EHB Transition Document

|  |  |  |
| --- | --- | --- |
| **Title:** EHB Transition Document | |  |
| **Description:**  Beginning January 1, 2015, Large Group non-grandfathered plans are mandated to combine/commingle cost shares for deductible, copay, and coinsurance for covered Essential Health Benefits among separate providers (Medical, Rx, Pediatric Dental, Pediatric Vision, and Behavioral Health) to a single out of pocket maximum. Exceptions are services that are deemed excepted benefits (voluntary for a member to opt in or out such as some vision and or dental plans.  Deductible, copay, and coinsurance cost shares for Large Group Non-Grandfathered plans covering Essential Health Benefits (EHBs) and Non-EHB covered services in-network and out-of-network emergency services must accumulate to the in network out of pocket following the product strategy.  The accumulator ODS will be used to store all accumulated dollars( Medical , Pharmacy, Vision, healthlink) for EHB products built on WPD and eWPD platforms .The ODS would be also used to track dollars for EHB products built on all of Anthem’s legacy medical claims platforms (ACES, HealthLink, CHIPS, CS90, NASCO and FACETS) | | |
| **Approved By:** | **Date Approved:** | |
| **Version History** | | |
| **Revised on:** | **Version #:**V0.1, Offshore | |
|  |  | |
|  |  | |
|  |  | |
|  |  | |
|  |  | |
|  |  | |
|  |  | |
|  |  | |

Table of Contents

[**Introduction** 3](#_Toc416352224)

[**Benefit feed load and Reports.** 6](#_Toc416352225)

[**EyeMed Synch Process** 34](#_Toc416352227)

[**ESI – Real time Transaction** 42](#_Toc416352228)

[**P2P Process – ISG to ISGHIX/ISGEHB Transfer** 54](#_Toc416352229)

[**EHB SCREENS** 63](#_Toc416352230)

[**Forward Sync and backward sync process** 86](#_Toc416352232)

[**Forward sync** 86](#_Toc416352233)

[Forward Sync Process 88](#_Toc416352234)

[Backward Sync Process 97](#_Toc416352235)

[Forward Sync Process 105](#_Toc416352236)

[Flow Diagram 105](#_Toc416352237)

[Backward Sync Process 106](#_Toc416352238)

[**Over Applied Report Generation.** 108](#_Toc416352239)

**Introduction**

This document lays down the changes done as part of EHB project. The changes are grouped under headings as follows

**Basic understanding of Accum ODS–**

Please find below the flow diagram for the EHB process that covers the Legacy and the Specialty systems that interact with Accum ODS. The membership information resides in the respective Legacy Systems and the data is passed as eligibility feed to ESI and Eyemed that process Pharmacy and Vision claims respectively. We also have a separate Database to store the Benefit Information for all the Legacy Systems. Also, all the HIX and PEX contract details that are available in the Spider tables would also be loaded to the new Benefit Tables by a batch process. This will enable both ESI and Eyemed accum sync process to refer the new Benefit Tables for both EHB as well as HIX/PEX contracts. All the Legacy Systems (except ACES and Healthlink) access the Accum ODS via batch process while ACES and Healthlink access in Realtime.



Some of the additional functionalities handled while updating accumulators in ODS are as follows which is based on the indicators for each contract in the new Benefit Database:

1. Cross Apply - This attribute of the contract does the cross application of PAR accums to NPAR and vice-versa.
2. Preferred Cross Apply - This attribute of the contract does the cross application of preferred accums to the corresponding PPO PAR/NPAR accums.
3. Dual Update - This attribute does the dual update from shared to non-shared and vice-versa.
4. Last Quarter - This attribute does the last quarter carryover of accums from the current year to the next year.
5. Embedded/Non-embedded – This attribute decides how to identify the maximum amount for each accumulator.

In addition, we also perform a duplicate check before updating the accums to the ODS DB. The details are covered separately under each section for Legacy and Specialty systems.

**Benefit Loading**

1. Receive Medical benefits from ACES, CHIPS, CS90, FACETS, HealthLink, NASCO and WGS-WPD, eWPD and Spider
2. Map the Medical Accum names to Generic Accum

**Eye Med process –**

1. Interaction with Eye Med for accumulator dollar transfer for VISION comingled members.
2. Reporting Accum met members to EyeMed

**ESI Real Time Process**

RX Claims accums will be sent by ESI in real time for all HIX/PEX/EHB members

**Forward and backward Sync**

Accumulator sync up for members moving from legacy Anthem system to WGS 2.0 on renewal.

**P2P process**

Systematic portability of accumulator dollars for members moving from non EHB to EHB, EHB to EHB

**Benefit feed load and Reports.**

**In Scope**

Scope is limited to processing of the EHB feed files received from different legacy systems and to invoke newly created IO routine to insert feed data in translation and history tables for benefit and Accum through batch processing and to perform DB IO operation like Update, Select & Insert on Benefit translation tables (benefit & Accum) and corresponding History tables (benefit & Accum). Listed below are the systems in scope.

