```
#!/bin/ksh
# Akadia AG, Arvenweg 4, CH-3604 Thun
my_backup.ksh
# ------------------------------------------------------------------
--
#
# File:       my_backup.ksh
#
# Autor:      Martin Zahn / 10.11.2002
#
# Purpose:    Backup Script
#
# Location:   /export/home/zahn/backup
#
# Certified:  Sun Solaris 7/8 with GNU Tools
#
# Restore:    cd /<any-directory-with-enough-space>
#
#             Restore entire Tape
#
#             dd if=/dev/rmt/0 ibs=20b | \
#             gunzip | gtar xvfb - 20
#
#             Restore single File
#
#             dd if=/dev/rmt/0 ibs=20b | \
#             gunzip | gtar xvfb - 20 "<file-pattern>"
# ------------------------------------------------------------------
--
#
##### ------------------------------------------------------------------
--
##### PART I: Local Variable Definition
##### ------------------------------------------------------------------
--
#
# Mail Recipient
#
SENDTO="Enter EMail Address here"
export SENDTO
#
# Location of Daily Backup Log Files
#
BACKUPDIR="/var/log/backup"
export BACKUPDIR
#
#
# The TAPEID File is the first File on the Archive. It contains a time
# stamp, so you can verify the date and time when the Archive was
created.
# Read the TAPEID File back with the following command:
# dd if=/dev/rmt/0 ibs=20b | gunzip | gtar xvfb - 20
"var/log/backup/tapeid"
#
TAPEID="${BACKUPDIR}/tapeid"
export TAPEID
#
# Directories to Backup. Note that Directories below /net/<machine-
name>
# are automatically mounted when used, see automount(1M). NEVER backup
# Directories with their absolute PATH (e.g. /home). The relative PATH
```

```
# must be directly below the root directory "/". The first file on the
# Tape must be the TAPEID File, so you can verify the Tape at anytime
later.
#
TOBACKUP="`echo ${TAPEID} | cut -c2-` net/cosmos users app export/home
usr/local etc"
export TOBACKUP
#
# Generated List of Files to backup
#
BACKUPLIST="${BACKUPDIR}/backuplist"
export BACKUPLIST
#
# Location of Daily Backup Log Files
#
BACKUPLOG="${BACKUPDIR}/my_backup-$(date '+%Y.%m.%d:%H:%M').log"
export BACKUPLOG
#
# Backup Host
#
BACKUPHOST="`uname -n`"
export BACKUPHOST
#
# Location of TAR Logfile
#
TARLOG="${BACKUPDIR}/tar-$(date '+%Y.%m.%d:%H:%M').log"
export TARLOG
#
# Location of GZIP Logfile
#
GZIPLOG="${BACKUPDIR}/gzip-$(date '+%Y.%m.%d:%H:%M').log"
export GZIPLOG
#
# Location of DD Logfile
#
DDLOG="${BACKUPDIR}/dd-$(date '+%Y.%m.%d:%H:%M').log"
export DDLOG
#
# Device Files of the Tape Drive
#
TAPEDEV="/dev/rmt/0"
export TAPEDEV
TAPEDEVNR="/dev/rmt/0cn"
export TAPEDEVNR
#
# Check entered Variables above
#
if [ ! -d ${BACKUPDIR} -o ! -w ${BACKUPDIR} ]
then
  echo "*** ${BACKUPDIR} doesn't exist or isn't writable ***" | \
  mailx -s "Backup ERROR on $${BACKUPHOST}" ${SENDTO}
  exit 1
fi
if [ ! -c ${TAPEDEV} ]
then
  echo "*** ${TAPEDEV} is not a character special device ***" | \
  mailx -s "Backup ERROR on $${BACKUPHOST}" ${SENDTO}
  exit 1
fi
if [ ! -c ${TAPEDEVNR} ]
then
```

```
  echo "*** ${TAPEDEVNR} is not a character special device ***" | \
  mailx -s "Backup ERROR on ${BACKUPHOST}" ${SENDTO}
  exit 1
fi
#
##### ---------------------------------------------------------------
--
##### PART II: Start the Backup
##### ---------------------------------------------------------------
--
#
# Rewind Tape
#
mt -f ${TAPEDEV} rewind
if [ ${?} -ne 0 ]
then
    echo "*** No Tape loaded or Drive offline ***" | \
    mailx -s "Backup ERROR on ${BACKUPHOST}" ${SENDTO}
    exit 1
fi
#
# Write EOF to Tape to test that it is writable
#
mt -f ${TAPEDEV} weof 1
if [ ${?} -ne 0 ]
then
    echo "*** Tape is write protected ***" | \
    mailx -s "Backup ERROR on ${BACKUPHOST}" ${SENDTO}
    exit 1
fi
#
# Tape is checked - start the Backup
#
cat > $BACKUPLOG 2>&1 <<-EOF
