#!/bin/ksh

########################################################################################

# Contents :

# - /aedc/etc/work/aedc/SCC/setsccenv.ksh

# - variables: CAE\_BIN,SCC\_BIN,SYB\_DIR,sybase\_path

# SCCROOT,SCCTMP,SCCCNF,SCCERR,HIS,daysec,opr\_list,offset

# - /aedc/etc/work/aedc/SCC/aedc\_SCC\_IOfunctions

# - shaftota

# - othr\_disply

# - multi\_print

# - asc\_print

# - \_PRT

# - undr\_prep

# - /aedc/etc/work/aedc/SCC/aedc\_SCC\_Timefunctions

# - creat\_rng\_statment

# - days\_in\_mon

# - sep\_date

# - enter\_range

# - check\_range

# - enter\_his\_date\_range

# - prep\_aux\_dates

# - add\_86400

# - createDir

# - zedit

# - zdxdiff

# - dcc\_ss\_replace

# - replacement

# - replacement1

# - select\_db\_serv

# - data\_src\_select

# - read\_pw

# - check\_master\_sys

# - filter\_acc\_date\_range

# - filter\_his\_date\_range

# - disp\_count

# - get\_Alarm

# - read\_fac

# - MkChkDir

########################################################################################

export CAE\_BIN=/aedc/bin

export SCC\_BIN=/aedc/etc/work/aedc/SCC/bin

export SYB\_DIR=/usr/local/sybase/sybase1002/bin

sybase\_path="/usr/local/sybase/sybase10/interfaces"

. /aedc/etc/work/aedc/SCC/setsccenv.ksh

. /aedc/etc/work/aedc/SCC/aedc\_SCC\_Timefunctions

. /aedc/etc/work/aedc/SCC/aedc\_SCC\_IOfunctions

if [[ ! -s $SCCTMP ]]

then

echo "creating directory /aedc/tmp/scc"

mkdir /aedc/tmp/scc

chmod 777 /aedc/tmp/scc

fi

id | cut -d")" -f1 | cut -d"(" -f2 | read user

if [ ! -d "/aedc/tmp/scc/${user}" ]

then

mkdir /aedc/tmp/scc/${user}

fi

SCCTMP=/aedc/tmp/scc/${user}

chmod 777 /aedc/tmp/scc/${user}

echo "Jan 1 1970 ` ${CAE\_BIN}/asctime 3600 |awk '{print $4}'|awk 'FS=":"{print $1}'`:00AM"|read offset

#echo "Jan 1 1970 ` ${CAE\_BIN}/asctime 0 |awk '{print $4}'|awk 'FS=":"{print $1}'`:00AM"|read offset

function createDir {

if [ ! -d $1 ]

then

mkdir $1

fi

}

function zedit {

comp\_fl=$1

shift

parmt=$\*

uncompress $comp\_fl

echo $comp\_fl | sed "s/.Z$//" | read uncomp\_fl

xedit $uncomp\_fl $parmt

compress $uncomp\_fl

}

function zdxdiff {

comp\_fl1=$1

comp\_fl2=$2

uncompress $comp\_fl1

uncompress $comp\_fl2

echo $comp\_fl1 | sed "s/.Z$//" | read uncomp\_fl1

echo $comp\_fl2 | sed "s/.Z$//" | read uncomp\_fl2

dxdiff $uncomp\_fl1 $uncomp\_fl2

if [ "$comp\_fl1" != "$uncomp\_fl1" ]

then

compress $uncomp\_fl1

fi

if [ "$comp\_fl2" != "$uncomp\_fl2" ]

then

compress $uncomp\_fl2

fi

}

function dcc\_ss\_replace {

#=======================================================================

# $1 : E-SS1 ,M-DP3 ,or, W-SS5 ,....etc

# $2 : input file

# It is used to replace $1 in the input file by its real name

#----------------------------------------------------------

# GROUP : SCC

# AUTHOR : Eng. Reham Sobhy Eng. Dalia El Emam Eng. Shahira

# DATE : 9/12/2001

#=======================================================================

function replacement {

awk 'FS="!"{print $1}' $SCCCNF/DCC\_SS\_DP|grep -w $item | awk 'FS="\t" {printf("%-24s#\n", substr($2,1,24))}'|read desc