* 1. WGS
  2. SPIDER
  3. HEALTHLINK
  4. CS90
  5. FACETS
  6. NASCO
  7. ACES
  8. CHIPS

Following is the process flow:

**Response file, Success report, Failure report, Request report**

**Processing of the feed files from legacy systems**

**Input feed file from different legacy systems**

**Insert/update**

**Translation Table**

* The input feed file will be received from different legacy systems thru FTP/NDM.
* The mandatory fields will be validated to see if it contains a valid value. If valid value is not present then it will be report to the response file with proper error message and error code.
* The optional fields will be validated to see if it contains a valid value. If valid value is not present then set the field to have default value.
* The feed file layout and the details of the fields, are specified in the below Excel sheet.



* After the validation, the IO routine is called to insert/update feed file data into the new benefit translation tables. The IO routine will be called only if the benefit record and all the corresponding Accum records are valid. If either the benefit or accum record has invalid value, both benefit and Accum records will not be inserted in the translation table.
* Response file will be created which will contain all the records present in the input feed file. In addition to this, the status code will be added with proper status message.
* Success report, failure report and requests report will be created based on the response messages in the response file and will be routed to dispatch.

**Out of Scope**

* SPIDER DB2 routines are not in scope of this project.
* Membership validation and Membership information population are not in the scope of this project.

**Component Details**

**GNCEHBFD (Copy book)**

* GNCEHBFD is the new copybook for EHB benefit and accum feed file layout.
* The GNCEHBFD record length is 500.



**GNC9846 (Input feed file validation)**

* Read the feed file.
* Validate the feed file.
* In trailer record, Benefit record count is compared with the total number of benefit records in the feed file.
* The Trailer accum count is compared with the total number accum records in the feed file.
* The previous request date and time is validated against the current request date and time to identify whether the file is valid or not.
* Check if the file is empty.
* If the feed file is empty, then the return code 5 is set. If the file is duplicate, or contains only header and trailer or there is count mismatches or then the return code is set to 8 and job will stop execution else it will write the data from benefit feed file to Output control file and will continue execution.

**GNC9843 (Driver Module)**

* Read the feed file in EHB layout.
* The input feed file will be received from different legacy systems.
* The mandatory fields will be validated to see if it contains a valid value. If valid value is not present then it will be report to the response file with proper status code and error message.
* The optional fields will be validated to see if it contains a valid value. If valid value is not present then set the field to have default value.
* After validation the IO routine is called to insert/update feed file data into the benefit translation tables. The IO routine will be called only if the benefit record and all the corresponding Accum records are valid. If one of the BENEFIT/ACCUM records contains invalid value, both benefit and Accum records will not be inserted/updated in the translation table.
* Response file will be created which will contain all the records present in the input feed file. In addition to this, the status code will be added with status message. The following response codes will be set.

|  |  |  |
| --- | --- | --- |
| **Status** | **Message** | **Triggering Condition** |
| 00 | TRANSACTION SUCCESSFUL | When the record is successfully inserted into the DB2 tables |
| 91 | INVALID RECORD TYPE | When the record type is not valid |
| INVALID RECORD SEQUENCE NUMBER | When sequence number is not valid |
| INVALID BENEFIT PRODUCT TYPE | When the product type is not valid for the benefit record |
| INVALID CONTRACT TYPE | When contract type is not valid |
| INVALID ACCUM NAME | When accum name is not valid |
| INVALID ACCUM TYPE | When accum type is not valid |
| INVALID ACCUM SUB TYPE | When accum sub type is not valid |
| BEN RECORD COUNTS ARE NOT MATCHING | When the benefit count in the trailer record doesn’t match the number of benefit records present in the feed file |
| ACM RECORD COUNTS ARE NOT MATCHING | When the accum count in the trailer record doesn’t match the number of accum records present in the feed file |
| INVALID SOURCE SYSTEM | When source system is not valid |
| COVERAGE PLAN INDICATOR IS SPACES | When plan id is spaces |
| TOTAL RECORD COUNTS ARE NOT MATCHING | When the total count in the trailer record doesn’t match the number of records present in the feed file (benefit record + accum records) |
| INVALID MEMBR TIER IND | When the member tier indicator is not valid |
| INVALID CDHP PROD TYPE | When the CDHP product type is not valid |
| INVALID CDHP VENDOR | When the  is not valid |
| INVALID CONT EMBD INDICATOR | When the embedded indicator is not valid |
| INVALID DED IN OOP INDICATOR | When the Ded included in OOP is not valid |
| INVALID DED IN TO OUT INDICATOR | When the Ded IN goes to OUT is not valid |
| INVALID DED OUT TO IN INDICATOR | When the Ded OUT goes to IN is not valid |
| INVALID OOP IN TO OUT INDICATOR | When the OOP IN Goes to OUT  is not valid |
| INVALID OOP OUT TO IN INDICATOR | When the OOP OUT goes to IN is not valid |
| INVALID DED PREFRD PAR INDICATOR | When the Preferred PAR Deductible cross apply indicator is not valid |
| INVALID OOP PREFRD PAR INDICATOR | When the  Preferred PAR out of pocket cross apply indicator is not valid |
| INVALID LAST QTR INDICATOR | When the Last Quarter is not valid |
| INVALID CARRY OVER INDICATOR | When the  CARRY OVER RESET IND is not valid |
| INVALID ADULT OOP INDICATOR | When the Adult Vision OOP Exclusion IND is not valid |
| INVALID ACCUM MAX AMT | When the Accum amount is not valid |
| INVALID ACCUM MAX COUNT | When the limit occurs is not valid |
| INVALID ACCUM MAX DAYS | When the limit days is not valid |
| INVALID MED CMNGL INDICATOR | When the Medical Commingle  is not valid |
| INVALID RX CMNGL INDICATOR | When the RX Commingle is not valid |
| INVALID VISION CMNGL INDICATOR | When the vision Commingle is not valid |
| INVALID DENTAL CMNGL INDICATOR | When the dental Commingle is not valid |
| 92 | DATE FORMAT IS NOT CORRECT | When date format for contract start and end date is not correct |
| 95 | ACM REC NOT RECEIVED FOR BEN RECORD | When Accum record is not present for a benefit record. |
| 96 | ODS HAS SAME REC VERSION SO THIS REC IGNORED | When the same Benefit and Accum records are present in the benefit translation tables |
| 99 | SYSTEM FAILURE | When the IO routine to insert the data into the translation table returns any return code other than zero |
| SPACES | No message is displayed | When previous record is invalid |
| 02 | DUPLICATE BENEFIT FEED FILE | When duplicate feed file is received. |
| 03 | FEED FILE CONTAINS ONLY HEADER AND TRAILER | When feed file received has only HDR and TRL records. |
| 04 | MISMATCH IN THE TRAILER RECORD COUNTS | When the benefit count, Accum count or total count in the trailer record doesn’t match the number of benefit records, Accum records and total records present in the feed file. |
| 10 | ODS SWITCHED OFF | When plan is migrated from EHB to non EHB. |
| 11 | TIER DEACTIVATED | When the particular tier level is deactivated. |
| 12 | DATE DEACTIVATED | When the date is deactivated. |

