop &
```

15	Prepare to restore from the BCVs. As we have changed the Volume Manager configuration on the BCV Disk Group, it recommended that the disk group meta data be flushed to disk afterwards. We will then unmount the file system on the BCV, umount all file systems in the disk group if they are mounted, and deport the diskgroup. On the secondary host: <pre># vxpd flush stdg<SGN> # cd / # umount /students/<SGN>/BCVLAB2FS # umount /students/<SGN>/IAB2BCV # vxpd deport stdg<SGN></pre>	
16	On the primary host, start the TimeFinder restore, import the disk group and mount the Standard Volume File systems. Verify that the new file system and data created on the BCVs is now available on the STD devices. <pre># symmir -g stdg<SGN> restore -nop # vxpd import stdg<SGN> # vxvol -g stdg<SGN> startall # mount /dev/vx/dsk/stdg<SGN>/myvol101 # students/<SGN>/IAB2STD # mkdir /students/<SGN>/BCVLAB2FS # mount /dev/vx/dsk/stdg<SGN>/myvol102 # students/<SGN>/BCVLAB2FS</pre> How many files are in the /students/<SGN>/BCVLAB2FS directory?	
17	Split the BCVs. <pre># symmir -g stdg<SGN> split -nop</pre>	

18	Clean up the server for the next lab group. Unmount file systems, detach the BCVs, destroy the Disk Groups, and remove the mount point directories. On the primary host <pre># cd / # umount /students/<SGN>/LAB2STD # umount /students/<SGN>/BCVLAB2FS # rm -r /students/<SGN>/LAB2STD # rm -r /students/<SGN>/BCVLAB2FS # vxdg deport stdg<SGN></pre> On the secondary host <pre># umount /students/<SGN>/LAB2BCV # rm -r /students/<SGN>/BCVLAB2FS</pre>		
	This concludes TimeFinder Lab 2 (SUN-VERITAS).		

Lab Exercise 3: TimeFinder Clone (SUN-Veritas)

Purpose:	This exercise is designed to lead the student through the setup and operations of the EMC TimeFinder Clone facility. Students will identify devices for use as the Clone Source and Targets. A disk group, Logical Volume and a file system will be created on the Clone Source device. Data will be written to the file system on the Clone Source, a Clone session with a Clone Target will be created and activated. The data on the Clone Target will then be accessed by the same host as the Clone Source.
-----------------	---

Objectives:	At the completion of this lab, using the Veritas Volume Manager and the SYMCLI command set, the students will be able to: ▪ Use Veritas Volume Manager to create a disk group, volumes and file system on a Clone Source Volume. ▪ Create a SYMCLI Device group and add a Clone Source and Clone Target Volumes to the same. ▪ Create, activate, recreate, and restore a Clone session between the Clone Source and Clone Target. ▪ Check the status of the Clone session. ▪ Perform Business Continuity Operations by accessing the data on the Clone Target volumes. ▪ Terminate the active Clone session.
--------------------	---

References:	EMC Solutions Enabler Symmetrix Timefinder CLI - Version 6.0 PRODUCT GUIDE P/N - 300-000-876 - REV A0 PowerPath Product Guidel - Version 4.4 P/N - 300-002-308 - REV A01
--------------------	--

Lab 3 Part 1: TimeFinder Clone Functionality

STEP	ACTION
------	--------

1	Create a Veritas disk group, Volume and UFS filesystem on the Clone Source Volume (Symmetrix Standard Device). Identify the Symmetrix Logical Volumes available to your host. # syminq more
---	--

2

Find the Standard Volume assigned as your Clone Source and complete the table.

Note: Standard Volumes are Symmetrix Logical Volumes which do not have any description under the "Type" column in the output of the syminq command.

Standard Volumes (Clone Source)	PdevName (c#t#d#s2)	SymDevName (eg. 031)
Device 1		

Create a Veritas disk group using your STD Volume. <pre># /etc/vx/bin/vxdisksetup -i c#t#d#</pre> <pre># vxdg init stddg<SGN> cxtvdzs2</pre> Create a volume in the disk group. The following command creates a 200 MB Volume named clonevol <pre># vxassist -g stddg<SGN> make clonevol 200m</pre> Create a UFS file system on the clonevol Volume. The UFS file system is created via the newfs command. Note that the "raw" disk access to the Volume must be used for creating a file system. <pre># newfs /dev/vx/rdisk/stddg<SGN>/clonevol</pre>	
---	--

Create a mount point directory and mount the file system. Note that the "block" disk access to the Volume must be used while mounting the file system. <pre># mkdir /students/<SGN>/clonefs # mount /dev/vx/dsk/stdg<SGN>/clonevol /students/<SGN>/clonefs</pre> Add data to the Filesystem on the Clone Source. The script generates 1000 files <pre># /students/scripts/createfiles.sh /students/<SGN>/clonefs 1000 -nop &</pre> Perform a listing of the files on the file system to confirm that you have created the desired files. <pre># ls -l /students/<SGN>/clonefs</pre> How many files are created?	3 On the Secondary host, identify the Device assigned as your Clone Target. Record the information about your assigned Clone Target in the table below. Note that in this example the Clone Target is a Standard Device. BCV's and SRDF devices can also be used as Clone Targets. # syminfo more	<table border="1"> <tr> <td>Clone Target</td><td>PdevName (c##d##s2)</td><td>SymDevName (eg. 031)</td></tr> <tr> <td>Device 1</td><td></td><td></td></tr> </table>	Clone Target	PdevName (c##d##s2)	SymDevName (eg. 031)	Device 1		
Clone Target	PdevName (c##d##s2)	SymDevName (eg. 031)						
Device 1								

4	Create a SYMCLI Device Group and add your Clone Source and Target <pre># symdg -type REGULAR create stdbg<SGN> # symld -g stdbg<SGN> add dev <CloneSourceSymDevName> # symld -g stdbg<SGN> add dev <CloneTargetSymDevName></pre> The Clone Source and Target SymDevNames are three digit numbers that were recorded earlier in this lab.									
