

Infotrac Programming Setup Steps:

Step1: source /home/master/func_set1.sh #allow you to use some common functions, such as get_segment

Step2: call get_segment function #gets segment from insert file. Removes datatrack.tmp file

Step3: call get_md5_sum.sh function #get file_id from scanner DB and store in datatrack.tmp

Step4: call isisdisk with seg_num parameter

Step5: update insert file:

- add segment to \$fn.

Control file settings:

- estmt_trac="y"
- track_by_rec="y"
- infotrac="m"

Step6: DFA - update account number so that

1. remove leading zeros
2. right justified
3. remove suffix

Statement setup example:

```
105) cid="nmcu"
    in_media="disk"
    echo -e "        Applications for $cid:\n"
    echo -e "        1 - Monthly Statement"
    echo -e "\n\n\n\n\n        Please enter appropriate number: \c"
    read choice
    clear
    case "$choice" in
        1) app="msl"
            start_time=`date "+Start %H:%M:%S %m%d%Y"`
            ${home_master}/master/nmcums1.sh $cid $app $job_sel
            chk_exit_status $? nmcums1.sh
            end_time=`date "+End %H:%M:%S %m%d%Y"`
            echo ${start_time}\|${end_time} >> ${home_keep}/keep/${cid}${app}.datatrack;;
```

```
#####
# Program      : nmcums1.sh
# Program date: 05/13/2010
# Programmer   : Peter Dang
# Description  : run NMCU monthly Statement
# Parameter    : 1. cid
#              : 2. app
#              : 3. job select
#####
```

```
if [ $# -lt 3 ]
then
    echo -e "Usage: $0 cid app job_sel"
    exit 1
fi

cid=$1
app=$2
job_sel=$3
```

```
. ${home_master}/master/func_set1.sh # STEP 1 source functions.
```

```
echo -e "\n\nPlease enter the original input file (pgp) with full path <enter>"
read infile
if [ ! -s $infile ]; then
    echo -e "$infile does not exist."
    exit 1
fi
```

```
get_segment $cid $app # STEP 2 get segment number from DT_JOB_ID
export bypass="1"
```

```
rm $prefix.datatrack.tmp # STEP 3
```

```
${home_master}/master/get_md5_value.sh $infile $cid $app # STEP 3 store md5 value for infotrac
chk_exit_status $? get_md5_value.sh
```

```
const_file1=`echo $infile|cut -d "." -f 1` # remove .zip.pgp
file_path=`dirname $infile`
proc_file="nmcu_member_?????.dat" # process file
```

```
/home/util/decrypt.sh ${infile} $const_file1.zip
if [ $? != 0 ]; then
    echo "Error decrypting $infile."
    exit 1
fi
```

```
if [ -e ${file_path}/${proc_file} ]
then
    rm ${file_path}/${proc_file}
```

```

fi

unzip -d${file_path}/ $const_file1.zip $proc_file

if [ $? != 0 ]; then
    echo "Error unzipping $const_file1.\n"
    exit 1
fi

in_file="${file_path}/${proc_file}"          # proc file with full path
chk_sum2 $in_file $job_sel
log_file $in_file

datafile="${in_file}.con" #set datafile
if [ -e $datafile ]; then
    rm -f $datafile
fi

${home_master}/master/symconcat.pl $in_file $datafile
chk_exit_status $? symconcat.pl

if [ $job_sel = "e" ];then
    rm -f ${d_dir}/nmcu/nmcums?8.afp #Update isisdisk to reference correct file name when copy to ISWK
fi

if [ $job_sel = "b" ];then
    chk_sum2 $proc_file $job_sel
    log_file $proc_file
    ${home_master}/master/isisdisk.sh "$cid$app" "s" ${datafile} $seg_num
    chk_sum2 $proc_file $job_sel
    log_file $proc_file
    ${home_master}/master/isisdisk.sh "$cid$app" "f" ${datafile} $seg_num
    echo "Generating check index file"
    ${home_master}/master/cid_chk_index.pl ${datafile} $cid $app
    ${home_master}/master/nmcu_ecl.pl $cid $app ${datafile}
    ${home_master}/master/nmcu_paperless.pl $cid $app ${datafile}
    chk_sum2 $proc_file $job_sel
    log_file $proc_file
    rm -f ${d_dir}/nmcu/nmcums?8.afp
    ${home_master}/master/isisdisk.sh "$cid$app" "e" ${datafile} $seg_num
elif [ $job_sel = "f" -o $job_sel = "t" ]; then
    echo "Generating check index file"
    ${home_master}/master/cid_chk_index.pl "${datafile}" $cid $app
    ${home_master}/master/nmcu_ecl.pl $cid $app ${datafile}
    ${home_master}/master/nmcu_paperless.pl $cid $app ${datafile}
    ${home_master}/master/isisdisk.sh "$cid$app" $job_sel ${datafile} $seg_num
else
    ${home_master}/master/isisdisk.sh "$cid$app" "$job_sel" ${datafile} $seg_num
fi
end_time=`date "+End %H:%M:%S %m%d%Y"`
rm $in_file

```

Control file

```

infotrac="m"
cycleno=s${1}`grep PROC_DATE: ${home_ins}/insert/${prefix}.ins | cut -d " " -f2 | cut -c3-4`
fn="${dir$cid$cycleno}"

#ESTMT_PAPERLESS
supprefix="${cid$cycleno}"

export dir tifffdir fn supprefix

```

(DSI Team can bypass insert file step).