* Response file is sent back to legacy system using WGS NDM process.

**GNCBNIOS (ODS insert or Update)**

A new IO routine GNCBNIOS will be created to do INPUT/OUTPUT operations. This routine will read, insert, update and delete data from new benefit translation tables and also from history tables. This routine can be invoked from batch as well as online (Benefit and Accum enquiry screens).

**DB2 table layout for the translation tables.**

**GNC.CONTR\_LEV\_INFO:**

|  |
| --- |
| Process Options Utilities Help |
| ---------------------------------------------------------------------- |
| FM/DB2 (DB2P) Table View 1 of unknown |
| Command ===> Scroll CSR |
| **TABLE DB2P.GNC.CONTR\_LEV\_INFO** Format SNGL |
| Top Line is 1 of 32 in Row 1 |
| Key Column Data |
| PU BEN\_SRC\_SYS 808 |
| PU COVG\_PLN\_ID AB95 |
| PU ODS\_BEN\_STRT\_DT 2015-01-01 |
| PU ODS\_BEN\_END\_DT 9999-12-31 |
| PU MBR\_TIER\_IND 0 |
| VERS\_NBR 3 |
| SRC\_SYS\_PROD\_TYPE MED |
| CHDP\_PROD\_TYPE X |
| CHDP\_VNDR |
| EMBD\_IND X |
| CONTR\_TYPE P |
| DED\_IN\_OOP\_IND A |
| DED\_IN\_TO\_OUT\_IND X |
| DED\_OUT\_TO\_IN\_IND X |
| OOP\_IN\_TO\_OUT\_IND X |
| OOP\_OUT\_TO\_IN\_IND X |
| DED\_PRF\_PAR\_IND X |
| OOP\_PRF\_PAR\_IND X |
| COPAY\_IN\_OOP\_IND |
| COPAY\_PRF\_PAR\_IND |
| COPAY\_IN\_TO\_OUT\_IND |
| COPAY\_OUT\_TO\_IN\_IND |
| COINS\_IN\_OOP\_IND |
| COINS\_PRF\_PAR\_IND |
| COINS\_IN\_TO\_OUT\_IND |
| COINS\_OUT\_TO\_IN\_IND |
| LST\_QTR\_IND B |
| CRY\_OVR\_RESET\_IND X |
| ADLT\_OOP\_EXCLD\_IND Y |
| LST\_UPDT\_DTM 2014-12-09-01.21.04.292859 |
| LST\_UDT\_USR\_ID GNC37ADD |
| DUAL\_UPD\_IND Y |
|  |

**GNC.ACCUM\_LEV\_INFO:**

|  |
| --- |
| Process Options Utilities Help |
| ------------------------------------------------------------------------------- |
| FM/DB2 (DB2P) Table View 1 of unknown |
| Command ===> Scroll CSR |
| TABLE DB2P.GNC.ACCUM\_LEV\_INFO Format SNGL |
| Top Line is 1 of 19 in Row 1 |
| Key Column Data |
| PU BEN\_SRC\_SYS 808 |
| PU COVG\_PLN\_ID AB95 |
| PU ODS\_BEN\_STRT\_DT 2015-01-01 |
| PU ODS\_BEN\_END\_DT 9999-12-31 |
| PU MBR\_TIER\_IND 0 |
| VERS\_NBR 3 |
| PU ACCUMR\_NM FAMDED |
| ACCUMR\_TYPE F |
| ACCUM\_SB\_TYPE A |
| LMT\_AMT 1800.00 |
| LMT\_OCR\_CNT 0 |
| LMT\_DYS\_CNT 0 |
| MED\_COMMINGLE\_IND Y |
| RX\_COMMINGLE\_IND N |
| VSN\_COMMINGLE\_IND N |
| DEN\_COMMINGLE\_IND N |
| LAST\_UPDT\_TS 2014-12-09-01.21.04.295114 |
| LST\_UPDT\_USR\_ID GNC37ADD |
| GEN\_ACCUM\_NM 010302 |
| \*\*\* End of record \*\*\* |
|  |