sed "s/$item /$desc /g;s/$item$/$desc/g;s/$item /$desc /g" $infile |sed "s/#/ /g"> $SCCTMP/${basefile}\_desc

chmod 666 $SCCTMP/${basefile}\_desc

cp $SCCTMP/${basefile}\_desc $infile

}

function replacement1 {

awk 'FS="!"{print $1}' $SCCCNF/DCC\_SS\_DP|grep -w $item | awk 'FS="\t" {printf("%s\n",$2)}'|read desc

sed "s/$item /$desc /g;s/$item$/$desc/g;s/$item /$desc /g" $infile > $SCCTMP/${basefile}\_desc

chmod 666 $SCCTMP/${basefile}\_desc

cp $SCCTMP/${basefile}\_desc $infile

}

if [[ $# -eq 2 ]]

then

infile=$2

echo $infile | sed 's-/- -g'|read list

for basefile in $list

do

print -n ""

done

grep -v "====" $SCCCNF/DCC\_SS\_DP | awk 'FS="!"{print $1}' | grep -v "S / S" |awk '{printf("%s ",$1)}' | read base\_list

case $1 in

"-a")

for item in $base\_list

do

if [ ! -z "`grep -w $item $infile`" ]

then

replacement

fi

done

;;

"-i")

for item in $base\_list

do

if [ ! -z "`grep -w $item $infile`" ]

then

replacement1

fi

done

;;

\*)

item=$1

replacement

esac

else

print "Usage : <function name> <-a> <inputfile name> :to replace all items in file

<function name> <item name> <inputfile name> :to replace one item at a time"

fi

cat $infile

}

function select\_db\_serv {

#=================================================

# To select between AEDC\_SYSTEM or AEDC\_TRAINER

#=================================================

if [[ -z $sev ]]

then

read sev?"

Enter DataBase Server Name ( AEDC\_SYSTEM or AEDC\_TRAINER ) >>> "

if [[ -z $sev ]]

then

sev=AEDC\_SYSTEM

fi

fi

}

function data\_src\_select {

#================================================================

# To select your data\_source : DataBase or SCC Historical files

#================================================================

read data\_src?"

Do you want to get data from :

1) DataBase [recomended for last week]

2) Historical files [recomended for far than last week]

enter your choice (<CR> for DB) >> "

if [[ -z $data\_src ]]

then

data\_src=1

fi

}

function read\_pw {

#===========================================================

# USAGE: read\_pw <message to be printed to the user>

#===========================================================

read pw?"$\* >> `tput font1` "

tput sgr0

}

function check\_master\_sys {

echo ===============================

server\_name=`rsh $SYS1 $SYBASE/install/showserver |awk -F"-s" '$2 ~ // {print $2}'|awk '$1 {print $1}'`

if [ "`echo $server\_name`" = "" ]

then

server\_name=`rsh $SYS2 $SYBASE/install/showserver |awk -F"-s" '$2 ~ // {print $2}'|awk '$1 {print $1}'`

if [ "`echo $server\_name`" != "" ]

then

DBA\_HOST=$SYS2

echo The master sys node is $DBA\_HOST

else

echo `tput blink`there is no master sys `tput sgr0`

fi

else

DBA\_HOST=$SYS1

echo The master sys node is $DBA\_HOST

fi

echo ===============================

}

function filter\_acc\_date\_range {

#===============================================================================================

# Searching for time range

# 1) FROM date1\_asc TO date3\_asc ------>> search in the first file where date>date1\_asc

# 2) FROM date3\_asc TO date4\_asc ------>> search in all files inside this rang

# 3) FROM date5\_asc TO date6\_asc ------>> search in the last file where date<date6\_asc

# 4) after start of today if required

# ----------------------------------------------------

# Usage : filter\_acc\_date\_range [[-t <acc|pkacc>]|[-i acc\_id]|[-n acc\_name]] [-f out\_filename] [-DB]

# -DB to get all data from DataBase and warning with the oldest time found in it

# defaultOutFile: $SCCTMP/Filter\_acc OR $SCCTMP/Filter\_pkacc

#===============================================================================================

