BounceBack

Bounce Back

If email got bounced send paper statement.

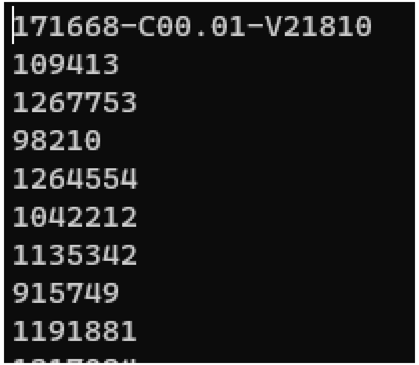
So for any cycle processed, if some email got bounced you may need to process that cycle again for those accounts to send paper statement.

The bounce back paper sample can be the original sample or different sample

Bounce Back

2 things needs to available:

1. The account list of customers for those, emails got bounced and some info by which you can pick the cycle to process (comes from isd, undeliverable file)



```
171668-C00.01-V21810
109413
1267753
98210
1264554
1042212
1135342
915749
1191881
```

apcu_ms2_undeliverable_1_02102022_100050.txt

1. The cycle itself (need to save by original script)

Bounce Back

Possible 2 processing scripts:

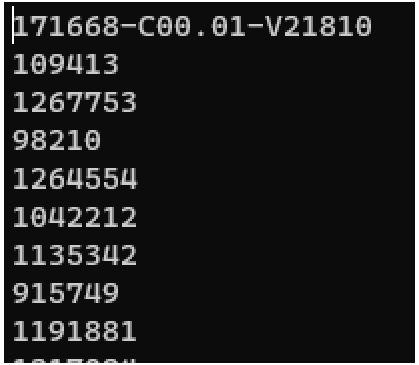
1. The regular one that will save the cycle in some way to be processed again for bounceback. Example: apcums2_process.sh

1. Bounce back script that will take the account list from undeliverable files comes from isd and process the original cycle for those accounts
   Example: apcuud1_process.sh (different prefix from original)

Bounce Back

Input: Undeliverable file comes from isd in ${d_dir}/ftpbu/${cid}bu/bounceback

Naming convention: apcu_ms2_undeliverable_169057-C00.01-V21810_1_11232021_134345.txt.pgp (***confirmed this with isd***)



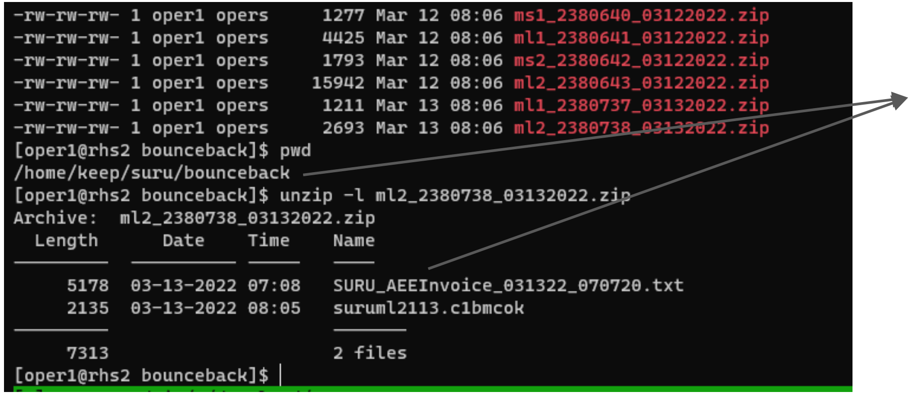
```
171668-C00.01-V21810
109413
1267753
98210
1264554
1042212
1135342
915749
1191881
```

Undeliverable file from isd contains **dt job id** and account no’s for which email got bounced

Bounce Back

From original processing script zip the c1bmcok file and original cycle to $home_keep/<cid>/bounceback folder.

