



Connecting to SAN Snapshot™ Copies on Sun™

Description of Connecting Network Appliance Snapshot™ Copies to a Sun System in a SAN Environment

by Richard Jooss, Network Appliance, Inc.

April, 2004 | TR 3317

TECHNICAL REPORT

Network Appliance, a pioneer and industry leader in data storage technology, helps organizations understand and meet complex technical challenges with advanced storage solutions and global data management strategies.

Table of Contents

- 1. Scope**
- 2. Intended Audience**
- 3. Requirements and Assumptions**
- 4. Mapping from File System to NetApp Filer Disks**
- 5. Overview of the Process**
- 6. Steps to Connect to LUNs Created with Snapshot Copies**
- 7. Example of Connecting to LUN from a Snapshot Copy**
- 8. Graphical Representation of Example**
- 9. Steps to Return to the Original State**
- 10. Example of Returning to the Original State**
- 11. Automation/Scripting**

1. Scope

This document covers the steps needed for mounting NetApp filer Snapshot copies from LUNs from Sun Solaris™ systems. In other words, it describes how to gain read/write access to Snapshot copies created for file systems on Sun systems. It does not expand on the multitude of reasons for wanting to gain access to these Snapshot copies such as restore, backup, consistency checking, data mining, etc.

2. Assumptions

This paper is intended for system and storage administrators.

3. Requirements and Assumptions

For the methods and procedures in this document to be useful to the reader, several assumptions are made:

- The reader has at least basic Sun Solaris administration skills and has root access to a UNIX® system.
- The reader has at least basic Network Appliance administration skills and has administrative access to the filer via the command-line interface.
- The filer and host have the necessary licenses to perform the activities outlined in this document.
- The target system has the required block-level and network protocol interconnects to perform the activities outlined in this document.
- Snapshot copies of the file system, or, more correctly stated. Snapshot copies of the NetApp filer volumes containing the LUNs for the file system, have already been created, and they have been created in such a way that the content of the LUNs is consistent. The recommended method is using NetApp SnapDrive™ for UNIX.
- The Snapshot copies will be connected to or made visible on a different system than the one where the original LUNs were connected. The LUNs cannot be made visible to the original system because VxVM will be confused by having multiple LUNs contain identical disk groups, since the LUNs from the Snapshot copy and the original LUNs appear to be identical disks.
- The system where the Snapshot copies will be connected must have the same versions of the volume manager and file system.
- In the examples in this report, all administrative commands are performed at the server or filer console for clarity. Web-based management tools can also be used.

4. Mapping from File System to NetApp Filer Disks

It is important to understand the logical stack that exists between the file system and the NetApp filer disks. There are many possible combinations, and the diagram below shows the most common configurations. The stack on the host consists of:

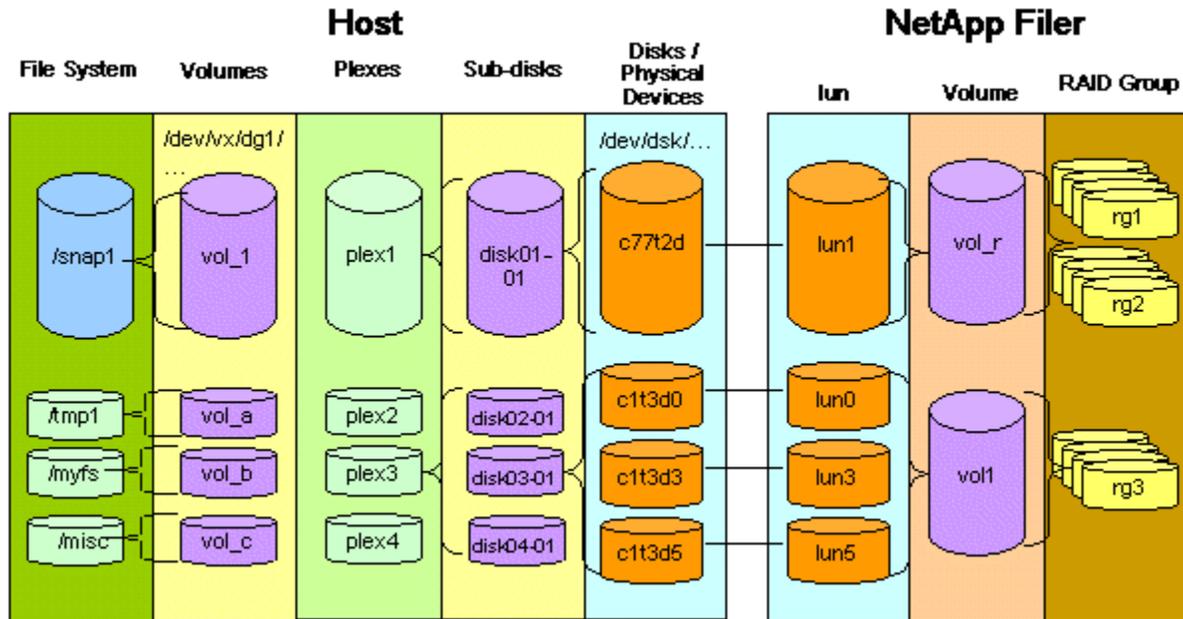
- A file system created on a VERITAS volume.
- The volume is built from one or more plexes.
- A plex is built from VERITAS subdisks.
- A VERITAS disk is built from disks that refer to physical device files.

