APPLICATION NOTE

Using Shared Virtual Array® (SVA™) disk systems with SnapShot software, SAP R/3, Oracle and NetWorker on Solaris

MARCH 2004

Using Shared Virtual Array® (SVA™) disk systems with SnapShot software, SAP R/3, Oracle and NetWorker on Solaris

TABLE OF CONTENTS

1 ABSTRACT 4

2 UNDERSTANDING SNAPSHOT 4

3 ARCHITECTURE EXAMPLE 5

4 SAP SPLIT MIRROR SCENARIO OVERVIEW 6

5 CONFIGURATION STEPS 7

5.1 PRODUCTION SERVER CONFIGURATION 8

5.1.1 SVA configuration 8

5.1.2 SAP partitioning 9

5.1.3 SVA target LUNs 9

5.1.4 SAP application – Oracle database configuration 10

5.1.4.1 User accounts and groups 10

5.1.4.2 UNIX security 11

5.1.4.3 Startup and shutdown processes 11

5.2 BACKUP SERVER CONFIGURATION 12

5.2.1 SVA target LUNs configuration 12

5.2.2 SAP and Oracle configuration 13

5.2.2.1 User accounts and groups 13

5.2.2.2 Oracle installation 13

5.2.2.3 Oracle configuration 13

5.2.2.4 Oracle SQL*Net configuration 14

5.2.2.5 BRTOOLS installation 15

5.2.3 Shared backup file systems 15

5.2.4 Legato NetWorker server and NetWorker module for SAP 15

5.2.4.1 Installation 15

5.2.4.2 NetWorker configuration 16

6 BACKUP AND RESTORE OPERATIONS 21

6.1 SAP ONLINE/OFFLINE SPLIT BACKUP USING A DEDICATED BACKUP SERVER 21

6.1.1 NetWorker for SAP 22

6.1.2 SAP tools 22

6.1.3 Split and resynchronisation scripts 22

6.1.4 The main backup script 23

6.1.5 Backup execution overview 24

6.1.5.1 Manual backup using the main brbackup script 24

6.1.5.2 Manual backup using brbackup command line 25

6.1.5.3 Manual backup from SAPDBA 25

6.1.6 Backup the archive logs 27
6.1.6.1 Archive logs from the SAPDBA utility 27
6.1.6.2 Archive logs from the brarchive command 28
6.2 HOW TO RESTORE? 28
6.2.1 Restore versus recovery 28
6.2.2 Restore from SAPDBA 29
6.2.3 Restore from the NetWorker interface 29

7 CRITICAL ANALYSIS 30

8 APPENDIX 31
APPENDIX A: TARGET LUNS CREATION SCRIPT 31
APPENDIX B: SCRIPT TO FORMAT TARGET LUNS 32
APPENDIX C: SOLARIS/ETC/SYSTEM ADD-ON 32
APPENDIX D: NET*8 CONFIGURATION FILES 33
APPENDIX E: LUN CONFIGURATION FILE 35
APPENDIX F: MAIN BACKUP SCRIPT (ONLINE SPLIT MIRROR) 36
APPENDIX G: MAIN BACKUP SCRIPT (OFFLINE SPLIT MIRROR) 37
APPENDIX H: SNAPSHOT SCRIPT 38
APPENDIX I: RESYNCHRONIZATION SCRIPT 43
APPENDIX J: EXAMPLE OF ONLINE SUCCESSFUL BACKUP 47
APPENDIX K: EXAMPLE OF OFFLINE SUCCESSFUL BACKUP 50

HARDWARE AND SOFTWARE APPLICATIONS USED

	Name	Vendor	Comments
ISV	SAP R/3 SAP Front end Oracle 8i NetWorker 6.1 NetWorker Module for SAP	SAP SAP Oracle Legato Legato	Tested with SAP 4.6C Optional*
Operation System	Solaris 2.7, 2.8, 2.9	SUN	Tested on Solaris 2.7 & 2.8 Should work with 2.9
SVA Software	SVAA 3.1.0 patch 12 SVAC 3.1.0 patch 11 SVA PATH 3.3	StorageTek	 Optional** Optional***
SVA Hardware	SVA 9500 and V960	StorageTek	Tested on SVA 9500
Backup Hardware	Quantum DLT drive DLT tape	Quantum	

*SAP Front end is the GUI that enables the user to connect to the SAP application. **SVAC is the graphical version of SVAA. The use of the SVAC is optional; the devices can be created using the CLI. ***SVA Path is the failover product for SVA. The scripts included in this document contain SVA Path command lines. Those lines have to be commented if you don't use SVA Path.

1 ABSTRACT

The StorageTek® Shared Virtual Array® (SVA™) has a great feature, called 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 following products: SAP R/3 application with an Oracle database and the Legato NetWorker backup software. The tests performed here are based on a Solaris environment. But, you can refer to the "Proof of Concept (POC) SAP Online Backup on NT using SVA SnapShot and NetWorker (June 2002)" for more information about the same scenario but based on a Windows platform. If the backup software used is the VERITAS NetBackup software, you can refer to the "POC SAP Online Backup on Solaris using SVA SnapShot and NetBackup (June 2002)".

It is assumed that the reader has a good knowledge of Solaris administration, Legato NetWorker Software, SAP R/3 application, Oracle database management, SVA principles and SVA administration.

The last part of this document describes the different kind of backup and restore for the SAP application and its Oracle database with SVA and NetWorker.

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

ASSUMPTIONS

It is supposed that the reader has a good knowledge of the SVA and SAN configuration. The purpose of this document is not to describe how to connect all the hardware components of this architecture, nor to help in making the SAP application and its Oracle database run, nor to detail how to configure the SVA to install SAP and Oracle on its devices, but to give solutions in making backup and restore operations of SAP and Oracle using the SVA SnapShot feature. These solutions are based on scripts.

So, this paper assumes that the SAP R/3 is already installed and its installation is "standard" (not configured for split mirror backup).

One of our objectives is to integrate the split mirror scenario into the production server without affecting the system availability.

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 LUN 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 that are 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.

HOW DOES IT WORK?

Lets take an example to concretely show the SnapShot principle **(Figure 1).**

In this client/server configuration from the backup server point of view, the production server is fully available in responding to requests. It is involved in the backup operation only to perform the snapshot copy of the data – that does not last too long.

> The data must be on the SVA devices in order to make a snapshot.
> The source and target LUNs must be on the same domain (D2) for the system to be able to perform a snapshot copy of the data.
> Both servers must share the same domain on the SVA in order to have access to the same LUNs.
> The backup server will use the SnapShot LUNs to perform the data backup.

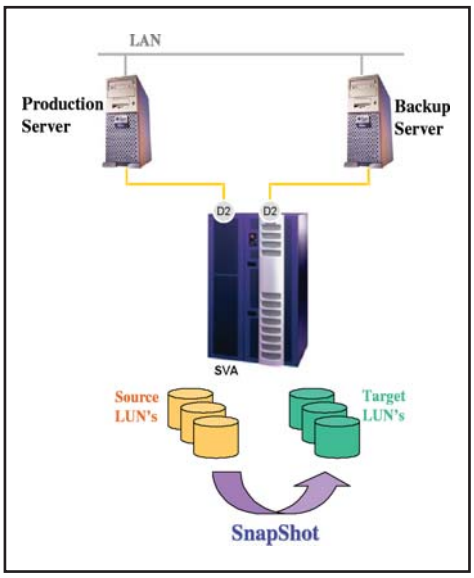

Figure 1. Snapshot mechanism.

Figure 1. Snapshot mechanism

The following operations are executed:

> The source LUNs are accessible from the production server (D2)
> The target LUNs must be created and made available – but not mounted – for the production server (D2)
> The SnapShot operation is performed on the production server (a few seconds)
> The target LUNs must be accessible from the backup server (D2)
> The backup server must mount those target LUNs
> The backup is done by the backup server using the target LUNs.

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 the Legato NetWorker software.

The figure below depicts the SAP and Oracle with NetWorker configuration. This basic configuration intends to reproduce a typical schema with a central instance, where the database and the application server are running on the same physical CPU and a separate backup server is used to minimize the performance influence of the backup on the productive system.

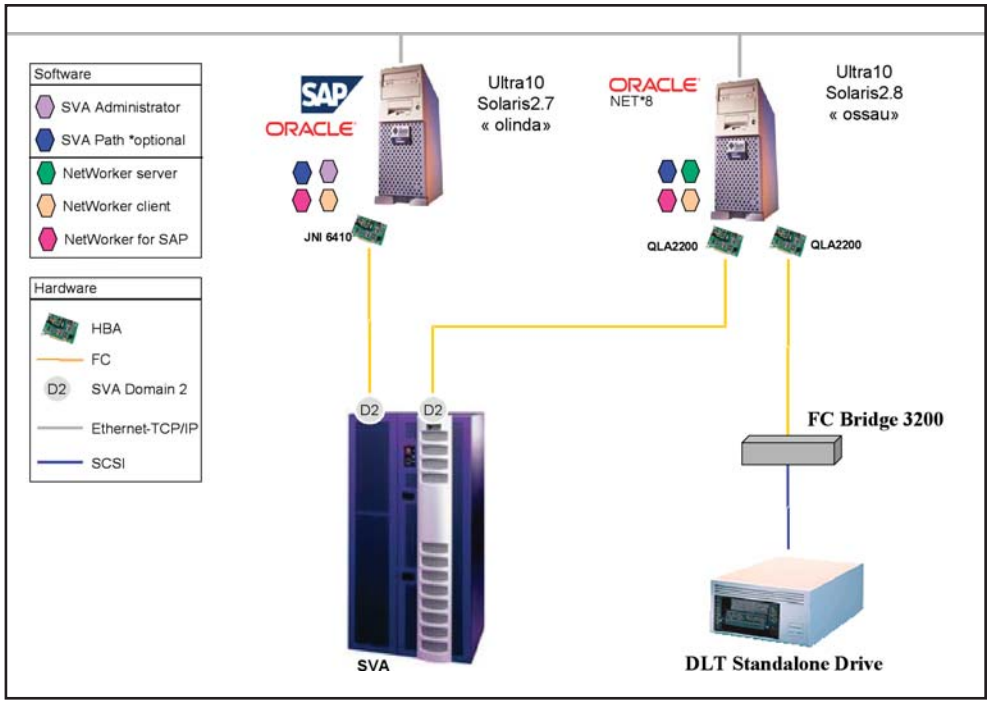

Figure 2. Single production/backup server configuration.

The **production server** (Solaris 2.7 olinda) uses 17 LUNs of the SVA to store the SAP application and all the data related to the Oracle database (including SAP executables and transport directories, Oracle redo-logs and archive directories…). So, the SAP application is installed on the production server. The snapshot copy uses 17 other LUNs on the same SVA with exactly the same characteristics as the source LUNs. The server uses a Fibre Channel connection (JNI 6410) to communicate with the SVA subsystem on the domain 2. The production server is configured as a NetWorker client.

The **backup server** (Solaris 2.8 ossau) is connected to the SVA with a Fibre Channel connection (QLogic QLA2200F). It is also connected to the tape drive via a Fibre Channel connection (QLogic QLA2200F). This server is configured as the NetWorker server. Oracle has to be installed and properly configured (the same as on the production server) on the backup server. All the SAP binary files have to be present on the backup server.

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

4 SAP SPLIT MIRROR SCENARIO OVERVIEW

There are two different ways to perform a backup:

> Offline backup (or cold backup): The database is shutdown before the effective backup and started again after the backup is finished. The SAP application is not available during the backup time (that may be long depending on the size of the database).

> Online backup (or hot backup): The database is switched into hot backup mode before the effective backup and this hot backup mode is switched off after the backup is finished. The SAP application is still available (but the performance may decrease).

The SAP tool BRBACKUP supports the backup of the database using a split mirror scenario. This mirror is split before the backup is performed. The data are snapped on the primary host and mounted on a secondary host that makes the backup. This technique is possible either for online or offline backups. This configuration is fully supported from SAP Release 4.0 and is recommended for large databases because the time the database is offline or in hot backup mode is considerably reduced.

BRBACKUP also communicates with the productive database and is able to save the results of the backup.

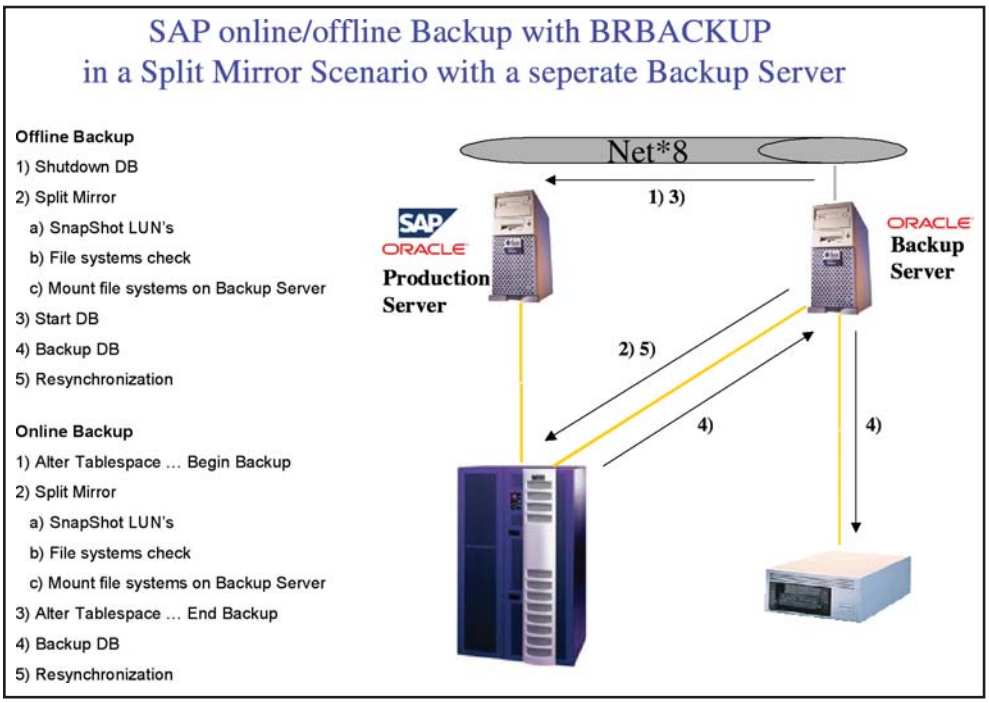

Figure 3. Backup process overview using a dedicated backup server.