zip example: ml2_2380738_03132022.zip



```
-rw-rw-rw- 1 oper1 opers     1277 Mar 12 08:06 ms1_2380640_03122022.zip
-rw-rw-rw- 1 oper1 opers     4425 Mar 12 08:06 ml1_2380641_03122022.zip
-rw-rw-rw- 1 oper1 opers     1793 Mar 12 08:06 ms2_2380642_03122022.zip
-rw-rw-rw- 1 oper1 opers    15942 Mar 12 08:06 ml2_2380643_03122022.zip
-rw-rw-rw- 1 oper1 opers     1211 Mar 13 08:06 ml1_2380737_03132022.zip
-rw-rw-rw- 1 oper1 opers     2693 Mar 13 08:06 ml2_2380738_03132022.zip
[oper1@rhs2 bounceback]$ pwd
/home/keep/suru/bounceback
[oper1@rhs2 bounceback]$ unzip -l ml2_2380738_03132022.zip
Archive:  ml2_2380738_03132022.zip
  Length      Date    Time    Name
---------  ---------- -----   ----
     5178  03-13-2022 07:08   SURU_AEEInvoice_031322_070720.txt
     2135  03-13-2022 08:05   suruml2113.c1bmcok
---------                     -------
     7313                     2 files
[oper1@rhs2 bounceback]$ 
```

For suru, stored the c1bmcok fiel and original cycle in /home/keep/suru/bounceback folder