5	Identify the Logical Device Names of the Clone Source and Target Volumes. A default logical name (ld) is assigned to each Symmetrix device when it is added or associated with a Device Group. This logical name can be changed if desired. Identify the Logical Device Names assigned to your Clone source and target volumes. <pre># symld -g stdbg<SGN> list</pre>									
	<table><tr><td>Device</td><td>Logical Name (eg DEV001)</td><td>SymDevName (eg. 031)</td></tr><tr><td>Clone Source</td><td></td><td></td></tr><tr><td>Clone Target</td><td></td><td></td></tr></table>	Device	Logical Name (eg DEV001)	SymDevName (eg. 031)	Clone Source			Clone Target		
Device	Logical Name (eg DEV001)	SymDevName (eg. 031)								
Clone Source										
Clone Target										

6	Create a Clone Session. This creates a Clone session between the Clone Source and Clone Target. Creating a Clone session makes the Clone Target "Not-Ready" and also puts a "hold" on that device so that the Target cannot participate in other copy activities like TimeFinder, Snap or Clone. <pre># symclone -g stdbg<SGN> create <CloneSourceLogicalDevName> sym ld <CloneTargetLogicalDevName> -differential -precopy</pre> What is the purpose of the -precopy option? _____ What is the difference between the Clone create command with and without the "-differential" option? _____ <pre># symclone -g stdbg<SGN> query</pre> What is the "State" of this Clone session? _____ List the number of Protected Tracks on the Source Volume. _____ <pre># symld -g stdbg<SGN> -held list -v</pre> The command lists the devices in the device group that have a "hold" on them. What is the Device Status of the Clone Target device? _____ Why is the Target device placed in a "Not-ready" state after the creation of a Clone Session? _____
---	--

7	On the Primary host, activate the Clone Session. <pre># symclone -g stdag<SGN> activate <SrcLogicalDevName> sym_id <TgtLogicalDevName> -consistent</pre> # symclone -g stdag<SGN> query What is the state of the Clone Session? _____ List the number of Protected Tracks on the Source Volume. _____ <pre># symld -g stdag<SGN> -held list</pre> What is the Device Status of the Clone Target device? _____ Did the Device Status change from the previous step? _____ What is the purpose of –consistent option? _____
8	Create additional testfiles on the Source Volume Filesystem on the Primary host. <pre># /students/scripts/createFiles.sh /students/<SGN>/clonefs 500 -nop &</pre> Perform a listing of the files on the file system to confirm that you have created the desired files. <pre># ls -l /students/<SGN>/clonefs wc -l</pre> How many files are in the directory? _____
9	Accessing the data on the Clone Target volume from the secondary host On the secondary host, import the disk group and start the volume. <pre># vxdbg -C import stdag<SGN> # vxvol -g stdag<SGN> start clonevol</pre>

10	Mount the file systems Clone Target Device and verify that the data is accessible. On the secondary host: <pre># mkdir /students/<SGN>/clonefs</pre> Remember that the Source file system was mounted at the time the Clone Session was activated. Thus a mount of the file system on the Clone device may fail and a "fsck" on the Target volume might be necessary. We will perform an fsck just to be sure. <pre># fsck -Y /dev/vx/rdisk/stdg<SGN>/clonevol1</pre> <pre># mount /dev/vx/disk/stdg<SGN>/clonevol1 /students/<SGN>/clonefs</pre> <pre># ls -l /students/<SGN>/clonefs</pre> Do you see any of the new test files on the secondary host? _____ Should you see the new test files on the secondary host? _____ On the primary host: <pre># ls -l /students/<SGN>/clonefs</pre> How many files are in the directory? _____
----	--

11	Recreate and activate the clone session to incrementally copy all subsequent changes made to the source device (made after the point-in-time copy initiated) to the target device. On the secondary host, unmount the file system and deport the disk group. <pre># umount /students/<SGN>/clonefs # vxkg deport stddg<SGN></pre> On the primary host, recreate and activate the clone session. <pre># symclone -g stddg<SGN> recreate <CloneSourceLogicalDevName>sym Id <CloneTargetLogicalDevName> # symclone -g stddg<SGN> activate <CloneSourceLogicalDevName> > sym Id <CloneTargetLogicalDevName></pre>
----	---

12	Accessing the data on the Clone Target volume again on the secondary host On the secondary host: Import the disk group and start the volume <pre># vxvg -C import stdg<SGN> # vxvol -g stdg<SGN> start clonevol</pre> Mount the file systems Clone Target Device and verify that the data is accessible. <pre># mkdir /students/<SGN>/clonefs</pre> Remember that the Source file system was mounted at the time the Clone Session was activated. Thus a mount of the file system on the Clone device may fail and a "fsck" on the Target volume might be necessary. <pre># fsck -Y /dev/vx/r<sk>/stdg<SGN>/clonevol # mount /dev/vx/<sk>/stdg<SGN>/clonevol /students/<SGN>/clonefs # ls -l /students/<SGN>/clonefs</pre> How many files are in the directory? _____
----	--

13	Restore the data on clone to source volume. The session must have been created with –differential for incremental restore On the secondary host: Create additional 200 files in the /students/<SGN>/clonefs directory <pre>#/students/scripts/CreateFiles.sh /students/<SGN>/clonefs 200 -nop</pre> Wait until all 200 files have been created. On the primary host: Restore the clone session, import the disk group, and mount the file system Unmount the file system and deport the disk group <pre># umount /students/<SGN>/clonefs # vxvg deport stdg<SGN></pre> Start the restore <pre># symclone -g stdg<SGN> restore DEV001 sym ld DEV002 -nop</pre> Import the disk group back and re-mount the file system <pre># vxvg -C import stdg<SGN> # vxvol -g stdg<SGN> start clonevol # mount /dev/vx/dsk/<SGN>/clonevol /students/<SGN>/clonefs</pre> How many files are in the /students/<SGN>/clonefs directory?
