

IBM EAS Technical Brief

Oracle Recovery (Local) with FlashCopy Services

Mark A. Hays
IBM EAS Advanced Technical Support

Version: 1.0
Date: 05 February 2003

Document Purpose

The following procedure is intended to explain the use of IBM Copy Services (specifically FlashCopy) to *locally* recover Oracle database environments in the event of lost (accidentally removed) or corrupt data files. The example provided herein utilizes a PeopleSoft application in conjunction with the Oracle8i database server on IBM pSeries hardware running the AIX operating environment: this procedure's use, however, is not necessarily restricted to PeopleSoft installations or a specific version of Oracle RDBMS. There is no reason that the steps used here couldn't be used to recover any Oracle database / application environment that is protected using IBM Advanced Copy Services.

In an effort to make the procedure clear, this document explains some basic steps to establish FlashCopy pairs and summarizes some features and implementation choices related to the Advanced Copy Services products. This document does not endeavor to summarize or explain all the functionality contained with the IBM Advanced Copy Services suite; for example, the procedures to create LUNs and allocate storage to attached servers is not detailed. Please see the Technical Reference Materials section of the document for publications that address specific storage administration issues.

Finally, the procedure introduces the 'recreatevg' command as a tool to simplify the technical issues related to creating a volume group using FlashCopy *targets* on the same server on which the *source* LUNs are used. The command is available in AIX operating environment as of version 4.3.3 ML ? and AIX 5.1 ML?.

Audience

This document is written for IT technical professionals that are responsible for supporting IBM RS/6000 or pSeries servers running an Oracle database and application environment attached to IBM storage products that utilize FlashCopy services.

The skills required to implement the procedure may exist in different departments within an IT organization. The standard professional disciplines required to affect the procedure include: Oracle database administrator, AIX/UNIX system administrator, and ESS/Storage Server administrator.

IBM ESS Advanced Copy Services (FlashCopy) Summary

FlashCopy is one product feature contained within the suite of IBM Advanced Copy Services for the ESS (and other storage subsystems in the future). Other products (Peer-to-Peer Remote Copy and PPRC-XD) in this suite are not addressed in this procedure as they are used for data recovery at a remote site. As stated earlier, this procedure focuses on the steps for *local* recovery of the Oracle database.

FlashCopy establishes an exact copy of the *source* LUN to an administrator-designated *target* LUN. The data blocks (512 bytes) that are written to the *target* LUN represent a point-in-time (T0); that time is the moment that the FlashCopy relationship is made. For each data block that is updated thereafter, FlashCopy generates an additional I/O operation to copy the original unmodified block to the *target* LUN.

FlashCopy provides a ‘background copy’ option at the time the pair is established, which affects the scheduling of the unmodified block copy to the *target* LUN. The background copy is intended to provide a complete image of the *source* LUN at the T0 time as quickly as possible regardless of update activity on the *source*. Once the complete copy is made, the FlashCopy relationship terminates and the administrator is free to use the copy on another server.

As stated previously the standard FlashCopy only copies the data blocks to the *target* LUN when updates are made to the *source*. The critical implication here is that any representation of the LUN at the T0 time requires that the FlashCopy relationship remain intact. For example, if one were to perform a backup against the *target* LUNs (that represent the frozen T0 time), the backup would get any unmodified data from the *source* LUN and any data that was modified after the FlashCopy pairing would be retrieved from the *target*; as such, it should be plain why the pair cannot be broken if one is to use the FlashCopy pair for any future endeavor – backup or otherwise.

Test Environment Specifications

The following hardware and software was used to generate the information contained in this procedure. The information provided here is for reference use only and is not necessarily central to the procedure.

Hardware:	IBM pSeries Model H80 2 GB RAM 2 x 18.2GB internal storage (rootvg) 15 x 8GB allocated ESS storage 4 Fiber Channel Adapters FC 6228 Fiber Channel Switch Model 2109
Software:	PeopleSoft Financials v8 PeopleTools v8.41 Oracle8i v8.1.7.3 IBM WebSphere Application Server v4 IBM HTTP Web Server v1.3 JAVA v1.3

Technical Reference Materials

IBM TotalStorage Enterprise Storage Server User’s Guide
IBM TotalStorage Enterprise Storage Server Web Interface User’s Guide
Mark A. Hays

IBM TotalStorage Enterprise Storage Server Command-Line Interface User's Guide
IBM TotalStorage Enterprise Storage Server Introduction and Planning Guide
IBM TotalStorage Enterprise Storage Server Configuration Planner for Open System
Hosts
IBM TotalStorage Enterprise Storage Server Host Systems Attachment Guide
IBM Subsystem Device Driver Installation and User's Guide
IBM TotalStorage Enterprise Storage Server – Implementing Copy Services in an Open
Systems Environment (Redbook)

Procedure Steps

The procedure is summarized with the following the overview:

- Create and save tasks to establish FlashCopy pairs for LUNs contained in Oracle volume group(s).
- Alter Oracle tablespaces in the database to BEGIN BACKUP mode.
- Initiate tasks to establish FlashCopy pairs.
- Alter Oracle tablespaces in the database to END BACKUP mode.
- Event occurs that causes data file loss or corruption (simulated with rm and dd commands).
- Assign FlashCopy target LUNs to the same server as the source LUNs
- Create a new volume group containing all hdisks from the source volume group.
- Copy the missing or corrupt data files from the FlashCopy target to the proper directory structure in the existing Oracle database and recover the data files using the appropriate Oracle procedure.

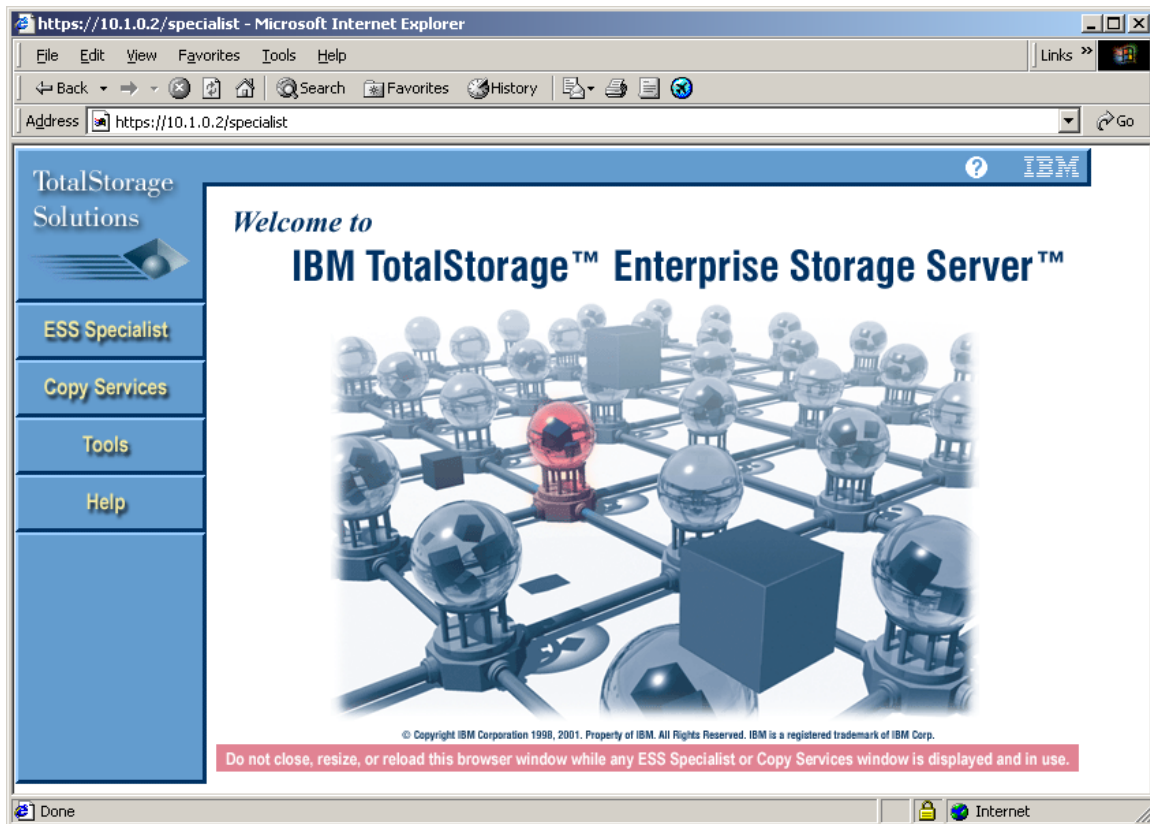
The first four bulleted points would be performed on a regular basis as one part of a comprehensive data backup and recovery solution. The other points are only used in the event that local recovery of the Oracle database is required.

WARNING: For Oracle recovery considerations, the FlashCopy pair must be established when all the database tables have been placed in the BEGIN BACKUP mode.