**GNC.CONTR\_LEV\_INFO\_HIST:**

|  |
| --- |
| Process Options Utilities Help |
| --------------------------------------------------------------------------- |
| FM/DB2 (DB2P) Table View 1 of unknown |
| Command ===> Scroll CSR |
| TABLE DB2P.GNC.CONTR\_LEV\_INFO\_HIST Format SNGL |
| Top Line is 1 of 32 in Row 1 |
| Key Column Data |
| U BEN\_SRC\_SYS 808 |
| U COVG\_PLN\_ID AB95 |
| U ODS\_BEN\_STRT\_DT 2015-01-01 |
| U ODS\_BEN\_END\_DT 9999-12-31 |
| U MBR\_TIER\_IND 0 |
| U VERS\_NBR 1 |
| SRC\_SYS\_PROD\_TYPE MED |
| CHDP\_PROD\_TYPE X |
| CHDP\_VNDR |
| EMBD\_IND X |
| CONTR\_TYPE P |
| DED\_IN\_OOP\_IND A |
| DED\_IN\_TO\_OUT\_IND X |
| DED\_OUT\_TO\_IN\_IND X |
| OOP\_IN\_TO\_OUT\_IND X |
| OOP\_OUT\_TO\_IN\_IND X |
| DED\_PRF\_PAR\_IND X |
| OOP\_PRF\_PAR\_IND X |
| COPAY\_IN\_OOP\_IND |
| COPAY\_PRF\_PAR\_IND |
| COPAY\_IN\_TO\_OUT\_IND |
| COPAY\_OUT\_TO\_IN\_IND |
| COINS\_IN\_OOP\_IND |
| COINS\_PRF\_PAR\_IND |
| COINS\_IN\_TO\_OUT\_IND |
| COINS\_OUT\_TO\_IN\_IND |
| LST\_QTR\_IND B |
| CRY\_OVR\_RESET\_IND X |
| ADLT\_OOP\_EXCLD\_IND Y |
| LAST\_UPDT\_DTM 2014-11-01-22.33.48.443328 |
| LST\_UPDT\_USR\_ID GNC37ADD |
| DUAL\_UPD\_IND |
|  |

**GNC.ACCUM\_LEV\_INFO\_HIST:**

|  |
| --- |
| Process Options Utilities Help |
| ---------------------------------------------------------------------- |
| FM/DB2 (DB2P) Table View 1 of unknown |
| Command ===> Scroll CSR |
| TABLE DB2P.GNC.ACCUM\_LEV\_INFO\_HIST Format SNGL |
| Top Line is 1 of 19 in Row 1 |
| Key Column Data |
| U BEN\_SRC\_SYS 808 |
| U COVG\_PLN\_ID AB95 |
| U ODS\_BEN\_STRT\_DT 2015-01-01 |
| U ODS\_BEN\_END\_DT 9999-12-31 |
| U MBR\_TIER\_IND 0 |
| U VERS\_NBR 1 |
| U ACCUMR\_NM FAMDED |
| ACCUMR\_TYPE F |
| ACCUM\_SB\_TYPE A |
| LMT\_AMT 1800.00 |
| LMT\_OCR\_CNT 0 |
| LMT\_DYS\_CNT 0 |
| MED\_COMMINGLE\_IND Y |
| RX\_COMMINGLE\_IND Y |
| VSN\_COMMINGLE\_IND N |
| DEN\_COMMINGLE\_IND N |
| LST\_UPDT\_USR\_ID GNC37ADD |
| LST\_UPDT\_TS 2014-11-01-22.33.48.444901 |
| GEN\_ACCUM\_NM |
| \*\*\* End of record \*\*\* |
|  |

The IO routine GNCBNIOS will be created to perform below functionality:

* **Select Benefit and ACCUM data from the benefit translation tables**:

The Benefit and Accum translation tables CONTR-LEV-INFO and ACCUM-LEV-INFO will be queried with the given key fields - benefit source system, coverage plan indicator, start date, end date(optional) and member tier indicator to get the benefit and accum data; if the record is not found then a non-zero return code will be passed to the calling program.

* **Select Benefit and ACCUM data from the benefit history tables**:

The Benefit and Accum translation tables CONTR-LEV-INFO\_HIST and ACCUM-LEV-INFO\_HIST will be queried with the given key fields - benefit source system, coverage plan indicator, start date, end date(optional) and member tier indicator to get the benefit and accum data; if the record is not found then a non-zero return code will be passed to the calling program.