note: better to keep name with jid ,
***ml2**_2380738_03132022.zip*

Bounce Back

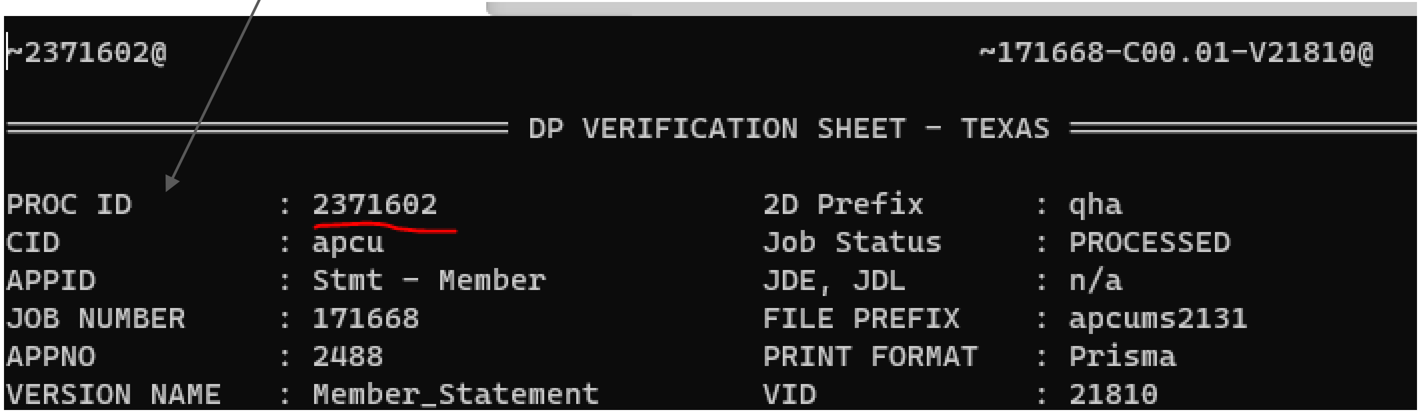
Zip file from original script

ml2_2380738_03132022.zip

Proc id from dpvs

Cycle date

Proc id is important here. It will ultimately help to find this file later



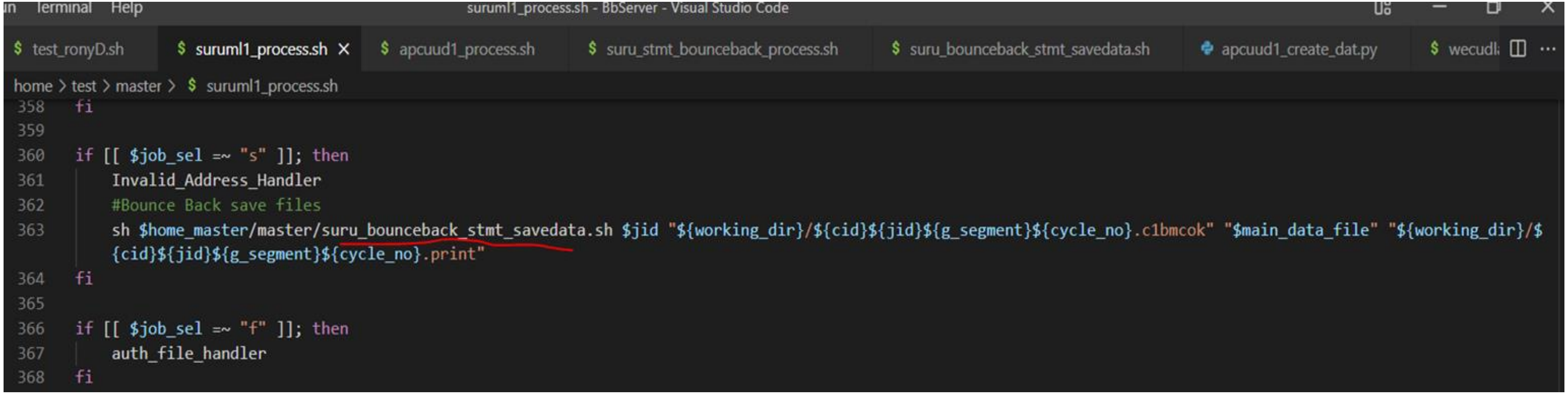
```
~2371602@                                              ~171668-C00.01-V21810@

═══════════════════════════ DP VERIFICATION SHEET - TEXAS ═══════════════════

PROC ID          : 2371602                 2D Prefix       : qha
CID              : apcu                    Job Status      : PROCESSED
APPID            : Stmt - Member           JDE, JDL        : n/a
JOB NUMBER       : 171668                  FILE PREFIX     : apcums2131
APPNO            : 2488                    PRINT FORMAT    : Prisma
VERSION NAME     : Member_Statement        VID             : 21810
```

Bounce Back

Sample code for saving cycle and c1bmcok for suru

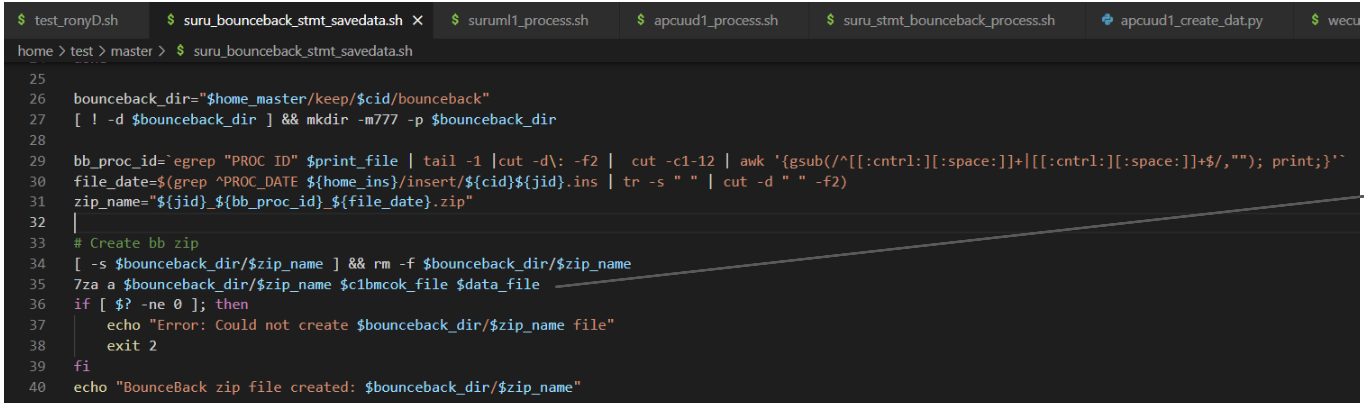


un Terminal Help
suruml1_process.sh - BbServer - Visual Studio Code
test_ronyD.sh
suruml1_process.sh
apcuud1_process.sh
suru_stmt_bounceback_process.sh
suru_bounceback_stmt_savedata.sh
apcuud1_create_dat.py
wecudl
home > test > master > suruml1_process.sh

```
fi

if [[ $job_sel =~ "s" ]]; then
    Invalid_Address_Handler
    #Bounce Back save files
    sh $home_master/master/suru_bounceback_stmt_savedata.sh $jid "${working_dir}/${cid}${jid}${g_segment}${cycle_no}.c1bmcok" "$main_data_file" "${working_dir}/${cid}${jid}${g_segment}${cycle_no}.print"
fi

if [[ $job_sel =~ "f" ]]; then
    auth_file_handler
fi
```