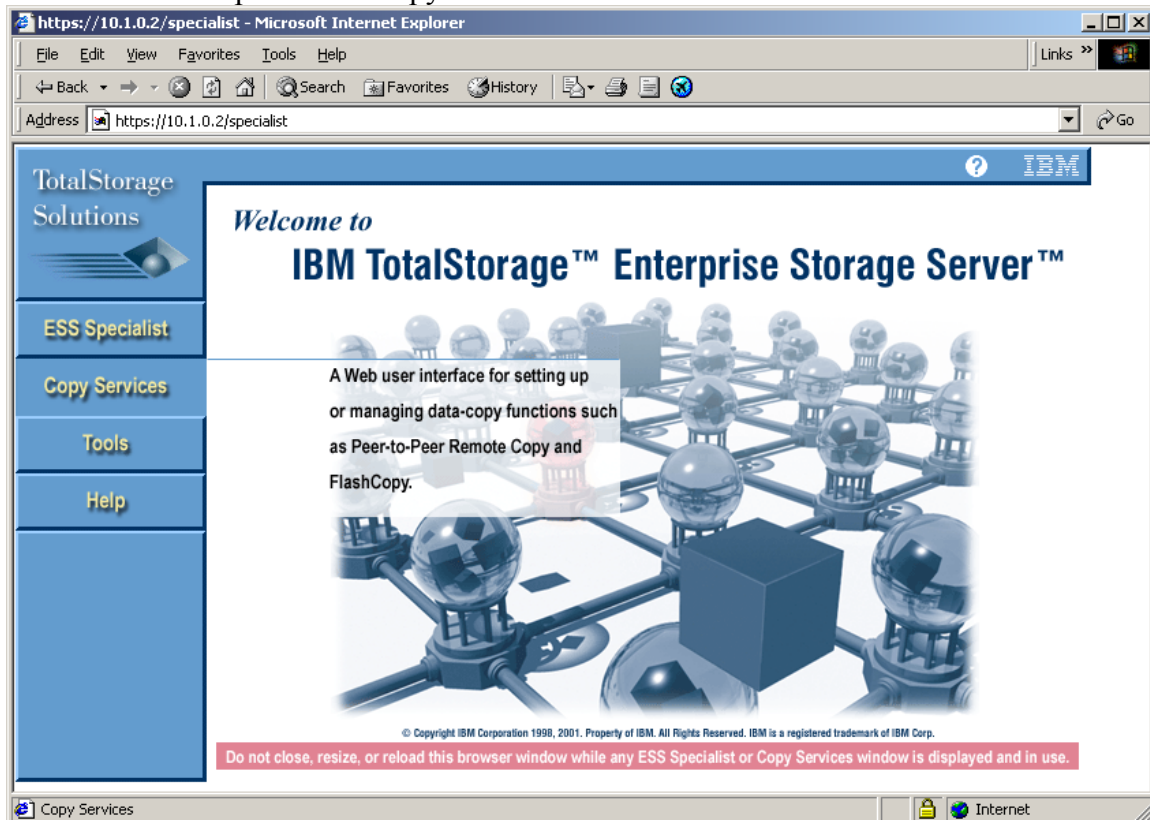
Create and save tasks to establish FlashCopy pairs for LUNs contained in Oracle volume group(s).

1. Start a web browser and point the URL to the ESS Storwatch Specialist

As an example: <http://10.1.0.2/>



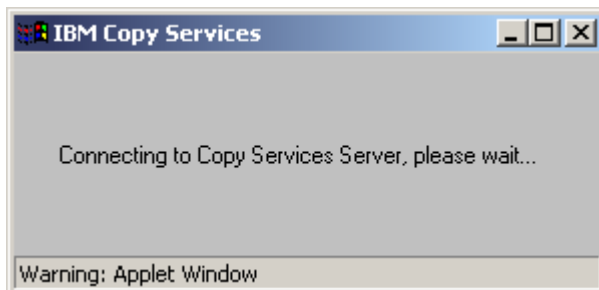
2. Select the panel for “Copy Services”.



3. The administrator will be prompted to enter the user and password. The default user is “storwatch” and the password is “specialist”. The storage or system administrators should know the actual user name and password.



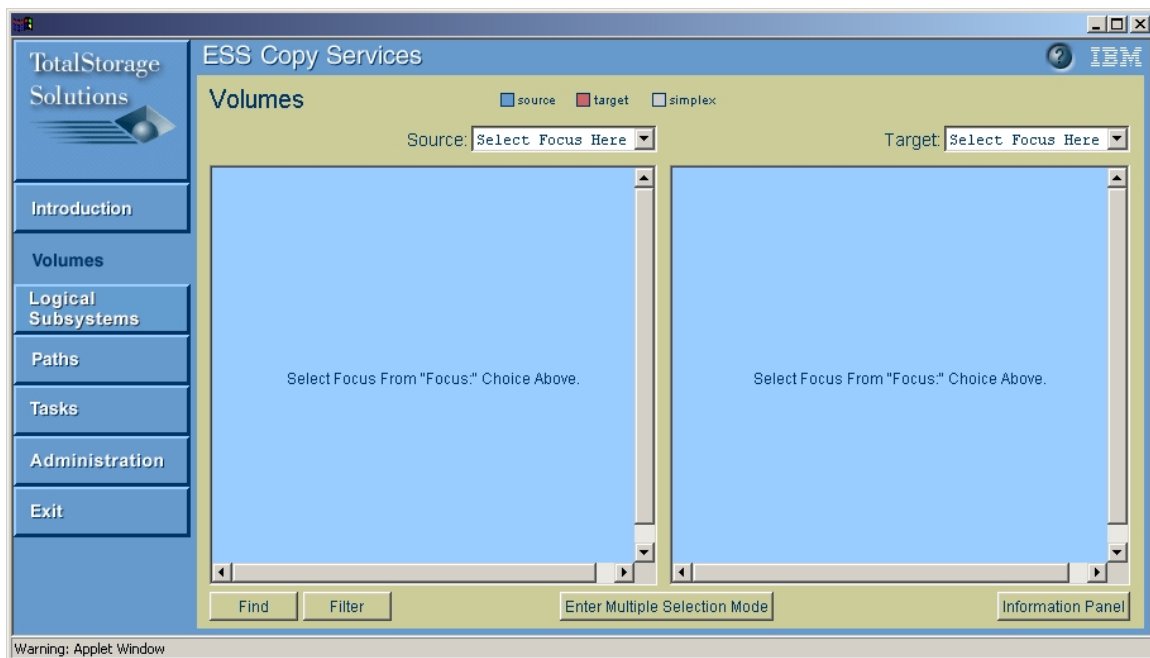
The following status box will be displayed as the connection is established on the Copy Services Server.



The ESS Copy Services panel will be displayed after successful logon.



4. Select the Volumes panel.

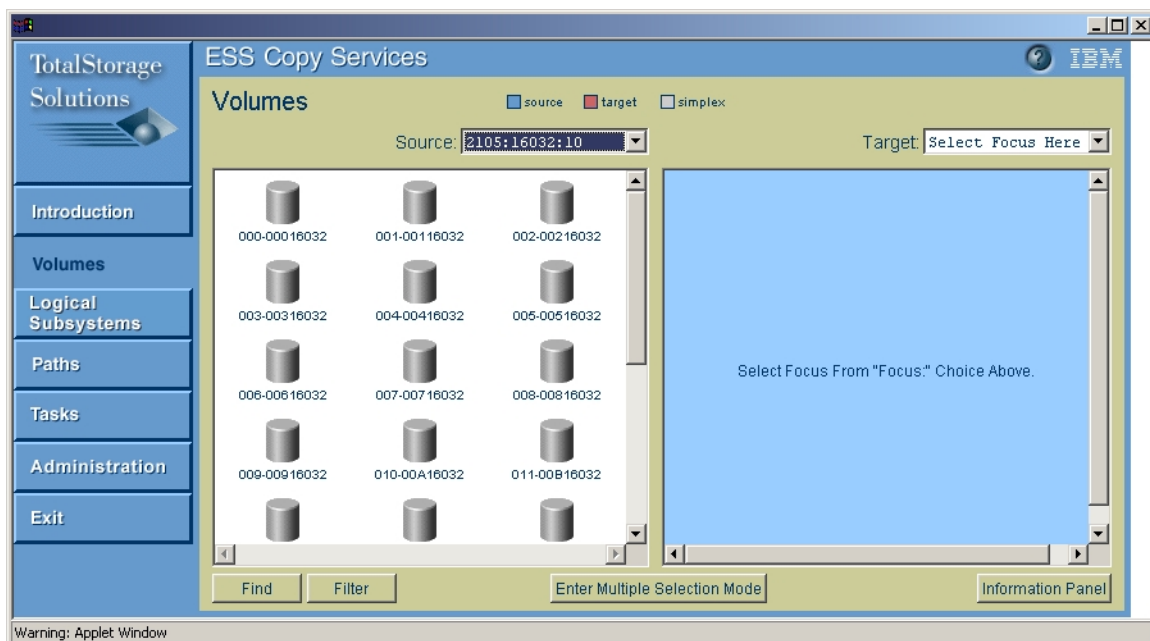


5. Choose the host name or the Logical SubSystem (LSS) on which the source LUNs reside from the "Source" pull-down dialogue box.

If you don't understand the term LSS, please do not proceed with these steps until you have familiarized yourself with the ESS Administration tasks. Please see the Technical Reference Materials section of this document for a list of related publications.

By default, the source and target dialogue boxes will display the LUNs in each LSS as shown below. The naming convention of each source LUN includes the sequential number in the LSS, a unique hexadecimal address of the LUN and the serial number of the ESS on which the LUN is configured.

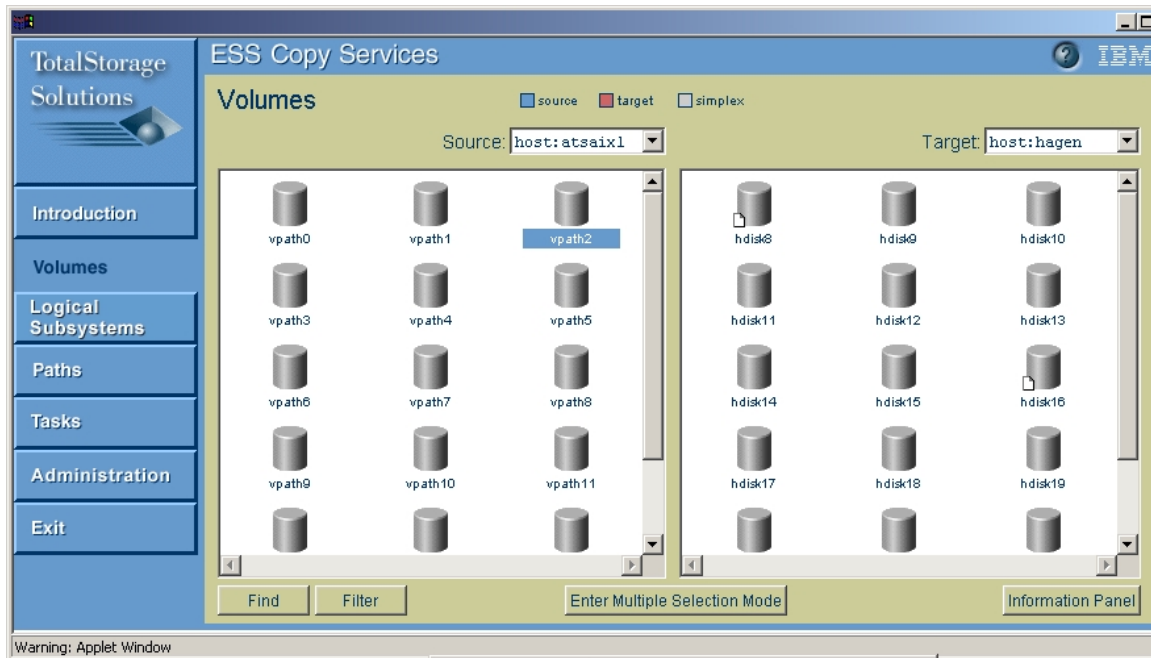
WARNING: The naming convention for the LUNs in this screen makes it difficult to map to the actual hdisk number on the host server. It is much safer to list the LUNs by host where the actual hdisk number is display. Unfortunately, this convenience will likely not exist when establishing the target LUNs.