function SQL\_filter\_acc {

# I/P variables : $AccFlType $strt\_tm $end\_tm $AccID $outAccFl

case $AccFlType in

acc)

echo "

select \* from T0432\_data

where C0432\_date between $strt\_tm and $end\_tm

and C0401\_aid = ${AccID}

go" | isql -Udbu -Pdbudbu |

grep -v "C0401" | grep -v "affected" | grep -v [-][-][-] | tee -a ${outAccFl}1

;;

pkacc)

echo "

select aid=str(C0401\_aid,6,0) , date=C0434\_date , value=str(C0434\_value,10,4) ,

ass\_val=str(C0434\_assoc\_value,10,4) , occ\_tm=C0434\_time\_of\_occurrence ,

sts=C0434\_status , ass\_sts=C0434\_assoc\_status

from T0434\_peak\_data

where C0434\_date between $strt\_tm and $end\_tm

and C0401\_aid = ${AccID}

go" | isql -Udbu -Pdbudbu |

grep -v "aid" | grep -v "affected" | grep -v [-][-][-] | tee -a ${outAccFl}1

;;

esac

}

Usage="Usage : $0 [[-t <acc|pkacc>]|[-i acc\_id]|[-n acc\_name]] [-f out\_filename] [-DB]"

${CAE\_BIN}/mktime `date +"%C%y/%m/%d:00:00:00"`|read tody\_strt\_sec

AccFlType=""

AccID=""

AccNm=""

outAccFl=""

DBflag=off

while [ ! -z "$1" ]

do

case $1 in

-t)

AccFlType=$2

echo "@${date1\_asc}@${date6\_asc}" > $SCCTMP/ID\_Tm\_list

shift

;;

-i)

AccID=$2

echo "select C0401\_name,C0411\_accnt\_type from T0401\_accounts

where C0401\_aid=${AccID}

go"|isql -U dbu -P dbudbu | head -6 | tail -2 | { read AccNm ; read AccTyp ; }

echo "${AccID}@${date1\_asc}@${date6\_asc}" > $SCCTMP/ID\_Tm\_list

shift

;;

-n)

AccNm=$2

echo "select C0401\_aid,C0411\_accnt\_type from T0401\_accounts

where C0401\_name='${AccNm}'

go"|isql -U dbu -P dbudbu | head -3 | tail -1 | awk 'OFS="\n"{print $1,$2}' |

{ read AccID ; read AccTyp ; }

echo "${AccID}@${date1\_asc}@${date6\_asc}" > $SCCTMP/ID\_Tm\_list

shift

;;

-f)

outAccFl=$2

shift

;;

-DB)

DBflag=on

;;

esac

shift

done

if [[ ! -z "$AccID" && ! -z "$AccNm" ]]

then

case $AccTyp in

1|3|8) AccFlType=acc ;;

2) AccFlType=pkacc ;;

esac

else

echo no specific Acc or Acc not found

fi

if [ -z "$outAccFl" ]

then

outAccFl=$SCCTMP/Filter\_${AccFlType}

fi

print -n "" > $outAccFl

while read ln # from $SCCTMP/ID\_Tm\_list #

do

echo $ln | awk 'FS="@",OFS="\n"{print $1,$2,$3}' | { read AccID ; read Date1\_asc ; read Date6\_asc ; }

${CAE\_BIN}/mktime `echo ${Date1\_asc} |cut -d":" -f1`:00:00:00 |read date3

${CAE\_BIN}/mktime `echo ${Date6\_asc} |cut -d":" -f1`:00:00:00 |read date5

${CAE\_BIN}/mktime $Date1\_asc | read date1\_sec

${CAE\_BIN}/mktime $Date6\_asc | read date6\_sec

if [ $DBflag = "off" ]

then

# Part ( 1 )

# ------------

${SCC\_BIN}/var\_asctime $date3 %d\_%m\_%Y.${AccFlType}.Z | read frst\_filename

zcat ${HIS}/${frst\_filename} | grep " ${AccID} " |

sed "s/^/${AccID}/" | awk ' $2==$1{print $0}' | sed "s/${AccID}//" |

sed "s/^/${date1\_sec}/" | awk ' $3>=$1{print $0}' | sed "s/${date1\_sec}//" > $SCCTMP/first\_file