       Backup started at: `date`
       ---------------------------------------------------------------
-------
EOF
echo "Backup on ${BACKUPHOST} for ${TOBACKUP} created at `date`"
1>${TAPEID} 2>&1
cd /
#
# Generate List of Files to Backup
#
find ${TOBACKUP} \( -type f -o -type l -o -type s -o -type p \) -depth
-print > ${BACKUPLIST}
#
# Start the Backup in a Subshell (...), catch the RETURN Code for the
dd Command
#
( /usr/local/bin/gtar --create --blocking-factor=20 --files-
from=${BACKUPLIST} 2>${TARLOG} | \
  /usr/local/bin/gzip -9 2>${GZIPLOG} | \
  dd of=${TAPEDEV} obs=20b 2>${DDLOG}
) 1>>${BACKUPLOG} 2>&1
#
# Check Return Code for last command in Pipe (dd)
#
if [ ${?} -ne 0 ]
then
  STATUS=1
```

```
else
  STATUS=0
fi
#
##### ---------------------------------------------------------------
--
##### PART III: Cleanup, verify the Backup, check Logfiles
##### ---------------------------------------------------------------
--
rm ${TAPEID}
rm ${BACKUPLIST}
#
# Get number of File Marks on Tape
#
mt -f ${TAPEDEVNR} eom
NUMFILE=`mt -f ${TAPEDEVNR} status | awk '/file no/ { print $3 }'`
if [ "${NUMFILE}" -lt "1" ]
then
  echo "*** No Archive File found on Tape ***" | \
  mailx -s "Backup ERROR on ${BACKUPHOST}" ${SENDTO}
  exit 1
fi
#
# Rewind Tape
#
mt -f ${TAPEDEV} rewind
if [ ${?} -ne 0 ]
then
    echo "*** Tape cannot be rewinded ***" | \
    mailx -s "Backup ERROR on ${BACKUPHOST}" ${SENDTO}
    exit 1
fi
#
# If Size of Logfiles for ${TARLOG} or ${GZIPLOG} is > 0 then we have
an Error,
# dd is already checked in the Subshell
#
if [ -s ${TARLOG} ] || [ -s ${GZIPLOG} ]
then
  STATUS=1
fi

#
##### ---------------------------------------------------------------
--
##### PART IV: Send Mail to ${SENDTO}
##### ---------------------------------------------------------------
--
#
if [ "${STATUS}" = 0 ]
then
  cat >> $BACKUPLOG 2>&1 <<-EOF

        Backup Host: ${BACKUPHOST}
        Backup Devive: ${TAPEDEV}
        Backup Tape-ID: ${TAPEID}
        Backup Log-Directory: ${BACKUPDIR}
        Files backed up: ${TOBACKUP}

        Output from Command: dd
```

```
        ---------------------------------------------------------------
-
        `cat ${DDLOG}`

        Restore Information
        ---------------------------------------------------------------
-
        cd /<any-directory-with-enough-space>

        Restore entire Tape:

        dd if=${TAPEDEV} ibs=20b | \
        gunzip | gtar xvfb - 20

        Restore single File:

        dd if=${TAPEDEV} ibs=20b | \
        gunzip | gtar xvfb - 20 "<file-pattern>"

        Restore Tape-ID:

        dd if=${TAPEDEV} ibs=20b | \
        gunzip | gtar xvfb - 20 "`echo ${TAPEID} | cut -c2-`"

        Backup successfully finished at: `date`
        ---------------------------------------------------------------
-------
EOF
  cat ${BACKUPLOG} | mailx -s "Backup OK on ${BACKUPHOST}" ${SENDTO}
  exit 0
else
  cat >> $BACKUPLOG 2>&1 <<-EOF

        Backup Host: ${BACKUPHOST}
        Backup Devive: ${TAPEDEV}
        Backup Tape-ID: ${TAPEID}
        Backup Log-Directory: ${BACKUPDIR}
        Files backed up: ${TOBACKUP}

        Output from Command: dd
        ---------------------------------------------------------------
-
        `cat ${DDLOG}`

        Output from Command: gtar
        ---------------------------------------------------------------
-
        `cat ${TARLOG}`

        Output from Command: gzip
        ---------------------------------------------------------------
-
        `cat ${GZIPLOG}`

        Backup failed at: `date`
        ---------------------------------------------------------------
-------
EOF
  cat ${BACKUPLOG} | mailx -s "Backup ERROR on ${BACKUPHOST}"
${SENDTO}
  exit 1
```

```
fi
```