Scrubbing Disks Using the Solaris™ Operating Environment Format Program

By Rob Snevely - Enterprise Technology Center

Sun BluePrints™ OnLine - June 2000

http://www.sun.com/blueprints
Scrubbing Disks Using the Solaris™ Operating Environment Format Program

A scriptable interface to the format utility

In this day of multi-terabyte databases, many people assume that disks are just a swappable, disposable commodity; and, for the most part that is true for many customers. However, there are several areas where being able to reuse disks is important. One example is for website or application hosting, in which case, a customer might decide to upgrade to a different machine or a newer type of storage. As the ASP, you would want to be able to reuse that disk for a new customer. But, you had better make sure that the old customers data is no longer on those drives. Another example is that a large company may want to sell old disks to a different unit inside the company, or to an outside party. This requires you to ensure that all of the division’s or company’s data is completely removed from the disks.

One might ask, why not just do “rm -r*” or newfs it. Well you could, and in a ideal world that would be sufficient. However, not everybody out there is honest, and if new data is not written to the sectors on those drives, the data is still there and could be recovered. The question is how much risk do you want to expose your company to? If you are in the military or a goverment contractor, you might even have requirements to purge data, as part of your contract.
**format: What is it? And why should you care?**

*format* is a disk partitioning and maintenance utility. It allows you to format, label, repair, analyze, and scrub data off of a disk. *format* also only works on SCSI and Fiber Channel disks that are directly seen by Solaris™ Operating Environment. So this procedure will not work with disks that are controlled by a HW Raid controller, since only the controller itself is seen by the *format* program.

For the purpose of this article, we will focus on the last item, Scrubbing disks.

**IMPORTANT NOTE:** *format* operations of the type defined here will destroy data that is on the disk. (After all that is what we are trying to do when we scrub a disk.) It is strongly recommended to NOT perform this on a production server. If you need to scrub disks, move them to a NON production machine. It may even be a good idea to dedicate a machine especially for this task.

Below is a session to scrub disk c11t4d0. First we start the *format* program. Select the disk we want to scrub, in this case c11t4d0. Enter the Analyze utility. Then run `verify`. `verify` will write data to the entire disk, twice. Then restore the original label to the disk.

---

**Note** – In the process of verifying the disk, all blocks on the disk are written to. Any disk block that has a non-recoverable error is marked as bad, added to the defect list for that drive, and thus made unavailable on the disk. So this process will also help ensure that only valid blocks on the disk are available for the next user of this disk.
*** Sessions Starts ***

root@rims: format
Searching for disks...done

AVAILABLE DISK SELECTIONS:

0. c0t2d0 <SUN4.2G cyl 3880 alt 2 hd 16 sec 135> /sbin@3,0/SUNW,fas@3,8800000/sd@2,0
1. c0t3d0 <SUN4.2G cyl 3880 alt 2 hd 16 sec 135> /sbin@3,0/SUNW,fas@3,8800000/sd@3,0
2. c0t4d0 <drive not available: formatting> /sbin@3,0/SUNW,fas@3,8800000/sd@4,0
3. c11t0d0 <SUN9.0G cyl 4924 alt 2 hd 27 sec 133> /sbin@b,0/SUNW,socal@0,0/sd@w2100002037079109,0
4. c11t1d0 <SUN9.0G cyl 4924 alt 2 hd 27 sec 133> /sbin@b,0/SUNW,socal@0,0/sd@w2100002037078952,0
5. c11t2d0 <SUN9.0G cyl 4924 alt 2 hd 27 sec 133> /sbin@b,0/SUNW,socal@0,0/sd@w2100002037079248,0
6. c11t3d0 <SUN9.0G cyl 4924 alt 2 hd 27 sec 133> /sbin@b,0/SUNW,socal@0,0/sd@w2100002037079285,0
7. c11t4d0 <SUN9.0G cyl 4924 alt 2 hd 27 sec 133> /sbin@b,0/SUNW,socal@0,0/sd@w2100002037078c2f,0
8. c11t5d0 <SUN9.0G cyl 4924 alt 2 hd 27 sec 133> /sbin@b,0/SUNW,socal@0,0/sd@w2100002037079259,0
9. c11t6d0 <SUN9.0G cyl 4924 alt 2 hd 27 sec 133> /sbin@b,0/SUNW,socal@0,0/sd@w2100002037079205,0
10. c11t7d0 <SUN9.0G cyl 4924 alt 2 hd 27 sec 133> /sbin@b,0/SUNW,socal@0,0/sd@w210000203707788c,0
11. c11t8d0 <SUN9.0G cyl 4924 alt 2 hd 27 sec 133> /sbin@b,0/SUNW,socal@0,0/sd@w210000203707921e,0
12. c11t9d0 <SUN9.0G cyl 4924 alt 2 hd 27 sec 133> /sbin@b,0/SUNW,socal@0,0/sd@w2100002037078f6d,0
13. c11t10d0 <SUN9.0G cyl 4924 alt 2 hd 27 sec 133> /sbin@b,0/SUNW,socal@0,0/sd@w210000203707914b,0
14. c11t11d0 <SUN9.0G cyl 4924 alt 2 hd 27 sec 133> /sbin@b,0/SUNW,socal@0,0/sd@w2100002037077f8b,0
15. c11t12d0 <SUN9.0G cyl 4924 alt 2 hd 27 sec 133> /sbin@b,0/SUNW,socal@0,0/sd@w2100002037078de0,0
16. c11t13d0 <SUN9.0G cyl 4924 alt 2 hd 27 sec 133> /sbin@b,0/SUNW,socal@0,0/sd@w2100002037079267,0