Typically, with a NetApp filer the admin builds a simple one-to-one relationship between disks, subdisks, plexes, and volumes.

On the NetApp filer the stack consists of:

- Each LUN is mapped to one and only one NetApp filer volume. This mapping can be seen using the NetApp SnapDrive for UNIX "storage show" command.
- Each NetApp filer volume is mapped to one or more RAID groups.
- Each RAID group consists of a number of disks.

For the sake of simplicity, the logical diagram below does not show the multiple paths to each LUN. This diagram only shows a single disk group; on a typical system there will be multiple disk groups.



5. Overview of the Process

Before going into the detailed steps necessary to mount the LUNs from the Snapshot copies, it makes sense to quickly describe the general process.

- **Making the LUNs in the NetApp Snapshot copies readable/writeable**
 After a Snapshot copy is created (recommended method is using NetApp SnapDrive for UNIX), the LUNs are created on the NetApp filer but not visible to any hosts. Before making the LUNs visible to another host, a readable/writable copy of the Snapshot copy needs to be created, since the LUNs in the original Snapshot copy are read-only. This is called a copy of the LUN; however, the only additional space required is for the changes that are actually written to the copy of the LUN. All operations on this readable/writable version of the LUN are performed on the same disk mechanisms as the original LUN, so very heavy activity can also affect performance.
- **Mapping of LUNs**
 The readable/writeable LUNs are then mapped the NetApp filer interfaces (FC or iSCSI) to make them visible to the host(s).
- **Making LUNs visible on the UNIX system**
 In order for the LUNs to be accessed by the UNIX system, the necessary device files need to be created.
- **Importing of disk/volume group**
 The LUNs already contain a valid disk/volume group, and after the necessary steps are performed on the host to gain access to the LUNs, they are imported into the volume manager. At that point, it is just necessary to mount the file systems contained on the LUNs to a mount point on the system.

6. Steps to Connect to LUNs Created with Snapshot Copies

The following table contains the steps necessary to connect LUNs created as part of a Snapshot copy of a Network Appliance filer volume.

Step	Command	Command Location	Description
1	<code>sanlun lun show -p</code>	Original Host	Shows a list of LUNs visible to the host and on which filer the LUNs reside. This information can be used to determine on which filer the follow-on steps need to be executed.
2	<code>lun show -m</code>	Filer	Shows a list of LUNs mapped on the filer giving LUN names, filer volume names, and to which host they are mapped. The filer volume names and LUN names will be used in future steps.
3	<code>snapdrive snap show filer:/vol/<vol>:*</code>	Original host	Lists the available Snapshot copies for the desired volume using either the snapdrive command on the host or filer commands.

Network Appliance Inc.

