Matching preprocessor logic is ONLY applied to CCRS source files (HCFA, UB04, DENT).

File\_name: CCRS]-[FileType]-[ClaimType]-[CCRS]-[yyyymmdd] .txt

General description.

If SSN is presented value, then claims must be loaded unless such record broke another validation requirement.

If SSN is missing value and ICN is missing, then claims would be rejected to load.

If SSN is missing value and ICN is presented, then preprocessor try to obtain SSN value by ICN match from tables DIM\_PATIENT (and MVI\_VETERAN).

* SSN value is found, claim must be loaded
* SSN value is not found, claim must be rejected

**Step 1. DIM\_PATIENT match (preprocessor).**

Pick up latest DIM\_PATIENT.member\_id with icn\_status = ‘MATCH.

‘MATCH’ status indicates only records which were passed mvi app validation or confirmed by update from CDW view.

* SSN value is found in DIM\_PATIENT: write value to file preparing as the input to DataStage.
* SSN value is not found in DIM\_PATIENT: go to step 3 (go to Step 2 is disable now)

**Step 2. MVI\_VETERAN match (preprocessor). Current state: disable**

Pick up latest MVI\_VETERAN.ssn by icn\_status\_modified\_date\_time

* SSN value is found in MVI\_VETERAN: write value to file preparing as the input for DataStage (CCRS-[ClaimType]-[concatenated.txt)
* SSN value is not found in MVI\_VETERAN: go to step 3

**Step 3. Data validation and rejected lines (preprocessor).**

If SSN value is still missing, then do not include such lines to input DataStage file.

All such lines with missing ssn includes:

1. in rejected file [CCRS]-[FileType]-[ClaimType]-[CCRS]-[yyyymmdd]-Rejects .txt

[\dropzone\in\CCRS\rejects](file:///\\vac20apppci301.va.gov\dropzone\in\CCRS\rejects)

1. etl.rejected\_lines table.

There are no missing ssn values on start ETL DataStage routine. All missing ssn lines are rejected or resolved.

**Step 4. Raw data population on CLAIM\_PATIENT (preprocessor).**

Ingest only not rejected claims.

Populate founded SSN and source table name DIM\_PATIENT (or MVI\_VETERAN) where match was found for initial missing ssn on source file.

**Step 5. Data stage ETL**

Insert new record to dim\_patient if not such record exists (ssn match to member\_id) including ICN if it is available. (the population of ICN need be changed to blank to avoid false reference to ssn in case mvi validator did not confirmed valid SSN)

All member\_id values are unique on dim\_patient.

Establish FK dim\_va\_claim.patient\_key ( ssn fields match with dim\_patient.member\_id .

**Step 6. Validation of new patient record (mvi validator app).**

GIT location: datamgr/src/main/java/pit/datamgr

datamgr/datamgr/doc/mvi/MVI\_Service\_Description.pdf

Update and validation every new record in DIM\_PATIENT on the fly.

The value of DIM\_PATIENT.ICN\_STUTUS is null before

Fields are presented in request to validate patent records and obtain addition data using mvi interface.

dim\_patient.date\_of\_birth

dim\_patient.last\_name

dim\_patient.member\_id

|  |  |  |
| --- | --- | --- |
| ICN\_STUTUS | Action |  |
| **MATCH** | If match found set status to **MATCH** and  update data on dim\_patient for particular record from MVI source including ICN |  |
| **NO\_MATCH** | If match not found set status to **NO\_MATCH**  update only the log related info in dim\_patient |  |

Long term solution options:

1. Make unique only ICN in DIM\_PATIENT table – no way since not all streams delivery ICN and it would be not one moment switch from SSN to ICN as major patient identifier.
2. Make unique pair ssn and ICN in DIM\_PATIENT table – hidden dupes are remaining after ETL. Any post update from mvi source make cause data update and establish dupes in DIM\_PATIENT table.
3. Get rid of dim\_patient as master ‘source of true’ for PIT.