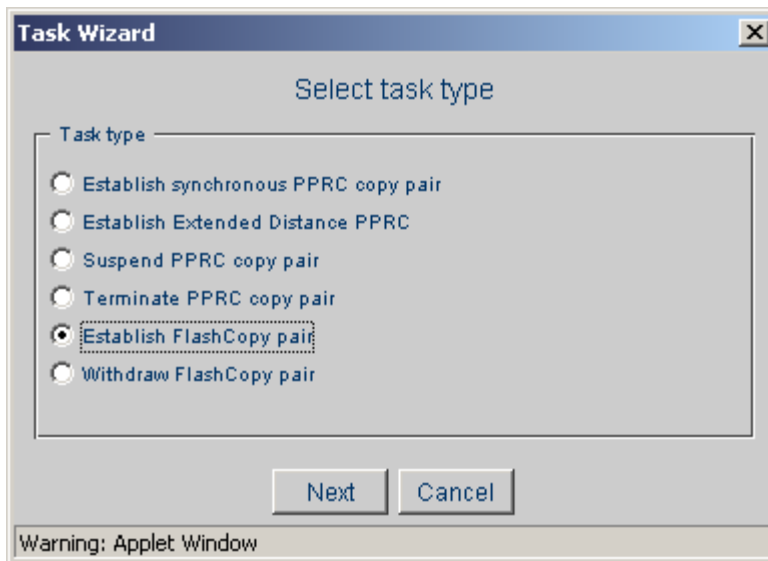
6. Select the appropriate entries from the Target drop-down dialogue box.

While conducting this exercise in the lab, I temporarily brought the target LUNs onto another server. This made the target select much easier later since they were all associated with the server with a hostname of 'hagen'. This convenience will likely not be available at a customer location and the administrator must rely on the cryptic LUN name described earlier.

WARNING: The interface for the Advanced Copy Services will not prevent the administrator from creating a FlashCopy target onto a LUN that is already assigned and in use on another server. It is critically important to correctly identify and select the target of the FlashCopy pair. **CUSTOMER DATA CAN AND WILL BE LOST IF DONE INCORRECTLY!**

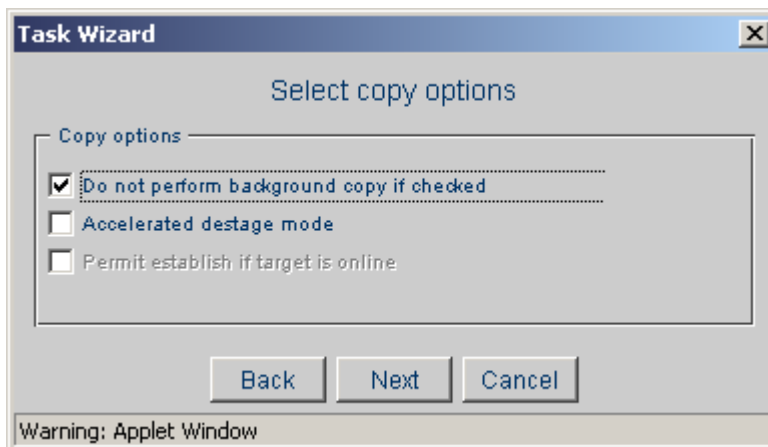


7. Once the source and target LUNs are displayed in each of the panels, the relationship is established with a sequence of mouse-clicks. It is important to perform the mouse-clicks in the correct sequence.
 - a. Left click once on the source LUN. The text description of the source LUN will be highlighted in *blue*.
 - b. Right click once on the target LUN. The text description of the target LUN will be highlighted in *red*.
 - c. Right click one more time on the target LUN and the “Task Wizard – Select Task Type” window is displayed.
8. Select “Establish FlashCopy pair” radio button and press “Next”.



9. Check the box as indicated below to establish a standard copy or leave it unchecked for a background copy. Press “Next”.

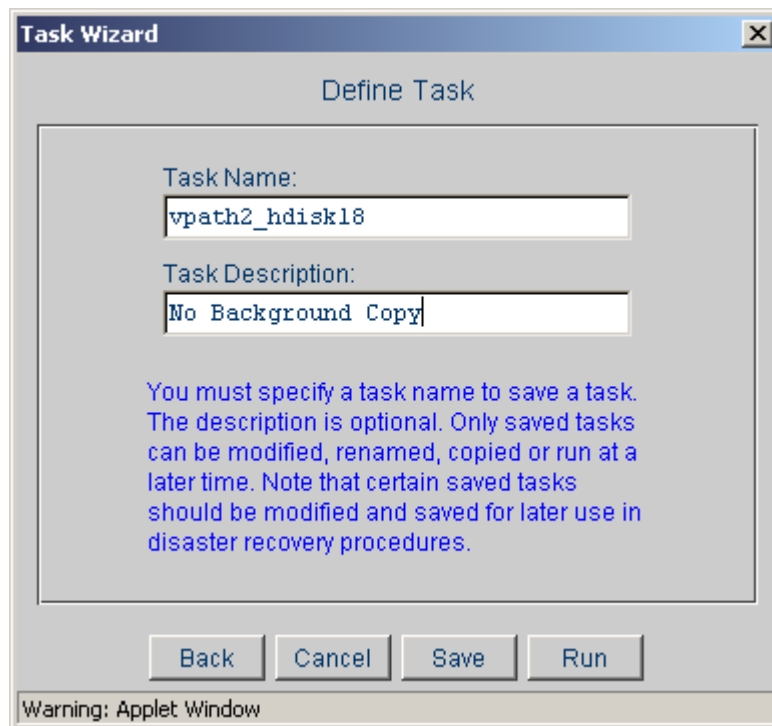
Note: If you’re really bored, you can ponder the wording of this option in the dialogue box. ☺



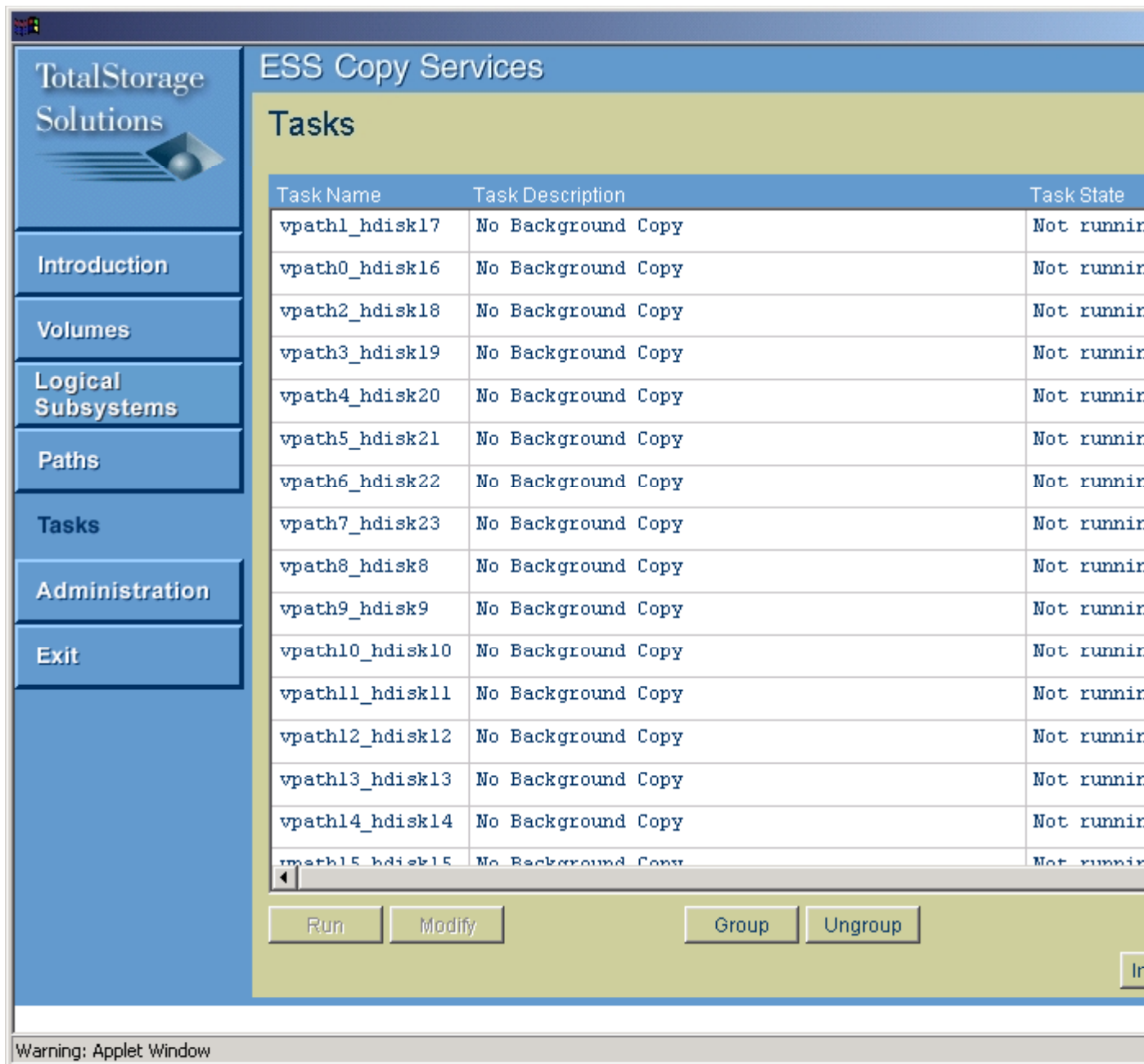
10. Enter the task name and description. Choose the “Save” option.

A FlashCopy pair can be started now with “Run” or saved as a task for a future start with the “Save” option. If the “Save” button is pressed, the window for the task name and description disappears and the user is returned to the Background Copy Services “Volumes” display. There is no visual indication that the two LUNs are associated in a FlashCopy relationship because it is not established until the task is actually run.

WARNING: For the purposes of this procedure, the FlashCopy pairs must be established with all LUNs that comprise the Oracle volume group(s), otherwise, the ‘recreatevg’ command will not run; additionally, the FlashCopy relationship must not be initiated for any LUN pair until ALL tablespaces in the database are in the BEGIN BACKUP mode. A FlashCopy in this manner is the equivalent of performing an Oracle hot backup – only this backup will complete in about one second!



11. Once all the tasks are created to establish pairs for the hdisks in the Oracle volume groups, the individual tasks can be grouped into a single task name. All individual tasks will be run at the same time when the group task is initiated. Perform this step by selecting the “Tasks” panel from the Copy Services main menu.



TotalStorage Solutions

ESS Copy Services

Tasks

Task Name	Task Description	Task State
vpath1_hdisk17	No Background Copy	Not running
vpath0_hdisk16	No Background Copy	Not running
vpath2_hdisk18	No Background Copy	Not running
vpath3_hdisk19	No Background Copy	Not running
vpath4_hdisk20	No Background Copy	Not running
vpath5_hdisk21	No Background Copy	Not running
vpath6_hdisk22	No Background Copy	Not running
vpath7_hdisk23	No Background Copy	Not running
vpath8_hdisk8	No Background Copy	Not running
vpath9_hdisk9	No Background Copy	Not running
vpath10_hdisk10	No Background Copy	Not running
vpath11_hdisk11	No Background Copy	Not running
vpath12_hdisk12	No Background Copy	Not running
vpath13_hdisk13	No Background Copy	Not running
vpath14_hdisk14	No Background Copy	Not running
vpath15_hdisk15	No Background Copy	Not running

Run Modify Group Ungroup In

Warning: Applet Window

Task Wizard [X]

Define Group

Group Name:

Group Description:

Warning: Applet Window

TotalStorage Solutions

Introduction
Volumes
Logical Subsystems
Paths
Tasks
Administration
Exit

ESS Copy Services

Tasks

Task Name	Task Description	Task State
atsaixl_hagen	PeopleSoft Oracle Flashcopy	Not running

Warning: Applet Window

Alter Oracle tablespaces in the database to BEGIN BACKUP mode.

Before the FlashCopy pairs are established, all the tablespaces in the database must be altered into a BEGIN BACKUP mode. The commands could be manually entered into SQL*Plus one by one; however, this will take more time and be more prone to error than creating a SQL script that automates the task.

Follow the listed steps to quickly create a SQL script with all the tablespaces.

```
1. oracle@atsaix1:/oracle=> sqlplus
```

```
SQL*Plus: Release 8.1.7.0.0 - Production on Mon Dec 30 17:18:00 2002
```

```
(c) Copyright 2000 Oracle Corporation. All rights reserved.
```

```
Enter user-name: system
```

```
Enter password: (password)
```

```
Connected to:
```

```
Oracle8i Enterprise Edition Release 8.1.7.0.0 - Production
```

```
JServer Release 8.1.7.0.0 - Production
```

```
SQL> set head off
```

```
SQL> set pagesize 500
```

```
SQL> spool beg_backup.sql
```

```
SQL> select 'alter tablespace '||tablespace_name||' begin backup ;'  
from dba_tablespaces ;
```

```
alter tablespace SYSTEM begin backup ;  
alter tablespace PSRBS begin backup ;  
alter tablespace PSTEMP begin backup ;  
alter tablespace AMAPP begin backup ;  
alter tablespace AMLARGE begin backup ;  
alter tablespace AMWORK begin backup ;  
alter tablespace APAPP begin backup ;  
alter tablespace APLARGE begin backup ;  
alter tablespace APWORK begin backup ;  
alter tablespace ARAPP begin backup ;  
alter tablespace ARLARGE begin backup ;  
alter tablespace ARWORK begin backup ;  
alter tablespace AUCAPP begin backup ;  
alter tablespace BDAPP begin backup ;  
alter tablespace BDLARGE begin backup ;  
alter tablespace BIAPP begin backup ;  
alter tablespace BILARGE begin backup ;  
alter tablespace BIWORK begin backup ;  
alter tablespace BNAPP begin backup ;  
alter tablespace CAAPP begin backup ;  
alter tablespace CAWORK begin backup ;  
alter tablespace CPAPP begin backup ;  
alter tablespace CUAUDIT begin backup ;  
alter tablespace CULARG1 begin backup ;
```

Mark A. Hays

```

alter tablespace CULARG2 begin backup ;
alter tablespace CULARG3 begin backup ;
alter tablespace CULARGE begin backup ;
alter tablespace DIAPP begin backup ;
alter tablespace DPAPP begin backup ;
alter tablespace EOAPP begin backup ;
alter tablespace EOLARGE begin backup ;
alter tablespace EXAPP begin backup ;
alter tablespace EXLARGE begin backup ;
alter tablespace FAAPP begin backup ;
alter tablespace FALARGE begin backup ;
alter tablespace FAWORK begin backup ;
alter tablespace FIAPP begin backup ;
alter tablespace FILARGE begin backup ;
alter tablespace FSAPP begin backup ;
alter tablespace FSAPP1 begin backup ;
alter tablespace FSLARGE begin backup ;
alter tablespace FSWORK begin backup ;
alter tablespace GIAPP begin backup ;
alter tablespace GLAPP begin backup ;
alter tablespace GLLARGE begin backup ;
alter tablespace GLWORK begin backup ;
alter tablespace GMAPP begin backup ;
alter tablespace HRAPP begin backup ;
alter tablespace HRLARGE begin backup ;
alter tablespace HTAPP begin backup ;
alter tablespace INAPP begin backup ;
alter tablespace INLARGE begin backup ;
alter tablespace INWORK begin backup ;
alter tablespace LCAPP begin backup ;
alter tablespace LCLARGE begin backup ;
alter tablespace MGAPP begin backup ;
alter tablespace MPAPP begin backup ;
alter tablespace MPLARGE begin backup ;
alter tablespace OEAPP begin backup ;
alter tablespace OMAPP begin backup ;
alter tablespace OMLARGE begin backup ;
alter tablespace OMWORK begin backup ;
alter tablespace PCAPP begin backup ;
alter tablespace PCLARGE begin backup ;
alter tablespace POAPP begin backup ;
alter tablespace POLARGE begin backup ;
alter tablespace POWORK begin backup ;
alter tablespace PSWORK begin backup ;
alter tablespace PTAMSG begin backup ;
alter tablespace PTAPP begin backup ;
alter tablespace PTAPPE begin backup ;
alter tablespace PTCMSTAR begin backup ;
alter tablespace PTLOCK begin backup ;
alter tablespace PTPRC begin backup ;
alter tablespace PTPRCS begin backup ;
alter tablespace PTAUDIT begin backup ;
alter tablespace PTRPTS begin backup ;
alter tablespace PSIMGR begin backup ;
alter tablespace PTTBL begin backup ;
alter tablespace PTTLRG begin backup ;
alter tablespace PTTREE begin backup ;

```

Mark A. Hays

DRAFT – December 27, 2002


```
alter tablespace PTWORK begin backup ;
alter tablespace PVAPP begin backup ;
alter tablespace PYAPP begin backup ;
alter tablespace PY0LRG begin backup ;
alter tablespace RSAPP begin backup ;
alter tablespace RSLARGE begin backup ;
alter tablespace TDAPP begin backup ;
alter tablespace TDLARGE begin backup ;
alter tablespace TDWORK begin backup ;
alter tablespace TLAPP begin backup ;
alter tablespace TLWORK begin backup ;
alter tablespace TRAPP begin backup ;
alter tablespace TRLARGE begin backup ;
alter tablespace TRWORK begin backup ;
alter tablespace WSAPP begin backup ;
alter tablespace PSIMAGE begin backup ;
alter tablespace PSINDEX begin backup ;
```

98 rows selected.

```
SQL> spool off
SQL> exit
```

2. Edit the resulting file and remove the informational lines such as “98 rows selected.”

```
oracle@atsaix1:/oracle=> vi $ORACLE_HOME/beg_backup.sql
```

3. Run the script with the following command.

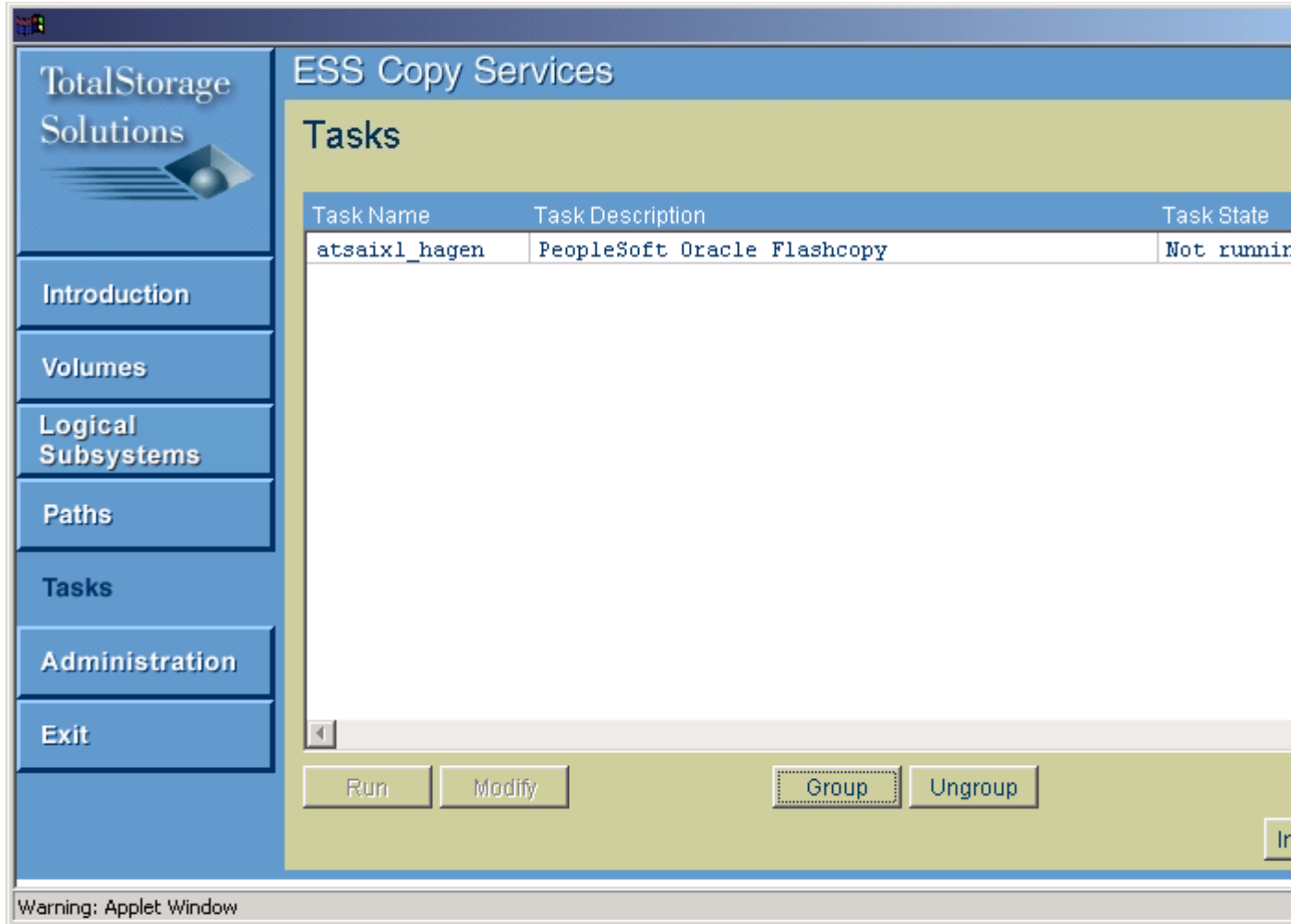
```
oracle@atsaix1:/oracle=> sqlplus system/(password)
@$ORACLE_HOME/beg_backup
```

```
Tablespace altered.
Tablespace altered.
Tablespace altered.
Tablespace altered.
Tablespace altered.
Tablespace altered.
...
```