14	Terminate the Clone Session. We will terminate the Clone session while the file system on the Target is still mounted. <pre># symclone -g stdg<SGN> terminate DEV001 sym ld DEV002 # symclone -g stdg<SGN> query # symld -g stdg<SGN> -held list</pre> Is there a “hold” on the Clone Target device?

15	Lab Cleanup: We will now clean up the server for the next lab group. On the primary host: <pre># umount /students/<SGN>/clonefs # symdg delete stdg<SGN>-force # vxdg deport stdg<SGN> # rm -r /students/<SGN>/clonefs</pre> On the secondary host: <pre># umount /students/<SGN>/clonefs # vxdg deport stdg<SGN> # rm -r /students/<SGN>/clonefs</pre>		
			This concludes the TimeFinder Lab 3 – Clone (SUN-Veritas).

Lab Exercise 4: TimeFinder Snap (Sun-Veritas)

Purpose: This exercise is designed to lead the student through the setup and operations of the EMC TimeFinder Snap facility. Students will identify devices for use as the Source and Snap Targets.

Objectives: At the completion of this exercise, using the Veritas Volume Manager and SYMCLI command set the students will be able to:

- Use Veritas Volume Manager to create a disk group, logical volume and file system on a source volume.
- Create a SYMCLI Device group and add a Source, and Snap Target Volumes.
- Create and use multiple save device pools
- Monitor save devices
- Create and activate a Snap session between the Source and Snap Target.
- Check the status of the Snap session.
- Perform Business Continuity Operations by accessing the data on the Snap volumes.
- Terminate the active Snap session.

References: EMC Solutions Enabler Symmetrix Timefinder CLI - Version 6.0 PRODUCT GUIDE P/N - 300-000-876 - REV A0

PowerPath Product Guidel - Version 4.4 P/N - 300-002-308 - REV A01

Lab 4 Part 1: TimeFinder Snap - Configuring Additional Save Pools.

STEP	ACTION
------	--------

1	Contact the instructor for the save pool devices to be used. New Save Pool Name _____ Symmetrix Device ID (Save Devices) _____
2	List the current save pools on the Symmetrix <pre># symsnap -pools list</pre> How Many save pools are there ? _____
3	List the save devices in the "DEFAULT_POOL" – save pool <pre># symsnap -sid <your symmetrix ID> show pool DEFAULT_POOL</pre> Are any of the save devices in an active state ? _____
4	In this step, the save device in the DEFAULT_POOL assigned to you will be disabled . Create a text file, <code>disable<SGN>.txt</code>, containing the following line: <pre>disable dev <Your Save Device1>:<Your Save Device2> in pool DEFAULT_POOL, type = SAVEDEV;</pre> Use the text file with <code>symconfigure</code> command. <pre># symconfigure -sid <Your Symmetrix ID> -f disable<SGN>.txt commit -nop -v</pre> Did the <code>symconfigure</code> command completed successfully? _____

5	Create a new save pool Create a text file, create_pool<SGN>.txt, containing the following line: <pre>create pool pool<SGN>, type = SAVEDEV;</pre> Use the text file with symconfigure command <pre># symconfigure -sid <Your Symmetrix ID> -f create_pool<SGN>.txt commit -nop</pre> Verify that a new save pool was created. <pre># symsnap -sid <Your Symmetrix ID> -pools list</pre> What is the state of the pool? _____
6	Add the disabled save device to the new save pool and enable it Create a text file, adddev<SGN>.txt, containing the following line: <pre>add dev <Your Save Device1>:<Your Save Device2>to pool pool<SGN>, type = SAVEDEV, member_state = ENABLE;</pre> Use the adddev<SGN>.txt file with the symconfigure command <pre># symconfigure -sid <Your Symmetrix ID> -f adddev<SGN>.txt commit -nop</pre> Verify that the device was added to the new pool and enabled. <pre># symsnap -sid <Your Symmetrix ID> show pool pool<SGN></pre> What is the state of the pool? _____ What is the state of the device? _____

Lab 4 Part 2: TimeFinder Snap Functionality

STEP	ACTION
------	--------

1	Create a Veritas disk group, Volume and UFS filesystem on the Source Volume (Symmetrix Standard Device). Identify the Symmetrix Logical Volumes available to the primary host. # syminq more Find the Standard Volume assigned as your Source and complete the table. # sympd list	Note: Standard Volumes are Symmetrix Logical Volumes which do not have any description under the "Type" column in the output of the syminq command. <table><tr><th>Standard Volumes (Source)</th><th>PdevName (c#t#d#s2)</th><th>SymDevName (eg. 031)</th></tr><tr><td>Device 1</td><td></td><td></td></tr></table>	Standard Volumes (Source)	PdevName (c#t#d#s2)	SymDevName (eg. 031)	Device 1			Create a veritas disk group with the device . The following command creates a Veritas disk group with a standard device . <pre># /etc/vx/bin/vxdisksetup -i c#t#d# # vxdg init stdbg<SGN> c#t#d#s2 # vxassist -g stdbg<SGN> make snapsrvcvol 400m MB Volume named snapsrvcvol</pre> Create a volume in the disk group. The following command creates a 400	
Standard Volumes (Source)	PdevName (c#t#d#s2)	SymDevName (eg. 031)								
Device 1										