Using SnapShot within a split mirror scenario does not require a permanent mirroring of the source to the target LUNs. Additional capacity is not necessary. In our case, resynchronization is not necessary but as the command has to be specified, the resync_cmd script only un-mounts the target LUNs from the backup server at the end on the backup.

5 CONFIGURATION STEPS

The aim of this paragraph is to detail the:

> Production server configuration
> Backup server configuration.

5.1 PRODUCTION SERVER CONFIGURATION

This section intends to present the global configuration of the production server. The major part of this configuration consists in:

> Configuring the SVA LUNs
> Installing and configuring SAP and Oracle.

It is considered that these operations have already been performed. Some more details are given on the configuration used in our example, but the installation and configuration processes will not be described.

The following describes:

> The SVA configuration
> The SAP partitioning in our example
> The SVA target LUNs configuration
> The SAP application and Oracle database configuration.

5.1.1 SVA configuration

This section describes the operations required to configure the SVA. Make sure that you can get access to the SVA subsystem.

> Install SVA Administrator, configure and start the server
> Make sure the SVA is correctly configured.

```
root@olinda: /opt/storagetek/SVA3.1.0/bin/sibadmin
SIB> dserver
SVAA Server: SAPDEMO           Date/Time: 10-31-2002 11:04:02 GMT

Version: 3.1
Description:
Trace Status: Disabled
Installed Features:
   SnapShot
Maintenance Level:
   0 PPFinfo PTF=L2P008T Patch=12 FIX=0 Issue=768241
Host:
   Name: olinda
   TCP/IP Port: 41248
Host OS Level:
     5.7
     Generic_106541-16
```

> Add a control path to the SVA and check the access to the subsystem.

```
root@olinda: /opt/storagetek/SVA3.1.0/bin/sibadmin
SIB> addsubsyspath –devpath /dev/rdsk/c1t8d5s2
SIB> qsubsys
SVAA Server: SAPDEMO           Date/Time: 10-31-2002 11:04:08 GMT

NOVA
```

> Install SVA Path (if used) and include the ECAM device you just added.
> Check with the format command.

```
root@olinda: format
       2. c1t8d5 <STK-9500-1127 cyl 3335 alt 2 hd 15 sec 96>
          /pci@1f,0/pci@1/fibre-channel@2/sd@8,5
```

You can now create the source LUNs that will contain the SAP and Oracle data. Refer to the paragraph below for details about SAP partitioning used in our configuration.

5.1.2 SAP partitioning

The Oracle home directory must correspond to the SAP standard central installation. The instance name in our example is TLS.

5.1.3 SVA target LUNs

The set of LUNs previously described is used by the original SAP application and its Oracle database. Once the source LUNs are created, the SAP application can be installed. It is recommended that only a SAP Technical Consultant do SAP installations.

Another set of LUNs is created for the SnapShot copies used for the backup. As only SAP data are snapped, only target LUNs for SAP data are created. The SVA SnapShot partitions must be identical to the initial partition.

Some scripts are used to make the LUNs creation easier. Please refer to **Appendix A: Target LUNs Creation Script.**

Note **Modify these scripts in order to reflect your configuration. Put the name of the SVA you use, the range of available fdid's, the LUNs number, the domain and target used on the SVA. Or you can create those LUNs from the SVA Console (Figure 4).**

Create all the required target LUNs:

> Go to menu Devices > Define New …
> Complete with the right 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.

File System Name	LUNs size
/oracle	7.4 GB
/oracle/TLS	7.4 GB
/oracle/TLS/origlogA	2.5 GB
/oracle/TLS/origlogB	2.5 GB
/oracle/TLS/mirrlogA	2.5 GB
/oracle/TLS/mirrlogB	2.5 GB
/oracle/TLS/sapdata1	14.8 GB
/oracle/TLS/sapdata2	14.8 GB
/oracle/TLS/sapdata3	14.8 GB
/oracle/TLS/sapdata4	14.8 GB
/oracle/TLS/sapdata5	14.8 GB
/oracle/TLS/sapdata6	14.8 GB
/oracle/TLS/sapbackup	14.8 GB
/oracle/TLS/saparch	14.8 GB
/oracle/TLS/sapreorg	14.8 GB
/usr/sap	14.8 GB
/sapmnt/TLS	14.8 GB

These sizes are based on the SAP requirements. Please refer to the SAP guide: "R/3 installation on UNIX: Oracle database."

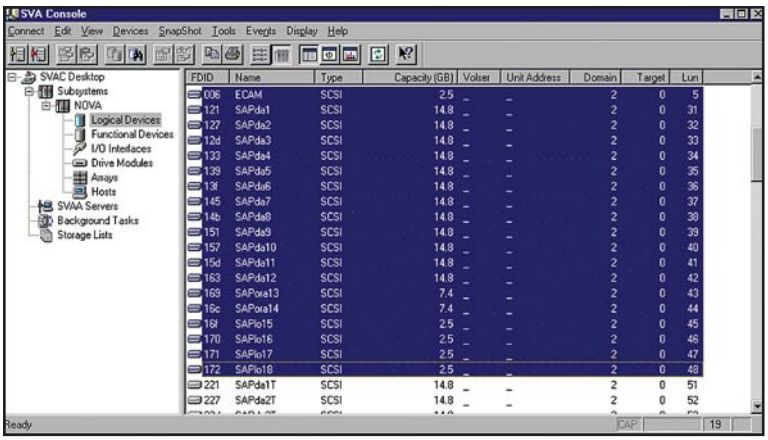

Figure 4.
SVA console interface (connection with NOVA).

Figure 5. Definition of a new device interface.

The Solaris server will discover these LUNs at boot time only if the target and LUN numbers are included in the /kernel/drv/sd.conf file. These lines correspond to the specific configuration we use. So, you may have to adapt the following lines to make it correspond to the bindings made. Refer to the /var/adm/messages file for detailed information on the bindings of the system or to the configuration file attached to the HBA cards where you can specify persistent bindings.

```
name="sd" class="scsi" target=8 lun=5;
name="sd" class="scsi" target=8 lun=31;
…
name="sd" class="scsi" target=8 lun=57;
```

All the LUNs created have to be formatted. Use the format label command to label all these LUNs.

These operations can be performed manually but can also be scripted. Using scripts can be useful if you have to create the target LUNs each time you perform a backup operation (the target LUNs are reuseable after the termination of the backup). Please refer to **Appendix B: Script to format target LUNs** for script examples to be adapted to your configuration.

5.1.4 SAP application – Oracle database configuration

The SAP application and its Oracle database are installed on the production server using the LUNs created on the SVA. The installation performed in our example is a central one. SAP needs some specific users to be created regarding the Oracle instance SID (TLS).

5.1.4.1 User accounts and groups

Make sure that the following two specific UNIX users and three groups exist on the system. If not, they must be created for SAP administration.

```
groupadd -g 102 dba
groupadd -g 100 sapsys
groupadd -g 101 oper
useradd -u 100 -g sapsys -G oper,dba -d /oracle/TLS -s /usr/bin/csh -
c "SAP Database Administration User" oratls
useradd -u 101 -g sapsys -G oper -d /export/home/tlsadm -s
/usr/bin/csh -c "SAP System Administrator" tlsadm
```

The user tlsadm is used to start and stop the SAP application.

The user oratls is used to administrate Oracle via svrmgrl and SAPDBA.

Note **If you have to change the users' password, always use the SAP tools.**

The following line must be present at the end of file /var/opt/oracle/oratab

```
TLS:/oracle/TLS:N
```

5.1.4.2 UNIX security

In order to allow the backup server to connect to the production server, the /.rhosts file has to be modified.

```
ossau root
ossau oratls
ossau tlsadm
```

5.1.4.3 Startup and shutdown processes

Use the following command to check whether the SAP system is started or not.

```
root@olinda: su – tlsadm
olinda:tlsadm 1% startsap check

Checking SAP R/3 TLS Database
------------------------------
 Database is running

Checking SAP R/3 TLS Instance DVEBMGS00
------------------------------
 Instance DVEBMGS00 is running
```

If the Oracle database is not running, connect with the oratls user account:

```
root@olinda: su – oratls
Sun Microsystems Inc.   SunOS 5.7       Generic October 1998
olinda:oratls 1% sapdba
```

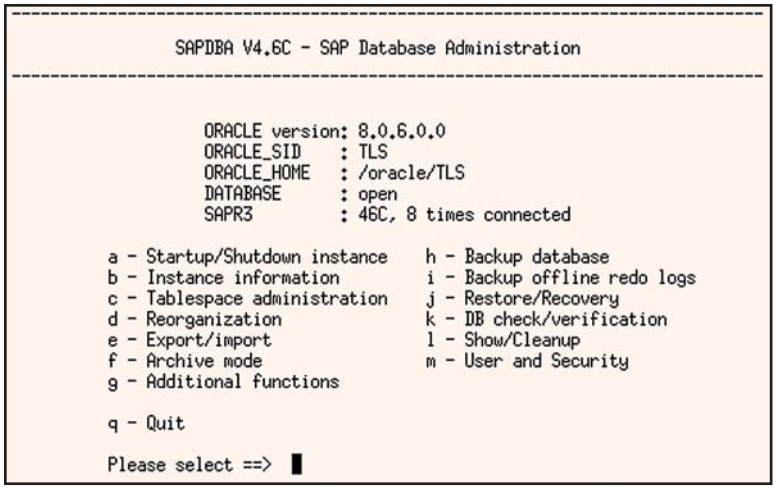

Figure 6. SAP database administration menu.

Troubleshooting startup errors:

> The listener may not run: use the tlsadm user account to check whether the listener is running or not (lsnrctl status)

> The file systems of the source LUNs may not be mounted

> You may have to shutdown the Oracle database and restart it if the SAP application has crashed (use SAPDBA).

Note **The startup logs are in /export/home/tlsadm.**

5.2 BACKUP SERVER CONFIGURATION

This section intends to present the way to configure the backup server in order to perform backups of the SAP application and its Oracle database using the split mirror scenario.

At the least, the backup server must:

> Have access to the target LUNs defined on the SVA for the SnapShot copy of the SAP data files

> Have SAP and Oracle configured. (See explanations for the following in the sections on next page.)

- User Accounts and Groups
- Oracle installation
- Oracle configuration
- Oracle SQL* Net configuration
- BRTOOLS installation

> Have the needed file systems shared.

5.2.1 SVA target LUNs configuration

The backup server needs to have access to the SnapShot LUNs. As the backup server and the production server are connected to the SVA on the same domain, the steps run on the production host are also applicable on the

backup server. The target LUNs were already created on the production server so you don't have to re-create them here. Perform only the following steps:

> Add the corresponding target LUNs in the /etc/sd.conf file

> Discover the target LUNs (reconfigure /devices).

```
root@ossau: devfsadm
```

> Update the SVA Path configuration file /etc/sppath.conf.

```
root@ossau: sppath -v
```

> Update /etc/spd.conf file.

```
root@ossau: setsp -g
```

> Start the spd driver for new devices.

```
root@ossau: setsp -S
```

> Create the mount points /oracle/TLS/sapdata1 to /oracle/TLS/sapdata6.

Note **It is recommended to use the same file system names as on the production server: all the SAP database files are mounted under /oracle/TLS. In case you want to perform the backup from the production server itself, you need to use a different anchor point for the SnapShot copy such as /oracle/SNAP.**

Note **If you are running SAP 4.5B or higher, you are able to mount the database files on the backup host in a different $SAPDATA_HOME directory by using the orig_db_home parameter in file initTLS.sap.**

5.2.2 SAP and Oracle configuration

5.2.2.1 User accounts and groups

Please refer to paragraph **5.1.4.1 User accounts and groups** for the steps to run to get the required UNIX users created on the backup server.

5.2.2.2 Oracle installation

The Oracle Client software and the Oracle Server Manager have to be installed on the backup server in a split mirror scenario. These applications are used to get connected to the production server using SQL*Net.

At least the client software and SRVMGR components of the Oracle database software must be installed on the backup server as well as SAP BRTOOLS.

5.2.2.3 Oracle configuration

The split mirror scenario requires a properly maintained Oracle password file and additional initialisation parameters. So, perform the following on the **production server**:

> Create the Oracle password file using the following command line:

```
root@olinda: pwd
/oracle/TLS/dbs
root@olinda: orapwd file=/oracle/TLS/dbs/orapwTLS password=my_password entries=100
```

> Ensure that the following lines are present and uncommented in the Oracle initialisation file at /oracle/TLS/dbs/initTLS.sap:

```
remote_login_passwordfile         = exclusive
```

Making changes in this file requires the database to be restarted for them to take effect.

> Set the authorizations for the Oracle user SYSTEM.

```
root@olinda: su – oratls
Sun Microsystems Inc.   SunOS 5.8       Generic February 2000
ossau:oratls 1% svrmgrl
SVRMGR> connect internal
Connected.
SVRMGR> grant SYSOPER to SYSTEM;
SVRMGR> grant SYSDBA to SYSTEM;
SVRMGR> exit
```

5.2.2.4 Oracle SQL*Net configuration

The backup server needs connections to the Oracle database on the production server during the backup process. To do so, files /oracle/TLS/dbs/tnsnames.ora, listener.ora and sqlnet.ora on the backup server have to be updated. Refer to **Appendix D: Net*8 configuration files**.