Initiate tasks to establish FlashCopy pairs.

If the previous session on the Copy Services Servers was discontinued in the first section of the procedure, you will need to restart the ESS StorWatch Specialist and log into the Copy Services Server once more.

1. Select the “Tasks” panel, select the group task name, and “Run” the group task that contains the FlashCopy tasks.



2. Verify that FlashCopy pairs have been established. Select the “Volumes” panel and entering the same source and target locations.

The icons for the LUNs have changed and include a lightning bolt within the disk icon – *blue* for source and *red* for target.



Alter Oracle tablespaces in the database to END BACKUP mode.

Before the FlashCopy pairs are established, all the tablespaces in the database must be altered into a BEGIN BACKUP mode. The commands could be manually entered into SQL*Plus one by one; however, this will take more time and be more prone to error than creating a SQL script that automates the task.

Follow the listed steps to quickly create a SQL script with all the tablespaces.

```
oracle@atsaix1:/oracle=> sqlplus
```

```
SQL*Plus: Release 8.1.7.0.0 - Production on Mon Dec 30 17:18:00 2002
```

```
(c) Copyright 2000 Oracle Corporation. All rights reserved.
```

```
Enter user-name: system
```

```
Enter password: (password)
```

```
Connected to:
```

```
Mark A. Hays
```



IBM Americas Advanced Technical Support

Oracle8i Enterprise Edition Release 8.1.7.0.0 - Production
JServer Release 8.1.7.0.0 - Production

```
SQL> set head off
SQL> set pagesize 500
SQL> spool end_backup.sql
SQL> select 'alter tablespace '||tablespace_name||' end backup ;' from
dba_tablespaces ;
```

```
alter tablespace SYSTEM end backup ;
alter tablespace PSRBS end backup ;
alter tablespace PSTEMP end backup ;
alter tablespace AMAPP end backup ;
alter tablespace AMLARGE end backup ;
alter tablespace AMWORK end backup ;
alter tablespace APAPP end backup ;
alter tablespace APLARGE end backup ;
alter tablespace APWORK end backup ;
alter tablespace ARAPP end backup ;
alter tablespace ARLARGE end backup ;
alter tablespace ARWORK end backup ;
alter tablespace AUCAPP end backup ;
alter tablespace BDAPP end backup ;
alter tablespace BDLARGE end backup ;
alter tablespace BIAPP end backup ;
alter tablespace BILARGE end backup ;
alter tablespace BIWORK end backup ;
alter tablespace BNAPP end backup ;
alter tablespace CAAPP end backup ;
alter tablespace CAWORK end backup ;
alter tablespace CPAPP end backup ;
alter tablespace CUAUDIT end backup ;
alter tablespace CULARG1 end backup ;
alter tablespace CULARG2 end backup ;
alter tablespace CULARG3 end backup ;
alter tablespace CULARGE end backup ;
alter tablespace DIAPP end backup ;
alter tablespace DPAPP end backup ;
alter tablespace EOAPP end backup ;
alter tablespace EOLARGE end backup ;
alter tablespace EXAPP end backup ;
alter tablespace EXLARGE end backup ;
alter tablespace FAAPP end backup ;
alter tablespace FALARGE end backup ;
alter tablespace FAWORK end backup ;
alter tablespace FIAPP end backup ;
alter tablespace FILARGE end backup ;
alter tablespace FSAPP end backup ;
alter tablespace FSAPP1 end backup ;
alter tablespace FSLARGE end backup ;
alter tablespace FSWORK end backup ;
alter tablespace GIAPP end backup ;
alter tablespace GLAPP end backup ;
alter tablespace GLLARGE end backup ;
alter tablespace GLWORK end backup ;
alter tablespace GMAPP end backup ;
alter tablespace HRAPP end backup ;
```

Mark A. Hays

DRAFT – December 27, 2002

Page 19 of 35

```

alter tablespace HRLARGE end backup ;
alter tablespace HTAPP end backup ;
alter tablespace INAPP end backup ;
alter tablespace INLARGE end backup ;
alter tablespace INWORK end backup ;
alter tablespace LCAPP end backup ;
alter tablespace LCLARGE end backup ;
alter tablespace MGAPP end backup ;
alter tablespace MPAPP end backup ;
alter tablespace MPLARGE end backup ;
alter tablespace OEAPP end backup ;
alter tablespace OMAPP end backup ;
alter tablespace OMLARGE end backup ;
alter tablespace OMWORK end backup ;
alter tablespace PCAPP end backup ;
alter tablespace PCLARGE end backup ;
alter tablespace POAPP end backup ;
alter tablespace POLARGE end backup ;
alter tablespace POWORK end backup ;
alter tablespace PSWORK end backup ;
alter tablespace PTAMSG end backup ;
alter tablespace PTAPP end backup ;
alter tablespace PTAPPE end backup ;
alter tablespace PTCMSTAR end backup ;
alter tablespace PTLOCK end backup ;
alter tablespace PTPRC end backup ;
alter tablespace PTPRCS end backup ;
alter tablespace PTAUDIT end backup ;
alter tablespace PTRPTS end backup ;
alter tablespace PSIMGR end backup ;
alter tablespace PTTBL end backup ;
alter tablespace PTTLRG end backup ;
alter tablespace PTTREE end backup ;
alter tablespace PTWORK end backup ;
alter tablespace PVAPP end backup ;
alter tablespace PYAPP end backup ;
alter tablespace PY0LRG end backup ;
alter tablespace RSAPP end backup ;
alter tablespace RSLARGE end backup ;
alter tablespace TDAPP end backup ;
alter tablespace TDLARGE end backup ;
alter tablespace TDWORK end backup ;
alter tablespace TLAPP end backup ;
alter tablespace TLWORK end backup ;
alter tablespace TRAPP end backup ;
alter tablespace TRLARGE end backup ;
alter tablespace TRWORK end backup ;
alter tablespace WSAPP end backup ;
alter tablespace PSIMAGE end backup ;
alter tablespace PSINDEX end backup ;
98 rows selected.

```

```

SQL> spool off
SQL> exit

```

Edit the resulting file and remove the informational lines such as “98 rows selected.”

```
oracle@atsaix1:/oracle=> vi $ORACLE_HOME/end_backup.sql
```

Run the script with the following command.

```
oracle@atsaix1:/oracle=> sqlplus system/(password)
@$ORACLE_HOME/end_backup
```

```
Tablespace altered.
Tablespace altered.
Tablespace altered.
Tablespace altered.
Tablespace altered.
Tablespace altered.
...
```

Event occurs that causes data file loss or corruption (simulated with rm and dd commands).

While testing this procedure, the author created two simple scenarios from which to recover the database. Two data files were removed, in the first case, to simulate an error in which someone with a user id with sufficient privileges accidentally removes files. The second case is set up with dd command. Random data (/dev/zero) are written into the file at arbitrary locations. The commands are listed here for reference.

Scenario 1:

```
root@atsaix1:/u02/oradata/FIN8=> ls
amapp.dbf      arapp.dbf      biapp.dbf      cpapp.dbf      diapp.dbf
faapp.dbf      fsapp1.dbf     amlarge.dbf    arlarge.dbf    bilarge.dbf
cuaudit.dbf    dpapp.dbf      falarge.dbf    fslarge.dbf    amwork.dbf
arwork.dbf     biwork.dbf     cularg1.dbf    eoapp.dbf      fawork.dbf
fswork.dbf     apapp.dbf      aucapp.dbf     bnapp.dbf      cularg2.dbf
eolarge.dbf    fiapp.dbf      lost+found     aplarge.dbf    bdapp.dbf
caapp.dbf      cularg3.dbf    exapp.dbf      filarge.dbf    psrbs01.dbf
apwork.dbf     bdlarge.dbf    cawork.dbf     cularge.dbf    exlarge.dbf
fsapp.dbf
```

```
root@atsaix1:/u02/oradata/FIN8=> rm -f apapp.dbf apwork.dbf
```

Scenario 2:

```
root@atsaix1:/ dd if=/dev/zero of=/u02/oradata/FIN8/apwork.dbf bs=8192\
seek=30 count=5
5+0 records in.
5+0 records out.
```

The author unsuccessfully attempted to create a data corruption or inconsistency condition that would only be detected while exporting the database. All efforts to overcome Oracle's detection of file corruption at startup were unsuccessful.

In the event that data file corruption or database inconsistency is found during normal operation or while exporting to /dev/null for consistency checks, the technical

Mark A. Hays

professionals responsible for recovering the database must consider the timing of the corruption and the last FlashCopy / Oracle hot backup procedure. It's possible – in some cases likely -- that the inconsistent state existed prior to the last FlashCopy operation and the inconsistent state is captured on the target LUNs making them useless for recovery. In this case the files must be recovered from previous tape backup operations. This situation should clearly point out that this FlashCopy procedure is not a complete backup and recovery solution – it is a valuable tool in a complete backup and recovery plan.

Assign FlashCopy target LUNs to the same server as the source LUNs

Individuals responsible for administering the ESS storage server should assign the target LUNs used for the FlashCopy to the same server as the targets.

As stated previously, this paper does not attempt to provide steps for all administration tasks; therefore, please refer to the Technical Reference Materials section of this document for the publications that provide the steps for assigning LUNs to host adapters.

Once the LUNs have been assigned to the host, the system administrator should run the 'cfgmgr' command. The new hdisks and vpaths should be seen with the 'lspv' or 'lsdev – Cc disk' command.

The following output illustrates what the administrator should see before and after running 'cfgmgr'.

Before:

```
root@atsaix1:/=> lspv | grep vpath
vpath0 Available Data Path Optimizer Pseudo Device Driver
vpath1 Available Data Path Optimizer Pseudo Device Driver
vpath2 Available Data Path Optimizer Pseudo Device Driver
vpath3 Available Data Path Optimizer Pseudo Device Driver
vpath4 Available Data Path Optimizer Pseudo Device Driver
vpath5 Available Data Path Optimizer Pseudo Device Driver
vpath6 Available Data Path Optimizer Pseudo Device Driver
vpath7 Available Data Path Optimizer Pseudo Device Driver
vpath8 Available Data Path Optimizer Pseudo Device Driver
vpath9 Available Data Path Optimizer Pseudo Device Driver
vpath10 Available Data Path Optimizer Pseudo Device Driver
vpath11 Available Data Path Optimizer Pseudo Device Driver
vpath12 Available Data Path Optimizer Pseudo Device Driver
vpath13 Available Data Path Optimizer Pseudo Device Driver
vpath14 Available Data Path Optimizer Pseudo Device Driver
vpath15 Available Data Path Optimizer Pseudo Device Driver
```

Command Usage:

```
root@atsaix1:/=> cfgmgr
```


After:

```

root@atsaix1:/=> lspv | grep vpath
vpath0      00086cbf2db409b2    psoftvg
vpath1      00086cbf2db413b3    psoftvg
vpath2      00086cbf3217ae7b    psoftvg
vpath3      00086cbf3217b04a    psoftvg
vpath4      00086cbf3217b20c    psoftvg
vpath5      00086cbf3217b3dd    psoftvg
vpath6      00086cbf3217b5ae    psoftvg
vpath7      00086cbf321839e2    psoftvg
vpath8      00086cbf32183bb6    psoftvg
vpath9      00086cbf32183d7c    psoftvg
vpath10     00086cbf32183f3d    psoftvg
vpath11     00086cbf321840ff    psoftvg
vpath12     00086cbf321842bb    psoftvg
vpath13     00086cbf32184482    psoftvg
vpath14     00086cbf3218e6fc    psoftvg
vpath15     00086cbf3218e8bd    psoftvg
vpath16     none                    None
vpath17     none                    None
vpath18     none                    None
vpath19     none                    None
vpath20     none                    None
vpath21     none                    None
vpath22     none                    None
vpath23     none                    None
vpath24     none                    None
vpath25     none                    None
vpath26     none                    None
vpath27     none                    None
vpath28     none                    None
vpath29     none                    None
vpath30     none                    None
vpath31     none                    None

```

Create a new volume group containing all hdisks from the source volume group.

The 'recreatevg' is an extremely useful tool when the targets LUNs in a FlashCopy pair are used as a new volume group on the same server. Since the target LUN is a copy of the source at the disk block level, FlashCopy is unaware of any AIX Logical Volume Manager (LVM) or JFS file system constructs; as such, when the target LUNs are used on the same server, problems occur due to identical physical volume ids (PVID), volume group descriptor areas (VGDA), logical volume control blocks (LVCB), logical volume and JFS file system names.

All vpath or hdisk names must be included in the recreatevg command line. If even one LUN is forgotten the recreatevg command will fail and the volume group will not be made available. This behavior is expected and normal. In the AIX operating

environment, the volume group is never varied online unless all disks that comprise the volume group are included.

Please see Appendix A for command usage and a listing of the shell script.

The following illustrates the command usage and expected output before and after the commands are run.

Before:

```
root@atsaix1:/=> lsvg -o
psoftvg
rootvg
```

```
root@atsaix1:/=> df
Filesystem      512-blocks      Free %Used    Iused %Iused Mounted on
/dev/hd4         262144       95496   64%     2567    4% /
/dev/hd2        4194304      883664   79%    48887   10% /usr
/dev/hd9var      327680      296168   10%     673    2% /var
/dev/hd3         524288      493312    6%     145    1% /tmp
/dev/oraclelv    4096000     197072   96%    41831    9% /oracle
/dev/psoftlv    12255232     6230272  50%     9128    1% /psoft
/dev/tuxedolv    4096000     3726248  10%     2823    1% /tuxedo
/dev/oradat01    8192000     6057784  27%        22    1% /u01/oradata/FIN8
/dev/oradat02    8192000     5174280  37%        57    1% /u02/oradata/FIN8
/dev/oradat03    8192000     5549520  33%        54    1% /u03/oradata/FIN8
/dev/oradat04    8192000     6474680  21%        35    1% /u04/oradata/FIN8
/dev/oradat05    8192000     6909744  16%        18    1% /u05/oradata/FIN8
/dev/oradat06    8192000     1641896  80%        23    1% /u06/oradata/FIN8
```

Command Usage:

```
root@atsaix1:/=> recreatevg -y copyvg -Y cpy_ vpath16 vpath17 vpath18 vpath19\
> vpath20 vpath21 vpath22 vpath23 vpath24 vpath25 vpath26 vpath27 vpath28 \
> vpath29 vpath30 vpath31
0516-010 lqueryvg: Volume group must be varied on; use varyonvg command.
copyvg
```

The `-y` option provides a name for the new volume group. If omitted, the AIX operating environment will name it `vgxx` (where `xx` is the next unused sequential number starting at 00). The `-Y` option provides a prefix for all logical volumes to make them unique. The mount point of the file systems will contain a directory prefix of `/fs/...`

```
root@atsaix1:/=> mount -a
mount: 0506-324 Cannot mount /dev/oraclelv on /oracle: The requested resource is busy.
mount: 0506-324 Cannot mount /dev/psoftlv on /psoft: The requested resource is busy.
mount: 0506-324 Cannot mount /dev/tuxedolv on /tuxedo: The requested resource is busy.
mount: 0506-324 Cannot mount /dev/oradat01 on /u01/oradata/FIN8: The requested resource
is busy.
mount: 0506-324 Cannot mount /dev/oradat02 on /u02/oradata/FIN8: The requested resource
is busy.
mount: 0506-324 Cannot mount /dev/oradat03 on /u03/oradata/FIN8: The requested resource
is busy.
mount: 0506-324 Cannot mount /dev/oradat04 on /u04/oradata/FIN8: The requested resource
is busy.
mount: 0506-324 Cannot mount /dev/oradat05 on /u05/oradata/FIN8: The requested resource
is busy.
mount: 0506-324 Cannot mount /dev/oradat06 on /u06/oradata/FIN8: The requested resource
is busy.
```

replaying log for /dev/cpy_oraclelv.

The warnings from the previous command are due to the original file systems already mounted on the server.

After:

```
root@atsaix1:/=> lsvg -o
copyvg
psoftvg
rootvg
```

```
root@atsaix1:/=> df -k
Filesystem      1024-blocks      Free %Used      Iused %Iused Mounted on
/dev/hd4          131072       47744   64%       2567    4% /
/dev/hd2         2097152      441832   79%      48887   10% /usr
/dev/hd9var       163840      148084   10%        673    2% /var
/dev/hd3          262144      246656    6%        145    1% /tmp
/dev/oraclelv     2048000       98536   96%      41831    9% /oracle
/dev/psoftlv      6127616     3115128  50%       9128    1% /psoft
/dev/tuxedolv     2048000     1863124  10%       2823    1% /tuxedo
/dev/oradat01     4096000     3028892  27%        22    1% /u01/oradata/FIN8
/dev/oradat02     4096000     2587140  37%        57    1% /u02/oradata/FIN8
/dev/oradat03     4096000     2774760  33%        54    1% /u03/oradata/FIN8
/dev/oradat04     4096000     3237340  21%        35    1% /u04/oradata/FIN8
/dev/oradat05     4096000     3454872  16%        18    1% /u05/oradata/FIN8
/dev/oradat06     4096000     820948   80%        23    1% /u06/oradata/FIN8
/dev/cpy_oraclelv 2048000       98556   96%      41831    9% /fs/oracle
/dev/cpy_psoftlv  6127616     3118896  50%       9115    1% /fs/psoft
/dev/cpy_tuxedolv 2048000     1863124  10%       2823    1% /fs/tuxedo
/dev/cpy_oradat01 4096000     3028892  27%        22    1% /fs/u01/oradata/FIN8
/dev/cpy_oradat02 4096000     2587140  37%        57    1% /fs/u02/oradata/FIN8
/dev/cpy_oradat03 4096000     2774760  33%        54    1% /fs/u03/oradata/FIN8
/dev/cpy_oradat04 4096000     3237340  21%        35    1% /fs/u04/oradata/FIN8
/dev/cpy_oradat05 4096000     3454872  16%        18    1% /fs/u05/oradata/FIN8
/dev/cpy_oradat06 4096000     820948   80%        23    1% /fs/u06/oradata/FIN8
```

```
root@atsaix1:/=> lspv | grep vpath
vpath0      00086cbf2db409b2    psoftvg
vpath1      00086cbf2db413b3    psoftvg
vpath2      00086cbf3217ae7b    psoftvg
vpath3      00086cbf3217b04a    psoftvg
vpath4      00086cbf3217b20c    psoftvg
vpath5      00086cbf3217b3dd    psoftvg
vpath6      00086cbf3217b5ae    psoftvg
vpath7      00086cbf321839e2    psoftvg
vpath8      00086cbf32183bb6    psoftvg
vpath9      00086cbf32183d7c    psoftvg
vpath10     00086cbf32183f3d    psoftvg
vpath11     00086cbf321840ff    psoftvg
vpath12     00086cbf321842bb    psoftvg
vpath13     00086cbf32184482    psoftvg
vpath14     00086cbf3218e6fc    psoftvg
vpath15     00086cbf3218e8bd    psoftvg
vpath16     00086cbf38ed959e    copyvg
vpath17     00086cbf38ed97e7    copyvg
vpath18     00086cbf38ed9a3c    copyvg
vpath19     00086cbf38ed9c7a    copyvg
vpath20     00086cbf38ed9ebb    copyvg
vpath21     00086cbf38eda0ed    copyvg
vpath22     00086cbf38eda333    copyvg
vpath23     00086cbf38eda572    copyvg
vpath24     00086cbf38eda7bc    copyvg
vpath25     00086cbf38edaa03    copyvg
```

Mark A. Hays

DRAFT – December 27, 2002

Page 25 of 35



IBM Americas Advanced Technical Support

vpath26	00086cbf38edac4c	copyvg
vpath27	00086cbf38edaea5	copyvg
vpath28	00086cbf38edb0fd	copyvg
vpath29	00086cbf38edb33c	copyvg
vpath30	00086cbf38edb585	copyvg
vpath31	00086cbf38edb7cc	copyvg

Copy the missing or corrupt data files from the FlashCopy target to the proper directory structure in the existing Oracle database and recover the data files using the appropriate Oracle procedure.