Create a UFS file system on the snapsrvol Volume. The UFS file system is created via the newfs command. Note that the "raw" disk access to the Volume must be used for creating a file system. <pre># newfs /dev/vx/rdisk/stdg<SGN>/snapsrvol</pre> Create a mount point directory, mount the file system, and add 50 testfiles <pre># mkdir /students/<SGN>/sunlab4</pre> <pre># mount /dev/vx/disk/stdg<SGN>/snapsrvol /students/<SGN>/students/<SGN>/sunlab4</pre> <pre># /students/scripts/CreateFiles.sh /students/<SGN>/sunlab4 50 -nop &</pre>	2 On the Secondary host, identify the Device assigned as your Snap device. Record the information about your assigned Virtual and BCV devices in the table below. <pre># sympl list grep VDEV</pre>	<table><tr><td></td><td>Device Type</td><td>PdevName (c##d##s2)</td><td>SymDevName (eg. 031)</td></tr><tr><td></td><td>Virtual device#1</td><td></td><td></td></tr></table>		Device Type	PdevName (c##d##s2)	SymDevName (eg. 031)		Virtual device#1			3 Create a SYMCLI Device Group and add the Source, BCV, and Virtual Device as members of the Device Group. Then, establish the BCV mirror. Create the device group and add source, and Virtual devices <pre># symdg -type REGULAR create stdg<SGN></pre> <pre># symld -g stdg<SGN> add dev <SourceSymDevName></pre> <pre># symld -g stdg<SGN> add dev <VirtualDevSymDevName></pre> The Source, and VirtualDev SymDevNames are three digit numbers that were recorded earlier in this lab.
	Device Type	PdevName (c##d##s2)	SymDevName (eg. 031)								
	Virtual device#1										

4	List the save pools. Snap session can then be assigned to a specific pool based on the usage pattern. This improves performance of snap volumes. <pre># symsnap list -pools</pre> How many save device pools are there? _____
5	Create a snap session using the save pool you have created. <pre># symsnap -g stddg<SGN> create -svp pool<SGN> <LogicalSourceSymDevName> sym DEV <VirtualDevSymDevName> -nop</pre> <pre># symsnap -g stddg<SGN> -pools query</pre> What is the status of the SRC⇌TGT column? _____ What the name of the save device pool used by this snap session? _____
6	Activate the snap session. <pre># symsnap -g stddg<SGN> activate <LogicalSourceSymDevName> sym Id <LogicalVirtualDevSymDevName> -nop</pre> <pre># symsnap -g stddg<SGN> query</pre> What is the status of the SRC⇌TGT column? _____

7	On the primary host, monitor the usage of save device pool Create a text file, /students/<SGN>/monitor.sh, action script containing the following lines: <pre>#/bin/sh echo "echo reached 10 percent">/students/<SGN>/monitor.out # chmod +x /students/<SGN>/monitor.sh</pre> Make the file executable Execute the monitoring command. <pre># symsnap -sid <SYM> monitor -percent 10 -i 5 -action "/students/<SGN>/monitor.sh" -svp pool<SGN> -norepeat &</pre>
8	On the secondary host , import the disk group, mount the file systems on the snap volume and verify that the data is accessible. <pre># vxvg -C import stddg<SGN> # vxvol -g <SGN> start snapsrcvol # mkdir /students/<SGN>/sunlab4_snap # fsck -Y /dev/vx/rdisk/stddg<SGN>/snapsrcvol # mount /dev/vx/dsk/stddg<SGN>/snapsrcvol /students/<SGN>/sunlab4_snap # ls -l /students/<SGN>/sunlab4_snap On the primary host: # ls -l /students/<SGN>/sunlab4 Are the contents of the two directories identical ?</pre>

9	On the primary host, create 500 test files on the file system on the source. <pre>/students/scripts/CreateFiles.sh /students/<SGN>/sunlab4 500 -nop &</pre> Perform a listing of the files on the file system to confirm that you have created the desired files. <pre># ls -l /students/<SGN>/sunlab4 wc -l</pre> How many files are in the directory? _____
10	Investigate the Save devices <pre># symsnap list -savedevs -svp pool<SGN></pre> What is the total used tracks on all save devices? _____ What is % Full of all save devices? _____ If the % Full is 10 percent or higher, check if the <code>/students/<SGN>/monitor.out</code> file was created by the <code>symbolsnap monitor</code> command. Was the file created? _____
11	On the secondary host, create 700 test files on the snap volume. <pre># /students/scripts/CreateFiles.sh /students/<SGN> /sunlab4_snap 700 -nop &</pre> <pre># symsnap list -savedevs -svp pool<SGN></pre> What is the total used tracks on all save devices? _____ How many files are in the <code>/students/<SGN>/hplab4_snap</code> directory? _____

12	Restore the virtual device to the standard device On the secondary host: <pre># umount /students/<SGN>/sunlab4_snap # vxkg deport stdg<SGN></pre> Unmount the file system and deport the disk group On the primary host: <pre># umount /students/<SGN>/sunlab4 # vxkg deport stdg<SGN></pre> Unmount the file system and deport the disk group Start the restore and verify the status of the snap sessions <pre># symsnap -g stdg<SGN> restore -nop # symsnap -g stdg<SGN> query -multi</pre> Note: the original snap session remains and the restore creates an additional snap session. The restore leaves the VDEV in a NOT READY state. What are the status of the 2 snap sessions?
