Model

1) **FacilityACH**

- Contains Details about account information, recurrence\_amount, recurrence\_type(monthly or threshold) etc.

***Fields***

* account = Foreign key to Account Model
* recurring\_payment\_details\_encrypted = Bank Account details in encrpyted format
* recurrence\_type = Whether Monthly or Threshold
* recurrence\_amount = Amount to recharge
* threshold\_account\_value = Account value for which account is eligible for recharge. Only Applicable for recurrence type THRESHOLD
* automatic\_recharge\_status = Active or Paused or Blocked. We block the account after 2 consecutive failures. Only Contractor Dealer can unblock it.
* cheque\_image\_url = s3 url of the cheque image uploaded

2) **ACHEntryLog**

- Contains Details about the all the recharge done through ACH

***Fields***

* account\_ach = Foreign key to FacilityACH model
* identification\_number = 15 digit unique identification number
* facility = Foreign key to Facility model
* response\_type = If the transaction was failure/warning we receive response type

1) RESPONSE\_TYPE\_RETURN = The transaction was failure

2) RESPONSE\_TYPE\_NOTIFICATION = The transaction was success but with some warning

* total\_amount\_payable = Total Amount Recharged. This is same as recurrence\_amount of FacilityACH model
* penalty\_fee = Penalty in case of Recharge failure
* status = Current Status of ACH transaction
  + - 1) NEW = ACHEntryLog is just created but amount is not credit yet
    - 2) CREDITED = We have added total\_amount\_payable to Facility Account balance.
    - 3) RIP\_FAILURE = RIP file was received containing above transaction. The transaction may either be failure or with warning.

3) **ACHFileLog**

- Contains Details about the File we create and send to citibank server

***Fields***

* file\_name = Name of the File
* ack\_file\_name = Name of the corresponding Acknowledgement file
* should\_retry = Flag that enables the File entries to be added again in new file
* parent\_file = If should\_retry is True for file F1 and F2 is the parent file it means that all the entries in F1 was added in F2 for retrial.
* status = Current Status of ACH file
  + 1) NEW = ACH file is just created
  + 2) SFTP\_SUCCESS = ACH file successfully sftp to citibank server
  + 3) SFTP\_FAILURE = ACH file unsuccessfully sftp to citibank server
  + 4) ACK\_SUCCESS = Acknowledgement file successfully received for the ACH file
  + 5) ACK\_FAILURE = Acknowledgement file unsuccessfully received for the ACH file

4) **ACHEntryFileRelationship**

- Contains information about which ACH entries were sent in which ACH file

***Fields***

* ach\_entry = Foreign key to ACHEntryLog
* ach\_file = Foreign key to ACHFileLog

CRON Info and Troubleshooting steps

1) **create\_threshold\_entry\_log\_data**

**FREQUENCY** = Every 1 Minute

**FUNCTIONALITY** = Recharge all accounts with their specified recharge amount for which available balance < threshold amount

**IMPACT of CRON FAILUR**E = No impact. Recharge will be done on next cron run

2) **create\_monthly\_entry\_log\_data**

**FREQUENCY** = Every 1st of Month at 12 PM EST

**FUNCTIONALITY** = Recharge all accounts with their specified recharge amount who opted for monthly recharge.

**IMPACT of CRON FAILURE** = If crons fails, we need to run this CRON again. There is no impact if this CRON run multiple times because we ensure if an account is recharged for the given month, he won't be recharged again for the same month, no matter how many times CRON is executed.

3) **send\_ach\_file**

**FREQUENCY** = Every day at 7 AM EST

**FUNCTIONALITY** = Create and Send an ACH file as per NACHA specification containing all the accounts info and their recharges which we need to send to CITIBANK to process. The file will contain all the entries which we have recharged since the last ACH file was sent.

**IMPACT of CRON failure** = If cron fails, we send a monitoring email. CRON can fail for 2 reasons

i) There is some problem in ACH file creation (Never happened so far)

- If this happens, then we need to fix our backend code so that files get created successfully. The file entries will automatically be picked on next CRON run

2) There is some problem in sending file to citibank via SFTP (This can happen sometime)

- No need to change code. Just send the file again

How to send file again

A) - If the file creation fails i.e File status is NEW, then the entries will be automatically picked when next cron is run. So no action needs to be taken.

Example

Day 1 - File created with 10 entries failed with status NEW

Day 2 - File created with 5 new entries and 10 previous entries failed with status NEW

Day 3 - File created with 5 new entries and 15 previous entries passed.

B) - If the file was created and SFTP was not successful i.e file status is SFTP\_FAILURE. We need to mark should\_retry flag True for the given file.

To see all the files which are either in SFTP\_FAILURE and ACK\_FAILURE, run '**Failed ACH files not posted to Citibank Report**' report.

To mark should\_retry to True use this API - {base\_url}/update\_ach\_file

Payload = {

'ach\_file': NAME\_OF\_THE\_FILE

}

To mark should\_retry to True using management command - python manage.py manually\_retry\_ach\_file --ach\_files FILE\_NAME\_1 FILE\_NAME\_2 FILE\_NAME3 FILE\_NAME\_N

Note : If we manually run the CRON again on the same day, it won’t create and send another ACH file to citibank since an ACH file is already created for today. If we don’t want to enable should\_retry flag and wait for 1 day, then we need to remove code that prevent ACH file to be created again on same day and then run the CRON manually.

4) **process\_ach\_acknowledgement\_file**

**FREQUENCY** = Every day at 1:00 PM EST

**FUNCTIONALITY** = Acknowledge that ACH file we send is successfully received and validated by citibank. If there's any acknowledgment file, we mark file status to ACK\_SUCCESS. If no acknowledgement file is received, we mark it to ACK\_FAILURE.

**IMPACT of CRON failure** = If cron fails, we send a monitoring email. In this case, We should run this CRON manually ASAP before the next ACH file is sent. In cases where we couldn't run it manually, we should directly update the status of file log to ACK\_SUCCESS after confirmation that it was actually processed by citibank.

5) **process\_ach\_response\_file**

**FREQUENCY** = Every day at 6:00 PM EST

**FUNCTIONALITY** = Deduct the amount and add penalty charge for all those account whose bank account were not able to be successfully debited.

**IMPACT of CRON failure** - If cron fails, we send a monitoring email. In this case, we need to run the CRON again within 1 day. RIP file only remains in the citibank server for 1 day. So we need to process it before they delete from their server. If we are not able to run CRON again before the file is deleted, we should request the RIP file again if there is any for that day. There is no impact on reading the same RIP file again. If the RIP file is processed, then it won't get processed again. We send monitoring email containing all the accounts which were received in RIP file.