Scenario 1:

```
root@atsaix1:/u02/oradata/FIN8=> ls
amapp.dbf      arapp.dbf      biapp.dbf      cpapp.dbf      diapp.dbf
faapp.dbf      fsapp1.dbf     amlarge.dbf    arlarge.dbf    bilarge.dbf
cuaudit.dbf    dpapp.dbf      falarge.dbf    fslarge.dbf    amwork.dbf
arwork.dbf     biwork.dbf     cularg1.dbf    eoapp.dbf      fawork.dbf
fswork.dbf     apapp.dbf      aucapp.dbf     bnapp.dbf      cularg2.dbf
eolarge.dbf    fiapp.dbf      lost+found     aplarge.dbf    bdapp.dbf
caapp.dbf      cularg3.dbf    exapp.dbf      filarge.dbf    psrbs01.dbf
apwork.dbf     bdlarge.dbf    cawork.dbf     cularge.dbf    exlarge.dbf
fsapp.dbf
```

```
root@atsaix1:/u02/oradata/FIN8=> rm -f apapp.dbf apwork.dbf
```

SQL*Plus: Release 8.1.7.0.0 - Production on Thu Dec 19 14:45:31 2002

(c) Copyright 2000 Oracle Corporation. All rights reserved.

Connected to:

Oracle8i Enterprise Edition Release 8.1.7.0.0 - Production

JServer Release 8.1.7.0.0 - Production

SQL> shutdown abort

ORACLE instance shut down.

SQL> startup

ORACLE instance started.

Total System Global Area 400363412 bytes

Fixed Size 73620 bytes

Variable Size 71577600 bytes

Database Buffers 327680000 bytes

Redo Buffers 1032192 bytes

Database mounted.

ORA-01157: cannot identify/lock data file 7 - see DBWR trace file

ORA-01110: data file 7: '/u02/oradata/FIN8/apapp.dbf'

```
root@atsaix1:/u02/oradata/FIN8=> cp -p /fs/u02/oradata/FIN8/apapp.dbf \
> /u02/oradata/FIN8/apapp.dbf
```

Mark A. Hays

DRAFT – December 27, 2002

Page 26 of 35

```
SQL> shutdown immediate
ORA-01109: database not open
Database dismounted.
ORACLE instance shut down.
```

```
SQL> startup
ORACLE instance started.
```

```
Total System Global Area  400363412 bytes
Fixed Size                  73620 bytes
Variable Size               71577600 bytes
Database Buffers           327680000 bytes
Redo Buffers                1032192 bytes
Database mounted.
ORA-01113: file 7 needs media recovery
ORA-01110: data file 7: '/u02/oradata/FIN8/apapp.dbf'
```

```
SQL> recover database
ORA-00283: recovery session canceled due to errors
ORA-01157: cannot identify/lock data file 9 - see DBWR trace file
ORA-01110: data file 9: '/u02/oradata/FIN8/apwork.dbf'
```

```
root@atsaix1:/u02/oradata/FIN8=> cp -p /fs/u02/oradata/FIN8/apwork.dbf\
> /u02/oradata/FIN8/apwork.dbf
```

```
SQL> shutdown abort
ORACLE instance shut down.
SQL> startup
ORACLE instance started.
```

```
Total System Global Area  400363412 bytes
Fixed Size                  73620 bytes
Variable Size               71577600 bytes
Database Buffers           327680000 bytes
Redo Buffers                1032192 bytes
Database mounted.
ORA-01113: file 7 needs media recovery
ORA-01110: data file 7: '/u02/oradata/FIN8/apapp.dbf'
```

```
SQL> recover database
Media recovery complete.
SQL> alter database open
Database altered.
```

Scenario 2:

```
root@atsaix1:/ dd if=/dev/zero of=/u02/oradata/FIN8/apwork.dbf bs=8192\
seek=30 count=5
5+0 records in.
5+0 records out.
```

```
oracle@atsaix1:/oracle=> sqlplus internal
Connected to:
```

Mark A. Hays



IBM Americas Advanced Technical Support

Oracle8i Enterprise Edition Release 8.1.7.0.0 - Production
JServer Release 8.1.7.0.0 - Production

```
SQL> shutdown immediate
ORA-01122: database file 9 failed verification check
ORA-01110: data file 9: '/u02/oradata/FIN8/apwork.dbf'
ORA-01251: Unknown File Header Version read for file number 9
```

```
SQL> shutdown abort
ORACLE instance shut down.
```

```
SQL> startup
ORA-01122: database file 9 failed verification check
ORA-01110: data file 9: '/u02/oradata/FIN8/apwork.dbf'
ORA-01251: Unknown File Header Version read for file number 9
```

oracle@atsaix1:/oracle=> sqlplus internal

SQL*Plus: Release 8.1.7.0.0 - Production on Thu Dec 19 15:42:15 2002

(c) Copyright 2000 Oracle Corporation. All rights reserved.

Connected to:
Oracle8i Enterprise Edition Release 8.1.7.0.0 - Production
JServer Release 8.1.7.0.0 - Production

```
SQL> shutdown immediate
Database closed.
Database dismounted.
ORACLE instance shut down.
```

```
SQL> startup
ORACLE instance started.
```

```
Total System Global Area  400363412 bytes
Fixed Size                  73620 bytes
Variable Size               71577600 bytes
Database Buffers            327680000 bytes
Redo Buffers                 1032192 bytes
Database mounted.
ORA-01122: database file 9 failed verification check
ORA-01110: data file 9: '/u02/oradata/FIN8/apwork.dbf'
ORA-01200: actual file size of 34 is smaller than correct size of 11520
blocks
```

```
SQL> recover database
Media recovery complete.
SQL> alter database open
Database altered.
```

Appendix A - AIX command 'recreatevg'

The usage and output of the shell command file are provided below for reference purposes.

Usage:

```
recreatevg [ -y VGname ] [ -p ] [ -f ] [ -Y lv_prefix | -l LvNameFile ]  
[ -L label_prefix ] [ -n ] PVname...  
Recreates a volume group that is already existing on the specified set  
of disks.
```

Location:

/usr/sbin/recreatevg

Output:

```
#!/bin/ksh  
# IBM_PROLOG_BEGIN_TAG  
# This is an automatically generated prolog.  
#  
# bos43S src/bos/usr/sbin/lvm/highcmd/recreatevg.sh 1.3  
#  
# Licensed Materials - Property of IBM  
#  
# Restricted Materials of IBM  
#  
# (C) COPYRIGHT International Business Machines Corp. 2000  
# All Rights Reserved  
#  
# US Government Users Restricted Rights - Use, duplication or  
# disclosure restricted by GSA ADP Schedule Contract with IBM Corp.  
#  
# IBM_PROLOG_END_TAG  
# @(#)97 1.3 src/bos/usr/sbin/lvm/highcmd/recreatevg.sh, cmdlvm, bos43S,  
s2000_22B3 6/1/00 14:03:05  
#  
# COMPONENT_NAME: (cmdlvm) Logical Volume Commands  
#  
# FUNCTIONS: importvg.sh  
#  
# ORIGINS: 27  
#  
# (C) COPYRIGHT International Business Machines Corp. 1989, 1998  
# All Rights Reserved  
# Licensed Materials - Property of IBM  
#  
# US Government Users Restricted Rights - Use, duplication or  
# disclosure restricted by GSA ADP Schedule Contract with IBM Corp.  
#  
## [End of PROLOG]  
  
# NAME: test_return()  
#  
# DESCRIPTION: Tests function return code. Will exit and output error message  
# if bad.  
#  
# INPUT:
```



```

#          $1 : Function return code
#
# OUTPUT:
#          Error messages (Standard Error)
#
# RETURN VALUE DESCRIPTION:
#          0      Successful
#
# NOTE: This function will not return if input value is invalid. (exit 1)
#
test_return()
{
    if [ "$1" != 0 ]
    then
        dspmsg -s 1 cmdlvm.cat 1320 "`lvmmmsg 1320`\n" recreatevg >& 2
        exit
    fi
}

##### cleanup #####
#
# NAME: cleanup()
#
# DESCRIPTION: Called from trap command to clean up environment and exit.
#
# INPUT: none
#
# OUTPUT:
#          Error messages (Standard Error)
#
# RETURN VALUE DESCRIPTION:
#          EXIT_CODE
#
# NOTE: This function will not return.
#
#
cleanup()
{
    trap '' 0 1 2 15

    if test $EXIT_CODE -eq 1
    then
        if test $GOTMAJOR -eq 1 -o $GOTMINOR -eq 1 -o $VGNAME_INODM -eq 1
        then
            # release the major and minor number and also the
            # VGNAME from the ODM
            putlvodm -G $VGNAME >/dev/null 2>&1
            GOTMAJOR=0
            GOTMINOR=0
            VGNAME_INODM=0
        fi

        # remove the vg device?
        if test $MADENODE -eq 1
        then
            rm -f /dev/$VGNAME
            MADENODE=0
        fi

        if [ -n "$lv_node_list" ]
        then
            for lv in $lv_node_list
            do
                rm -f /dev/$lv
                rm -f /dev/r$lv
            done
        fi

        fi

    rm -f /tmp/pvidlist$$ /tmp/lvidlist$$
    umask $OLD_UMASK      # restore umask
}