print -n "" > ${outAccFl}1

if [ $date5 -gt $date3 ]

then

cat $SCCTMP/first\_file | tee ${outAccFl}1

# Part ( 2 )

# ------------

date=`expr $date3 + 86400`

while [ "$date" -lt "$date5" ]

do

${SCC\_BIN}/var\_asctime $date %d\_%m\_%Y.${AccFlType}.Z | read filename

zcat ${HIS}/${filename} | grep " $AccID " |

sed "s/^/${AccID}/" | awk ' $2==$1{print $0}' | sed "s/${AccID}//" |

tee -a ${outAccFl}1

date=`expr $date + $daysec`

done

# Part ( 3 )

# ------------

${SCC\_BIN}/var\_asctime $date5 %d\_%m\_%Y.${AccFlType}.Z | read last\_filename

zcat ${HIS}/${last\_filename} | grep " ${AccID} " |

sed "s/^/${AccID}/" | awk ' $2==$1{print $0}' | sed "s/${AccID}//" |

> $SCCTMP/last\_file

else

cat $SCCTMP/first\_file > $SCCTMP/last\_file

fi

sed "s/^/${date6\_sec}/" $SCCTMP/last\_file | awk ' $3<=$1{print $0}' |

sed "s/${date6\_sec}//" | tee -a ${outAccFl}1

else # if [ $DBflag = "off" ]

# part ( 1-3 )

# -----------

strt\_tm=${date1\_sec}

if [[ "${date6\_sec}" -lt "${tody\_strt\_sec}" ]]

then

end\_tm=${date6\_sec}

else

end\_tm=${tody\_strt\_sec}

fi

print -n "" > ${outAccFl}1

SQL\_filter\_acc | head -1 | awk '{print $2}' | read oldest\_tm

echo "warning: oldest data for AccID=$AccID is from `asctime $oldest\_tm` " > $outAccFl

fi # if [ $DBflag = "off" ]

# part ( 4 )

# -----------

end\_tm=0

strt\_tm=0

if [[ "${date6\_sec}" -gt "${tody\_strt\_sec}" ]]

then

end\_tm=${date6\_sec}

if [[ "${date1\_sec}" -gt "${tody\_strt\_sec}" ]]

then

strt\_tm=${date1\_sec}

else

strt\_tm=${tody\_strt\_sec}

fi

SQL\_filter\_acc

fi

cat ${outAccFl}1 >> $outAccFl

done < $SCCTMP/ID\_Tm\_list

}

function filter\_his\_date\_range {

#===============================================================================================

# Searching for time range

# 1) FROM date1\_asc TO date3\_asc ------>> search in the first file where date>date1\_asc

# 2) FROM date3\_asc TO date4\_asc ------>> search in all files inside this rang

# 3) FROM date4\_asc TO date6\_asc ------>> search in the last file where date<date6\_asc

# 4) after start of today if required

# ----------------------------------------------------

# Usage : filter\_his\_date\_range [search\_string]

# OutFile: $SCCTMP/"rtu\_test\_$\*"

#===============================================================================================

if [[ -z ${date1\_asc} && -z ${date6\_asc} ]]

then

date1\_asc=$2

date6\_asc=$3

fi

${SCC\_BIN}/var\_mktime `echo ${date1\_asc} |cut -d":" -f1` '%Y/%m/%d'|read date3

${SCC\_BIN}/var\_mktime `echo ${date6\_asc} |cut -d":" -f1` '%Y/%m/%d'|read date5