* **Update/insert data into benefit translation and history tables:**

The Benefit and Accum translation tables CONTR-LEV-INFO and ACCUM-LEV-INFO will be queried with the given key fields. If record is present then the comparison will be done with the data from feed file and existing values from DB. If there is only change in the benefit termination date, then the termination date of existing record is updated benefit tables. If there are changes in the benefit fields, then the following actions are performed:

* + - Verify there are no date overlap issues with new feed data and existing date from table. If the dates are overlapping then reject the record with ’97’ status code. If the dates are not overlapping:
      * Insert the obtained record into the corresponding history tables CONTR-LEV-INFO-HIST and ACCUM-LEV-INFO-HIST.
      * Delete the record from ACCUM-LEV-INFO table.
      * Delete the record from the CONTR-LEV-INFO.
      * Insert the data into CONTR-LEV-INFO and ACCUM-LEV-INFO after incrementing the version number.
      * If the feed is already processed, then populate the status message ‘96’ and no insert/update will occur.
      * In case of any DB/environment issues populate the status message, ‘99’.
      * If insert/update was successful, populate the status code ‘00’.

**GNC3742, GNC3743, GNC3744 (REPORTS)**

* EHB response file generated through the above process are used for creating the reports.
* GNC3742 - Success report program.
* GNC3743 - Failure report program.
* GNC3744 - Request report program.
* Report process validates the response and corresponding Success, Failure and request reports are created.
* Success and failure reports are to be created based on the Transaction status codes.
* If the response record’s transaction status is 00(success), 10(ODS switched off), 11(tier deactivated), 12(date deactivated), those records are routed to Success report.
* All other records are routed to Failure reports except records with 96 status which are excluded from both success and failure reports.
* Request report contains all the records in response file.
* All the Reports routed to dispatch. Dispatch report Id’s for different systems are mentioned below.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **System name** | **Job Name** | **Proc** | **Report ID** | **Remarks** |
| FACETS | GNC37BDD | GNC37BDD | GNC37011 | Success report |
| GNC37012 | Failure report |
| GNC37013 | Response report |
| WGS | GNC37ADD | GNC370DD | GNC37021 | Success report |
| GNC37022 | Failure report |
| GNC37023 | Response report |
| HIX/SPIDER | GNC37HDD | GNC370DD | GNC37031 | Success report |
| GNC37032 | Failure report |
| GNC37033 | Response report |
| CS90 | GNC37EDD | GNC371DD | GNC37041 | Success report |
| GNC37042 | Failure report |
| GNC37043 | Response report |
| HEALTHLINK | GNC37GDD | GNC371DD | GNC37051 | Success report |
| GNC37052 | Failure report |
| GNC37053 | Response report |
| ACES | GNC37CDD | GNC372DD | GNC37061 | Success report |
| GNC37062 | Failure report |
| GNC37063 | Response report |
| CHIPS | GNC37DDD | GNC372DD | GNC37071 | Success report |
| GNC37072 | Failure report |
| GNC37073 | Response report |
| NASCO | GNC37FDD | GNC37FDD | GNC37081 | Success report |
| GNC37082 | Failure report |
| GNC37083 | Response report |

**Job details to process the feed file.**

**Health Link:**

The following are the jobs executed:

* + 1. **GNC36GTD**
    2. **GNC37GDD**
    3. **GNC37GTD**

**Job GNC36GTD**

* This job executes the Proc GNC36GTD.

1. Step to get the health link file for processing via FTP.
2. Step to delete the trigger file.

* Input file : Text file
* Output file : MF file –

/wellpoint/ehb\_healthlink/Inbound/ehb\_healthlink\_feed.txt

GNCP.NONX.GNC36GTD.HLTLNK.FEED

**Job GNC37GDD**

**Purpose: -** To process the feed file from health link. Job will be triggered once the feed file is received in WGS side.

**Input file:** GNCP.NONX.GNC36GTD.HLTLNK.FEED

**Output files:** GNCP.NONX.GNC37GDD.GNC9843.WGS.RESP.G\*

GNCP.NONX.GNC37GDD.GNC3742.WGS.SRPT.G\*

GNCP.NONX.GNC37GDD.GNC3743.WGS.FRPT.G\*

GNCP.NONX.GNC37GDD.GNC3744.WGS.RRPT.\*G

* This job executes the PRC GNC371DD.
  + - 1. STEPP005: GNC9846 benefit feed file validation program.
* Input file : GNCP.NONX.GNC36GTD.HLTLNK.FEED
* Output file: GNCP.NONX.GNC37GDD.HLTLNK.CNTLFIL.G\*
  + - 1. STEPS010: Sort to split the feed file to remove the header record.
      2. STEPS015: Sort step to merge the new header with feed file.
      3. STEPU017: IDCAMS step to set the return code 08.
      4. STEPU020: IEBGENER take the backup of benefit feed file.
      5. STEPU025: IEFBR14 to allocate the GSAM file.
      6. STEPS026: SORT to split the header, detailed and trailer records.
      7. STEPS027: SORT the detail records based on start date and coverage plan id.
      8. STEPS028: SORT to merger the header sorted detail and trailer records.
      9. STEPP030: GNC9843 EHB benefit feed file processing program.
* Input file :

GNCP.NONX.GNC37GDD.WGS.SOUT.G\*

* Output file : GNCP.NONX.GNC37GDD.GNC9843.WGS.RESP.G\*
  + - 1. STEPU040: IEFBR14 to delete the benefit feed file.
      2. STEPS050: SORT step for sorting the files based on status code.
* Input file : GNCP.NONX.GNC37GDD.GNC9843.WGS.RESP.G\*
* Output files :