----	---

13	Verify the contents of both source and target are identical On the primary host: Import the disk group, mount the file system, and list the contents <pre># vxkg import snapvg<SGN> # mount /dev/vx/dsk/<SGN>/snapsrcvol /students/<SGN>/sunlab4 # ls -la /students/<SGN>/sunlab4 wc -l</pre> How many files are in the students/<SGN>/sunlab4 directory? _____ Is the number of files equal to that in Step 11? _____
14	Terminate the Snap Session. The restore session must first be terminated using the -restore flag. On the primary host: <pre># symsnap -g stdg<SGN> terminate <LogicalSourceSymDevName> sym ld <LogicalVirtualDevSymDevName> -restore # symsnap -g stdg<SGN> terminate <LogicalSourceSymDevName> sym ld <LogicalVirtualDevSymDevName></pre> Note: The restore session must first be terminated using the -restore flag.

15	Lab Cleanup: We will now clean up the server for the next lab group. On the primary host: <pre># umount /students/<SGN>/sunlab4 # rm -r /students/<SGN>/sunlab4 # vxkg deport stdg<SGN> # symdg delete stdg<SGN> -force</pre> On the secondary host: <pre># umount /students/<SGN>/sunlab4_snap # rm -r /students/<SGN>/sunlab4_snap # vxkg deport stdg<SGN></pre>	This concludes the TimeFinder – Snap (SUN-VERITAS) Lab.
----	--	---

File - BCL LG_Sun_V21a.Doc
Date - 26-Jul-2005
Ver - 2.1a

Symmetrix Business Continuity - TimeFinder Solutions Lab Guide – HP-UX



Global Education and Productivity

**26-Jul-2005
Ver-2.1a**

Copyright :

Copyright © 1996, 2000, 2001, 2002, 2003, 2004, 2005 EMC Corporation. All Rights Reserved.

EMC believes the information in this publication is accurate as of its publication date. The information is subject to change without notice.

THE INFORMATION IN THIS PUBLICATION IS PROVIDED "AS IS." EMC CORPORATION MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WITH RESPECT TO THE INFORMATION IN THIS PUBLICATION, AND SPECIFICALLY DISCLAIMS IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Use, copying, and distribution of any EMC software described in this publication requires an applicable software license.

EMC, ICDA® (Integrated Cached Disk Array), and EMC²® (the EMC logo), and Symmetrix®, are registered trademarks of EMC Corporation. EMC™ and SRDF™ are trademarks of EMC Corporation.

All other trademarks used herein are the property of their respective owners.

Trademark Information:

EMC Trademarks	EMC?, EMC, Symmetrix, Celerra, CLARiiON, CLARAlert, Connectix, Dantz, Documentum, HighRoad, Legato, Navisphere, PowerPath, ResourcePak, SnapView/IP, SRDF, TimeFinder, VisualSAN, and where information lives are registered trademarks and EMC Automated Networked Storage, EMC ControlCenter, EMC Developers Program, EMC OnCourse, EMC Proven, EMC Snap, Access Logix, AutoAdvice, Automated Resource Manager, AutoSwap, AVALONidm, C-Clip, Celerra Replicator, Centera, CLARevent, CopyCross, CopyPoint, DatabaseXtender, Direct Matrix, Direct Matrix Architecture, EDM, E-Lab, Enginuity, FarPoint, FLARE, GeoSpan, InfoMover, MirrorView, NetWin, OnAlert, OpenScale, Powerlink, PowerVolume, Replicare, SafeLine, SAN Architect, SAN Copy, SAN Manager, SDMS, SnapSure, SnapView, StorageScope, SupportMate, SymmAPI, SymmEnabler, Symmetrix DMX, Universal Data Tone, and VisualSRM are trademarks of EMC Corporation. All other trademarks used herein are the property of their respective owners.
Third Party Trademarks	AIX is a registered trademark of International Business Machines Corporation. Brocade, SilkWorm, SilkWorm Express, and the Brocade logo are trademarks or registered trademarks of Brocade Communications Systems, Inc., in the United States and/or in other countries. Compaq and the names of Compaq products referenced herein are either trademarks and/or service marks or registered trademarks and/or service marks of Compaq. Hewlett-Packard, HP, HP-UX, OpenView, and OmniBack are trademarks, or registered trademarks of Hewlett-Packard Company. McDATA, the McDATA logo, and ES-2500 are registered trademarks of McDATA Corporation. Microsoft, Windows, and Windows NT are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. NobleNet is a registered trademark of Rogue Wave Software, Inc. SANbox is a trademark of QLogic Corporation. Sun, Sun Microsystems, the Sun Logo, SunOS and all Sun-based trademarks and logos, Java, the Java Coffee Cup Logo, and all Java-based trademarks and logos, Solaris, and NFS, are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries. UNIX is a registered trademark of The Open Group.