	or snap list <filer volume>	or filer	Typically all the LUNs will be contained within one filer volume. If there are multiple filer volumes, this step needs to be repeated for each volume.
4	lun create -b <lun_snapshot_name> <new_lun_name>	Filer	Creates a copy or Snapshot copy of the original LUN from the Snapshot copy. Originally the LUN from the Snapshot copy is read-only, and in order to mount/activate the host volume group (VG), a writeable Snapshot copy is needed. The original "snapshotted" LUN is not accessible during the time this LUN created with "lun create -b" exists. The original Snapshot copy state is returned after this LUN is removed. This command has to be repeated for each needed LUN.
5	sanlun fcp show adapter -c	Secondary host	This command will show the command needed to be entered on the filer in the next step. Note that the name shown in quotes can be changed to your preferred name.
6	igroup create -f -t solaris <group name> WWN#1 WWN#2	Filer	This command will create an igroup to which the newly created LUNs will be mapped. The WWNs from the secondary host will be entered here. It is important not to enter the original host WWNs to avoid having the Snapshot copy LUNs in addition to the primary LUNs visible to the original host.
7	lun map <new_lun_name> <igroup> <LUN_ID>	Filer	Maps the newly created LUNs so that they are visible for the secondary host. It is important not to enter an igroup where the original host WWNs are present to avoid having the Snapshot copy LUNs in addition to the primary LUNs visible to the original host.
8	Create persistent bindings between the host and the filer. Instructions on how to do this are available as part of the host attach kit (HAK) documentation on the NetApp On the Web (NOW) site	Secondary host	It is necessary to create a persistent binding between the host (initiator) and the filer (target) so that the LUNs mapped from the filer are always available with the correct SCSI target IDs. This operation requires a reboot of the system.
9	Create entries in the sd.conf for the mapped LUNS. Instructions on how to do this are available as part of the host attach kit (HAK) documentation on the NetApp On the Web (NOW) site	Secondary host	In order for the host to see the newly mapped LUNs, the appropriate entries need to be made in the /kernel/drv/sd.conf file. This operation requires a reboot of the host.
10	fsdevadm	Secondary host	Scans for and creates I/O devices for configured devices (LUNs in the case of

	<code>vxdctl enable</code>		FC/iSCSI devices) and makes them visible to the VERITAS Volume Manager.
11	<code>sanlun lun show -p</code>	Secondary host	Shows a list of LUNs visible to the host. The newly created and mapped LUNs should appear. The "-p" option can be left off to get different presentation of the device files.
12	<code>vxdg -tC -n <new_dg_name> import <disk group></code>	Secondary host	This command scans the available device files and imports the disk group to the volume manager on this host. The disk group name is changed to <new_dg_name>. <p>After this command is executed, the directory /dev/vx/<new_dg_name> should be present with its appropriate subdirectories.</p>
13	<code>vxvol -g <new_dg_name> start all</code>	Secondary host	This command activates all the volumes in the disk group just imported.
14	<code>mkdir 777/<snap></code>	Secondary host	This command creates a directory for mounting the file system.
15	<code>mount /dev/vx/<new_dg_name> /<volume></code>	Secondary host	This command mounts the file system on the directory that was just created.

7. Example of Connecting to a LUN from a Snapshot Copy

The following table shows an example in which a file system is built using a single LUN located on a single filer. Typically, a logical volume will consist of multiple LUNs, but for simplicity only a single LUN is here. If multiple LUNs are used, steps 4, 5, 9, and 13 need to be repeated or expanded to include the addition LUNs. If the Snapshot copies are from multiple filers, steps 2, 3, 4, and 5 will need to be executed on each filer.