Specify disk (enter its number): 7
selecting c11t4d0
[disk formatted]
FORMAT MENU:
  disk - select a disk
  type - select (define) a disk type
  partition - select (define) a partition table
  current - describe the current disk
  format - format and analyze the disk
  repair - repair a defective sector
  label - write label to the disk
  analyze - surface analysis
  defect - defect list management
  backup - search for backup labels
  verify - read and display labels
  save - save new disk/partition definitions
  inquiry - show vendor, product and revision
  volname - set 8-character volume name
  !<cmd> - execute <cmd>, then return
  quit

format> analyze

ANALYZE MENU:
  read - read only test (doesn’t harm SunOS)
  refresh - read then write (doesn’t harm data)
  test - pattern testing (doesn’t harm data)
  write - write then read (corrupts data)
  compare - write, read, compare (corrupts data)
  purge - write, read, write (corrupts data)
  verify - write entire disk, then verify (corrupts data)
  print - display data buffer
  setup - set analysis parameters
  config - show analysis parameters
  !<cmd> - execute <cmd>, then return
  quit

analyze> verify
Ready to verify (will corrupt data). This takes a long time, but is interruptable with CTRL-C. Continue? y

  pass 0
  4923/26/7
  pass 1
  4923/26/7

Total of 0 defective blocks repaired.

analyze> quit
FORMAT MENU:

disk - select a disk
type - select (define) a disk type
partition - select (define) a partition table
current - describe the current disk
format - format and analyze the disk
repair - repair a defective sector
label - write label to the disk
analyze - surface analysis
defect - defect list management
backup - search for backup labels
verify - read and display labels
save - save new disk/partition definitions
inquiry - show vendor, product and revision
volname - set 8-character volume name
!<cmd> - execute <cmd>, then return
quit
format> backup
Disk has a primary label, still continue? y
Searching for backup labels...found.
Restoring primary label.

format> q
root@rims:

*** Session Ends ***

Now that you know how to do this by hand, you may want to automate the process. Do this by using the -f flag to format and a format command file, and passing a disk name as an argument on the command line.

A format command file allows format to take its command input from the file rather than from the standard input. The file must contain commands that appear just as they would if they had been entered from the keyboard. A disk name is the name of a disk: i.e c1t4d0.
So if we create a format command file called /opt/FORMATSCRIPTS/analyzecmd and put the following entries in it:

```
analyze
verify
quit
backup
quit
```

**Note** – Since running `format` will not happen interactively, you might also want to log a transcript of the `format` session. You do this with the `-l` flag.

Now run `format` and do the scrub of c11t4d0 and log the session as shown below.

```
format -f /opt/FORMATSCRIPTS/analyzecmd -l /opt/FORMATSCRIPTS/flog.c11t4d0 c11t4d0
```

To scrub all of the disks on a controller, get the disk names in a disk-list and a `format` command file and iterate through the disk-list.

First a list is required of the target disk names that need to be scrubbed. In this case, we will take all the disks on c11.

```
ls /dev/rdsk/c11* | sed 's/\//dev/\//rdsk//\///g' | sed 's/s2//g'>/opt/FORMATSCRIPTS/c11.disk.list
```

We now have a disk-list and a `format` command file.
Use the following script to iterate through the disk-list and scrub the disks.

```bash
#!/bin/sh
#
# Copyright (c) 2000 by Sun Microsystems, Inc.
# All rights reserved.
#
# WARNING: This script will destroy all data on the disk.
#    USE AT YOUR OWN RISK
#
# scrubdisks.sh
# A shell script to scrub disks using a disk-list and a format
# command file
case $# in
  0) echo "need more args\n Usage: $0 disklistfile log_dir
          formatcmdfile" 1>&2; exit 2 ;;
  1) echo "need more args\n Usage: $0 disklistfile log_dir
          formatcmdfile" 1>&2; exit 2 ;;
  2) echo "need more args\n Usage: $0 disklistfile log_dir
          formatcmdfile" 1>&2; exit 2 ;;
esac
# Define the Disk List file
DISKLIST=$1
# Define where the logfiles directory is
LOGDIR=$2
# Define the location of the format command file
FMTCMD=$3
#
# Make the log directory
mkdir -p $LOGDIR
#
# Iterate through the disk list
for DISKS in 'cat $DISKLIST'
do
  # Run the format command and check the exit status
  # to see if it worked correctly
  if format -f $FMTCMD -I $LOGDIR/formatlog.$DISKS $DISKS; then
    echo "format of disk $DISKS completed" >> /opt/FORMATSCRIPTS/
passedanalyze
  else
    echo "format of disk $DISKS failed" >> /opt/FORMATSCRIPTS/
failedanalyze
    exit 2
  fi
done
```
Using the examples above you can now run the following.

```
./scrubdisks.sh /opt/FORMATSCRIPTS/c11.disk.list /opt/FORMATSCRIPTS/c11.fmt.log /opt/FORMATSCRIPTS/analyzecmd
```

## Conclusion

There are many cases where disks need to be scrubbed of all data in order to be reused within a company, or possibly sold externally. `format` is a disk partitioning and maintenance utility that allows you to effectively remove all data from a disk or multiple disks on a controller with a disk-list. This article contains all of the information needed to scrub disks on a Solaris Operating Environment system.

**Author’s Bio: Rob Snevely**

Rob is a member of Sun’s Enterprise Technology Center technical staff. He has over 10 years experience in UNIX® system administration, networking, and performance tuning. His major responsibilities include architecting and designing data center and network architectures.