GNCP.NONX GNC37GDD.SORT.WGS.SUSS.G\*

GNCP.NONX GNC37GDD.SORT.WGS.FAIL.G\*

* + - 1. STEPP060: GNC3742 EHB success reports file processing.
* Input file :

GNCP.NONX GNC37GDD.SORT.WGS.SUSS.G\*

* Output file : GNCP.NONX.GNC37GDD.GNC3742.WGS.SRPT.G\*
  + - 1. STEPP070: GNC3743 EHB failure reports file processing.
* Input file :

GNCP.NONX GNC37GDD.SORT.WGS.FAIL.G\*

* Output file : GNCP.NONX.GNC37GDD.GNC3743.WGS.FRPT.G\*
  + - 1. STEPP080: GNC3744 EHB request reports file processing.
* Input file : GNCP.NONX.GNC37GDD.GNC9843.WGS.RESP.G\*
* Output file : GNCP.NONX.GNC37GDD.GNC3744.WGS.RRPT.G\*
  + - 1. STEPU090: PSUTDUMP route the success report to dispatch.
      2. STEPU100: PSUTDUMP route the success report to dispatch.
      3. STEPU110: PSUTDUMP route the success report to dispatch.
      4. ABENDCHK : ABEND CHECK ROUTINE

**Job GNC37GTD**

**Purpose:** - To send the response file to Health link via FTP.

* This job will execute Proc GNC37GTD.

STEPU010: FTP the file to health link.

* Input file: GNCP.NONX.GNC37GDD.GNC9843.WGS.RESP.G\*
* Output file: /outgoing/ehb\_healthlink/ehb\_healthlink\_resp.txt

**CS90:**

The following are the jobs executed:

* + 1. **GNC36ETD**
    2. **GNC37EDD**
    3. **GNC37ETD**

**Job GNC36ETD**

* This job executes the PRC GNC36ETD.
  + 1. Step to get the CS90 file for processing via FTP.
    2. Step to delete the trigger file.
* Input file : Text file
* Output file : MF file –

/wellpoint/MOM9PWVWBS002/FTP-ODS\_BATCH/inbound /ehb\_cs90obis\_feed.txt

GNCP.NONX.GNC36ETD.CS90.FEED

**Job GNC37EDD**

**Purpose: -** To process the feed file from CS90. Job will be triggered once the feed file is received in WGS side.

**Input file:** GNCP.NONX.GNC36ETD.CS90.FEED

**Output files:** GNCP.NONX.GNC37EDD.GNC9843.WGS.RESP.G\*

GNCP.NONX.GNC37EDD.GNC3742.WGS.SRPT.G\*

GNCP.NONX.GNC37EDD.GNC3743.WGS.FRPT.G\*

GNCP.NONX.GNC37EDD.GNC3744.WGS.RRPT.\*G

* This job executes the PRC GNC371DD.
  + - 1. STEPP005: GNC9846 benefit feed file validation program.
* Input file : GNCP.NONX.GNC36ETD.CS90.FEED
* Output file : GNCP.NONX.GNC37EDD.CS90.CNTLFIL.G\*
  + - 1. STEPS010: SORT splitting the feed file to remove the header record.
      2. STEPS015: SORT step to merge the new header with feed file.
      3. STEPU017: IDCAMS step to set the return code 08.
      4. STEPU020: IEBGENER take the back up of benefit feed file.
      5. STEPU025: IEFBR14 to allocate the GSAM file.
      6. STEPS026: SORT to split the header, detailed and trailer records.
      7. STEPS027: SORT the detail records based on start date and coverage plan id.
      8. STEPS028: SORT to merge the header sorted detail and trailer records.
      9. STEPP030: GNC9843 EHB benefit feed file processing program.
* Input file :

GNCP.NONX.GNC37EDD.WGS.SOUT.G\*

* Output file : GNCP.NONX.GNC37EDD.GNC9843.WGS.RESP.G\*
  + - 1. STEPU040: IEFBR14 to delete the benefit feed file.
      2. STEPS050: SORT step for sorting the files based on status code.
* Input file : GNCP.NONX.GNC37EDD.GNC9843.WGS.RESP.G\*
* Output files :

GNCP.NONX GNC37EDD.SORT.WGS.SUSS.G\*GNCP.NONX GNC37EDD.SORT.WGS.FAIL.G\*

* + - 1. STEPP060: GNC3742 EHB success reports file processing.
* Input file :

GNCP.NONX GNC37EDD.SORT.WGS.SUSS.G\*

* Output file : GNCP.NONX.GNC37EDD.GNC3742.WGS.SRPT.G\*
  + - 1. STEPP070: GNC3743 EHB failure reports file processing.
* Input file :

GNCP.NONX GNC37EDD.SORT.WGS.FAIL.G\*

* Output file : GNCP.NONX.GNC37EDD.GNC3743.WGS.FRPT.G\*
  + - 1. STEPP080: GNC3744 EHB request reports file processing.
* Input file : GNCP.NONX.GNC37EDD.GNC9843.WGS.RESP.G\*
* Output file : GNCP.NONX.GNC37EDD.GNC3744.WGS.RRPT.G\*
  + - 1. STEPU090: PSUTDUMP route the success report to dispatch.
      2. STEPU100: PSUTDUMP route the failure report to dispatch.
      3. STEPU110: PSUTDUMP route the request report to dispatch.
      4. ABENDCHK: ABEND routine.