> In the profile file initTLS.sap, update the following parameters:

```
remote_host = olinda
remote_user = root
primary_db = TLS.WORLD
```

> When calling the SAP brbackup command, use the option –u internal/my_password

> Test the connection from the backup server to the production server:

```
root@ossau: su – oratls
Sun Microsystems Inc.   SunOS 5.8       Generic February 2000
ossau:oratls 1% svrmgrl
SVRMGR> connect internal@TLS.WORLD
Password: my_password
Connected.
```

5.2.2.5 BRTOOLS installation

The SAP backup utilities (BRBACKUP, BRARCHIVE, BRRESTORE…) have to be properly installed on the backup server. You can either share the /usr/sap and /sapmnt/TLS directories of the production server or make SnapShot copies of that directory.

5.2.3 Shared backup file-systems

The SAPBACKUP directory on the production server must be accessible from the backup server. To make sure it is shared for remote network mounts, run the following command on the production server:

```
root@olinda: share
-               /oracle/TLS/sapbackup    rw    ""
-               /oracle/TLS/sapreorg    rw    ""
```

If these directories are not shared, use the following command on the production server:

```
root@olinda: share /oracle/TLS/sapbackup
root@olinda: share /oracle/TLS/sapreorg
```

On the backup server the main backup script will use the following commands:

```
mount olinda:/oracle/TLS/sapbackup /oracle/TLS/sapbackup
umount /oracle/TLS/sapbackup

mount olinda:/oracle/TLS/sapreorg /oracle/TLS/sapreorg
umount /oracle/TLS/sapreorg
```

Note **Make sure that the directories /oracle/TLS/sapbackup and /oracle/TLS/sapreorg are created on the backup server. After a reboot of the system, you need to execute the share command again unless you include those file systems in the /etc/dfs/dfstab file.**

5.2.4 Legato NetWorker server and NetWorker module for SAP

5.2.4.1 Installation

Install the NetWorker Server on the backup server using the standard procedure. Then, check that the NetWorker client is also installed to be able to install the NetWorker module for SAP.

Copy the backint program to the directory where the SAP BRTOOLS reside:

```
root@ossau: cp /nsr/tmp/backint /usr/sap/TLS/SYS/exe/run/
```

Copy the init.utl file to the Oracle home directory. Rename this file adding the Oracle instance SID.

```
root@ossau: cp /nsr/tmp/init.utl /oracle/TLS/dbs/initTLS.utl
```

Note **The NetWorker client and the module for SAP are installed on the production server.**

5.2.4.2 NetWorker configuration

Launch the NetWorker Administrator Interface in order to use the backup utilities. If the NetWorker daemon and processes are not yet started, use the command /etc/init.d/networker start. If you need to stop them, type /etc/init.d/networker stop.

```
root@ossau: /usr/bin/nwadmin &
```

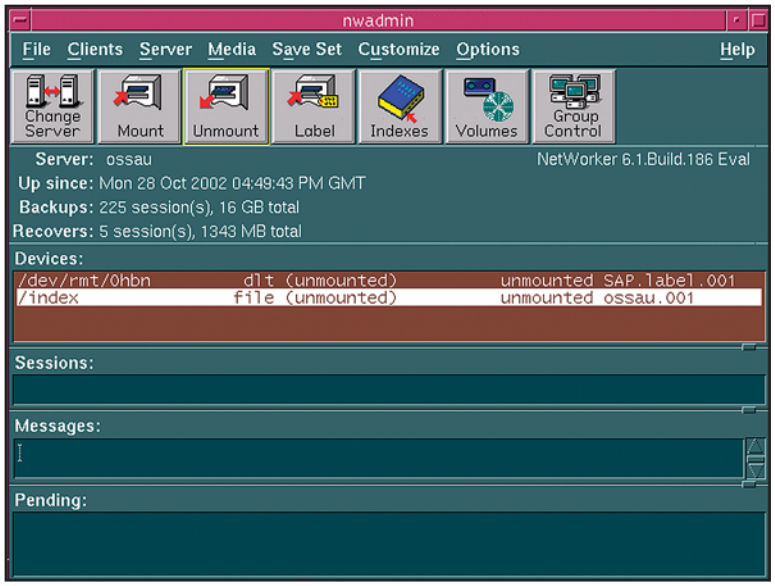

Figure 7. The NetWorker administration interface.

Configuration Steps

The NetWorker Backup tool must be configured manually. Perform the following:

> Create a new group. Go to the menu Customize > Groups… Choose the create function and complete the fields with the appropriate information. Create the group SAPgroup.

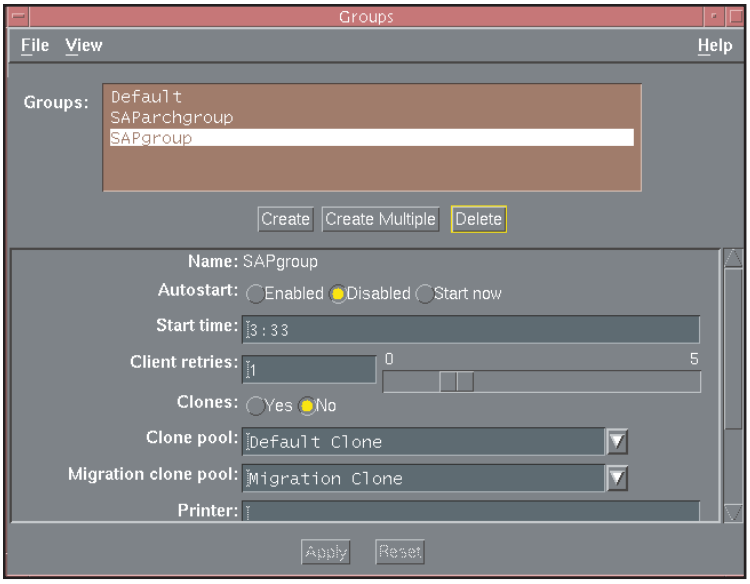

Figure 8. NetWorker groups interface.

> Create a new label template for the pool to be created. Go to the menu Customize > Label Templates… Choose the create function and complete the fields with the appropriate information. Create the label SAPlabel.

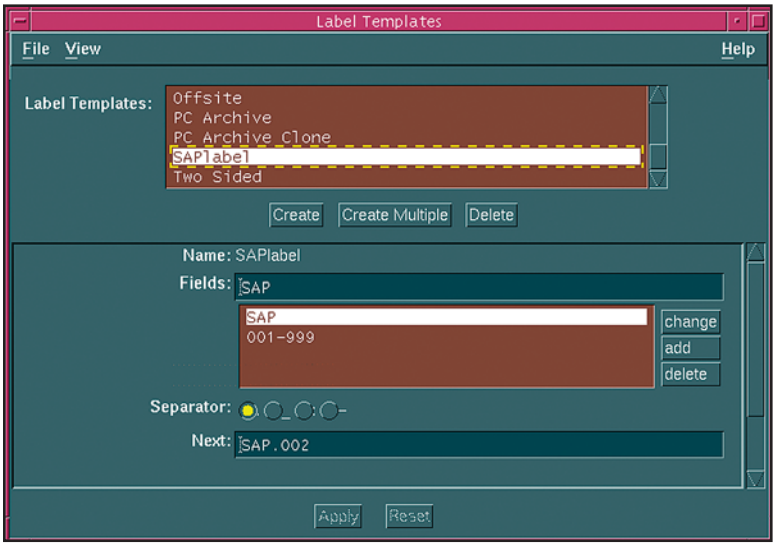

Figure 9. NetWorker label template interface.

The label template will be used to label the device for the pool that will be used to do the backup. The field "Next" will be automatically completed during creation.

> Create a new device for the pool to be created. Go to the menu Media > Devices… Choose the create function and complete the fields with the appropriate information. Create the device /dev/rmt/0hbn for the tape drive connected.

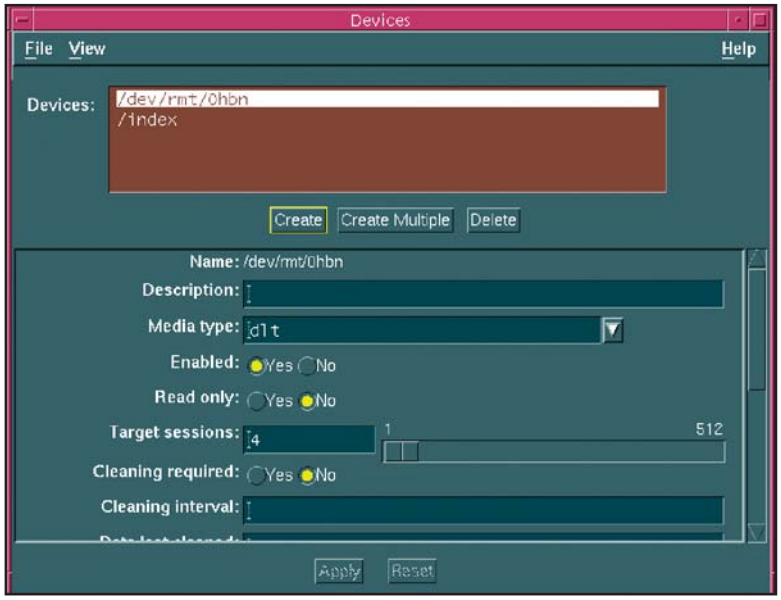

Figure 10. NetWorker devices interface.

Before configuring the new device, make sure that the tape drive is correctly configured in terms of hardware.

```
root@ossau: /usr/sbin/jbverify

 Jbverify is running on host ossau
 no enabled jukeboxes found in database.
 Processing stand-alone devices...
 Processing /dev/rmt/0hbn
 Finished processing /dev/rmt/0hbn
 Processing /index
 Finished processing /index
 Finished processing stand-alone devices.

*********************************************************************
                  Summary report of jbverify
                  ======= ====== == ========
Hostname    Device Handle     Blocksize   Jukebox          Drv No. Status
--------    -------------     ---------   -------          ------- ------
ossau       /dev/rmt/0hbn     98304       N/A               N/A      Pass
ossau       /index             32768       N/A               N/A      Pass
*********************************************************************
 Exiting jbverify successfully.
```

Note If needed, use the /usr/sbin/jbconfig tool to configure the Juke Box used.

> Create the new pool in charge of the backup operation. Go to the menu Media > Pools… Choose the create function and complete the fields with the appropriate information. Create the pool SAPpool for the SAPgroup using the device /dev/rmt/0hbn previously created for the client olinda.

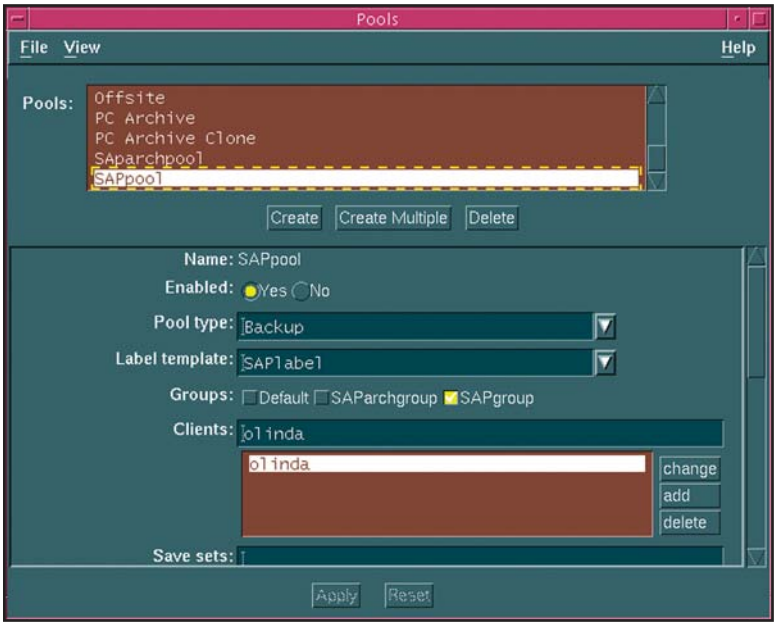

Figure 11. NetWorker pools interface.

Do not complete the Clients field yet. The client must be configured for the pool first. Use the NetWorker interface > Media Management > Manage Pools, go to the menu Operations > Create… and complete the fields with the appropriate information.

> Create the client that will have save sets to be backed up. Go to the menu Clients > Client Setup… Choose the create function and complete the fields with the appropriate information. Create the client olinda for the SAPgroup.

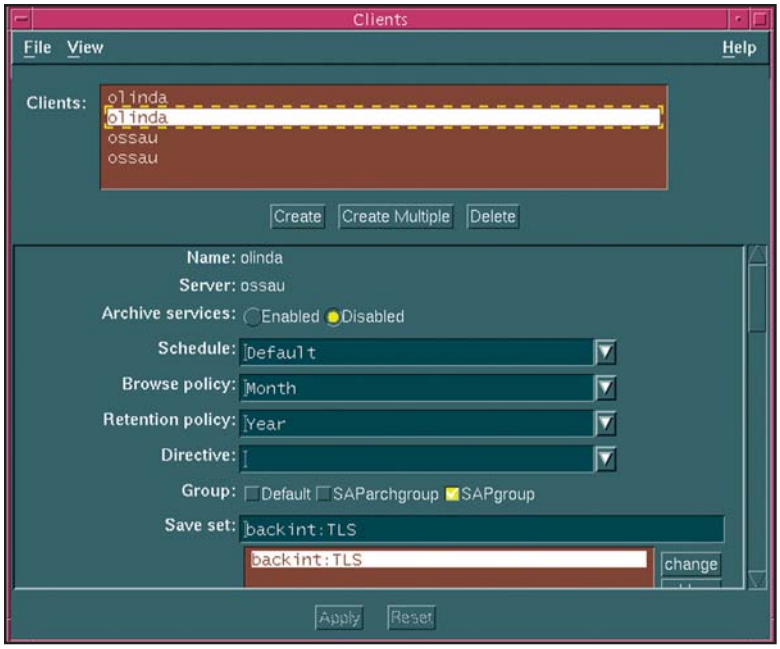

Figure 12. NetWorker clients interface.