test_ronyD.sh
suru_bounceback_stmt_savedata.sh
suruml1_process.sh
apcuud1_process.sh
suru_stmt_bounceback_process.sh
apcuud1_create_dat.py
wecu
home > test > master > suru_bounceback_stmt_savedata.sh

```

bounceback_dir="$home_master/keep/$cid/bounceback"
[ ! -d $bounceback_dir ] && mkdir -m777 -p $bounceback_dir

bb_proc_id=`egrep "PROC ID" $print_file | tail -1 |cut -d\: -f2 |  cut -c1-12 | awk '{gsub(/^[[:cntrl:][:space:]]+|[[:cntrl:][:space:]]+$/,""); print;}'`
file_date=$(grep ^PROC_DATE ${home_ins}/insert/${cid}${jid}.ins | tr -s " " | cut -d " " -f2)
zip_name="${jid}_${bb_proc_id}_${file_date}.zip"

# Create bb zip
[ -s $bounceback_dir/$zip_name ] && rm -f $bounceback_dir/$zip_name
7za a $bounceback_dir/$zip_name $c1bmcok_file $data_file
if [ $? -ne 0 ]; then
    echo "Error: Could not create $bounceback_dir/$zip_name file"
    exit 2
fi
echo "BounceBack zip file created: $bounceback_dir/$zip_name"
```

Zipping c1bmcok and original cycle

Bounce Back

For bounceback script use different prefix to process for bounceback. Say

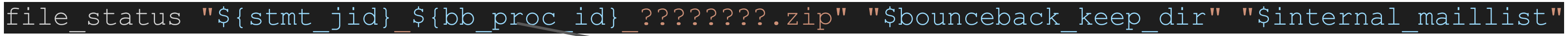
***apcums2**_process.sh(original) -> **apcuud1**_process.sh(bounceback)*

In the bounceback script:

Get proc id from that dt job id of undeliverable file(that contains **dt job id** and account list) comes from isd

bb_proc_id=`perl ${home_master}/master/**pull_procid_cycle.pl $dt_job_id** | cut -d\| -f1`

Find the zip file that you stored with proc id from original processing script.



```
file_status "${stmt_jid}_${bb_proc_id}_????????.zip" "$bounceback_keep_dir" "$internal_maillist"
```

It's that zip file in which stored the c1bmcok file and the cycle data

Bounce Back

At this point you have:

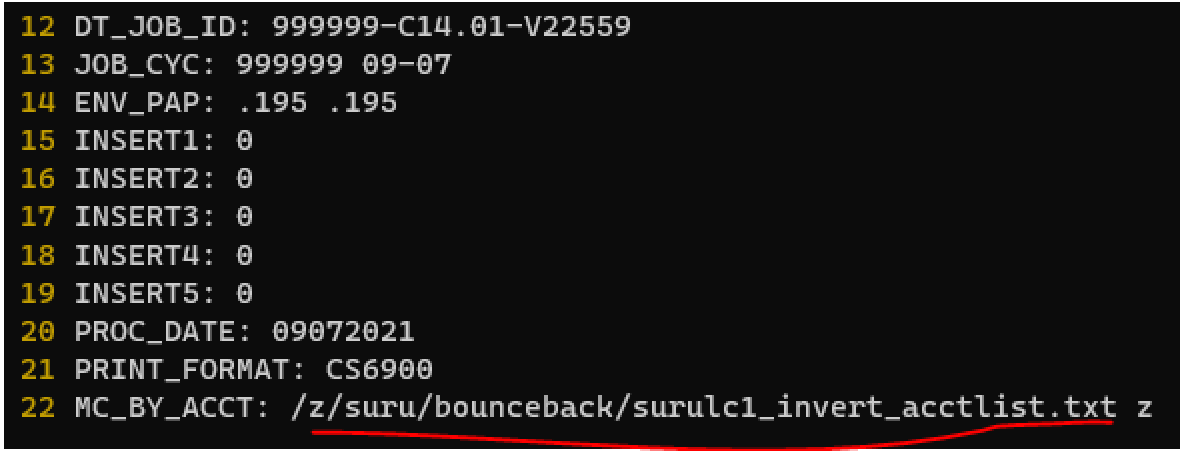
- Account list (for those email got bounced) from undeliverable file(let’s call those undeliverable accounts) from isd for which we need to process

