```
#!/bin/ksh
uptim()
{
upt=`last reboot|head -1|awk '{ print "Last " $2, $3, "on " $4, $5, $6, "at " $7 }'`
up=`uptime|sed s/\:/" "/g|awk '{ print "System has been up for " $4, $5, "and " $6, "hours, " $7, "minutes" }'`
echo "<center><table width=50% border=1>
<tr>
  <td><b>$up</b></td>
</tr>
<tr>
  <td><b>$upt</b></td>
</tr>
</table> </center>" >> $INDEX
}
tivoli()
{
TIV=`[ -d /opt/tivoli ] && pkginfo -l TIVsmCba |grep VERSION|awk '{ print $3"."$5"."$7 }'`
[ $? != 0 ] && TIV="Not Installed"
}
clearcase()
{
CLC=`[ -d /opt/rational ] && /opt/rational/clearcase/bin/cleartool -ver |grep clearcase_|tail -1|awk '{ print $1 }'`
if [ "$CLC" = "" ];then
CLC="Not Installed"
fi
}
samba()
{
smb1=0
smb2=0
smb=`[ -f /usr/sfw/sbin/smbd ] && /usr/sfw/sbin/smbd -V |awk '{ print $2 }'`
[ $? != 0 ] && smb1=1
smb="$smb `[ -f /usr/local/samba/bin/smbd ] && /usr/local/samba/bin/smbd -V |awk '{ print $2 }'`"
[ $? != 0 ] && smb2=2
smb3=`expr $smb1 + $smb2`
if [ $smb3 = 3 ];then
smb="Not Installed"
fi

}
sybase()
{
ds=`ps -ef|grep ASE|awk '{ print $9 }'|grep dataserver|sort -u`
dserv=`echo $ds`
if [ "$dserv" != "" ];then
syb=`/usr/local/bin/sudo -u sybase $dserv -v|head -1`
if [ $? != 0 ];then
syb="Not Installed"
else
syb=`echo $syb|sed s/"\/"/" "/g|awk '{ print $4 }'`
fi
else
syb="Not Installed"
fi
}
mqm()
{
mq=`pkginfo |grep mqm `
if [ $? != 0 ];then
 mq="Not Installed"
else
mq=`pkginfo -l mqm|grep VERSION|awk '{ print $2 }'`
fi
}
jdk()
{
java -fullversion > /tmp/jd 2>&1
```

```
if [ $? != 0 ] ; then
 jd="Not Installed"
else
jd=`tail -1 /tmp/jd|sed s/\"//g|awk '{ print $4 }'`
rm /tmp/jd
fi
}
obp()
{
OS=`uname -r`
case $OS in
5.7)
PRTDIAG=/usr/platform/`uname -i`/sbin/prtdiag
SOL="Solaris 7"
;;
5.8)
PRTDIAG=/usr/platform/`uname -i`/sbin/prtdiag
SOL="Solaris 8"
;;
5.9)
PRTDIAG=/usr/sbin/prtdiag
SOL="Solaris 9"
;;
5.10)
PRTDIAG=/usr/sbin/prtdiag
SOL="Solaris 10"
;;
esac
ob=`$PRTDIAG -v|grep OBP|awk '{ print $2 }'`
[ $? != 0 ] && ob="Problem Reading"
}
model()
{
mod=`uname -i|sed s/SUNW\,//g`
}
patch()
{
case `uname -r` in
5.10)
pat=`showrev |tail -1|awk '{ print $5 }'`
;;
*)
pat=`showrev |tail -1|awk '{ print $6 }'`
;;
esac
}
weblog()
{
wl=`[ -d /export/home/weblogic/bea ] && dir=true`
[ $? = 0 ] && wl=`grep WebLogic /export/home/weblogic/bea/registry.xml|grep Server|sed s/\"/" "/g|awk '{ print $6
}'`
wl1=`[ -d /opt/bea ] && wl1=true`
[ $? = 0 ] && wl1=`grep WebLogic /opt/bea/registry.xml|grep Server|sed s/\"/" "/g|awk '{ print $6 }'`
wl="$wl $wl1"
echo $wl
if [ "$wl" = " " ];then
  wl="Not Installed"
fi
}
maestro()
{
tws=`[ -d /opt/tws ] && /opt/tws/maestro/version/version|head -1 |awk '{ print $4,$5 }'`
[ $? != 0 ] && tws="Not Installed"
}
oracle()
{
ps -ef|grep pmon|grep -v grep > /dev/null 2>&1
```

```
if [ $? = 0 ];then
for ora_sid in `ps -ef|grep pmon|grep -v grep|nawk -F_ '{ print $3 }'`
do
        ORAENV_ASK=NO
        ORACLE_SID=$ora_sid
        . /usr/local/bin/oraenv
        ora=`sqlplus -v|awk '{ print $3 }'` > /dev/null 2>&1
done
else
        ora="Not Installed"