# part ( 1 )

# -----------

${SCC\_BIN}/var\_asctime $date3 %d\_%m\_%Y|read frst\_filename

if [[ -z "$1" ]]

then

zcat /aedc/data/nfs/historical/${frst\_filename}.his.Z > $SCCTMP/first\_file

else

zcat /aedc/data/nfs/historical/${frst\_filename}.his.Z | grep "$\*" > $SCCTMP/first\_file

fi

sep\_date $date1\_asc

if [ ${h1} != 0 ]

then

H1=`expr $h1 - 1`

grep -v " [0-${H1}][0-9]:[0-5][0-9]:[0-5][0-9]" $SCCTMP/first\_file > $SCCTMP/first\_file1

mv $SCCTMP/first\_file1 $SCCTMP/first\_file

fi

if [ ${h2} != 0 ]

then

H2=`expr $h2 - 1`

grep -v " ${h1}[0-${H2}]:[0-5][0-9]:[0-5][0-9]" $SCCTMP/first\_file > $SCCTMP/first\_file1

mv $SCCTMP/first\_file1 $SCCTMP/first\_file

fi

if [ ${m1} != 0 ]

then

M1=`expr $m1 - 1`

grep -v " ${h1}${h2}:[0-${M1}][0-9]:[0-5][0-9]" $SCCTMP/first\_file > $SCCTMP/first\_file1

mv $SCCTMP/first\_file1 $SCCTMP/first\_file

fi

if [ ${m2} != 0 ]

then

M2=`expr $m2 - 1`

grep -v " ${h1}${h2}:${m1}[0-${M2}]:[0-5][0-9]" $SCCTMP/first\_file > $SCCTMP/first\_file1

mv $SCCTMP/first\_file1 $SCCTMP/first\_file

fi

if [ ${s1} != 0 ]

then

S1=`expr $s1 - 1`

grep -v " ${h1}${h2}:${m1}${m2}:[0-${S1}][0-9]" $SCCTMP/first\_file > $SCCTMP/first\_file1

mv $SCCTMP/first\_file1 $SCCTMP/first\_file

fi

if [ ${s2} != 0 ]

then

S2=`expr $s2 - 1`

grep -v " ${h1}${h2}:${m1}${m2}:${s1}[0-${S2}]" $SCCTMP/first\_file > $SCCTMP/first\_file1

mv $SCCTMP/first\_file1 $SCCTMP/first\_file

fi

print -n "" > $SCCTMP/"rtu\_test\_$\*"

if [ $date5 -gt $date3 ]

then

cat $SCCTMP/first\_file > $SCCTMP/"rtu\_test\_$\*"

# part ( 2 )

# -----------

date=$date3

while [ "$date" -lt "$date5" ]

do

${SCC\_BIN}/var\_asctime $date %d\_%m\_%Y|read filename

if [[ -z "$1" ]]

then

zcat /aedc/data/nfs/historical/$filename.his.Z >> $SCCTMP/"rtu\_test\_$\*"

else

zcat /aedc/data/nfs/historical/$filename.his.Z | grep "$\*" >> $SCCTMP/"rtu\_test\_$\*"

fi

date=`expr $date + $daysec`

done

# part ( 3 )

# -----------

${SCC\_BIN}/var\_asctime $date5 %d\_%m\_%Y|read last\_filename

if [[ -z "$1" ]]

then

zcat /aedc/data/nfs/historical/${last\_filename}.his.Z > $SCCTMP/last\_file

else

zcat /aedc/data/nfs/historical/${last\_filename}.his.Z | grep "$\*" > $SCCTMP/last\_file

fi

else

cat $SCCTMP/first\_file > $SCCTMP/last\_file

fi

sep\_date $date6\_asc

if [ ${h1} != 2 ]

then

H1=`expr $h1 + 1`

grep -v " [${H1}-2][0-9]:[0-5][0-9]:[0-5][0-9]" $SCCTMP/last\_file > $SCCTMP/last\_file1

mv $SCCTMP/last\_file1 $SCCTMP/last\_file

fi

if [ ${h2} != 9 ]

then

H2=`expr $h2 + 1`

grep -v " ${h1}[${H2}-9]:[0-5][0-9]:[0-5][0-9]" $SCCTMP/last\_file > $SCCTMP/last\_file1

mv $SCCTMP/last\_file1 $SCCTMP/last\_file

fi

if [ ${m1} != 5 ]

then

M1=`expr $m1 + 1`

grep -v " ${h1}${h2}:[${M1}-5][0-9]:[0-5][0-9]" $SCCTMP/last\_file > $SCCTMP/last\_file1