```

```

        exit $EXIT_CODE
    }

##### getlv_list #####
#
# NAME: getlv_list()
#
# DESCRIPTION: For each lv in the vg get the new name, and creates the lv
# special file.
#
# RETURNS:
#     returns on success
#     exits on error
#
getlv_list()
{
    no_of_lvs=`lqueryvg -p $PVNAME -n`
    lvlist=`lqueryvg -p $PVNAME -L | awk '{ print $2 }'`

    for lv in $lvlist
    do
        newlv=

        # If the lFLAG is specified (i.e, new lvnames are specified on the
        # command line)
        if [ -n "$lFLAG" ]
        then
            newlv=`grep -w ^$lv $lVAL | awk 'BEGIN{FS=":"} {print $2}'`

            # If the new name of lv is not specified on command line
            # get the default lvname
            if [ -z "$newlv" ]
            then
                newlv=`getlvname -N $VGNAME`
            else
                # If the new name of lv is specified on command
                # line but it is not null

                newlv=`getlvname -n $newlv -N $VGNAME 2>/dev/null`
                if [ -z "$newlv" ]
                then
                    newlv=`getlvname -N $VGNAME`
                fi
            fi
        else
            newlv=`getlvname -n $lv_prefix$lv -N $VGNAME 2>/dev/null`
            if [ -z "$newlv" ]
            then
                newlv=`getlvname -N $VGNAME`
            fi
        fi

        echo "$lv $newlv" >> /tmp/lvidlist$$

        #
        # Two devices must be created for each logical volume, one for block
        # i/o and the other for raw i/o.  Call mknod to do this.
        #
        mknod /dev/$newlv b $MAJOR $MINOR      # block i/o device
        test_return $?                         # check for error return from mknod
        mknod /dev/r$newlv c $MAJOR $MINOR     # raw i/o device -- lvname prefixed with r
        test_return $?                         # check for error return from mknod
        lv_node_list="$lv_node_list $newlv"

    done
}

##### check_input #####
#
# NAME: check_input()
#

```

```

# DESCRIPTION: Checks validity of input arguments.
#
# RETURNS:
#     returns on success
#     exits on error
#
check_input()
{
while [ "$#" -ne 0 ]    # While there are still arguments to process
do

    case "$1" in
        -l) if [ ! -r "$2" ]
            then
                # -l LvMapFile doesn't exist or is not readable
                dspmsg -s 1 cmdlvm.cat 1321 "`lvmmmsg 1321`\n" recreatevg >& 2
                exit
            fi
        *) shift;;      # shift past this option flag/value pair
        # unknown option, skip it
    esac

done    #end - while there are still arguments to process
}

Usage()
{
    dspmsg -s 1 cmdlvm.cat 1319 "`lvmmmsg 1319`\n" recreatevg >& 2
}

##### main #####
# Recreates a volume group that is already existing on the specified set of disks.
# Input:
#     Command line options and arguments:
#     recreatevg [ -y VGname ] [ -p ] [ -f ] [ -Y lv_prefix | -l LvNameFile ]
#     [ -L label_prefix ] [ -n ] PVname...
# Output:
#     Error Messages (Standard error)
#

PATH=/usr/bin:/etc:/usr/sbin:/usr/ucb:/usr/bin/X11:/sbin:$PATH

EXIT_CODE=1          #Initialize exit code. This will be reset to 0 before
                     #exiting only if mkvg completes successfully.

GOTMINOR=0
GOTMAJOR=0
MADENODE=0
VGNAME_INODM=0
lv_node_list=

OLD_UMASK=`umask`    # save old umask value
umask 117             # set umask for rw-rw----

#
# Trap on exit/interrupt/break to clean up
#
trap 'cleanup' 0 1 2 15

#
# Parse command line options
#
set -- `getopt y:pfY:l:L:n $*`

if [ $? != 0 ]        # Determine if there is a syntax error.
then
    Usage
    exit
fi

```

```

yFLAG= ; pFlag= ; FFlag= ; YFLAG= ; lFLAG= ; Lflag= ; nFLAG=
yVAL= ; lv_prefix=fs ; lVAL= ; label_prefix=fs ; PVNAME= ; PVLIST= ;
no_of_pvs=0 ; lv_node_list=

>/tmp/pvidlist$$
>/tmp/lvidlist$$

while [ $1 != -- ]           # While there is a command line option
do
    case $1 in
        -y) yFLAG='-y' ; yVAL="-n $2"; VGNAME=$2; shift; shift;; # VGname
        -p) pFlag='-p' ; shift;;
        -f) FFLAG='-f' ; shift;;
        -Y) YFLAG='-Y' ; lv_prefix=$2; shift; shift;;
        -l) lFLAG='-l' ; lVAL=$2; shift; shift;;
        -L) LFLAG='-L' ; label_prefix=$2; shift; shift;;
        -n) nFLAG='-n' ; shift;;
    esac
done #end - while there is a command line option

#
# Parse command line arguments
#

shift      # skip past "--" from getopt

if [ -n "$1" ] #if pvname argument on command line
then
    PVNAME=$1
    while [ -n "$1" ]
    do
        #for each pvname on the command line

        # Check for PV that is already in a VG
        getlvodm -j $PVNAME >/dev/null 2>&1
        if [ $? = 0 ]
        then
            dspmsg -s 1 cmdlvm.cat 696 "`lvmmmsg 696`\n" recreatevg $PVNAME >& 2
            test_return 1
        fi

        PVLIST="$PVLIST $1 " #save pvnames in string variable pvname
        ((no_of_pvs=$no_of_pvs + 1))
        shift                #skip to next command line argument
    done                    #end - for each pvname on the command line
else
    dspmsg -s 1 cmdlvm.cat 604 "`lvmmmsg 604`\n" recreatevg >& 2
    Usage
    exit
fi

#
# Check for invalid input
#

check_input $lFLAG $lVAL

# check for illegal combination of command line options
if [ -n "$YFLAG" -a -n "$lFLAG" ]
then
    dspmsg -s 1 cmdlvm.cat 1323 "`lvmmmsg 1323`\n" recreatevg >& 2
    Usage
    exit
fi

actualpvno=`lqueryvg -p $PVNAME -c`
if [ "$actualpvno" -gt "$no_of_pvs" -a -z "$FFLAG" ]
then
    dspmsg -s 1 cmdlvm.cat 1322 "`lvmmmsg 1322`\n" recreatevg >& 2

```

```

        exit
    fi

    OLD_VGID=`lqueryvg -p $PVNAME -v 2>/dev/null`
    ORIG_LANG=$LANG

    # check to see if the user specified a VGNAME first...
    if [ -n "$VGNAME" ]
    then
        LANG=C
        case $VGNAME in
            *[\ -,/:-@[-\^\`~\{~}~]*)
                #Invalid character in vgname
                LANG=$ORIG_LANG
                dspmsg -s 1 cmdlvm.cat 874 "`lvmmmsg 874`\n" recreatevg $VGNAME >& 2 ;
                test_return 1;;
            esac
        LANG=$ORIG_LANG
    fi

    if [ -n "$yVAL" ]
    then
        VGNAME=`getvgname $yVAL`
        if [ $? -ne 0 ]
        then
            test_return 1
        fi
    fi

    if [ -z "$VGNAME" ]
    then
        VGNAME=`getvgname`
        if [ $? -ne 0 ]
        then
            test_return 1
        fi
    fi

    VGNAME_INODM=1

    # Get major number.
    MAJOR=`lvgenmajor $VGNAME`
    test_return $?
    GOTMAJOR=1

    # Get minor number ( should be zero )
    MINOR=`lvgenminor $MAJOR $VGNAME`
    test_return $?
    GOTMINOR=1

    #
    # Create special device file for volume group with mknod
    #
    mknod /dev/$VGNAME c $MAJOR $MINOR
    if test $? -ne 0
    then
        dspmsg -s 1 cmdlvm.cat 870 "`lvmmmsg 870`\n" recreatevg /dev/$VGNAME >& 2 ;
    fi

    exit
fi
MADENODE=1

# Check if the specified lv prefix is valid
if [ -n "$YFLAG" ]
then
    VALID_CHECK=`odmget -q "prefix= $lv_prefix" PdDv`
    if [ -n "$VALID_CHECK" ]
    then
        dspmsg -s 1 cmdlvm.cat 693 "`lvmmmsg 693`\n" recreatevg "Y" >& 2
        exit
    fi
fi

```

```

# For all pv's in the input list change the pvid and
# create a file with the pvname, newpvid
for pv in $PVLIST
do
    if [ -z "$pFlag" ]
    then
        chdev -l $pv -a pv=clear 1>/dev/null
        test_return $?
        chdev -l $pv -a pv=yes 1>/dev/null
        test_return $?
    fi
    NEW_PVID=`getlvodm -p $pv`
    test_return $?
    echo "$pv $NEW_PVID" >> /tmp/pvidlist$$
done

# For each lv in the vg get the new name, and create the lv special file.
# create a file with the oldlvname newlvname
getlv_list

NEW_VGID=`lrecreatevg -y $VGNAME -j $MAJOR -v $OLD_VGID -p $no_of_pvs -l $no_of_lvs \
-r $label_prefix -P /tmp/pvidlist$$ -L /tmp/lvidlist$$`
test_return $?

importvg $FFLAG -N -y $VGNAME $PVNAME
test_return $?

# We no longer want to remove the vg should we be terminated
EXIT_CODE=0

if [ -n "$FFLAG" ]
then
    >/tmp/lvidlist1$$
    no_of_lvs=`lqueryvg -p $PVNAME -n`
    newlvlist=`lqueryvg -p $PVNAME -L | awk '{ print $2 }'`

    for lv in $newlvlist
    do
        grep -w $lv /tmp/lvidlist$$ >> /tmp/lvidlist1$$
    done

    mv /tmp/lvidlist1$$ /tmp/lvidlist$$

    for lv in $lv_node_list
    do
        grep -w $lv /tmp/lvidlist$$ >/dev/null
        if [ $? -ne 0 ]
        then
            rm -f /dev/$lv
            rm -f /dev/r$lv
            putlvodm -N $lv $NEW_VGID >/dev/null 2>&1
        fi
    done
fi

lrecreatelvcb -y $VGNAME -v $NEW_VGID -l $no of lvs \
-r $label_prefix -L /tmp/lvidlist$$

imfs $VGNAME
test_return $?

if [ -n "$nFLAG" ]
then
    varyoffvg $VGNAME
fi

```