#!/bin/ksh

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

# TITLE :aedc\_display\_new\_seg

# PURPOSE :to report new rooms and cables in specified period

## GROUP :IS

# AUTHOR :Alaa Nagy ,RANIA SAAD HEIKAL ( SCC S/W group )

# DATE :20 MARCH 2001

# MODIFIED BY :Alaa Nagy ( SCC S/W group )

#PURPOSE OF MODIFICATION:It can repeat itself in order not to exit from the first time

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

. $SCCROOT/aedc\_SCC\_functions

check () {

find /tmp/ -name $1 |read output

if [[ -z ${output} ]]

then

echo " There isn't new $type in this period "

else

echo ${header}>/tmp/final\_$1

cat /tmp/$1 >> /tmp/final\_$1

cat /tmp/final\_$1|more

echo ""

asc\_print /tmp/final\_$1

fi

}

function room {

echo "

use iutddb

go

set nocount on

select rtrim(CSCC1\_seg\_name)+'('+rtrim(CSCC1\_seg\_description)+')'

+'|'+substring(CSCC1\_sld1,1,40)+'|'+str(CSCC1\_mod\_day,2,0)

+'|'+str(CSCC1\_mod\_month,2,0)+'|'+str(CSCC1\_mod\_year,4,0)

+'|'+substring(CSCC1\_dcc,1,7)

from TSCC1\_new\_seg

where (CSCC1\_mod\_year\*12+CSCC1\_mod\_month)>= ${y1}\*12+${m1}

and (CSCC1\_mod\_year\*12+CSCC1\_mod\_month)<= ${y2}\*12+${m2}

and CSCC1\_seg\_type = 'R'

and upper(CSCC1\_dcc) like upper('${center}%')

order by upper(CSCC1\_dcc),CSCC1\_mod\_year,CSCC1\_mod\_month,CSCC1\_mod\_day

go" | isql -Udbu -Pdbudbu|grep [0-9]|sed 's/()//'|

awk 'FS="|"{printf"%-25s%-25s %2s/%2s/%4s\t %7s\n",$1,$2,$3,$4,$5,$6}' >/tmp/new\_room

}

function cable {

echo "

use iutddb

go

set nocount on

go

select

str(CSCC1\_seg\_id,4,0)+'|'+substring(CSCC1\_seg\_name,1,12)+'|'+substring(CSCC1\_sld1,1,40)+'|'+str(CSCC1\_mod\_day,2,0)

+'|'+str(CSCC1\_mod\_month,2,0)+'|'+str(CSCC1\_mod\_year,4,0)+'|'+substring(CSCC1\_dcc,1,7)

from TSCC1\_new\_seg

where (CSCC1\_mod\_year\*12+CSCC1\_mod\_month)>= ${y1}\*12+${m1} and (CSCC1\_mod\_year\*12+CSCC1\_mod\_month)<= ${y2}\*12+${m2}

and CSCC1\_seg\_type = 'C' and CSCC1\_dcc like upper('${center}%')

order by upper(CSCC1\_dcc),CSCC1\_mod\_year,CSCC1\_mod\_month,CSCC1\_mod\_day

go" | isql -Udbu -Pdbudbu|grep [0-9]|awk 'FS="|"{printf"%-20s%-32s%2s/%2s/%4s\t%7s\n",$2,$3,$4,$5,$6,$7}' >/tmp/new\_cable

}

#/\*'|'+substring(CSCC1\_sld2,1,40)\*/

#while true

#do

clear

function main {

read first\_date?"echo enter first range mm/yyyy >> "

read second\_date?"echo enter second range mm/yyyy >> "

if [[ -z ${first\_date} ]]

then

first\_date="0/0"

second\_date="12/2037"

fi

echo $first\_date|awk 'FS="/"{printf"%d",$1}'|read m1

echo $first\_date|awk 'FS="/"{printf"%d",$2}'|read y1

echo $second\_date|awk 'FS="/"{printf"%d",$1}'|read m2

echo $second\_date|awk 'FS="/"{printf"%d",$2}'|read y2

function DCC\_TYPE {

read cc?" select area to report its segments ( EAST , MIDDLE , WEST , ALL ) >> "

if [[ -z $cc ]]

then

exit

fi

case $cc in

ALL|all|a)

cc=''

center=$cc

;;

EAST|E|e|MIDDLE|M|m|WEST|W|w)

center=$cc

;;

\*) exit

esac

echo""

read type?" enter the type do you want to display ( R , C , A ) >> "

case $type in

R|r|room|ROOM)

room

type='Room'

header="Room description

\t Network\t\t\tDate\t DCC\n==================== ============================ ============= ========\n"

check new\_room

;;

C|c|cable|CABLE)

# rm /tmp/new\_cable

cable

type='Cable'

echo""

header="Cable Name \t Network\t\t\tDate\t\tDCC\n============== ============================ =============== ================\n"

check new\_cable

;;

A|a|all)

echo""

echo "<<<<<<<<<< new \_Rooms >>>>>>>>>>>"

room

type='Room'

echo ""

header="Room description

\t Network\t\t\tDate\t DCC\n==================== ============================ ============== =========\n"

check new\_room

sleep 3

echo ""

echo "<<<<<<<<<< new \_cables >>>>>>>>>>>"

cable

type='Cable'

echo""

header="Cable Name \t Network\t\t\tDate\t\tDCC\n============= ============================= ================ ===============\n"

check new\_cable

;;

\*) exit

esac

}

}

main

DCC\_TYPE

echo ""

while true

do

echo ""

read cont?"Would you like to continue ? (y/Y) ( HIT <CR> to exit ) >>"

if [[ -z $cont ]]

then

exit

else

echo ""

read RANGE?"Would you like to enter new range [y/Y] ( HIT <CR> if not ) >>"

if [[ -z $RANGE ]]

then

echo ""

DCC\_TYPE

else

echo ""

main

fi

fi

#break

done