Document Revision History:

Rev #	File Name	Date	Activity / Note
1.0a	BCL_LG_HP_V10a.Doc	05-May-05	
1.1a	BCL_LG_HP_V11a.Doc	25-May-05	Update 2
1.1a	BCL_LG_HP_V11a.Doc	15-Jun-05	Update 3
1.1a	BCL_LG_HP_V11a.Doc	05-Jul-05	Update 4
			Added Create
			Save Pool
2.1a	BCL_LG_HP_V21a.Doc	26-Jul-05	Changed Name

Table of Contents: - For Symmetrix Business Continuity – TimeFinder Solutions

Copyright :	2
Trademark Information:	3
Document Revision History:	4
Table of Contents: - For Symmetrix Business Continuity – TimeFinder Solutions	5
Lab Exercise 1: TimeFinder Set up Lab.	6
Lab 1 Part 1: Verify your setup Environment.	7
Lab Exercise 2: TimeFinder Integration Lab.	13
Lab 2 Part 1: TimeFinder – HP-UX LVM Integration Lab.	14
Lab Exercise 3: TimeFinder Clone (HP-UX LVM)	23
Lab 3 Part 1: TimeFinder Clone Functionality.	24
Lab Exercise 4: TimeFinder Snap (HP-UX LVM)	34
Lab 4 Part 1: TimeFinder Snap - Configuring Additional Save Pools.	35
Lab 4 Part 2: TimeFinder Snap Functionality.	37

Notes:

SGN is the abbreviation for **Student Group Number**. This is the number that will be assigned to you by the instructor before starting the lab exercise.

Lab Exercise 1: TimeFinder Set up Lab.

Purpose:

In this lab exercise students will explore the configuration and basic setup of the Symmetrix TimeFinder facility using the EMC Solutions Enabler Command Line Interface (SYMCLI). Business Continuance Volumes (BCVs) will be identified and associated with a SYMCLI Device Group of type REGULAR. The contents of the Standard Volumes will be copied over to the BCVs. After synchronization, the BCVs will be split from the Standard Volumes, giving a point-in-time copy of the data.

After completion of this exercise, participants will be able to:

1. Display and identify the configured BCVs.
2. Create a SYMCLI device group of type REGULAR.
3. Associate BCVs with a SYMCLI device group.
4. Synchronize the BCVs with the Standard Volumes belonging to the SYMCLI device group.
5. Monitor the status and synchronization of the BCVs
6. Split the BCVs from the Standard Volumes to obtain point-in-time copy of the data.

References:

EMC Solutions Enabler Symmetrix Timefinder CLI - Version 6.0 PRODUCT GUIDE P/N - 300-000-876 - REV A06

Lab 1 Part 1: Verify your setup Environment.

STEP	ACTION
------	--------

1	Verify or setup your user environment for SYMCLI. Check to see if the directory for SYMCLI executables is included in your PATH variable and if not set, and export it. <pre># echo \$PATH</pre> Is the directory /usr/symcli/bin included in the PATH? <u>If not add /usr/symcli/bin to your PATH and export it, using the commands below.</u> <pre># PATH=\$PATH:/usr/symcli/bin # export PATH</pre> Check to see if the directory for SYMCLI manual pages is included in your MANPATH variable and if not, set and export it. <pre># echo \$MANPATH</pre> Is the directory /usr/symcli/man included in the MANPATH? <u>If not add /usr/symcli/man to your PATH and export it, using the commands below.</u> <pre># MANPATH=\$MANPATH:/usr/symcli/man # export MANPATH</pre>
2	Initialize the SYMAPI Database - (The file) <pre>/var/symapi/db/sympi_db.bin) # symctg discover</pre>

3	1. Identify the volumes accessible to your host. This and following TimeFinder exercises require two standard volumes and two BCV volumes. In the output of the <code>syminq</code> command, locate the devices assigned to you and complete the table below. a) List all the volumes available to your host. # syminq more Identify the two Standard Volumes and record the host device names (PdevName) and the Symmetrix device names (SymDevName) under the "Type" column.	<table border="1"> <thead> <tr> <th>Standard Volumes</th> <th>PdevName (c##d##s2)</th> <th>SymDevName (eg. 031)</th> </tr> </thead> <tbody> <tr> <td>Device 1</td> <td></td> <td></td> </tr> <tr> <td>Device 2</td> <td></td> <td></td> </tr> </tbody> </table>	Standard Volumes	PdevName (c##d##s2)	SymDevName (eg. 031)	Device 1			Device 2			b) List the BCVs configured on the Symmetrix and identify and record the pair of BCVs assigned to you. i) Display summary information about all BCVs in the attached Symmetrix. # symbcv list dev more ii) Display summary information about all BCVs accessible to your host. # symbcv list pd more iii) Identify the two BCV volumes assigned to you, and record the host device names (PdevName), the Symmetrix device names (SymDevName), attribute, and status in the table below.
Standard Volumes	PdevName (c##d##s2)	SymDevName (eg. 031)										
Device 1												
Device 2												

BCVs	PdevName (c#t#d#s2)	SymDevName (eg. 03C)	Attribute	Status
Device 1				
Device 2				

4

Create a device group, add standard devices and associate BCV devices. The device group type should be of type REGULAR.