- c1bmcok file and the original cycle which includes all accounts

Now if you need to process the cycle data to produce paper output for those undeliverable accounts so you have to suppress other accounts from the cycle data from print

Bounce Back

Build MC_BY_ACCT file for insert file from undeliverable account list and c1bmcok account list.



```
DT_JOB_ID: 999999-C14.01-V22559
JOB_CYC: 999999 09-07
ENV_PAP: .195 .195
INSERT1: 0
INSERT2: 0
INSERT3: 0
INSERT4: 0
INSERT5: 0
PROC_DATE: 09072021
PRINT_FORMAT: CS6900
MC_BY_ACCT: /z/suru/bounceback/surulc1_invert_acctlist.txt z
```

- *This contains accounts those are in c1bmcok (and hence cycle data file) but not in undeliverable file from isd.*
- *Now if you process the cycle data for paper all but undeliverable accounts will be suppressed from print. Hence only undeliverable accounts will get the print.*

Bounce Back

Overall flow:

- Zip data file and c1bmcok file (let’s call it bounceback file) in original processing script with process id

```
${jid}_bb_${bb_proc_id}_??????.zip
```

- Parse undeliverable files from isd in bounceback script to get dt job id and undeliverable account list

```
${jid}_undeliverable_<dt_job_id>_*.txt.pgp
```

- Get the process id by dt job id

```
bb_proc_id=`perl
${home_master}/master/pull_procid_cycle.pl
$dt_job_id | cut -d\| -f1`
```

- Get the correct bounceback file by process id

```
file_status "${jid}_bb_${bb_proc_id}_??????.zip"
"$bb_keep_dir" "$internal_maillist"
```

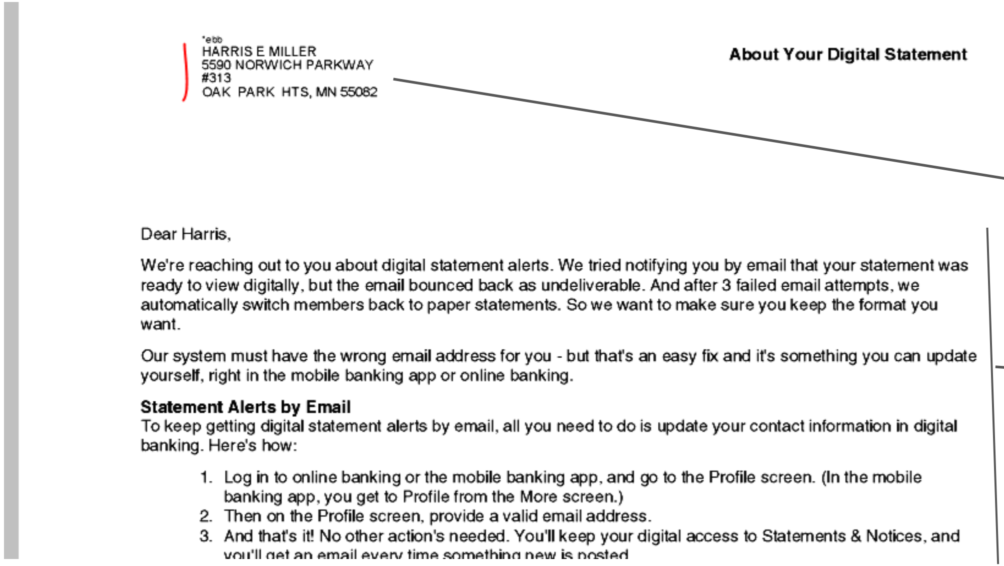
- Build mc_by_acct file to suppress accounts other than undeliverable accounts

```
MC_BY_ACCT:
/z/suru/bounceback/surulc1_invert_acctlist.txt z
```