Now that the client is configured for the pool NWKPool, modify the pool and complete the Clients field with the client name.

> Specify a device used by the Default group for the NetWorker indexes and bootstrap to be backed up (cannot be the same media as the one used to do the data backups, a disk space can be specified).

Complete the server's administrator list with the users that can administrate the backup server (at least user oratls@ossau).

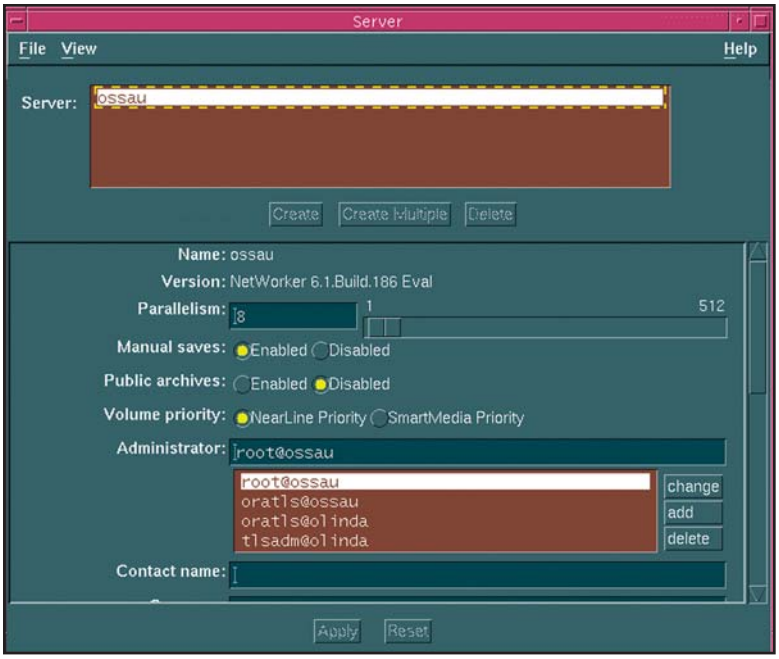

Figure 13. NetWorker server setup interface.

6 BACKUP AND RESTORE OPERATIONS

This section gives detailed steps on how to perform a backup or a restore of the SAP application using the Split Mirror Scenario.

This covers the following:

> SAP online/offline backup from a backup server
> Restore operations.

6.1 SAP ONLINE/OFFLINE SPLIT BACKUP USING A DEDICATED BACKUP SERVER

This part of the document describes how to perform online or offline backups using the SnapShot feature:

> NetWorker configuration
> SAP tools configuration
> Split and resync scripts.

6.1.1 NetWorker for SAP

Edit the parameter file **initTLS.utl** you copied in directory /oracle/TLS/dbs and complete with the following lines (refer to your configuration).

```
compress = yes
saveset = 50
parallelism = 8
pool = SAPpool
server = ossau
client = olinda
level_full = yes
```

Note **The server option indicates the hostname of the NetWorker server that the NetWorker Module will use for each save session. The client option indicates the client name that will be used for the client file index for the current backup. This facilitates the restore operation from the NetWorker restore interface.**

Configure the **initTLS.ora** file to activate the OPS$ mechanism. This file is located in /oracle/TLS/dbs

```
remote_os_authen = true
os_authen_prefix = OPS$
```

6.1.2 SAP tools

The parameter file used by SAP when performing backups is /oracle/TLS/dbs/**initTLS.sap**. This file can be configured for the online split backup or the offline split backup.

```
backup_type = online_split | offline_split
backup_mode = all
backup_dev_type = util_file
backup_root_dir = /oracle/TLS/sapbackup
compress_dir = /oracle/TLS/sapreorg
archive_copy_dir = /oracle/TLS/sapbackup
orig_db_home = /oracle/TLS
remote_host = olinda
remote_user = root
split_cmd = "/oracle/TLS/sapscript/snap_cmd [$]"
resync_cmd = "/oracle/TLS/sapscript/resync_cmd [$]"
util_par_file = /oracle/TLS/dbs/initTLS.utl
primary_db = TLS.WORLD
```

Note **If there is no additional backup server used, the primary_db must be set to LOCAL.**

6.1.3 Split and resynchronisation scripts

Copy the snap_cmd and resync_cmd scripts into directory /oracle/TLS/sapscript.

These scripts are referenced in the SAP parameter file initTLS.sap.

Note **The split_cmd and resync_cmd scripts are not allowed to display any output because messages are considered as error messages. Information messages must start with string #INFO.**

The snap_cmd script executes the following steps:

> On the production server, including the target LUNs
> On the production server, syncing the file systems
> On the production server, snapping the LUNs from the source to the target
> On the production server, excluding the target LUNs
> On the backup server, including the target LUNs
> On the backup server, mounting each file system.

Please refer to **Appendix H: SnapShot script** for the detailed script.

The usage of the SVA SnapShot does not require any resynchronization but the resync_cmd is required in the initTLS.sap file for online/offline split backups.

The resync_cmd script executes the following steps:

> On the backup server, dismounting each file system
> On the backup server, excluding the target LUNs.

Please refer to **Appendix I: Resynchronization script** for the detailed script.

Note **These scripts include command lines from SVA Path. But if you do not use SVA Path in your configuration, do not forget to comment all those command lines in both scripts. Please refer to the script itself for information about concerned lines.**

The paragraph **6.1.5** details the different way to perform backups. When the backup is initiated from the backup server, these scripts are located in the backup server directory. When the backup is launched from the production server, the split and resynchronization scripts used are located on the production server. Those scripts are identical in their algorithm but different in the use of the rsh commands. Refer to **Appendix H**: and **Appendix I**: for the snap_cmd and resync_cmd versions for the backup server and those for the production server.

Note **Do not forget to put the surootcmd executable in the directory where the scripts are located. This command is used to get root rights to execute some commands. You can try the sudo command: refer to http://www.sudo.ws/sudo/sudo.html.**

6.1.4 The main backup script

Create an SAP script that initiates the online or offline backup. This script will call the SAP backup tool BRBACKUP. This file is named brbackup_online or brbackup_offline in our example.

Copy this script in directory /oracle/TLS/sapscript.

See **Appendix F: Main Backup Script (Online Split Mirror)** and **Appendix G: Main Backup Script (Offline Split Mirror)** for a copy of the scripts used. Basically this script calls SAP brbackup with the following command line:

```
brbackup –u internal/password –p initTLS_online.sap –c –l E –d
util_file –r /oracle/TLS/dbs/initTLS.utl –t online_split
```

or

```
brbackup –u internal/password –p initTLS_offline.sap –c –l E –d
util_file –r /oracle/TLS/dbs/initTLS.utl –t offline_split
```

This script not only executes a brbackup command. It also mounts the shared file system of the production server specified in **5.2.3 Shared backup file-systems** and dismounts them after the backup termination.

All the results are reported in file: **/tmp/result_online or /tmp/result_offline.**

Note **You can also specify the file you want in the desired directory: change the ON_OFF_RESULT variable value in file /oracle/TLS/sapscript/brbackup_online or in file /oracle/TLS/sapscript/brbackup_offline.**

6.1.5 Backup execution overview

There are two main scenarios of backup: the online split backup and the offline split backup. But the backup operation can be done using different methods. This paragraph describes the possible methods:

> Make a manual backup using the main brbackup script
> Make a manual backup using the SAP brbackup command line
> Make a manual backup from SAPDBA.

The data backed up using one of these methods are effective only when the redo-logs are backed up. So the last section in this paragraph **(6.1.6 Backup the archive logs)** treats the backup of the archive logs.

Files Names	Directories on the Server
initTLS.utl	/oracle/TLS/dbs
initTLS_<online/offline>.sap	/oracle/TLS/dbs
brbackup_<online/offline>	/oracle/TLS/sapscript
snap_cmd	/oracle/TLS/sapscript
resync_cmd	/oracle/TLS/sapscript
lun_config	/oracle/TLS/sapscript
brbackup_<online/offline>	/oracle/TLS/sapscript (backup server only)

Table 1. File locations for the backup operation.

6.1.5.1 Manual backup using the main brbackup script

The main brbackup script is executed on the backup server. Before making further operations, make sure the operations below are located in the right directories on the **backup server**, refer to **Table 1**.

After all the configuration steps completed, perform the following (the principle is identical for an online and an offline split backup):

> Change the lun_config file to reflect your configuration. Please refer to **Appendix E: LUN configuration file** for an example of LUN configuration file. Copy this file in directory /oracle/TLS/sapscript.

> Execute the main script on the backup server. Refer to

```
root@ossau: /oracle/TLS/sapscript/brbackup_online
```

> View the result file specified in the brbackup_online script.

```
root@ossau: tail –f /tmp/result_online
```

Please refer to **Appendix J: Example of online successful backup** for the detailed logs of a successful online backup.

Please refer to **Appendix K: Example of offline successful backup** for the detailed logs of a successful offline backup.

6.1.5.2 Manual backup using brbackup command line

Although it is more efficient to use the previous script to perform a backup, you can perform a backup using SAP brbackup command from a command line on the **backup server**, refer to **Table 1**.

Before executing the brbackup command, mount the shared file systems from the production server:

```
root@ossau: mount olinda:/oracle/TLS/sapbackup /oracle/TLS/sapbackup
root@ossau: mount olinda:/oracle/TLS/sapreorg /oracle/TLS/sapreorg
```

Then, execute the brbackup command:

```
root@ossau: su – oratls
Sun Microsystems Inc.   SunOS 5.8       Generic February 2000
ossau:oratls 1% brbackup –u internal/password -p
/oracle/TLS/dbs/initTLS_online.sap -c -l E -d util_file -r
/oracle/TLS/dbs/initTLS.utl -t  online_split
```

Note **The principle is the same for an offline backup. Make sure that the initTLS_offline.sap and the initTLS.utl files are properly configured.**

After the backup is finished, dismount the shared file-system.

```
root@ossau: umount /oracle/TLS/sapbackup
root@ossau: umount /oracle/TLS/sapreorg
```

6.1.5.3 Manual backup from SAPDBA

You can also perform backup using SAPDBA from the production server.

Before executing the following operations, make sure the scripts below are saved in the right directories on the production server, refer to **Table 1**.

Note **Be careful when choosing the snap_cmd and resync_cmd scripts. Those to be used here are different from the ones used in the 6.1.5.1 and in the 6.1.5.2 paragraphs.**

Then, execute the SAP's SAPDBA database utility:

```
root@olinda: su - oratls
Sun Microsystems Inc.   SunOS 5.7       Generic October 1998
olinda:oratls 1% sapdba
Enter password (8 char. max.) for 'system' ==>
```

The general menu of the SAPDBA utility appears **(Figure 14)**.

```
--------------------------------------------------------------------------------
                SAPDBA V4.6C - SAP Database Administration
--------------------------------------------------------------------------------

                  ORACLE version: 8.0.6.0.0
                  ORACLE_SID    : TLS
                  ORACLE_HOME   : /oracle/TLS
                  DATABASE      : open
                  SAPR3         : 46C, 8 times connected

        a - Startup/Shutdown instance    h - Backup database
        b - Instance information         i - Backup offline redo logs
        c - Tablespace administration    j - Restore/Recovery
        d - Reorganization               k - DB check/verification
        e - Export/import                l - Show/Cleanup
        f - Archive mode                 m - User and Security
        g - Additional functions

        q - Quit

        Please select ==>
```

Figure 14. SAPDBA general menu.

> Select the Backup database function (h). The backup options menu is displayed **(Figure 15)**.

> Change the default values of the options to specify the backup you want to perform (Parameter file, backup device type, backup type…). When selecting a parameter file such as initTLS_online.sap, the displayed values correspond to the ones specified in the parameter file. If you change them, the new entered values are effective.

> Initiate the backup operation by selecting the Start BRBACKUP function (S).

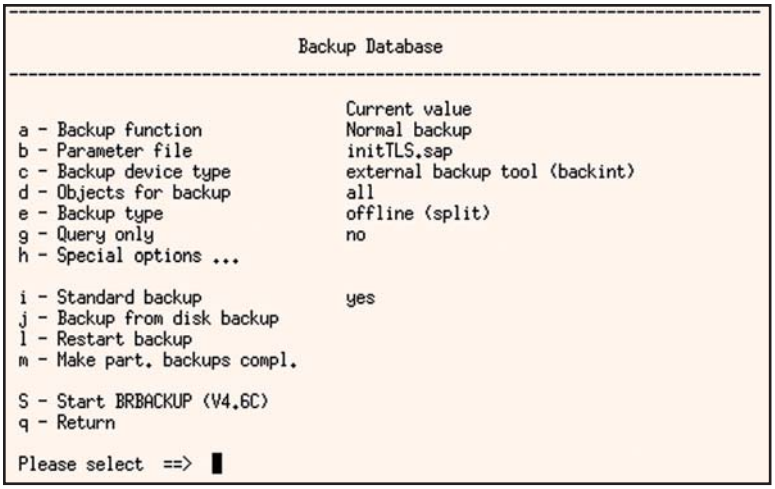

Figure 15. Backup database menu.

6.1.6 Backup the archive logs

The archive logs can be backed up from the primary host using SAP's SAPDBA utility or the SAP brarchive command from the command line.

6.1.6.1 Archive logs from the SAPDBA utility

To backup the archive logs using the SAPDBA utility:

> Start the SAPDBA program on the primary host to get access to the general menu **(Figure 14)**.

> Select i – Backup offline redo logs from this menu to have the following menu:

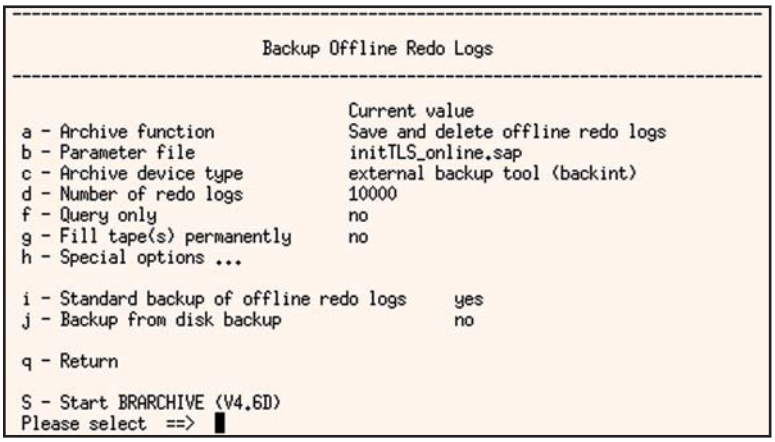

Figure 16. Backup database menu.

> Choose the Archive function (a).

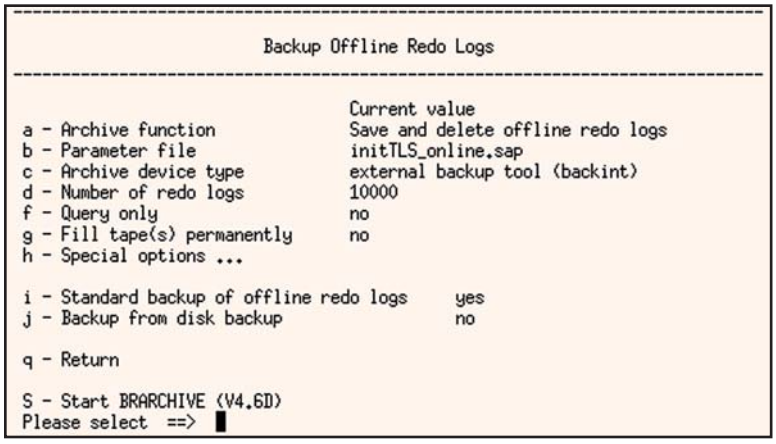

Figure 17. Backup offline redo logs menu.

> From the Archive function window, select Save Archive Logs (a) or Save and Delete (d)

> From the Backup archive Logs windows, select Start BRARCHIVE (S)

> Follow the instructions.

6.1.6.2 Archive logs from the brarchive command

To backup the archive logs from the command line:

> Enter the following line on the primary host:

```
olinda : oratls 1% brarchive –u internal/password -s –p initTLS.sap -
d util_file –r initTLS.utl
```

> Deleting the archive logs from the Oracle server after a backup operation frees space in the /oracle/TLS/saparch directory.

```
olinda : oratls 1% brarchive –u internal/password -sd –p initTLS.sap
-d util_file –r initTLS.utl
```

6.2 HOW TO RESTORE?

The procedure to restore files or directories is the same using either the single-server configuration or the dedicated backup server architecture.

The operations described below restore the files in their last backup. The redo logs are needed to properly recover the database.

6.2.1 Restore versus recovery

For Oracle, there is a difference between restoring and recovering a database.

> Restore means replacing files of the database from the backup media
> Recover means applying the redo logs to roll forward the redo log transaction to make the database consistent.

This document uses this Oracle terminology.

6.2.2 Restore from SAPDBA

To restore a file or directory, perform the following:

> Start the SAPDBA program on the primary host to display the general menu **(Figure 14)**
> Choose j - Restore/Recovery
> From the Restore/Recovery menu, select a - Partial restore and complete recovery
> Choose the Automatic Recovery function (g) and complete the recovery options. When the options are correctly set, choose to quit the current menu (q)
> Choose (g) again to start the Restore and Recovery of the database
> Follow the instructions.

This method will restore all the missing files and will recover the database.

Note **It is recommended to perform an Automatic Recovery of the database rather than manual restore and recovery.**

6.2.3 Restore from the NetWorker interface

From the NetWorker client olinda, perform the following:

> Launch the menu to make the restore by typing: nwrecover
> The NetWorker recover interface is displayed **(Figure 18)**
> For the client olinda, choose the files to be restored and mark them.

Note **If the client displayed is not the one you are interested in, go to the menu Change > Clients.**

> Then click on the Start button to initiate the restore operation.
> A window will appear to inform you of the restore status **(Figure 19)**.

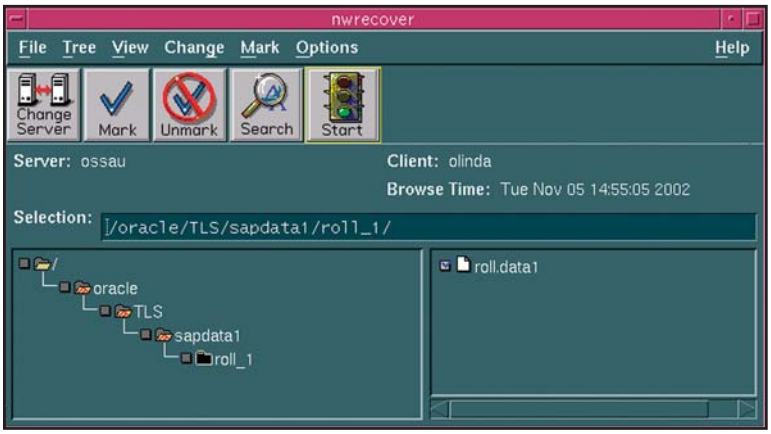

Figure 18. NetWorker restore interface.

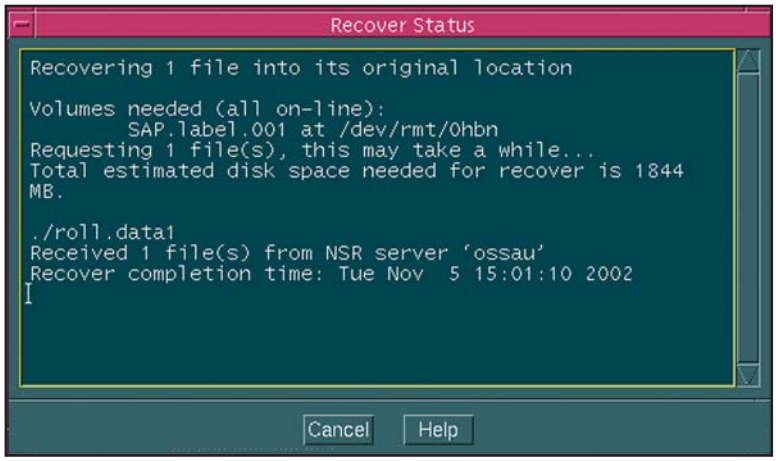

Figure 19. NetWorker restore status interface.

When the restore is complete, the message "Recover completion" is displayed.

Note **This method can only be used if you know which files have to be restored.**

After the restoration of the missing files, you **must recover the database** in order to make it consistent. Perform the same operations as in **6.2.2 Restore from SAPDBA** to execute an automatic recovery of the database.

7 CRITICAL ANALYSIS

Advantages	Drawbacks
Quick: Either performing online backup or offline backup, the time the database is in backup mode or shutdown is considerably reduced (about 3 seconds per LUNs used).	The target LUNs must be correctly created and configured before executing backup commands. It is important that the LUN configuration file contains the correct values.
The data can be backed up to tape using the same or another server that will mount the SVA snapshot.	
Simple: There is only one SVA command per file system in order to backup all database files.	
No additional physical space has to be provided when creating target LUNs.	

THE ONLINE SPLIT BACKUP PROCESS

To execute the snapshot we need to put the database in backup mode (online backup) during approximately 2–5 seconds, no matter how much data is copied. You save a lot of time.

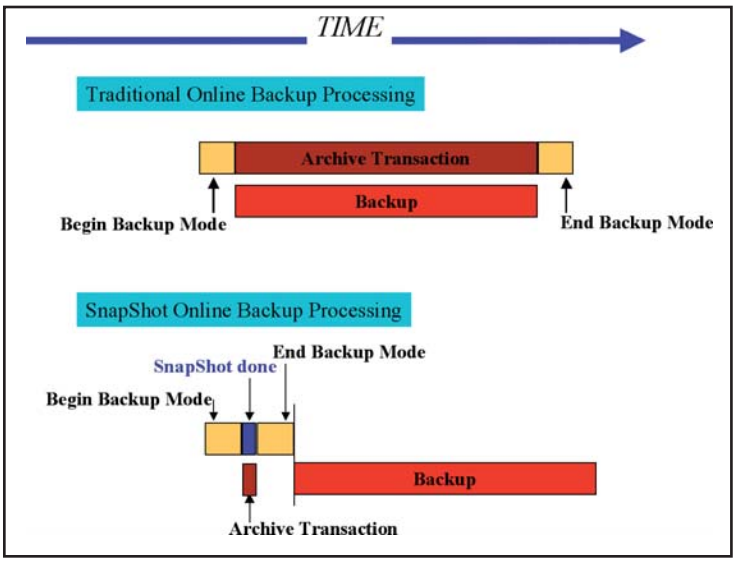

Figure 20. The SnapShot real advantage.

The archive logs contain all the modifications made on the database while this one is in backup mode. For a traditional online backup, the database is in backup mode during the whole backup operation. The archive logs in a snapshot online backup scenario are less bulky, thanks to the snapshot.

8 APPENDIX

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

APPENDIX A: TARGET LUNS CREATION SCRIPT

```
/opt/storagetek/SVAA3.1.0/bin/sibadmin<<EOF
defdev -subsys NOVA -fdid 253 -devtyp SCSIA -scsiblks 8192 -scsiaddr 2.0.51 -name
SAPda1T -fcapa 14g -fdidpool 201:2FF
defdev -subsys NOVA -fdid 259 -devtyp SCSIA -scsiblks 8192 -scsiaddr 2.0.52 -name
SAPda2T -fcapa 14g -fdidpool 201:2FF
defdev -subsys NOVA -fdid 25f -devtyp SCSIA -scsiblks 8192 -scsiaddr 2.0.53 -name
SAPda3T -fcapa 14g -fdidpool 201:2FF
defdev -subsys NOVA -fdid 265 -devtyp SCSIA -scsiblks 8192 -scsiaddr 2.0.54 -name
SAPda4T -fcapa 14g -fdidpool 201:2FF
defdev -subsys NOVA -fdid 26b -devtyp SCSIA -scsiblks 8192 -scsiaddr 2.0.55 -name
SAPda5T -fcapa 14g -fdidpool 201:2FF
defdev -subsys NOVA -fdid 271 -devtyp SCSIA -scsiblks 8192 -scsiaddr 2.0.56 -name
SAPda6T -fcapa 14g -fdidpool 201:2FF
EOF
```

Modify this script with your values of:

> SVA subsystem name (e.g. NOVA)
> Each LUNs fdid (e.g. 253)
> SVA Domain, Target and LUN (e.g. 2.0.51)
> LUNs name (e.g. SAPda1T)
> Fdid pool to use (e.g. 201:2FF).

Note **If the values you specify are already used, errors occur during the execution of this script.**

APPENDIX B: SCRIPT TO FORMAT TARGET LUNS

```
format -f format_label.cmd -d c1t9d51
format -f format_label.cmd -d c1t9d52
format -f format_label.cmd -d c1t9d53
format -f format_label.cmd -d c1t9d54
format -f format_label.cmd -d c1t9d55
format -f format_label.cmd -d c1t9d56
```

format_label.cmd

```
label
```

APPENDIX C: SOLARIS/ETC/SYSTEM ADD-ON

```
*******************************************************************
*
*         Necessary modifications for SAP
*
*******************************************************************
*
*         Shared memory parameters for SAP
*
set shmsys:shminfo_shmmin=1
set shmsys:shminfo_shmmax=536870912

set shmsys:shminfo_shmmni=256
set shmsys:shminfo_shmseg=100

*
*         Semaphores parameters set for SAP
*
set semsys:seminfo_semmap=256
set semsys:seminfo_semmni=4096
set semsys:seminfo_semmns=4096
set semsys:seminfo_semmnu=4096
set semsys:seminfo_semume=64
set semsys:seminfo_semmsl=2048
set semsys:seminfo_semopm=50

*
*         IPC message queues parameters for SAP
*
```

```
set msgsys:msginfo_msgmap=2048
set msgsys:msginfo_msgmax=8192
set msgsys:msginfo_msgmnb=16384
set msgsys:msginfo_msgmni=50
set msgsys:msginfo_msgssz=32
set msgsys:msginfo_msgtql=2048
set msgsys:msginfo_msgseg=32767

*
*          File descriptor parameters set for SAP
*
set rlim_fd_cur=1024
set rlim_fd_max=1536
*
******************************************************************************
*     End of modifications for SAP
******************************************************************************
```

APPENDIX D: NET*8 CONFIGURATION FILES

listener.ora on the production server

```
LISTENER =
  (ADDRESS_LIST =
         (ADDRESS=
           (PROTOCOL=IPC)
           (KEY= TLS.WORLD)
         )
         (ADDRESS=
           (PROTOCOL=IPC)
           (KEY= TLS)
         )
         (ADDRESS =
           (PROTOCOL = TCP)
           (HOST = olinda)
           (PORT = 1527)
         )
  )
STARTUP_WAIT_TIME_LISTENER = 0
CONNECT_TIMEOUT_LISTENER = 10
TRACE_LEVEL_LISTENER = OFF
SID_LIST_LISTENER =
  (SID_LIST =
    (SID_DESC =
      (SDU = 32768)
      (SID_NAME = TLS)
      (ORACLE_HOME = /oracle/TLS)
    )
  )
```

tnsnames.ora on the production server

```
TLS.WORLD=
  (DESCRIPTION =
    (SDU = 32768)
    (ADDRESS_LIST =
        (ADDRESS =
```

```
          (PROTOCOL = TCP)
          (HOST = olinda)
          (PORT = 1527)
        )
    )
    (CONNECT_DATA =
       (SID = TLS)
       (GLOBAL_NAME = TLS.WORLD)
    )
  )
```

listener.ora on the backup server

```
LISTENER =
  (ADDRESS_LIST =
        (ADDRESS=
          (PROTOCOL=IPC)
          (KEY= TLS.BACKUP)
        )
        (ADDRESS=
          (PROTOCOL=IPC)
          (KEY= TLS)
        )
        (ADDRESS =
          (PROTOCOL = TCP)
          (HOST = ossau)
          (PORT = 1527)
        )
  )
STARTUP_WAIT_TIME_LISTENER = 0
CONNECT_TIMEOUT_LISTENER = 10
TRACE_LEVEL_LISTENER = OFF
SID_LIST_LISTENER =
  (SID_LIST =
    (SID_DESC =
      (SDU = 32768)
      (SID_NAME = TLS)
      (ORACLE_HOME = /oracle/TLS)
    )
  )
```

tnsnames.ora on the backup server

```
TLS.WORLD=
  (DESCRIPTION =
    (SDU = 32768)
    (ADDRESS_LIST =
        (ADDRESS =
          (PROTOCOL = TCP)
          (HOST = olinda)
          (PORT = 1527)
        )
    )
    (CONNECT_DATA =
       (SID = TLS)
       (GLOBAL_NAME = TLS.WORLD)
```

```
    )
  )

TLS.BACKUP=
  (DESCRIPTION =
    (SDU = 32768)
    (ADDRESS_LIST =
        (ADDRESS =
          (PROTOCOL = TCP)
          (HOST = ossau
          (PORT = 1527)
        )
    )
    (CONNECT_DATA =
       (SID = TLS)
       (GLOBAL_NAME = TLS.BACKUP)
    )
  )
```

sqlnet.ora on both servers

```
AUTOMATIC_IPC = ON
TRACE_LEVEL_CLIENT = OFF
SQLNET.EXPIRE_TIME = 0
NAMES.DEFAULT_DOMAIN = WORLD
NAME.DEFAULT_ZONE = WORLD
```

protocol.ora on both servers

```
tcp.nodelay = true
```

APPENDIX E: LUN CONFIGURATION FILE

lun_config

```
#
#
#      PRIMARY HOST                    BACKUP HOST
#=======================         ==========================
#SOURCE         TARGET            DISKNAME          MOUNT POINT
#
c1t8d31         c1t8d51           c1t9d51           /oracle/TLS/sapdata1
c1t8d32         c1t8d52           c1t9d52           /oracle/TLS/sapdata2
c1t8d33         c1t8d53           c1t9d53           /oracle/TLS/sapdata3
c1t8d34         c1t8d54           c1t9d54           /oracle/TLS/sapdata4
c1t8d35         c1t8d55           c1t9d55           /oracle/TLS/sapdata5
c1t8d36         c1t8d56           c1t9d56           /oracle/TLS/sapdata6
c1t8d37         c1t8d57           c1t9d57           /oracle/TLS/sapdata7
```

The SnapShot and Resynchronization scripts use this configuration file. Before executing the main backup script (brbackup_online or brbackup_offline), make sure that this script reflects exactly the current configuration.

APPENDIX F: MAIN BACKUP SCRIPT (ONLINE SPLIT MIRROR)

brbackup_online

```
#!/bin/sh

ON_OFF_INIT_SAP=/oracle/TLS/dbs/initTLS_online.sap
ON_OFF_INIT_UTL=/oracle/TLS/dbs/initTLS.utl
ON_OFF_RESULT=/tmp/result_online
ON_OFF_SPLIT=online_split

echo "#INFO ******************** Calling brbackup_online `date` **********************"
> $ON_OFF_RESULT
#
#Mount the sapbackup and the sapreorg directories from the production server
#
echo "#INFO Mounting the sapbackup directory from the production server to the backup
server">> $ON_OFF_RESULT
mount olinda:/oracle/TLS/sapbackup /oracle/TLS/sapbackup
echo "#INFO Mounting the sapreorg directory from the production server to the backup
server">> $ON_OFF_RESULT
mount olinda:/oracle/TLS/sapreorg /oracle/TLS/sapreorg

RETURN_STATUS=0

EX_CMD_LINE=""

EX_CMD_LINE="$EX_CMD_LINE setenv SAPDATA_HOME /oracle/TLS; setenv SAPBACKUP
/oracle/TLS/sapbackup; setenv SAPARCH /oracle/TLS/sap
arch; setenv SAPREORG /oracle/TLS/sapreorg;"

#
# Full online backup with dynamic BEGIN/END BACKUP switch
#
CMD_LINE="$EX_CMD_LINE /usr/sap/TLS/SYS/exe/run/brbackup -u internal/password -p
${ON_OFF_INIT_SAP} -c -l E -d util_file -r ${ON_O
FF_INIT_UTL} -t  ${ON_OFF_SPLIT} "

#
# The username on the "su" command needs to be replaced with the correct user
# name.
#
echo "Execute $CMD_LINE" >> $ON_OFF_RESULT
su - oratls -c "$CMD_LINE" >> $ON_OFF_RESULT

RETURN_STATUS=$?
echo "Status: $RETURN_STATUS " >> $ON_OFF_RESULT

#
# UnMount the sapbackup and sapreorg directories from the backup server
#
echo "#INFO UnMounting the sapbackup directory from the backup server">>
$ON_OFF_RESULT
umount /oracle/TLS/sapbackup
echo "#INFO UnMounting the sapreorg directory from the backup server">> $ON_OFF_RESULT
umount /oracle/TLS/sapreorg

echo "Status: $RETURN_STATUS " >> $ON_OFF_RESULT
exit $RETURN_STATUS
```

APPENDIX G: MAIN BACKUP SCRIPT (OFFLINE SPLIT MIRROR)

brbackup_offline

```
#!/bin/sh

ON_OFF_INIT_SAP=/oracle/TLS/dbs/initTLS_offline.sap
ON_OFF_INIT_UTL=/oracle/TLS/dbs/initTLS.utl
ON_OFF_RESULT=/tmp/result_offline
ON_OFF_SPLIT=offline_split

echo "#INFO ******************** Calling brbackup_offline `date`
**********************" > $ON_OFF_RESULT
#
#Mount the sapbackup and the sapreorg directories from the production server
#
echo "#INFO Mounting the sapbackup directory from the production server to the backup
server">> $ON_OFF_RESULT
mount olinda:/oracle/TLS/sapbackup /oracle/TLS/sapbackup
echo "#INFO Mounting the sapreorg directory from the production server to the backup
server">> $ON_OFF_RESULT
mount olinda:/oracle/TLS/sapreorg /oracle/TLS/sapreorg

RETURN_STATUS=0

EX_CMD_LINE=""

EX_CMD_LINE="$EX_CMD_LINE setenv SAPDATA_HOME /oracle/TLS; setenv SAPBACKUP
/oracle/TLS/sapbackup; setenv SAPARCH /oracle/TLS/sap
arch; setenv SAPREORG /oracle/TLS/sapreorg;"

#
# Full offline backup with dynamic BEGIN/END BACKUP switch
#
CMD_LINE="$EX_CMD_LINE /usr/sap/TLS/SYS/exe/run/brbackup -u internal/garonne -p
${ON_OFF_INIT_SAP} -c -l E -d util_file -r ${ON_O
FF_INIT_UTL} -t  ${ON_OFF_SPLIT} "

#
# The username on the "su" command needs to be replaced with the correct user
# name.
#
echo "Execute $CMD_LINE" >> $ON_OFF_RESULT
su - oratls -c "$CMD_LINE" >> $ON_OFF_RESULT

RETURN_STATUS=$?
echo "Status: $RETURN_STATUS " >> $ON_OFF_RESULT

#
# UnMount the sapbackup and sapreorg directories from the backup server
#
echo "#INFO UnMounting the sapbackup directory from the backup server">>
$ON_OFF_RESULT
umount /oracle/TLS/sapbackup
echo "#INFO UnMounting the sapreorg directory from the backup server">> $ON_OFF_RESULT
umount /oracle/TLS/sapreorg

echo "Status: $RETURN_STATUS " >> $ON_OFF_RESULT
exit $RETURN_STATUS
```

APPENDIX H: SNAPSHOT SCRIPT

snap_cmd on the backup server

```
#!/bin/sh

#
# Script Variables
#

STK_SCRIPTS_DIR=`dirname $0`
SIBADMIN=/opt/storagetek/SVAA3.1.0/bin/sibadmin
PRODUCTION_SERVER=olinda
SVA_LUNCONFIG_FILE=$STK_SCRIPTS_DIR/lun_config

#
# Logging methods
#
logerrors()
{
    case $* in
        "")
        while read line
        do
                logger -p daemon.notice -t `basename $0` $line
                echo \#INFO $line
        done
        ;;
        *)
                logger -p daemon.notice -t `basename $0` $*
                echo \#INFO $*
        ;;
    esac
}

loginfo()
{
    case $* in
        "")
        while read line
        do
                logger -p daemon.notice -t "`basename $0` NOTICE" $line
                echo \#INFO $line
        done
        ;;
        *)
                logger -p daemon.notice -t "`basename $0` NOTICE" $*
                echo \#INFO $*
        ;;
    esac
}

mysurootcmd()
{
        if [ `id | grep -c "uid=0"` = "1" ]
        then
         # I'm root user
         $*
        else
         # I'm not root user
```

```
          echo . | $STK_SCRIPTS_DIR/surootcmd "$*"
         fi

         return
}

loginfo Beginning of script: snap_cmd

loginfo LUN Configuration File:
egrep -v '^#|^$' $SVA_LUNCONFIG_FILE |
while read source target backupdisk mountpoint
do
     echo source=$source, target=$target, backupdisk=$backupdisk,
mountpoint=$mountpoint
done | loginfo

#
# Checking access to the PRODUCTION_SERVER (root)
#
rsh -n -l root $PRODUCTION_SERVER ls > /dev/null 2>/dev/null
if [ $? -ne 0 ]
then
     logerrors "rsh $PRODUCTION_SERVER failed! Check file /.rhosts"
     exit 1
fi

#
# If NOT USING SVAPATH -> comment these lines
# On the Production Server, Including each target LUN from the server view
loginfo On Production Server, Including each target LUN from the server view
egrep -v '^#|^$' $SVA_LUNCONFIG_FILE |
while read source target backupdisk mountpoint
do
     target_lun=`rsh $PRODUCTION_SERVER -n -l root /usr/sbin/setsp|grep ${target}|awk
'{print $1}'|cut -dd -f2`
     echo /usr/sbin/setsp -l${target_lun} -e0
done | rsh $PRODUCTION_SERVER -l root sh | logerrors

#
# On the Primary Host, Syncing the file systems
#
loginfo On Production Server, Syncing the file systems
rsh -n -l root $PRODUCTION_SERVER "sync; sync"

#
# On the Primary Host, Snapping each individual LUN
#
loginfo On Production Server, Snapping each individual LUN
egrep -v '^#|^$' $SVA_LUNCONFIG_FILE |
while read source target backupdisk mountpoint
do
     echo snapshot -source /dev/rdsk/${source}s2 -target /dev/rdsk/${target}s2
done | rsh $PRODUCTION_SERVER -l root $SIBADMIN 2>&1 | grep ERROR | logerrors

#
# If NOT USING SVAPATH -> comment these lines
# On the Production Server, Excluding each target LUN from the server view
#
```

```
loginfo On Production Server, Excluding each target LUN from the server view
egrep -v '^#|^$' $SVA_LUNCONFIG_FILE |
while read source target backupdisk mountpoint
do
    target_lun=`rsh $PRODUCTION_SERVER -n -l root /usr/sbin/setsp|grep ${target}|awk
'{print $1}'|cut -dd -f2`
    echo /usr/sbin/setsp -l${target_lun} -e1
done | rsh $PRODUCTION_SERVER -l root sh | logerrors

#
# If NOT USING SVAPATH -> comment these lines
# On the Backup Server, Including each LUN to the server view
#
loginfo On Backup Server, Including each LUN to the server view
egrep -v '^#|^$' $SVA_LUNCONFIG_FILE |
while read source target backupdisk mountpoint
do
    backuplun=`mysurootcmd /usr/sbin/setsp 2>&1 | grep ${backupdisk}|awk '{print
$1}'|cut -dd -f2`
    mysurootcmd /usr/sbin/setsp -l${backuplun} -e0 1>/dev/null 2>&1
done | logerrors

#
# On the Backup Host, Mounting each file system
#
loginfo On Backup Server, Checking and mounting each file system
egrep -v '^#|^$' $SVA_LUNCONFIG_FILE |
while read source target backupdisk mountpoint
do
    needs_checking=`mysurootcmd /usr/sbin/fsck -m /dev/rdsk/${backupdisk}s2 2>&1 |
grep -c needs`
    if [ $needs_checking -ne 0 ]
    then
        loginfo Checking /dev/rdsk/${backupdisk}s2
        mysurootcmd /usr/sbin/fsck -y /dev/rdsk/${backupdisk}s2 1>/dev/null 2>&1
    fi
    mysurootcmd /usr/sbin/mount /dev/dsk/${backupdisk}s2 ${mountpoint} > /dev/null
2>&1
done

loginfo Script successfully completed.
```