Your device group name: _____

a) Create a regular device group with the name stddg<SGN>.

```
# symdg -type REGULAR create stddg<SGN>
```

b) Add the two standard devices to the device group. Add the first device using its Physical Device Name (c#t#d#s2), and the second device using its Symmetric Device Name (eg. 088).

```
# symld -g stddg<SGN> add pd PdevName
```

```
# symld -g stddg<SGN> add dev SymDevName
```

c) Verify that the device group was added to the SYMAPI database

```
# symdg list
```

d) Show detailed information about your device group and record the Logical Device Names (LdevName) given to your two Standard Volumes.

```
# symdg show stddg<SGN> | more
```

Std. Volumes	PdevName	SymDevName	LdevName
Device 1			
Device 2			

	2) Associate the BCVs you identified, with the device group stddg<SGN>. <pre># symbcv -g stddg<SGN> associate dev SymDevName</pre> <pre># symbcv -g stddg<SGN> associate dev SymDevName</pre> a) Verify association and record the Logical Device name given to your BCVs. <pre># symdg show stddg<SGN> more</pre>												
<table border="1"> <thead> <tr> <th>BCVs</th> <th>PdevName</th> <th>SymDevName</th> <th>LdevName</th> </tr> </thead> <tbody> <tr> <td>Device 1</td> <td></td> <td></td> <td></td> </tr> <tr> <td>Device 2</td> <td></td> <td></td> <td></td> </tr> </tbody> </table>	BCVs	PdevName	SymDevName	LdevName	Device 1				Device 2				b) Set and export the variable that holds the value of your device group name. If this variable is set, the device group name need not be specified in the command line for subsequent steps. Note: Variables are case sensitive. <pre># export SYMCLI_DG=stddg<SGN></pre>
BCVs	PdevName	SymDevName	LdevName										
Device 1													
Device 2													
5	Query the BCV devices in your device group. Because you set the environment variable SYMCLI_DG, it is not necessary to specify the device group with the -g flag. <pre># symmlr query</pre> Did the command run successfully? _____												

6	Establish the pairing between the BCVs and the Standard Volumes. This synchronizes the contents of the BCVs with those of the Standard Volumes. If this is the first ever establish operation between the BCVs and the Standard Volumes, the "-full" option may have to be specified. This is because, the default is an incremental establish. c) Establish a BCV with each standard device in your device group. Use the -full and -opt options. # symmir establish -full -opt d) Query the status of the BCV devices. Obtain the synchronization rate, estimated time to completion. The command below repeats the query every 5 seconds, 10 times. # symmir query -i 5 -c 10 Synchronization rate: _____ Estimated time to completion: _____ BCV Device Status: _____
---	--

7	Split the BCV. A BCV must be fully synchronized before it can be split. e) Before synchronization is complete, attempt to split the BCV pairs. <pre># symmir split</pre> Did the split action succeed? _____ f) Run the "query" again and when the synchronization completes, attempt the split again. <pre># symmir query -i 5 # symmir split # symmir query</pre> You now have a point-in-time copy of your "data"! g) Reestablish your BCV you split above and again query your devices. <pre># symmir establish # symmir query</pre> Why did this establish complete so quickly? _____
8	TimeFinder Lab 2 will explore the use of a logical volume manager to present BCV volumes to the same host. Before proceeding to the next lab, ensure that the Standard-BCV pairs are split. Split the BCV and delete the SYMCLI device group "stdg<SGN>" that you created <pre># symmir split # symdg -force delete stdg<SGN></pre> This concludes TimeFinder Lab 1.

Lab Exercise 2: TimeFinder Integration Lab

Purpose:	In this lab exercise students will learn to manipulate the BCV Volume Group / Logical Volumes / File systems, and the Standard Volume Logical/Volumes and File systems..
Objectives:	After completion of this exercise, the students will be able to ▪ Establish the BCVs with the Standard Volumes and verify that they are fully synchronized. ▪ Split the BCVs and the Standard Volumes. ▪ Perform Business Continence Operations. ▪ Re-establish the BCVs with the Standard Volumes. ▪ Restore from the BCVs to the Standard Volumes after performing Business Continence Operations.