Insert File:

Add DT_JOB_ID i.e: {Job#}-C{cycle}.{segnum}-V{version#} DT_JOB_ID: 99494-C00.01-V20286

Daily Letters setup:

STEP1: \$cid_process.sh

- 1) run argument checking
`. ${home_master}/master/proc_arg_verify.sh $1 $2` Note: \$1=segment \$2= 0 or bypass
this program ensure no interactive job is running and return variable segment and bypass value.
- 2) export global variables
`export bypass=$g_bypass #allow program to update DT_JOB_ID`
`export g_segment=$g_segment`
- 3) delete *.datatrack.tmp file
- 4) call get_md5_value.sh #to get md5 key

STEP2: \$cid_process.pl

- 1) add segment value to combined .txt file
`system ("cat ${dir}${cid}dl?.txt > ${dir}${cid}d$ENV{g_segment}$a_time.txt");`
- 2) call isisdisk_daily.sh s option via passing segment value as the last argument
`system ("${ENV{home_master}}/master/isisdisk_daily.sh ${cid}dla s $ENV{d_dir}/daily/${cid}/${cid}d$ENV{g_segment}$a_time.txt $ENV{g_segment}");`
- 3) copy .clbmco file to .txt and call isisdisk_daily w/ f option
`system ("cp $ENV{dir}${cid}d$ENV{g_segment}$a_time.clbmco $ENV{dir}${cid}d$ENV{g_segment}$a_time.txt");`
#add segment to data file
`system ("${ENV{home_master}}/master/isisdisk_daily.sh ${cid}dla f $ENV{d_dir}/daily/${cid}/${cid}d$ENV{g_segment}$a_time.txt $ENV{g_segment}");`

STEP3: DataTrac (DSI team can ignore this step)

- update the DataTRAC w/ Version ID: <http://msgps>
Note: call /home/util/get_appno_from_datatrack.pl to get version number.

STEP4: control file

- 1) add infotrak flag
`infotrak="m"`
- 2) add segment value to fn
`if [$1 -a $1 != "format"]; then`
`cycleno=d${2}`grep PROC_DATE: ${home_ins}/insert/${cid}dla.ins | cut -d " " -f2 | cut -c3-4``
`fn="${dir}${cid}$cycleno"`
`prefix="${cid}$cycleno"`
`sam_jid="dl"`
`fi`

STEP 5: insert file (DSI can bypass this step)

- add DT_JOB_ID to the insert file
DT_JOB_ID: 123456-C24.01-V20785

STEP 1 DAILY NOTICE SETUP

Example of pacu_process.sh

```
main ()
{
    if [ $# -lt 1 ]
    then
        echo "Usage: $0 segment [0 | bypass]"
        exit
    fi

    cd ~
    cid="pacu"
    jid="dla"

    export cid

    . /home/control/daily.control          # define codebase, afptarget, barrsource, barrtarget

    inDir="${d_dir}/ftpbu/pacubu"          # production env

    targetBase="${d_dir}/daily"
    targetDir="$targetBase/$cid"

    jobid=`grep ^JOB_CYC: ${home_ins}/insert/${cid}${jid}.ins | tr -s " " | cut -d " " -f2`
    . ${home_env}/${keep_dir}/maillist/${cid}_mail.lis

    if [ $jobid != "888888" -a $jobid != "999999" ]; then
        daily_log="daily_process.log"
    else
        daily_log="daily_process_test.log"
    fi

    body="PACU"
    extension=".pgp"

    divider="*****"

    ${home_master}/master/proc_arg_verify.sh $1 $2

    export bypass=$g_bypass #allow program to update DT_JOB_ID
    export g_segment=$g_segment

    rm ${home_env}/${keep_dir}/${cid}${jid}.datatrack.tmp

    echo "" >> ${targetBase}/${daily_log}
    echo "$divider" >> ${targetBase}/${daily_log}
    echo `date` >> ${targetBase}/${daily_log}
    echo "Removing old files from ${targetDir}" >> ${targetBase}/${daily_log}
    echo "" >> ${targetBase}/${daily_log}
    echo "Process started....."
    echo "Removing old files from ${targetDir}"

    if [ $test -gt 0 ]; then
        rm -rf ${targetDir}/*          # remove old files first
        mkdir ${targetDir}/enotices

        filename=${inDir}/${body}_${file_cycle}_DL${extension}
```

```

${home_master}/master/get_md5_value.sh $filename ${cid} ${jid}

outfile=${targetDir}/${body}_${file_cycle}_DL
${home_var}/util/decrypt.sh "$filename" "$outfile" 2>> ${targetBase}/${daily_log}

if [ $? -ne 0 ]; then
    message=`tail -2 ${targetBase}/${daily_log} | tr "\n" " " | tr " " "~`

    subject="PACU - Decryption Error Alert for ${outfile}.gpg"

    perl ${codebase}/daily_mail2.pl 1 "$subject" "$message" "$maillist"

    echo "decrypt $filename .... failed"
    echo "decrypt $filename .... failed" >> ${targetDir}/today.log
    echo "decrypt $filename .... failed" >> ${targetBase}/${daily_log}
    exit 1
fi
else
    echo "No $anyfile from PACU, please check!"
    echo "No $anyfile from PACU, please check!" >> ${targetDir}/today.log
    echo "No $anyfile from PACU, please check!" >> ${targetBase}/${daily_log}

    subject="PACU $jid - No Data File Alert"
    perl ${codebase}/daily_mail2.pl 1 "$subject" "Alert" "$maillist"
    exit 1
fi