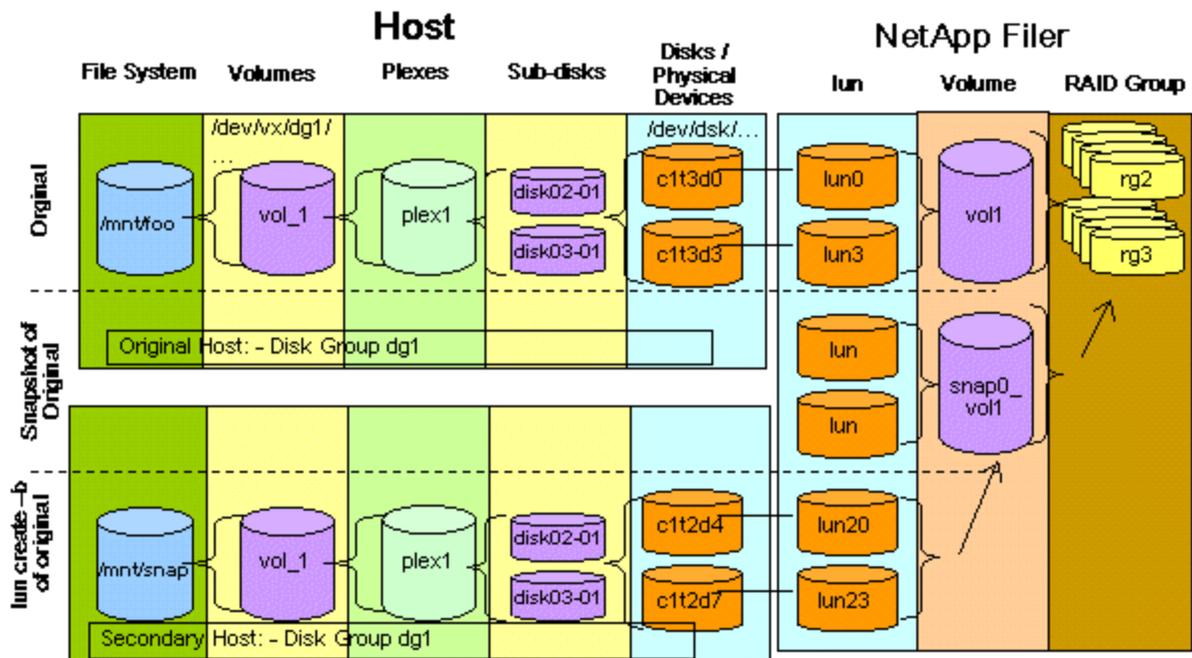
Step	Example
1	<pre> root> sanlun lun show -p filer1:/vol/vol1/sh-sun-lun11 (LUN 11) /dev/rdisk/c1t0d11s2 (c1t0d11s2) 100m (104857600) lun state: GOOD ----- path path device host target partner Multipath state type filename HBA port port policy ----- enabled primary /dev/rdisk/c1t0d11s2 lpfc0 5b A/A enabled primary /dev/rdisk/c2t0d11s2 lpfc1 7b A/A root> </pre>
2	<pre> filer1> lun show -m LUN path Mapped to LUN ID ----- /vol/vol1/sh-hp2-lun-010 sh-hp1 10 /vol/vol1/sh-sun-lun11 sun 11 filer1> </pre>

3	<pre> root> # snapdrive snap show filer1:/vol/voll:* host name filer path date name snapped ----- filer1 filer1:/vol/voll Feb 29 12:52 test_sun_snap -dg db2dg root> or filer1> snap list voll Volume voll working... %/used %/total date name ----- 0% (0%) 0% (0%) Feb 29 12:52 test_sun_snap filer1> </pre>
4	<pre> filer1> lun create -b /vol/voll/.snapshot/test_sun_snap/sh-sun-lun11 /vol/voll/snap-sh-sun-lun11 filer1> </pre>
5	<pre> root> sanlun fcp show adapter -c Enter this filer command to create an initiator group for this system: igroup create -f -t solaris "sunny" 10000000c938ed86 10000000c9391b6b root> </pre>
6	<pre> filer1> igroup create -f -t solaris sun_snap 10000000c938ed86 10000000c9391b6b filer1> </pre>
7	<pre> filer1> lun map /vol/voll/snap-sh-sun-lun11 sun_snap 13 filer1> </pre>
8	<p>Create persistent binding as described in host attach kit (HAK) documentation.</p>
9	<p>Edit /kernel/drv/sd.conf file for the newly created LUNs as described in host attach kit (HAK) documentation.</p>
10	<pre> root> devfsadm root> vxdctl enable root> </pre>
11	<pre> root> sanlun lun show -p filer1:/vol/voll/snap-sh-sun-lun11 (LUN 13) /dev/rdisk/c1t0d11s4 (c1t0d11s4) 100m (104857600) lun state: GOOD ----- path path device host target partner Multipath state type filename HBA port port policy ----- enabled primary /dev/rdisk/c1t0d11s4 lpfc0 5b A/A enabled primary /dev/rdisk/c2t0d11s4 lpfc1 7b A/A </pre>
12	<pre> root> vxdg -tC -n db2dg_new import db2dg </pre>

13	root> vxvol -g db2dg_new startall
14	root> mkdir 777 /mnt/snap
15	root> mount /dev/vx/dsk/db2dg_new/vol1 /mnt/snap

8. Graphical Representation of Example

The diagram below depicts the result of the steps above. The first row, Original, shows how the original file system is mounted. The second row, Snapshot copy of original, shows how the Snapshot copy exist on the filer but the LUNs are not visible to the host. The LUNs in this row are read only and not available during the time the LUNs in the third row exist. The third line, lun create -b of original, displays the read/writable LUNs which were made visible to the host to allow the mounting of the file system. Any changes made to the LUNs in this third row will be lost as soon as the LUNs are removed or destroyed. The diagram below differs from the example in that it shows two LUNs, which make up the file system, and the names are not representative.