snap_cmd on the production server

```
#!/bin/sh

#
# Script Variables
#

STK_SCRIPTS_DIR=`dirname $0`
SIBADMIN=/opt/storagetek/SVAA3.1.0/bin/sibadmin
PRODUCTION_SERVER=olinda
SVA_LUNCONFIG_FILE=$STK_SCRIPTS_DIR/lun_config

#
# Logging methods
#
```

```
logerrors()
{
    case $* in
        "")
        while read line
        do
                logger -p daemon.notice -t `basename $0` $line
                echo \#INFO $line
        done
        ;;
        *)
                logger -p daemon.notice -t `basename $0` $*
                echo \#INFO $*
        ;;
    esac
}

loginfo()
{
    case $* in
        "")
        while read line
        do
                logger -p daemon.notice -t "`basename $0` NOTICE" $line
                echo \#INFO $line
        done
        ;;
        *)
                logger -p daemon.notice -t "`basename $0` NOTICE" $*
                echo \#INFO $*
        ;;
    esac
}

mysurootcmd()
{
        if [ `id | grep -c "uid=0"` = "1" ]
        then
         # I'm root user
         $*
        else
         # I'm not root user
         echo . | $STK_SCRIPTS_DIR/surootcmd "$*"
        fi

        return
}

loginfo Beginning of script: snap_cmd

loginfo LUN Configuration File:
/usr/bin/egrep -v '^#|^$' $SVA_LUNCONFIG_FILE |
while read source target backupdisk mountpoint
do
    echo source=$source, target=$target, backupdisk=$backupdisk,
mountpoint=$mountpoint
done | loginfo
```

```
#
# Checking access to the BACKUP_SERVER (root)
#
/usr/bin/rsh -n -l root $BACKUP_SERVER ls > /dev/null 2>/dev/null
if [ $? -ne 0 ]
then
     logerrors "/usr/bin/rsh $BACKUP_SERVER failed! Check file /.rhosts"
     exit 1
fi

#
# If NOT USING SVAPATH -> comment these lines
# On the Production Server, Including each target LUN from the server view
#
loginfo On Production Server, Including each target LUN from the server view
/usr/bin/egrep -v '^#|^$' $SVA_LUNCONFIG_FILE |
while read source target backupdisk mountpoint
do
     target_lun=`mysurootcmd /usr/sbin/setsp 2>&1 | /usr/bin/grep ${target}|awk '{print
$1}'|cut -dd -f2`
     mysurootcmd /usr/sbin/setsp -l${target_lun} -e0 1>/dev/null 2>&1
done | logerrors

#
# On the Primary Host, Syncing the file systems
#
loginfo On Production Server, Syncing the file systems
/usr/sbin/sync;/usr/sbin/sync

#
# On the Primary Host, Snapping each individual LUN
#
loginfo On Production Server, Snapping each individual LUN
/usr/bin/egrep -v '^#|^$' $SVA_LUNCONFIG_FILE |
while read source target backupdisk mountpoint
do
     echo snapshot -source /dev/rdsk/${source}s2 -target /dev/rdsk/${target}s2
done | mysurootcmd $SIBADMIN 1>/dev/null 2>&1 | /usr/bin/grep ERROR | logerrors

#
# If NOT USING SVAPATH -> comment these lines
# On the Production Server, Excluding each target LUN from the server view
#
loginfo On Production Server, Excluding each target LUN from the server view
/usr/bin/egrep -v '^#|^$' $SVA_LUNCONFIG_FILE |
while read source target backupdisk mountpoint
do
     target_lun=`mysurootcmd /usr/sbin/setsp 2>&1 | /usr/bin/grep ${target}|awk '{print
$1}'|cut -dd -f2`
     mysurootcmd /usr/sbin/setsp -l${target_lun} -e1 1>/dev/null 2>&1
done | logerrors

#
# If NOT USING SVAPATH -> comment these lines
# On the Backup Server, Including each LUN to the server view
#
loginfo On Backup Server, Including each LUN to the server view
/usr/bin/egrep -v '^#|^$' $SVA_LUNCONFIG_FILE |
while read source target backupdisk mountpoint
```

```
do
    target_lun=`rsh $BACKUP_SERVER -n -l root /usr/sbin/setsp|grep ${backupdisk}|awk
'{print $1}'|cut -dd -f2`
    echo /usr/sbin/setsp -l${target_lun} -e0
done | rsh $BACKUP_SERVER -l root sh | logerrors

#
# On the Backup Host, Mounting each file system
#
loginfo On Backup Server, Checking and mounting each file system
/usr/bin/egrep -v '^#|^$' $SVA_LUNCONFIG_FILE |
while read source target backupdisk mountpoint
do
    needs_checking=`/usr/bin/rsh $BACKUP_SERVER -n -l root /usr/sbin/fsck -m
/dev/rdsk/${backupdisk}s2 2>&1 | /usr/bin/grep -c needs`
    if [ $needs_checking -ne 0 ]
    then
        loginfo Checking /dev/rdsk/${backupdisk}s2
        /usr/bin/rsh $BACKUP_SERVER -n -l root /usr/sbin/fsck -y
/dev/rdsk/${backupdisk}s2 1>/dev/null 2>&1
    fi
    /usr/bin/rsh $BACKUP_SERVER -n -l root /usr/sbin/mount /dev/dsk/${backupdisk}s2
${mountpoint} > /dev/null 2>&1
done

loginfo Script successfully completed.
```

APPENDIX I: RESYNCHRONIZATION SCRIPT

resync_cmd on the backup server

```
#!/bin/sh

#
# Script Variables
#
STK_SCRIPTS_DIR=`dirname $0`
SVA_LUNCONFIG_FILE=$STK_SCRIPTS_DIR/lun_config

#
# Logging methods
#
logerrors()
{
    case $* in
        "")
        while read line
        do
                logger -p daemon.notice -t `basename $0` $line
                echo \#INFO $line
        done
        ;;
        *)
                logger -p daemon.notice -t `basename $0` $*
                echo \#INFO $*
        ;;
    esac
}
```

```
loginfo()
{
    case $* in
        "")
        while read line
        do
                logger -p daemon.notice -t "`basename $0` NOTICE" $line
                echo \#INFO $line
        done
        ;;
        *)
                logger -p daemon.notice -t "`basename $0` NOTICE" $*
                echo \#INFO $*

        ;;
    esac
}


mysurootcmd()
{
        if [ `id | grep -c "uid=0"` = "1" ]
        then
         # I'm root user
         $*
        else
         # I'm not root user
         echo . | $STK_SCRIPTS_DIR/surootcmd "$*"
        fi
}


loginfo Beginning of script: resync_cmd

loginfo LUN Configuration File:
egrep -v '^#|^$' $SVA_LUNCONFIG_FILE |
while read source target backupdisk mountpoint
do
    echo source=$source, target=$target, backupdisk=$backupdisk,
mountpoint=$mountpoint
done | loginfo


#
# On the Backup Host, Un-Mounting each file system
#
loginfo On Backup Server, unmounting each file system
egrep -v '^#|^$' $SVA_LUNCONFIG_FILE |
while read source target backupdisk mountpoint
do
    mysurootcmd /usr/sbin/umount ${mountpoint} > /dev/null 2>&1
done

#
# If NOT USING SVAPATH -> comment these lines
# On the Backup Server, Excluding each LUN to the server view
#
loginfo On Backup Server, Excluding each LUN to the server view
egrep -v '^#|^$' $SVA_LUNCONFIG_FILE |
```

```
while read source target backupdisk mountpoint
do
    backuplun=`mysurootcmd /usr/sbin/setsp 2>&1 | grep ${backupdisk}|awk '{print
$1}'|cut -dd -f2`
    mysurootcmd /usr/sbin/setsp -l${backuplun} -e1 1>/dev/null 2>&1
done | logerrors

loginfo Script successfully completed.
```

resync_cmd on the production server

```
#!/bin/sh


#
# Script Variables
#
STK_SCRIPTS_DIR=`dirname $0`
SVA_LUNCONFIG_FILE=$STK_SCRIPTS_DIR/lun_config

#
# Logging methods
#
logerrors()
{
    case $* in
        "")
        while read line
        do
                logger -p daemon.notice -t `basename $0` $line
                echo \#INFO $line
        done
        ;;
        *)
                logger -p daemon.notice -t `basename $0` $*
                echo \#INFO $*
        ;;
    esac
}


loginfo()
{
    case $* in
        "")
        while read line

        do
                logger -p daemon.notice -t "`basename $0` NOTICE" $line
                echo \#INFO $line
        done
        ;;
        *)
                logger -p daemon.notice -t "`basename $0` NOTICE" $*
                echo \#INFO $*

        ;;
    esac
}
```

```
mysurootcmd()
{
         if [ `id | grep -c "uid=0"` = "1" ]
         then
          # I'm root user
          $*
         else
          # I'm not root user
          echo . | $STK_SCRIPTS_DIR/surootcmd "$*"
         fi
}

loginfo Beginning of script: resync_cmd




loginfo LUN Configuration File:
/usr/bin/egrep -v '^#|^$' $SVA_LUNCONFIG_FILE |
while read source target backupdisk mountpoint
do
     echo source=$source, target=$target, backupdisk=$backupdisk,
mountpoint=$mountpoint
done | loginfo


#
# On the Backup Host, Un-Mounting each file system
#
loginfo On Backup Server, unmounting each file system
/usr/bin/egrep -v '^#|^$' $SVA_LUNCONFIG_FILE |
while read source target backupdisk mountpoint
do
     /usr/bin/rsh $BACKUP_SERVER -n -l root /usr/sbin/umount ${mountpoint} > /dev/null
2>&1
done


#
# If NOT USING SVAPATH -> comment these lines
# On the Backup Server, Excluding each LUN to the server view
#
loginfo On Backup Server, Excluding each LUN to the server view
/usr/bin/egrep -v '^#|^$' $SVA_LUNCONFIG_FILE |
while read source target backupdisk mountpoint
do
     backuplun=`/usr/bin/rsh $BACKUP_SERVER -n -l root /usr/sbin/setsp 2>&1 |
/usr/bin/grep ${backupdisk}|awk '{print $1}'|cut -dd -f2`
     echo /usr/sbin/setsp -l${backuplun} -e1
done | /usr/bin/rsh $BACKUP_SERVER -n -l root sh | logerrors

loginfo Script successfully completed.
```