```

STEP#2 DAILY NOTICE SETUP

Example of pacu_process.pl

```

#!/usr/bin/perl -w
# Program Id      : pacu_process.pl
#
# Programmer      : Jimmy Li
# Created on      : 10/07/2008
#
# Purpose : Each data file is being process by format_only.sh to create
#          pacudla.txt. Then the combined file is ran with isisdisk_daily.sh
#          to create the print image.

use File::Copy;
umask 0000;
$home_master=$ENV{"home_master"};
$daily_log="$ENV{d_dir}/daily/daily_process.log"; # production
#$daily_log="$ENV{d_dir}/daily/daily_proc_test.log"; # for testing purpose
$cid='pacu';
$dir="$ENV{d_dir}/daily/${cid}/";

@type_arr=("a","b","c","d","e","f","g","h","i","j","k","l","m","n","o","p","q","r","s","t","u","v","w","x","y","z","0","1","2","3","4","5","6"); # add letter type to type_arr and input file to letter

open (LOGFILE, ">>${daily_log}"), or die "Can't open daily_process.log file $!\n";

```

```

$logstart_time=localtime($^T);
$start_mess="$${cid} started on $logstart_time\n";
print LOGFILE $start_mess;

print LOGFILE "Formatting each letter ...\n";
print "Formatting each letter ...\n";
for ($i=0;$i<=$#type_arr;$i++) {
    $infile=`ls ${dir}${cid}dl${type_arr[$i]}.dat`;
    chop $infile;

    if (-s ${infile}) {
        &format_step($type_arr[$i]);          # Formatting each letter
    }
}                                              # Create ${cid}dl?.txt

($cd_month, $a_time)=&cycle_num("${cid}dla"); # define which insert file to use

@date_tmp=localtime(time);

print LOGFILE "Processing print DFA ...\n";    # Combine all letters
print "Processing print DFA ...\n";           # and process print DFA

system ("cat ${dir}${cid}dl?.txt > ${dir}${cid}d$ENV{g_segment}$a_time.txt");
system ("${home_master}/master/isisdisk_daily.sh ${cid}dla s $ENV{d_dir}/daily/${cid}/${cid}d$a_time.txt
$ENV{g_segment}");

#####
print "Processing eNotice DFA ...\n";          # and process eNotice DFA
#####

system ("cp ${dir}${cid}d$ENV{g_segment}$a_time.clbmco ${dir}${cid}d$ENV{g_segment}$a_time.txt");
system ("${home_master}/master/isisdisk_daily.sh ${cid}dla f $ENV{d_dir}/daily/${cid}/${cid}d$a_time.txt
$ENV{g_segment}");

close ("LOGFILE");
exit 0;

```

STEP #3

Example of control file:

```

if [ $1 -a $1 != "format" ]; then
    if [ $2 ]; then
        cycleno=d${2}`grep PROC_DATE: ${home_ins}/insert/${cid}dla.ins | cut -d " " -f2 | cut -c3-4`
    else
        cycleno=d`grep PROC_DATE: ${home_ins}/insert/${cid}dla.ins | cut -d " " -f2 | cut -c3-4`
    fi

    fn="$dir${cid}$cycleno"
    prefix="${cid}$cycleno"
    sam_jid="dl"
fi

```

To test: Setup cron job user "opertest".**Note:** See `nmcudl_process.sh` and `nmcums1.sh` for examples.

***** Check List/Test ***:**

1. `${fn}` contain segment
2. eStmt test - Work with ISD to confirm job status appear correctly
3. Ensure online proofing works properly if applicable.
4. Ensure DVD/PDF archival work properly if applicable.