fi
}
filesystems()
{
WARN=85
CRIT=90
echo "<center><b><h2>Filesystem Health</h2></b></center>
<table width=100% border=1>
<tr>
  <td><h2><b>Filesystem</h2></td>
  <td><h2>1024_blocks</h2></td>
  <td><h2>Used</h2></td>
  <td><h2>Available</h2></td>
  <td><h2>Capacity</h2></td>
  <td><h2>Mount Point</h2></b></td>
</tr>
" >> $INDEX
df -F ufs -k|grep -v Filesystem > /tmp/$$
while read inputline
do
        filesystem="$(echo $inputline |awk '{ print $1 }')"
        kbytes="$(echo $inputline |awk '{ print $2 }')"
        used="$(echo $inputline |awk '{ print $3 }')"
        avail="$(echo $inputline |awk '{ print $4 }')"
        capacity="$(echo $inputline |awk '{ print $5 }')"
        mount_pt="$(echo $inputline |awk '{ print $6 }')"
        cap1=`echo $capacity|sed s/\%//g`
        if [ $cap1 -ge $WARN -a $cap1 -lt $CRIT ];then
                bgcol="orange"
        elif [ $cap1 -ge $WARN -a $cap1 -ge $CRIT ];then
                bgcol="red"
        else
                bgcol="green"
        fi
echo "<tr>
        <td>$filesystem</td>
        <td>$kbytes</td>
        <td>$used</td>
        <td>$avail</td>
        <td bgcolor=$bgcol>$capacity</td>
        <td>$mount_pt</td>
        </tr>" >> $INDEX
        done < /tmp/$$
rm /tmp/$$
echo "</table>
" >> $INDEX
}
netcheck()
{
echo "</table>
" >> $INDEX
echo "<center><b><h2>Network Health</h2></b></center>
<table width=100% border=1>
<tr>
  <td><h2><b>Interface</h2></td>
  <td><h2>MAC Address</h2></td>
  <td><h2>Speed</h2></td>
```

```
  <td><h2>Duplex</h2></b></td>
</tr>" >> $INDEX
# Determine the speed and duplex for each live NIC on the system
for INTERFACE in `netstat -i | egrep -v "^Name|^lo0" \
   | awk '{ print $1 }' | cut -d: -f1 | sort | uniq`
do
   # Only gather information for active interfaces
   # Note: "ce" interfaces can be "UP" in "ifconfig" but have link down
   /usr/sbin/ifconfig $INTERFACE | grep "^$INTERFACE:.*<UP," > /dev/null 2>&1 || continue
   # Skip "cip" ATM interfaces
   echo $INTERFACE | grep "^cip" > /dev/null 2>&1 && continue
   # "ce" interfaces
   if [ "`echo $INTERFACE | awk '/^ce[0-9]+/ { print }'`" ] ; then
      kstat > /dev/null 2>&1
      if [ $? -ne 0 ] ; then
         echo "The \"kstat\" command failed for interface $INTERFACE."
         continue
      fi
      # Determine the ce interface number
      INSTANCE=`echo $INTERFACE | cut -c 3-`
      DUPLEX=`kstat ce:$INSTANCE | grep link_duplex | awk '{ print $2 }'`
      case "$DUPLEX" in
         0) DUPLEX="link down" ;;
         1) DUPLEX="half" ;;
         2) DUPLEX="full" ;;
      esac
      SPEED=`kstat ce:$INSTANCE | grep link_speed | awk '{ print $2 }'`
      case "$SPEED" in
         0) SPEED="link down" ;;
         10) SPEED="10 Mbit/s" ;;
         100) SPEED="100 Mbit/s" ;;
         1000) SPEED="1 Gbit/s" ;;
      esac
      elif [ "`echo $INTERFACE | awk '/^bge[0-9]+/ { print }'`" ] ; then
      kstat > /dev/null 2>&1
      if [ $? -ne 0 ] ; then
         echo "The \"kstat\" command failed for interface $INTERFACE."
         continue
      fi
      # Determine the bge interface number
      INSTANCE=`echo $INTERFACE | cut -c 4-`
      DUPLEX=`kstat bge:$INSTANCE | grep link_duplex |head -1| awk '{ print $2 }'`
      case "$DUPLEX" in
         0) DUPLEX="link down" ;;
         1) DUPLEX="half" ;;
         2) DUPLEX="full" ;;
      esac
      SPEED=`kstat bge:$INSTANCE | grep link_speed | awk '{ print $2 }'`
      case "$SPEED" in
         0) SPEED="link down" ;;
         10) SPEED="10 Mbit/s" ;;
         100) SPEED="100 Mbit/s" ;;
         1000) SPEED="1 Gbit/s" ;;
      esac

   # "dmfe" interfaces
   elif [ "`echo $INTERFACE | awk '/^dmfe[0-9]+/ { print }'`" ] ; then
      # Only the root user should run "ndd"
      if [ "`id | cut -c1-5`" != "uid=0" ] ; then
         echo "You must be the root user to determine \
${INTERFACE_TYPE}${INSTANCE} speed and duplex information."
         continue
      fi
      DUPLEX=`/usr/sbin/ndd /dev/${INTERFACE} link_mode`
      case "$DUPLEX" in
         0) DUPLEX="half" ;;
         1) DUPLEX="full" ;;
```

```
    esac
    SPEED=`/usr/sbin/ndd /dev/${INTERFACE} link_speed`
    case "$SPEED" in
      10) SPEED="10 Mbit/s" ;;
      100) SPEED="100 Mbit/s" ;;
      1000) SPEED="1 Gbit/s" ;;
    esac
  # "bge" and "iprb" interfaces
  elif [ "`echo $INTERFACE | awk '/^iprb[0-9]+|bge[0-9]+/ { print }'`" ] ; then
    kstat > /dev/null 2>&1
    if [ $? -ne 0 ] ; then
      DUPLEX="The \"kstat\" command failed for interface $INTERFACE."
      continue
    fi
    # Determine the bge|iprb interface number
    INSTANCE=`echo $INTERFACE | tr -d '[a-z]'`
    INTERFACE=`echo $INTERFACE | tr -d '[0-9]'`
    DUPLEX=`kstat $INTERFACE:$INSTANCE | grep duplex | awk '{ print $2 }'`
    SPEED=`kstat $INTERFACE:$INSTANCE | grep ifspeed | awk '{ print $2 }'`
    case "$SPEED" in
      10000000) SPEED="10 Mbit/s" ;;
      100000000) SPEED="100 Mbit/s" ;;
      1000000000) SPEED="1 Gbit/s" ;;
    esac
  elif [ "`echo $INTERFACE | awk '/^e1000g[0-9]+/ { print }'`" ] ; then
    INSTANCE=`echo $INTERFACE | cut -c7-`
    # The duplex for e1000g devices can only be found with "dladm"
    DUPLEX=`dladm show-dev $INTERFACE | awk '{ print $NF }'`
    SPEED=`kstat e1000g:$INSTANCE | grep ifspeed | awk '{ print $2 }'`
    case "$SPEED" in
      10000000) SPEED="10 Mbit/s" ;;
      100000000) SPEED="100 Mbit/s" ;;
      1000000000) SPEED="1 Gbit/s" ;;
    esac
  # le interfaces are always 10 Mbit half-duplex
  elif [ "`echo $INTERFACE | awk '/^le[0-9]+/ { print }'`" ] ; then
    DUPLEX="half"
    SPEED="10 Mbit/s"
  # All other interfaces
  else
    INTERFACE_TYPE=`echo $INTERFACE | sed -e "s/[0-9]*$//"`
    INSTANCE=`echo $INTERFACE | sed -e "s/^[a-z]*//"`
    # Only the root user should run "ndd"
    if [ "`id | cut -c1-5`" != "uid=0" ] ; then
      echo "You must be the root user to determine \
${INTERFACE_TYPE}${INSTANCE} speed and duplex information."
      continue
    fi
    /usr/sbin/ndd -set /dev/$INTERFACE_TYPE instance $INSTANCE
    SPEED=`/usr/sbin/ndd -get /dev/$INTERFACE_TYPE link_speed`
    case "$SPEED" in
      0) SPEED="10 Mbit/s" ;;
      1) SPEED="100 Mbit/s" ;;
      1000) SPEED="1 Gbit/s" ;;
    esac
    DUPLEX=`/usr/sbin/ndd -get /dev/$INTERFACE_TYPE link_mode`
    case "$DUPLEX" in
      0) DUPLEX="half" ;;
      1) DUPLEX="full" ;;
      *) DUPLEX="" ;;
    esac
  fi
  ETHER=`/usr/sbin/ifconfig $INTERFACE|grep ether|awk '{ print $2 }'` > /dev/null 2>&1
  echo "<tr>
      <td>$INTERFACE</td>
      <td>$ETHER</td>
      <td>$SPEED</td>
```

```
        <td>$DUPLEX</td>
        </tr>" >> $INDEX

done
echo "</table>
" >> $INDEX
}
sds()
{
echo "<center><h2>Disk Suite</h2></center>" >> $INDEX
metastat > /dev/null 2>&1
if [ $? = 0 ];then
metastat -p|grep "\-m" > /tmp/metastat.$$
tmpfil=/tmp/metastat.$$

while read inputline
do
        mirror="$(echo $inputline |awk '{ print $1 }')"
        sub1="$(echo $inputline |awk '{ print $3 }')"
        sub2="$(echo $inputline |awk '{ print $4 }')"
        state1="`metastat $sub1|grep Okay|awk '{ print $4 }'`"
        if [ "$state1" = "Okay" ];then
                state1="Okay"
                bgcol1="green"
        else
                state1="Maintenance"
                bgcol1="red"
        fi
        state2="`metastat $sub2|grep Okay|awk '{ print $4 }'`"
        if [ "$state2" = "Okay" ];then
                state2="Okay"
                bgcol2="green"
        else
                state2="Maintenance"
                bgcol2="red"
        fi
        echo "
<center><table width=50% border=1>
<tr>
  <td><b>Mirror</b></td>
  <td>$mirror</td>
  <td><b>State</b></td>
</tr>
<tr>
  <td><b>Sub Mirror</b></td>
  <td>$sub1</td>
  <td bgcolor=$bgcol1><b>$state1</b></td>
</tr>
<tr>
  <td><b>Sub Mirror</b></td>
  <td>$sub2</td>
  <td bgcolor=$bgcol2><b>$state2</b></td>
</tr>
</table></center>
" >> $INDEX
done < $tmpfil
echo "</body>
</html>" >> $INDEX
rm $tmpfil
else
        echo "
<center><table width=50% border=1>
<tr>
  <td><b>SDS not installed</b></td>
</tr>" >> $INDEX
fi
}
```

```
top20()
{
echo "<center><h2>Top 20 Processes</h2>
<table width=60% border="1">
<tr>
  <td align=left><pre>" >> $INDEX
prstat -n 20 1 1 >> $INDEX
echo "</pre>
  </td>
</tr></table></pre>" >> $INDEX
}
messages()
{
echo "<center><h2>System Messages</h2>
<table width=80% border="1">
<tr>
  <td align=left><pre>" >> $INDEX
cat /var/adm/messages|egrep -vi 'adjust|ntpdate|anonymous' >> $INDEX
echo "</pre>
  </td>
</tr></table></pre>" >> $INDEX
}
sargraf()
{
echo "<br>
<center><h2>Sar Graph</h2>
<img width=80% src=/sar/$host/${MONTH}${DATE}_SolariscpuSar.jpg></center>
<br>" >> $INDEX
}


#####################################################################
#main
#####################################################################
host=`uname -n`
PATH=$PATH:/usr/local/bin:/usr/sbin:/usr/bin
INDEX=/tmp/index.html
dat=`date`
DATE=`TZ=$TZ+24 date "+%d"`
MONTH=`TZ=$TZ+24 date "+%b"`

echo "<html>
<body bgcolor=#ffffff>
<h3><center>$dat </center></h3>
<h1><ul><center>System Information for $host.</center></ul></h1> " > $INDEX
uptim
echo "
<h2><center>Application Releases and Versions </center></h2>
<table width=100% border="1">
<tr>
  <td><h2><b>System Name</h2></td>
  <td><h2>Model</h2></td>
  <td><h2>Operating System</h2></td>
  <td><h2>Patch Level</h2></td>
  <td><h2>OBP Level</h2></td>
  <td><h2>Java </h2></td>
  <td><h2>MQ Series</h2></td>
  <td><h2>Sybase</h2></td>
  <td><h2>TSM</h2></td>
  <td><h2>TWS</h2></td>
  <td><h2>Weblogic</h2></td>
  <td><h2>Samba</h2></td>
  <td><h2>Oracle</h2></b></td>
</tr>
" >> $INDEX

        clearcase
```

```
        model
        tivoli
        samba
        sybase
        mqm
        jdk
        obp
        patch
        weblog
        maestro
        oracle
echo "<tr>
        <td>$host</td>
        <td>$mod</td>
        <td>$SOL</td>
        <td>$pat</td>
        <td>$ob</td>
        <td>$jd</td>
        <td>$mq</td>
        <td>$syb</td>
        <td>$TIV</td>
        <td>Not Installed</td>
        <td>Not Installed</td>
        <td>$smb</td>
        <td>$ora</td>
        </tr>
</table><br>
" >> $INDEX
filesystems
netcheck
sds
top20
sargraf
messages
echo "</body>
</html>" >> $INDEX


ADDRESSEE="sysinfo"
TARGET="lnuxdthm"


cat $INDEX | /usr/sbin/mailx -s ${ADDRESSEE}@${TARGET}

rm $INDEX
```