Row patient data should be accumulated to claim\_patient table for all strems

Refactoring scoring and report query

Allow consume record without ssn and ICN value (claims initial reject by TPA)

Step 1 DIM\_PATIENT match query:

select member\_id, icn from

(select row\_number() over (partition by icn order by patient\_key desc) rn, member\_id, icn from dim\_patient

where icn in (?) and

iif (icn is null, '', icn ) != '' and member\_id is not null and ICN\_status = 'MATCH') upd

where upd.rn = 1

Step 2 MVI\_VETERAN query (step 2 is disable now):

select ssn, icn\_full

from (select row\_number() over (partition by icn\_full order by icn\_status\_modified\_date\_time desc) rn, ssn, icn\_full

from mvi\_veteran where icn\_full in (?))

Data validation requirement: CRITACAL and DUPLICATES (deduplication) are rejected to load.

The source ssn file field names are marked as yellow.

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  | **HCFA** |  |  |
|  | **CRITICAL Fields** | HCFAUniqueClaimID, Box25, Box1A, Box24AFrom , Box24B, Box24DCPT, Box24E |  |
|  | **WARNING Fields** | Box23, HCFAStatus, RenderingProvider, Rendering\_Facility\_NPI |  |
|  | **Duplicate IDs** | HCFAUniqueClaimID, HCFALines\_ID, DatabaseID |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  | **UB04** |  |  |
|  | **CRITICAL Fields** | UB92UniqueClaimID, FederalTaxID, PatientName, Box60A |  |
|  | **WARNING Fields** | UBL\_Box42, Status, ProviderID, PreAuth, TypeOfBill |  |
|  | **Duplicate IDs** | UBL\_ID, UB92UniqueClaimId, DatabaseID |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  | **DENT** |  |  |
|  | **CRITICAL Fields** | HCFAUniqueClaimID, Box25, Box1A, Box24AFrom, Box24B, Box24DCPT |  |
|  | **WARNING Fields** | Box23, HCFAStatus, RenderingProvider, Rendering\_Facility\_NPI |  |
|  | **Duplicate IDs** | HCFAUniqueClaimID, HCFALines\_ID, DatabaseID |  |
|  |  |  |  |
|  |  |  |  |

**Mapping config files to populate claim\_patient:**

HCFA\Dental:

job.name=ccrs\_hcfa\_claim\_patient\_load

target.table=claim\_patient

dedupe.by=HCFAUniqueClaimID,DatabaseID

map.etl\_batch\_id=

map.derived.claim\_id=

map.Box1A=insured\_id

map.InsuredIDA=insured\_id\_a

map.InsuredIDB=insured\_id\_b

map.InsuredIDC=insured\_id\_c

map.InsuredsNameA=insured\_name\_a

map.InsuredsNameB=insured\_name\_b

map.InsuredsNameC=insured\_name\_c

map.RelationshipToInsuredA=relationship\_to\_insured\_a

map.RelationshipToInsuredB=relationship\_to\_insured\_b

map.RelationshipToInsuredC=relationship\_to\_insured\_c

map.Box4=patient\_name

map.Box3Sex=sex

map.Box3DOB=date\_of\_birth

map.VistaPatKey=vista\_key

map.Box26=account\_number

map.ICN=ICN

map.ssn\_match\_source=ssn\_match\_source

UB04:

job.name=ccrs\_ub\_claim\_patient\_load

target.table=claim\_patient

dedupe.by=UB92UniqueClaimId,DatabaseID

map.etl\_batch\_id=

map.derived.claim\_id=

map.PatientSSN=patient\_id

map.PatientName=patient\_name

map.Sex=sex

map.DOB=date\_of\_birth

map.Box60A=insured\_id\_a

map.Box60B=insured\_id\_b

map.Box60C=insured\_id\_c

map.PatientControlNumber=account\_number

map.VistaPatKey=vista\_key

map.RelationshipToInsuredA=relationship\_to\_insured\_a

map.RelationshipToInsuredB=relationship\_to\_insured\_b

map.RelationshipToInsuredC=relationship\_to\_insured\_c

map.ICN=ICN

map.ssn\_match\_source=ssn\_match\_source

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |