

Mirror / unmirror Solaris Volume manager (SCRIPT)

Explanation of the unmirror Script

The unmirror script generates three additional scripts:

- unmirror_cont
- remirror
- remirror_cont

The specific contents of the unmirror_cont, remirror, and remirror_cont scripts vary based on your volumes.

How to Use the unmirror Script

1. As root, run the following command:

```
# ./unmirror
```

2. Perform your system upgrades or changes.

3. If the upgrades or changes succeed, as root, run the following command:

```
# ./remirror
```

Or, instead, if the upgrades or changes fail, boot from the root disk of the second system.

After detaching submirrors, the unmirror script creates a symbolic link file in /etc/rc3.d for unmirror_cont and reboots the system

so that the link file continues to finish the unmirroring after the reboot. After finishing upgrades or changes, you can then run the

remirror script to re-create the mirrors that existed before on the system. The remirror script creates a symbolic link file

in /etc/rc3.d for remirror_cont before rebooting the system, and the link file continues to finish the mirroring after the reboot.

Here is the script, unmirror.sh

```
#!/bin/sh
# script name: unmirror
#
# The unmirror script automates the following tasks:
#
# 1. Unmirror the volumes so that one system is split into two systems.
# 2. Prepare the second system to be ready for use in the event of a failure.
# 3. Re-create the mirrors for the first system as they existed before.
#
# See the tech tip:
# http://www.sun.com/bigadmin/content/submitted/unmirror_remirror.jsp
# to explain further.
#
```

```
Unmount_volume ()
{
  if mount | grep /dev/md/dsk/$1 > /dev/null
  then
    if umount /dev/md/dsk/$1
    then
      :
    else
      echo "umount /dev/md/dsk/$1 failed"
      exit 2
    fi
  fi
}
```

```
Sanity_Check ()
{
  if metastat -p > md.cf
  then
    :
  else
    echo "SVM volume does not exist!"
    exit 2
  fi

  if metastat | nawk '/State:/{print $2}' | grep -v Okay
  then
    echo "Some volume's State is not Okay, run metastat to check it"
    exit 2
  fi
}
```

```
check_result=0
rows=0
```

```
exec < $MD_CF
while read line
do
  set $line > /dev/null

  if [ "$2" = -m ]
  then
    mirror=$1
    if [ $# -eq 5 ]
    then
      rows=2
      submirror1=$3
      submirror2=$4
    elif [ $# -eq 4 ]
    then
      echo "There is oneway mirror: $line"
      check_result=1
    else
```

```

        echo "Error: $line"
        exit 2
    fi
else
    if [ $rows -eq 2 ]
    then
        if [ $1 = $submirror1 ]
        then
            rows=`expr $rows - 1`
            if [ $2 -eq 1 -a $3 -eq 1 ]
            then
                :
            else
                Unmount_volume $mirror $submirror1
            fi
        else
            echo "The first field in $line mismatched $submirror1"
            exit 2
        fi
    elif [ $rows -eq 1 ]
    then
        rows=`expr $rows - 1`
    else
        echo "There is non-mirror: $line"
        check_result=1
    fi
fi
done
exec < /dev/tty

if [ $check_result -eq 1 ]
then
    echo "Do you want to continue to un-mirror?[y]"
    read answer
    [ "$answer" != y -a "$answer" != Y -a "$answer" != "" ] && exit
fi
}

```

```
FindDskMdSlices ()
```

```

{
    if [ $slice1_type = SLICE ]
    then
        dskslice1=/dev/dsk/$slice1
        rdskslice1=/dev/rdisk/$slice1
    elif [ $slice1_type = VOLUME ]
    then
        dskslice1=/dev/md/dsk/$slice1
        rdskslice1=/dev/md/rdisk/$slice1
    fi

    if [ $slice2_type = SLICE ]
    then
        dskslice2=/dev/dsk/$slice2
        rdskslice2=/dev/rdisk/$slice2
    elif [ $slice2_type = VOLUME ]
    then
        dskslice2=/dev/md/dsk/$slice2
        rdskslice2=/dev/md/rdisk/$slice2
    fi
}

```

```
DoUnmirror ()
```

```

{
    if [ "$1" = / ]
    then

```

```

metadetach $mirror $submirror2
metaroot $dskslice1

subsys_dskroot=$dskslice2
else
metadetach $mirror $submirror2

sed "s,/dev/md/dsk$slash_m>,$dskslice1;; \
s,/dev/md/rdisk$slash_m>,$rdkslice1," $VFSTAB > tmp$$
mv tmp$$ $VFSTAB
fi

if [ $slice1_type = VOLUME ]
then
metaclear $mirror
else
echo "metaclear $mirror\n" >> unmirror_cont
fi

sed "s,/dev/md/dsk$slash_m>,$dskslice2;; \
s,/dev/md/rdisk$slash_m>,$rdkslice2," $VFSTAB2 > tmp$$
mv tmp$$ $VFSTAB2
}

PrepareRemirror()
{
if [ $slice1_type = VOLUME ]
then
echo "if mount | grep $dskslice1" >> remirror
echo "then" >> remirror
echo "  if umount $dskslice1" >> remirror
echo "    then" >> remirror
echo "      :" >> remirror
echo "    else" >> remirror
echo "      echo \"umount $dskslice1 failed.\"\" >> remirror
echo "    exit 2" >> remirror
echo "  fi" >> remirror
echo "fi\n" >> remirror
fi

echo "metainit $mirror -m $submirror1\n" >> remirror

if [ "$1" = / ]
then
echo "metaroot $mirror\n" >> remirror
else
echo "sed \"s,$dskslice1>, /dev/md/dsk$slash_m,; \\\" >> remirror
echo "  s,$rdkslice1>, /dev/md/rdisk$slash_m,\" \"$VFSTAB > tmp\\$\" >> remirror
echo "mv tmp\\$\\$ $VFSTAB\n" >> remirror
fi

echo "metattach $mirror $submirror2" >> remirror_cont
}

Unmirror_PrepareRemirror()
{
# parameter VFSTAB, mirror, submirror1, submirror2, slice1, slice2 are from main()
slash_m="/$mirror"
slash_m_space="/${mirror} |/${mirror} "
if [ `egrep "$slash_m_space" $VFSTAB | wc -l` = 1 ]
then
set `egrep "$slash_m_space" $VFSTAB`
else
echo "mirror $mirror does not exist in $VFSTAB"
exit 2

```

```

fi

FindDskMdSlices
DoUnmirror $3
PrepareRemirror $3
}

# main()
MD_CF=$PWD/md.cf

Sanity_Check

cp /etc/vfstab ./vfstab2
metadb | nawk '/^ *a/{print $NF}' > metadb.txt

PWD=`pwd`
VFSTAB=/etc/vfstab
VFSTAB2=$PWD/vfstab2
METADB=$PWD/metadb.txt

grep -v "#" $VFSTAB > tmp$$
mv tmp$$ $VFSTAB

grep -v "#" $VFSTAB2 > tmp$$
mv tmp$$ $VFSTAB2

echo "#!/bin/sh" > unmirror_cont
echo "#Script name: unmirror_cont\n" >> unmirror_cont
echo "[ `dumpadm | nawk -F'[()]' '/device/{print $2}` = dedicated ] && dumpadm -d swap\n" >> unmirror_cont

echo "#!/bin/sh" > remirror
echo "#Script name: remirror\n" >> remirror
echo "VFSTAB=/etc/vfstab" >> remirror
echo "PWD=`pwd`\n" >> remirror

echo "#!/bin/sh" > remirror_cont
echo "#Script name: remirror_cont\n" >> remirror_cont

rows=0

exec < $MD_CF
while read line
do
    set $line > /dev/null

    if [ "$2" = -m ]
    then
        mirror=$1
        if [ $# -eq 5 ]
        then
            rows=2
            submirror1=$3
            submirror2=$4
        fi
    else
        if [ $rows -eq 2 ]
        then
            rows=`expr $rows - 1`
            if [ $2 = 1 -a $3 = 1 ]
            then
                slice1_type=SLICE
                slice1=$4
            else
                slice1_type=VOLUME
                slice1=$submirror1
            fi
        fi
    fi
done

```

```

    fi
    elif [ $rows -eq 1 ]
    then
        rows=`expr $rows - 1`
        if [ $2 = 1 -a $3 = 1 ]
        then
            slice2_type=SLICE
            slice2=$4
        else
            slice2_type=VOLUME
            slice2=$submirror2
        fi
        Unmirror_PrepareRemirror
    fi
fi
done
exec < /dev/tty

```

```

echo "mount $subsys_dskroot /mnt" >> unmirror_cont
echo "cp $VFSTAB2 /mnt/etc/vfstab" >> unmirror_cont
echo "cp /etc/system /mnt/etc/" >> unmirror_cont
echo "cp /etc/lvm/md.cf /mnt/etc/lvm/" >> unmirror_cont
echo "umount /mnt\n" >> unmirror_cont
echo "[ -f /etc/rc3.d/S99unmirror_cont ] && rm /etc/rc3.d/S99unmirror_cont" >> unmirror_cont

```

```

echo "echo \"The output of metastat -p is: \"" >> remirror
echo "metastat -p" >> remirror
echo "echo Press any key to reboot system to finish mirroring." >> remirror
echo "read answer" >> remirror
echo "chmod u+x remirror_cont" >> remirror
echo "ln -s \"$PWD/remirror_cont /etc/rc3.d/S99remirror_cont" >> remirror
echo "init 6" >> remirror
chmod u+x remirror

```

```

echo "[ `dumpadm | nawk -F'[()]' '/device/{print \$2}` = dedicated ] && dumpadm -d swap" >> remirror_cont
echo "[ -f /etc/rc3.d/S99remirror_cont ] && rm /etc/rc3.d/S99remirror_cont" >> remirror_cont

```

```

echo "Press any key to reboot system. After rebooting, check $PWD/log/unmirror_cont.log for result."
read answer
chmod u+x unmirror_cont
ln -s $PWD/unmirror_cont /etc/rc3.d/S99unmirror_cont
init 6

```

```

#####
### This script is submitted to BigAdmin by a user of the BigAdmin community.
### Sun Microsystems, Inc. is not responsible for the
### contents or the code enclosed.
###
###
### Copyright Sun Microsystems, Inc. ALL RIGHTS RESERVED
### Use of this software is authorized pursuant to the
### terms of the license found at
### http://www.sun.com/bigadmin/common/berkeley_license.jsp
#####

```