9. Steps to Return to the Original State

The following table contains the steps necessary to undo what was done to connect to the Snapshot copy and return to the original state. To correlate the steps, these steps are number r for reverse steps.

Step	Command	Command Location	Description
r15	umount / <snap>	Host	This command unmounts the file system.
r14	rmdir /<snap>	Host	This command removes the

			directory where the file system was mounted.
r13	<code>vxvol -g <disk group> stopall</code>	Host	This command will deactivate all the volumes within the disk group.
r12-r10	<code>vxdg deport <disk group></code>	Host	This command will disable access to the disk group from this system.
r9	Remove the entries from the <code>/kernel/drv/sd.conf</code> file.		This will disable access to the LUNs from this host. Note that after editing this file the system needs to be rebooted for the changes to take effect.
r8	Remove the persistent bindings from the <code>/kernel/drv/lpfc.conf</code> file using the <code>lputil</code> command. It is not recommended to edit the file directly because it is extremely white space sensitive.		This will remove the bindings between the host and the filer. Note that after editing this file the system needs to be rebooted for the changes to take effect.
r7	<code>lun unmap <new_lun_name><igroup></code>	Filer	This command removes the mapping of the LUNs from the igroup.
r6	<code>igroup destroy <igroup></code>	Filer	This command will remove the igroup from the filer.
r5-r1	<code>lun destroy <new_lun_name></code>	Filer	This command removes the created LUN. After this command is executed, the LUN within the Snapshot copy is returned to its original state.

10. Example of Returning to the Original State

The following table shows the steps necessary to return to the original state from the example steps shown above.

Step	Example
r15	<code>root> umount /mnt/snap</code>
r14	<code>root> rmdir /mnt/snap</code>
r13	<code>root> vxvol -g db2dg_new stopallroot></code>
r12-r10	<code>root> vxdg deport db2dg_newroot></code>
r9	Remove entries from <code>/kernel/drv/sd.conf</code> file.
r8	Remove bindings from <code>/kernel/drv/lpfc.conf</code> file with <code>lputil</code> and reboot host.
r7	<code>filer1> lun unmap /vol/vol1/snap-sh-sun-lun11 sun_snapfiler1></code>

r6	filer1> group destroy sun_snap
r5-r1	Filer1> lun destroy /vol/voll/snap-sh-sun-lun11

11. Automation/Scripting

This is a procedure that typically is done repeatedly and is automated through scripts where the data is then backed up to tape, analyzed, checked for consistency, etc. Typically, the Snapshot copies are created and renamed using the same names each time to make the scripting process easier. Also, by using the same mappings each time, the steps of creating the file path names in the /kernel/drv files and creating the persistent bindings using the lputil command can be spared. This, of course, also eliminates the need to reboot the system each time.

In a scripted solution, normally only steps 4, 5, 7, 10, 12, 13, and 15 will need to be done each time to gain access to the Snapshot copies, and steps r15, r13, r12, r7, and r5 are necessary to return to the starting state.

Network Appliance, Inc.

© 2004 Network Appliance, Inc. All rights reserved. Specifications subject to change without notice. NetApp, the Network Appliance logo, DataFabric, FAServer, FilerView, NearStore, NetCache, SecureShare, SnapManager, SnapMirror, SnapRestore, and WAFL are registered trademarks and Network Appliance, ApplianceWatch, BareMetal, Camera-to-Viewer, Center-to-Edge, ContentDirector, ContentFabric, Data ONTAP, EdgeFiler, HyperSAN, InfoFabric, MultiStore, NetApp Availability Assurance, NetApp ProTech Expert, NOW, NOW NetApp on the Web, RoboCache, RoboFiler, SecureAdmin, Serving Data by Design, Smart SAN, SnapCache, SnapCopy, SnapDirector, SnapDrive, SnapFilter, SnapMigrator, Snapshot, SnapSuite, SnapVault, SohoCache, SohoFiler, The evolution of storage, Vfiler, VFM, Virtual File Manager, and Web Filer are trademarks of Network Appliance, Inc. in the U.S. and other countries. All other brands or products are trademarks or registered trademarks of their respective holders and should be treated as such.