APPENDIX J: EXAMPLE OF ONLINE SUCCESSFUL BACKUP

result_online

```
#INFO ***** Calling brbackup_online Wednesday October 30 14:02:12 GMT 2002 **********
#INFO Mounting the sapbackup directory from the production server to the backup server
#INFO Mounting the sapreorg directory from the production server to the backup server
Execute setenv SAPDATA_HOME /oracle/TLS; setenv SAPBACKUP /oracle/TLS/sapbackup;
setenv SAPARCH /oracle/TLS/saparch; setenv SAP
REORG /oracle/TLS/sapreorg; /usr/sap/TLS/SYS/exe/run/brbackup -u internal/garonne -p
/oracle/TLS/dbs/initTLS_online.sap -c -l E
-d util_file -r /oracle/TLS/dbs/initTLS.utl -t  online_split
Sun Microsystems Inc.   SunOS 5.8       Generic February 2000
BR051I BRBACKUP 4.6C (4)
BR055I Start of database backup: bdjfbluj.pnf 2002-10-30 14.02.13

BR319I Control file copied to /oracle/TLS/sapbackup/cntrl.dbf

BR280I Time stamp 2002-10-30 14.02.15
BR315I 'Alter tablespace PSAPROLL begin backup' successful

BR280I Time stamp 2002-10-30 14.02.15
BR295I Splitting mirror disks...
BR278I Command output of '/oracle/TLS/sapscript/snap_cmd
[/oracle/TLS/sapbackup/.bdjfbluj.lst]':
#INFO Beginning of script: backup_snap_online
#INFO LUN Configuration File:
#INFO source=c1t8d31, target=c1t8d51, backupdisk=c1t9d51,
mountpoint=/oracle/TLS/sapdata1
#INFO source=c1t8d32, target=c1t8d52, backupdisk=c1t9d52,
mountpoint=/oracle/TLS/sapdata2
#INFO source=c1t8d33, target=c1t8d53, backupdisk=c1t9d53,
mountpoint=/oracle/TLS/sapdata3
#INFO source=c1t8d34, target=c1t8d54, backupdisk=c1t9d54,
mountpoint=/oracle/TLS/sapdata4
#INFO source=c1t8d35, target=c1t8d55, backupdisk=c1t9d55,
mountpoint=/oracle/TLS/sapdata5
#INFO source=c1t8d36, target=c1t8d56, backupdisk=c1t9d56,
mountpoint=/oracle/TLS/sapdata6
#INFO source=c1t8d37, target=c1t8d57, backupdisk=c1t9d57,
mountpoint=/oracle/TLS/sapdata7
#INFO On Production Server, Syncing the file systems
#INFO On Production Server, Snapping each individual LUN
#INFO On Backup Server, Checking and mounting each file system
#INFO Script successfully completed.

BR280I Time stamp 2002-10-30 14.02.49
BR296I Split of mirror disks successful

BR280I Time stamp 2002-10-30 14.02.51
BR317I 'Alter tablespace PSAPROLL end backup' successful

BR340I Switching to next online redo log file for database instance TLS...
BR321I Switch to next online redo log file for database instance TLS successful

BR280I Time stamp 2002-10-30 14.02.52
BR057I Backup of database: TLS
BR058I BRBACKUP action ID: bdjfbluj
BR059I BRBACKUP function ID: pnf
BR110I Backup mode: PARTIAL
```

```
BR060I Tablespace for backup: PSAPROLL
BR077I Database file for backup: /oracle/TLS/sapbackup/cntrl.dbf
BR061I 2 files found for backup, total size 1804.672 MB
BR143I Backup type: online_split
BR130I Backup device type: util_file
BR109I Files will be saved by backup utility
BR126I Unattended mode active - no operator confirmation required

BR280I Time stamp 2002-10-30 14.02.52
BR229I Calling backup utility...
BR278I Command output of '/usr/sap/TLS/SYS/exe/run/backint -u TLS -f backup -i
/oracle/TLS/sapbackup/.bdjfbluj.lst -t file -p
/oracle/TLS/dbs/initTLS.utl -c':

BR280I Time stamp 2002-10-30 14.08.12
#FILE..... /oracle/TLS/sapdata1/roll_1/roll.data1
#SAVED.... 1035990172

BR280I Time stamp 2002-10-30 14.08.12
#FILE..... /oracle/TLS/sapbackup/cntrl.dbf
#SAVED.... 1035990173

BR280I Time stamp 2002-10-30 14.08.12
BR232I 2 of 2 files saved by backup utility
BR230I Backup utility called successfully

BR280I Time stamp 2002-10-30 14.08.14
BR229I Calling backup utility...
BR278I Command output of '/usr/sap/TLS/SYS/exe/run/backint -u TLS -f backup -i
/oracle/TLS/sapbackup/.bdjfbluj.lst -t file -p
/oracle/TLS/dbs/initTLS.utl -c':

BR280I Time stamp 2002-10-30 14.08.27
#INLOG.... /oracle/TLS/dbs/initTLS.dba
#SAVED.... 1035990495

BR280I Time stamp 2002-10-30 14.08.27
#INLOG.... /oracle/TLS/dbs/initTLS_online.sap
#SAVED.... 1035990496

BR280I Time stamp 2002-10-30 14.08.27
#INLOG.... /oracle/TLS/sapbackup/bdjfbluj.pnf
#SAVED.... 1035990497

BR280I Time stamp 2002-10-30 14.08.27
#INLOG.... /oracle/TLS/dbs/initTLS.ora
#SAVED.... 1035990498

BR280I Time stamp 2002-10-30 14.08.27
#INLOG.... /oracle/TLS/sapreorg/structTLS.log
#SAVED.... 1035990499

BR280I Time stamp 2002-10-30 14.08.27
#INLOG.... /oracle/TLS/sapbackup/backTLS.log
#SAVED.... 1035990500

BR280I Time stamp 2002-10-30 14.08.27
#INLOG.... /oracle/TLS/dbs/initTLS.utl
#SAVED.... 1035990501
```

```
BR280I Time stamp 2002-10-30 14.08.27
#INLOG.... /oracle/TLS/sapreorg/reorgTLS.log
#SAVED.... 1035990502

BR280I Time stamp 2002-10-30 14.08.27
#INLOG.... /oracle/TLS/dbs/initora.addon
#SAVED.... 1035990503

BR280I Time stamp 2002-10-30 14.08.27
#INLOG.... /oracle/TLS/dbs/init_806.ora
#SAVED.... 1035990504

BR280I Time stamp 2002-10-30 14.08.27
BR232I 10 of 10 files saved by backup utility
BR230I Backup utility called successfully

BR280I Time stamp 2002-10-30 14.08.28
BR297I Resynchronizing mirror disks...
BR278I Command output of '/oracle/TLS/sapscript/resync_cmd
[/oracle/TLS/sapbackup/.bdjfbluj.lst]':
#INFO Beginning of script: resync_cmd
#INFO LUN Configuration File:
#INFO source=c1t8d31, target=c1t8d51, backupdisk=c1t9d51,
mountpoint=/oracle/TLS/sapdata1
#INFO source=c1t8d32, target=c1t8d52, backupdisk=c1t9d52,
mountpoint=/oracle/TLS/sapdata2
#INFO source=c1t8d33, target=c1t8d53, backupdisk=c1t9d53,
mountpoint=/oracle/TLS/sapdata3
#INFO source=c1t8d34, target=c1t8d54, backupdisk=c1t9d54,
mountpoint=/oracle/TLS/sapdata4
#INFO source=c1t8d35, target=c1t8d55, backupdisk=c1t9d55,
mountpoint=/oracle/TLS/sapdata5
#INFO source=c1t8d36, target=c1t8d56, backupdisk=c1t9d56,
mountpoint=/oracle/TLS/sapdata6
#INFO source=c1t8d37, target=c1t8d57, backupdisk=c1t9d57,
mountpoint=/oracle/TLS/sapdata7
#INFO On Backup Server, unmounting each file system
#INFO Script successfully completed.

BR280I Time stamp 2002-10-30 14.08.29
BR298I Resynchronization of mirror disks successful

BR056I End of database backup: bdjfbluj.pnf 2002-10-30 14.08.27
BR280I Time stamp 2002-10-30 14.08.29
BR052I BRBACKUP terminated successfully
Status: 0
#INFO UnMounting the sapbackup directory from the backup server
#INFO UnMounting the sapreorg directory from the backup server
Status: 0
```

APPENDIX K: EXAMPLE OF OFFLINE SUCCESSFUL BACKUP

result_offline

```
#INFO ****** Calling brbackup_offline Monday November  4 14:31:29 GMT 2002 **********
#INFO Mounting the sapbackup directory from the production server to the backup server
#INFO Mounting the sapreorg directory from the production server to the backup server
Execute setenv SAPDATA_HOME /oracle/TLS; setenv SAPBACKUP /oracle/TLS/sapbackup;
setenv SAPARCH /oracle/TLS/saparch; setenv SAP
REORG /oracle/TLS/sapreorg; /usr/sap/TLS/SYS/exe/run/brbackup -u internal/garonne -p
/oracle/TLS/dbs/initTLS_offline.sap -c -l E
 -d util_file -r /oracle/TLS/dbs/initTLS.utl -t  offline_split
Sun Microsystems Inc.   SunOS 5.8       Generic February 2000
BR051I BRBACKUP 4.6C (4)
BR055I Start of database backup: bdjgadlj.pff 2002-11-04 14.31.31

BR280I Time stamp 2002-11-04 14.31.39
BR307I Shutting down the database instance TLS/PRIMARY...

BR280I Time stamp 2002-11-04 14.32.49
BR308I Shutdown of database instance TLS/PRIMARY successful

BR280I Time stamp 2002-11-04 14.32.49
BR295I Splitting mirror disks...
BR278I Command output of '/oracle/TLS/sapscript/snap_cmd
[/oracle/TLS/sapbackup/.bdjgadlj.lst]':
#INFO Beginning of script: backup_snap_online
#INFO LUN Configuration File:
#INFO source=c1t8d31, target=c1t8d51, backupdisk=c1t9d51,
mountpoint=/oracle/TLS/sapdata1
#INFO source=c1t8d32, target=c1t8d52, backupdisk=c1t9d52,
mountpoint=/oracle/TLS/sapdata2
#INFO source=c1t8d33, target=c1t8d53, backupdisk=c1t9d53,
mountpoint=/oracle/TLS/sapdata3
#INFO source=c1t8d34, target=c1t8d54, backupdisk=c1t9d54,
mountpoint=/oracle/TLS/sapdata4
#INFO source=c1t8d35, target=c1t8d55, backupdisk=c1t9d55,
mountpoint=/oracle/TLS/sapdata5
#INFO source=c1t8d36, target=c1t8d56, backupdisk=c1t9d56,
mountpoint=/oracle/TLS/sapdata6
#INFO source=c1t8d37, target=c1t8d57, backupdisk=c1t9d57,
mountpoint=/oracle/TLS/sapdata7
#INFO On Production Server, Syncing the file systems
#INFO On Production Server, Snapping each individual LUN
#INFO On Backup Server, Checking and mounting each file system
#INFO Script successfully completed.

BR280I Time stamp 2002-11-04 14.33.25
BR296I Split of mirror disks successful

BR280I Time stamp 2002-11-04 14.33.26
BR304I Starting the database instance TLS/PRIMARY...

BR280I Time stamp 2002-11-04 14.33.33
BR305I Startup of database instance TLS/PRIMARY successful

BR280I Time stamp 2002-11-04 14.33.33
BR057I Backup of database: TLS
BR058I BRBACKUP action ID: bdjgadlj
BR059I BRBACKUP function ID: pff
```

```
BR110I Backup mode: PARTIAL
BR060I Tablespace for backup: PSAPROLL
BR077I Database file for backup: /oracle/TLS/sapdata1/cntrl/cntrlTLS.dbf
BR061I 2 files found for backup, total size 1804.672 MB
BR143I Backup type: offline_split
BR130I Backup device type: util_file
BR109I Files will be saved by backup utility
BR126I Unattended mode active - no operator confirmation required

BR280I Time stamp 2002-11-04 14.33.34
BR229I Calling backup utility...
BR278I Command output of '/usr/sap/TLS/SYS/exe/run/backint -u TLS -f backup -i
/oracle/TLS/sapbackup/.bdjgadlj.lst -t file -p
/oracle/TLS/dbs/initTLS.utl -c':

BR280I Time stamp 2002-11-04 14.38.49
#FILE..... /oracle/TLS/sapdata1/roll_1/roll.data1
#SAVED.... 1036424014

BR280I Time stamp 2002-11-04 14.38.49
#FILE..... /oracle/TLS/sapdata1/cntrl/cntrlTLS.dbf
#SAVED.... 1036424015

BR280I Time stamp 2002-11-04 14.38.49
BR232I 2 of 2 files saved by backup utility
BR230I Backup utility called successfully

BR280I Time stamp 2002-11-04 14.38.51
BR229I Calling backup utility...
BR278I Command output of '/usr/sap/TLS/SYS/exe/run/backint -u TLS -f backup -i
/oracle/TLS/sapbackup/.bdjgadlj.lst -t file -p
/oracle/TLS/dbs/initTLS.utl -c':

BR280I Time stamp 2002-11-04 14.39.16
#INLOG.... /oracle/TLS/dbs/initTLS.dba
#SAVED.... 1036424332

BR280I Time stamp 2002-11-04 14.39.16
#INLOG.... /oracle/TLS/dbs/initTLS_offline.sap
#SAVED.... 1036424337

BR280I Time stamp 2002-11-04 14.39.16
#INLOG.... /oracle/TLS/sapbackup/bdjgadlj.pff
#SAVED.... 1036424333

BR280I Time stamp 2002-11-04 14.39.16
#INLOG.... /oracle/TLS/dbs/initTLS.ora
#SAVED.... 1036424331

BR280I Time stamp 2002-11-04 14.39.16
#INLOG.... /oracle/TLS/sapreorg/structTLS.log
#SAVED.... 1036424336

BR280I Time stamp 2002-11-04 14.39.16
#INLOG.... /oracle/TLS/sapbackup/backTLS.log
#SAVED.... 1036424338

BR280I Time stamp 2002-11-04 14.39.16
#INLOG.... /oracle/TLS/dbs/initTLS.utl
#SAVED.... 1036424334
```

```
BR280I Time stamp 2002-11-04 14.39.16
#INLOG.... /oracle/TLS/sapreorg/reorgTLS.log
#SAVED.... 1036424335

BR280I Time stamp 2002-11-04 14.39.16
#INLOG.... /oracle/TLS/dbs/initora.addon
#SAVED.... 1036424351

BR280I Time stamp 2002-11-04 14.39.16
#INLOG.... /oracle/TLS/dbs/init_806.ora
#SAVED.... 1036424352

BR280I Time stamp 2002-11-04 14.39.16
BR232I 10 of 10 files saved by backup utility
BR230I Backup utility called successfully

BR280I Time stamp 2002-11-04 14.39.17
BR297I Resynchronizing mirror disks...
BR278I Command output of '/oracle/TLS/sapscript/resync_cmd
[/oracle/TLS/sapbackup/.bdjgadlj.lst]':
#INFO Beginning of script: resync_cmd
#INFO LUN Configuration File:
#INFO source=c1t8d31, target=c1t8d51, backupdisk=c1t9d51,
mountpoint=/oracle/TLS/sapdata1
#INFO source=c1t8d32, target=c1t8d52, backupdisk=c1t9d52,
mountpoint=/oracle/TLS/sapdata2
#INFO source=c1t8d33, target=c1t8d53, backupdisk=c1t9d53,
mountpoint=/oracle/TLS/sapdata3
#INFO source=c1t8d34, target=c1t8d54, backupdisk=c1t9d54,
mountpoint=/oracle/TLS/sapdata4
#INFO source=c1t8d35, target=c1t8d55, backupdisk=c1t9d55,
mountpoint=/oracle/TLS/sapdata5
#INFO source=c1t8d36, target=c1t8d56, backupdisk=c1t9d56,
mountpoint=/oracle/TLS/sapdata6
#INFO source=c1t8d37, target=c1t8d57, backupdisk=c1t9d57,
mountpoint=/oracle/TLS/sapdata7
#INFO On Backup Server, unmounting each file system
#INFO Script successfully completed.

BR280I Time stamp 2002-11-04 14.39.18
BR298I Resynchronization of mirror disks successful
BR056I End of database backup: bdjgadlj.pff 2002-11-04 14.39.16
BR280I Time stamp 2002-11-04 14.39.18
BR052I BRBACKUP terminated successfully

Status: 0
#INFO UnMounting the sapbackup directory from the backup server
#INFO UnMounting the sapreorg directory from the backup server
Status: 0
```

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 9151 B 03/04