- Process the cycle

Bounce Back

For apcu we needed to send a notice to undeliverable accounts (not the original statement) which consists of client address and hard coded text



*ebb
HARRIS E MILLER
5590 NORWICH PARKWAY
#313
OAK PARK HTS, MN 55082

About Your Digital Statement

Dear Harris,

We're reaching out to you about digital statement alerts. We tried notifying you by email that your statement was ready to view digitally, but the email bounced back as undeliverable. And after 3 failed email attempts, we automatically switch members back to paper statements. So we want to make sure you keep the format you want.

Our system must have the wrong email address for you - but that's an easy fix and it's something you can update yourself, right in the mobile banking app or online banking.

Statement Alerts by Email
To keep getting digital statement alerts by email, all you need to do is update your contact information in digital banking. Here's how:

1. Log in to online banking or the mobile banking app, and go to the Profile screen. (In the mobile banking app, you get to Profile from the More screen.)
2. Then on the Profile screen, provide a valid email address.
3. And that's it! No other action's needed. You'll keep your digital access to Statements & Notices, and you'll get an email every time something new is posted

Address from c1bmcok file

Hard coded text

Hence we didn't need to store original cycle data or mc_by_acct file, just storing c1bmcok file was enough
And we extracted the address from c1bmcok for those undeliverable account

Bounceback references:

1. apcuud1_process.sh   -> for all apcu app

1. suru_stmt_bounceback_process.sh -> for surums1-ms7, ml1-ml2
   bounceback jid - mc1-mc7,lc1-lc2

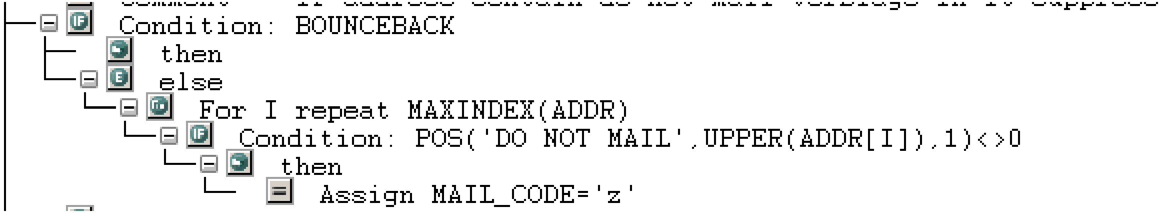
1. spcujba_process.sh -> for spcudna

1. Spcujbb_process.sh -> for spcudnb

Bounce Back

Careful with paper suppression logic in dfa for bounceback

- Your target is to send paper sample for those email got bounced. Paper suppression should be disable here. (confirm from client)
- You might want to use the original dfa if you need to send original paper statement
- You might need to disable the original paper suppression logic from the original dfa (confirm this from client)



```
    Comment    if address contain do not mail verbiage in it suppress
├─⊟ Condition: BOUNCEBACK
│ ├─    then
│ └─⊟ else
│   └─⊟ For I repeat MAXINDEX(ADDR)
│     └─⊟ Condition: POS('DO NOT MAIL',UPPER(ADDR[I]),1)<>0
│       └─⊟ then
│         └─    Assign MAIL_CODE='z'
```

*Check the **prefix** to determine if this dfa is running for bounceback*

Bounce Back

How do you get data for bounceback script while testing?

- Ask test undeliverable files from isd.

  Or,

- Process the original script build the zip file with process id containing the c1bmcok and cycle.
- Build a undeliverable file on your won with naming convention confirmed with isd
- Include the dt job id and some accounts from the cycle in the undeliverable file

Bounce Back

Any question?

- We are we using process id to name the zip ? Can’t we use dt job id instead since undeliverable file from isd contain it?

${jiid}_bb_${bb_proc_id}_??????.zip