References:	EMC Solutions Enabler Symmetrix Timefinder CLI - Version 6.0 PRODUCT GUIDE P/N - 300-000-876 - REV A06

Lab 2 Part 1: TimeFinder – HP-UX LVM Integration Lab

STEP	ACTION
------	--------

1	Create an HP-UX Volume Group / Logical Volume and a filesystem on your assigned Standard Volumes. Identify the Symmetrix Logical Volumes available to your host. <pre># syminq more Or # syminq > dev_list.txt</pre> Find the two Standard Volumes assigned to you and complete the tables. Ensure that the Volumes you have chosen are not already a part of a HP-UX Volume Group. <pre># ioscan -fnc disk > dev_list2.txt</pre>												
2	Note: Standard Volumes are those which do not have any description under the "Type" column in the output of the syminq command. <table><tr><td>Standard Volumes</td><td>PdevName (c#t#d#)</td><td>SymDevName (eg. 031)</td></tr><tr><td>Device 1</td><td></td><td></td></tr><tr><td>Device 2</td><td></td><td></td></tr><tr><td></td><td></td><td></td></tr></table>	Standard Volumes	PdevName (c#t#d#)	SymDevName (eg. 031)	Device 1			Device 2					
Standard Volumes	PdevName (c#t#d#)	SymDevName (eg. 031)											
Device 1													
Device 2													

3	Create an HP-UX Volume Group. Create the directory /dev/stdvg<SGN> with the character special file called group <pre># mkdir /dev/stdvg<SGN></pre> <pre># mknod /dev/stdvg<SGN>/group c 64 0x01000</pre> The minor number for the group file should be unique among all the volume groups on the system. It has the format 0xNN0000, where <i>NN</i> runs from 00 to 09. The maximum value of <i>NN</i> is controlled by the kernel tunable parameter <i>maxvgs</i>. The default for <i>maxvgs</i> is 10 and the maximum is 256. Verify the minor number of Initialize the disks <pre># pvcreate /dev/rdisk/cxtxdx</pre> <pre># pvcreate /dev/rdisk/cxtxdx</pre> Create, extend, and verify the information of a Volume Group <pre># vgcreate stdvg<SGN> /dev/dsk/c#t#d#</pre> <pre># vgextend stdvg<SGN> /dev/dsk/c#t#d#</pre> <pre># vgdisklay -v stdvg<SGN></pre> Create a Logical Volume (Striped across both disks.) (Note - To stripe across your two disks, and you decide on a stripe size of 32 kilobytes, and a logical volume size is 300 megabytes - to create the striped logical volume, you would enter.) <pre># lvcreate -i 2 -I 32 -L 300 -n stdlv<SGN> stdvg<SGN></pre> <pre># vgdisklay -v stdvg<SGN></pre> Create a file system on the Volume. Note that the "raw" disk access to the Volume must be used for creating a file system. <pre># newfs /dev/stdvg<SGN>/rstdlv<SGN></pre>
---	---

4	Create a mount point directory and mount the file system. The "block" disk access should be used for mounting the file system. <pre># mkdir -p /students/<SGN>/LAB2STD # mount /dev/stdvg<SGN>/stdlv<SGN> /students/<SGN>/LAB2STD</pre> List the mounted file systems <pre># bdf</pre>										
5	Add data to the Standard Volumes. The script create 3000 files. <pre># /students/scripts/CreateFiles.sh /students/<SGN>/LAB2STD 3000 -nop &</pre>										
6	Identify a pair of BCVs on the same Symmetrix that are accessible by the secondary host. Record the information about your assigned BCVs in the table below. <pre># symbcv list pd</pre> <table border="1"> <thead> <tr> <th>BCVs</th><th>PdevName (c#t#d#s2)</th><th>SymDevName (eg. 031)</th></tr> </thead> <tbody> <tr> <td>Device 1</td><td></td><td></td></tr> <tr> <td>Device 2</td><td></td><td></td></tr> </tbody> </table>	BCVs	PdevName (c#t#d#s2)	SymDevName (eg. 031)	Device 1			Device 2			
BCVs	PdevName (c#t#d#s2)	SymDevName (eg. 031)									
Device 1											
Device 2											

7	Create the device group, add the standard volume, and associate BCV pairs. <pre># symdg create stdbg<SGN> -type regular # symld -g stdbg<SGN> add dev <STDsymDevName1> # symld -g stdbg<SGN> add dev <STDsymDevName2> # symbcv -g stdbg<SGN> associate <dev BCVsymDevName1> # symbcv -g stdbg<SGN> associate dev <BCVsymDevName2></pre>	8	Confirm the creation of the device group. <pre># symmir -g stdbg<SGN> query</pre> What is the "State" of your device pairs? _____ Are there any invalid tracks? _____	9	Establishing the BCVs. This synchronizes the contents of the BCVs with that of the Standard Volumes. <pre># symmir -g stdbg<SGN> est -full -exact -nop</pre>	10	Query the status of the devices. Verify synchronization. <pre># symmir-g stdbg<SGN> verify -synchronized</pre>	11	Split the BCVs and the Standard Volumes. This split uses PowerPath to briefly hold the I/O to create a consistent image on the BCV volumes <pre># symmir -g stdbg<SGN> split -ppath STDDEVs -instant -nop</pre>
---	---	---	---	---	--	----	--	----	---

12	Accessing data on the BCL from the Secondary host. On the Primary Host: Create a map file. The vxexport creates a mapfile. With the -p option the command will not export the volume group. However, you must unmount the file system and deactivate the volume group before running vxexport command. <pre># unmount /students/<SGN>/LAB2STD # vgchange -a n stdvg<SGN> # vxexport -p -m /stdvg<SGN>.map stdvg<SGN> Copy to the Secondary Host # rcp /stdvg<SGN>.map <secondary_host>: /</pre> On the Secondary Host: Create the directory /dev/bcvgBCLLab2<SGN> with the character special file called group <pre># mkdir /dev/bcvgBCLLab2<SGN> # mknode /dev/bcvgBCLLab2<SGN> c 64 0x01000</pre> The minor number for the group file should be unique among all the volume groups on the system. Import the volume group using the mapfile <pre># vgimport -m /stdvg<SGN>.map bcvgBCLLab2<SGN> /dev/dsk/c#t#d# /dev/dsk/c#t#d#</pre> Activate the Volume Group and mount the file system <pre># vgchange -a y bcvgBCLLab2<SGN> # mkdir -p /students/<SGN>/LAB2BCV # fsck -p /dev/ bcvgBCLLab2<SGN>/rstdlv<SGN> # mount /dev/bcvgBCLLab2<SGN>/rstdlv<SGN> /students/<SGN>/LAB2BCV</pre> How many files are in the /students/<SGN>/LAB2BCV directory? _____
----	---