**Job GNC37ETD**

Purpose: To send the response file to CS90 via FTP.

* This job executes the Proc GNC37ETD

STEPU010: FTP the file to CS90.

* Input file: GNCP.NONX.GNC37EDD.GNC9843.WGS.RESP.G\*
* Output file:

/wellpoint/MOM9PWVWBS002/FTP-ODS\_BATCH/outbound/ehb\_cs90obis\_resp.txt

**FACETS:**

The following are the jobs executed:

* + 1. **GNC37BDD**
    2. **GNC37BTD**

**Job GNC37BDD**

**Purpose: -** To process the feed file from FACETS. Job will be triggered once the feed file is received in WGS side.

**Input file:** GNCP.NONX.GNC37BDD.FACETS.FEED.G\*

**Output files:** GNCP.NONX.GNC37BDD.GNC9843.WGS.RESP.G\*

GNCP.NONX.GNC37BDD.GNC3742.WGS.SRPT.G\*

GNCP.NONX.GNC37BDD.GNC3743.WGS.FRPT.G\*

GNCP.NONX.GNC37BDD.GNC3744.WGS.RRPT.\*G

* + This job executes the PRC GNC37BDD.

1. STEPP010: JVMLDM50 program for benefit feed file.

* Input file : GNCP.NONX.GNC37BDD.FACETS.FEED(0)
* Output file : GNCP.NONX.GNC37BDD.FACETS.CNTLFIL.G\*
* Return code 0 and 150 are acceptable.

1. STEPS020: SORT to split the header, detailed and trailer records.
2. STEPS030: SORT the detail records based on start date and coverage plan id.
3. STEPS040: SORT to merger the header sorted detail and trailer records.
4. STEPU050: IEBGENER take the back up of benefit feed file.
5. STEPU060: IEFBR14 to allocate the GSAM file.
6. STEPP070: GNC9843 EHB benefit feed file processing.

* Input file :

GNCP.NONX.GNC37BDD.WGS.SOUT.G\*

* Output file : GNCP.NONX.GNC37BDD.GNC9843.WGS.RESP.G\*

1. STEPP080: JVMLDM50 infogix program for response file.
2. STEPS090: SORT step for sorting the files based on status code.

* Input file : GNCP.NONX.GNC37BDD.GNC9843.WGS.RESP.G\*
* Output files :

GNCP.NONX GNC37BDD.SORT.WGS.SUSS.G\*

GNCP.NONX GNC37BDD.SORT.WGS.FAIL.G\*

1. STEPP100: GNC3742 EHB success reports file processing.

* Input file : GNCP.NONX GNC37BDD.SORT.WGS.SUSS.G\*
* Output file : GNCP.NONX.GNC37BDD.GNC3742.WGS.SRPT.G\*

1. STEPP110: GNC3743 EHB failure reports file processing.

* Input file : GNCP.NONX GNC37BDD.SORT.WGS.SUSS.G\*
* Output file : GNCP.NONX.GNC37BDD.GNC3743.WGS.FRPT.G\*

1. STEPP120: GNC3744 EHB request reports file processing.

* Input file : GNCP.NONX.GNC37BDD.GNC9843.WGS.RESP.G\*
* Output file : GNCP.NONX.GNC37BDD.GNC3744.WGS.RRPT.G\*

1. STEPU130: PSUTDUMP route the success report to dispatch.
2. STEPU140: PSUTDUMP route the failure report to dispatch.
3. STEPU150: PSUTDUMP route the request report to dispatch.
4. ABENDCHK: ABEND routine.

**Job GNC37BTD**

**Purpose:** To send the response file to FACETS via NDM.

* + - This job executes the Proc GNC37BTD

STEPS005: SORT step to append the date to NDM file name.

STEPU010: DMBATCH NDM file transmission of facets response file to CT mainframe.

* Input file :

GNCP.NONX.GNC37BDD.GNC9843.WGS.RESP.G\*

* Output file :

/aims/data/GNCP.NONX.GNC37BDD.FACETS.FEED.RESPONSE.DDMMYYYY

**ACES:**

The following are the jobs executed:

* + 1. **GNC37CDD**
    2. **GNC37CTD**

**Job GNC37CDD**

**Purpose:** To process the feed file from ACES. Job will be triggered once the feed file is received in WGS side.

**Input file:** GNCP.NONX.GNC37CDD.ACES.FEED.G\*

**Output files:** GNCP.NONX.GNC37CDD.GNC9843.WGS.RESP.G\*

GNCP.NONX.GNC37CDD.GNC3742.WGS.SRPT.G\*

GNCP.NONX.GNC37CDD.GNC3743.WGS.FRPT.G\*

GNCP.NONX.GNC37CDD.GNC3744.WGS.RRPT.\*G

* + - This job executes the Proc GNC372DD.