mv $SCCTMP/last\_file1 $SCCTMP/last\_file

fi

if [ ${m2} != 9 ]

then

M2=`expr $m2 + 1`

grep -v " ${h1}${h2}:${m1}[${M2}-9]:[0-5][0-9]" $SCCTMP/last\_file > $SCCTMP/last\_file1

mv $SCCTMP/last\_file1 $SCCTMP/last\_file

fi

if [ ${s1} != 5 ]

then

S1=`expr $s1 + 1`

grep -v " ${h1}${h2}:${m1}${m2}:[${S1}-5][0-9]" $SCCTMP/last\_file > $SCCTMP/last\_file1

mv $SCCTMP/last\_file1 $SCCTMP/last\_file

fi

if [ ${s2} != 9 ]

then

S2=`expr $s2 + 1`

grep -v " ${h1}${h2}:${m1}${m2}:${s1}[${S2}-9]" $SCCTMP/last\_file > $SCCTMP/last\_file1

mv $SCCTMP/last\_file1 $SCCTMP/last\_file

fi

cat $SCCTMP/last\_file >> $SCCTMP/"rtu\_test\_$\*"

# part ( 4 )

# -----------

${CAE\_BIN}/mktime `date +"%C%y/%m/%d:00:00:00"`|read tody\_strt\_sec

${CAE\_BIN}/mktime ${date6\_asc} |read date6\_sec

${CAE\_BIN}/mktime ${date1\_asc} |read date1\_sec

end\_tm=0

strt\_tm=0

if [[ "${date6\_sec}" -gt "${tody\_strt\_sec}" ]]

then

end\_tm=${date6\_sec}

if [[ "${date1\_sec}" -gt "${tody\_strt\_sec}" ]]

then

strt\_tm=${date1\_sec}

else

strt\_tm=${tody\_strt\_sec}

fi

echo "select C0439\_alm\_text from T0439\_almhc

where C0438\_date between $strt\_tm and $end\_tm

go"|isql -U dbu -P dbudbu |grep -v "C0439\_alm\_text"|grep -v "affected)"|grep -v [-][-][-]| grep "$\*" >> $SCCTMP/"rtu\_test\_$\*"

fi

rm -f $SCCTMP/last\_file

rm -f $SCCTMP/first\_file

}

function disp\_count {

read fnt?"count ( C ) or display ( D ) >> "

case $fnt in

D|d ) fnct=disp

;;

C|c ) fnct=count

;;

\*) exit

esac

}

function get\_Alarm {

his=/aedc/data/nfs/historical

zcat /aedc/data/nfs/historical/$1|grep -e $2 -e $3 >/tmp/$2-$3

l=`wc -l /tmp/$2-$3|awk '{print $1}'`

j=`expr $l - 1`

tail -$j /tmp/$2-$3 |sed "s/$2//g" >/tmp/$2-$3.2

paste /tmp/$2-$3 /tmp/$2-$3.2 |grep $2 |grep $3 |grep ADD >/tmp/$2-$3.final

echo "$3 Maintenance $1

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_">>$his/report/Maintence/$2-$3

#awk 'OFS="\t" {print $5,$1,$2,$6,$8,$9,$10,$11,$12,$13,$14,$15}' /tmp/$2-$3.final >>$his/report/Maintence/$2-$3

awk '{printf "%s %s %s %s %s %s %s %s %s %s %s" ,$5,$1,$2,$6,$9,$10,$11,$12,$13,$14,$15}' /tmp/$2-$3.final >>$his/report/Maintence/$2-$3

echo "

===============================================================================================

">>$his/report/Maintence/$2-$3

rm -f $2-$3.2 $2-$3

}

function read\_fac {

fac=58

case ${SS} in

E-SS1)

fac="90" ;;

W-OBORG|W-SS6|W-OMID)

fac="30" ;;

ALEX)

fac="1" ;;

esac

}

function MkChkDir {

# Usage : MkChkDir <dirname>

# check if the directory is existing , if not : create it

if [ ! -z "$1" ]

then

if [ ! -d $1 ]

then

mkdir $1

fi

else

echo "Usage : MkChkDir <dirname>"

fi

}