1. STEPP005: GNC9846 benefit feed file validation program.

* Input file : GNCP.NONX.GNC37CDD.ACES.FEED
* Output file : GNCP.NONX.GNC37CDD.ACES.CNTLFIL.G\*

1. STEPS010: SORT splitting the feed file to remove the header record.
2. STEPS015: SORT step to merge the new header with feed file.
3. STEPU017: IDCAMS step to set the return code 08.
4. STEPU020: IEBGENER take the back up of benefit feed file.
5. STEPU025: IEFBR14 to allocate the GSAM file.
6. STEPS026: SORT to split the header, detailed and trailer records.
7. STEPS027: SORT the detail records based on start date and coverage plan id.
8. STEPS028: SORT to merger the header sorted detail and trailer records.
9. STEPP030: GNC9843 EHB benefit feed file processing program.

* Input file :

GNCP.NONX.GNC37CDD.WGS.SOUT.G\*

* Output file : GNCP.NONX.GNC37CDD.GNC9843.WGS.RESP.G\*

1. STEPS050: SORT step for sorting the files based on status code.

* Input file : GNCP.NONX.GNC37CDD.GNC9843.WGS.RESP.G\*
* Output files :

GNCP.NONX GNC37CDD.SORT.WGS.SUSS.G\*

GNCP.NONX GNC37CDD.SORT.WGS.FAIL.G\*

1. STEPP060: GNC3742 EHB success reports file processing.

* Input file : GNCP.NONX GNC37CDD.SORT.WGS.SUSS.G\*
* Output file : GNCP.NONX.GNC37CDD.GNC3742.WGS.SRPT.G\*

1. STEPP070: GNC3743 EHB failure reports file processing.

* Input file : GNCP.NONX GNC37CDD.SORT.WGS.SUSS.G\*
* Output file : GNCP.NONX.GNC37CDD.GNC3743.WGS.FRPT.G\*

1. STEPP080: GNC3744 EHB request reports file processing.

* Input file : GNCP.NONX.GNC37CDD.GNC9843.WGS.RESP.G\*
* Output file : GNCP.NONX.GNC37CDD.GNC3744.WGS.RRPT.G\*

1. STEPU090: PSUTDUMP route the success report to dispatch.
2. STEPU100: PSUTDUMP route the failure report to dispatch.
3. STEPU110: PSUTDUMP route the request report to dispatch.
4. ABENDCHK: ABEND routine.

**Job GNC37CTD**

**Purpose:** To send the response file to ACES via NDM.

* + - This job executes the Proc GNC37CTD.

STEPU010: DMBATCH SEND THE RESPONSE FILE TO ACES VIA NDM

* Input file: GNCP.NONX.GNC37CDD.GNC9843.WGS.RESP.G\*
* Output file: C.P.FD81337.ACES.FEED

**CHIPS:**

The following are the jobs executed:

* + 1. **GNC37DDD**
    2. **GNC37DTD**

Each is explained in detail below.

**Job GNC37DDD**

**Purpose:** To process the feed file from CHIPS. Job will be triggered once the feed file is received in WGS side.

**Input file:** GNCP.NONX.GNC37DDD.CHIPS.FEED.G\*

**Output files:** GNCP.NONX.GNC37DDD.GNC9843.WGS.RESP.G\*

GNCP.NONX.GNC37DDD.GNC3742.WGS.SRPT.G\*

GNCP.NONX.GNC37DDD.GNC3743.WGS.FRPT.G\*

GNCP.NONX.GNC37DDD.GNC3744.WGS.RRPT.\*G

* + - This job executes the PRC GNC372DD.

1. STEPP005: GNC9846 benefit feed file validation program.

* Input file : GNCP.NONX.GNC37DDD.CHIPS.FEED
* Output file : GNCP.NONX.GNC37DDD.CHIPS.CNTLFIL.G\*

1. STEPS010: SORT splitting the feed file to remove the header record.
2. STEPS015: SORT step to merge the new header with feed file.
3. STEPU017: IDCAMS step to set the return code 08.
4. STEPU020: IEBGENER take the back up of benefit feed file.
5. STEPU025: IEFBR14 to allocate the GSAM file.
6. STEPS026: SORT to split the header, detailed and trailer records.
7. STEPS027: SORT the detail records based on start date and coverage plan id.
8. STEPS028: SORT to merge the header sorted detail and trailer records.
9. STEPP030: GNC9843 EHB benefit feed file processing program.

* Input file :

GNCP.NONX.GNC37DDD.WGS.SOUT.G\*

* Output file : GNCP.NONX.GNC37DDD.GNC9843.WGS.RESP.G\*

1. STEPS050: SORT step for sorting the files based on status code.

* Input file : GNCP.NONX.GNC37DDD.GNC9843.WGS.RESP.G\*
* Output files :

GNCP.NONX GNC37DDD.SORT.WGS.SUSS.G\*

GNCP.NONX GNC37DDD.SORT.WGS.FAIL.G\*

1. STEPP060: GNC3742 EHB success reports file processing.

* Input file : GNCP.NONX GNC37DDD.SORT.WGS.SUSS.G\*
* Output file : GNCP.NONX.GNC37DDD.GNC3742.WGS.SRPT.G\*

1. STEPP070: GNC3743 EHB failure reports file processing.

* Input file : GNCP.NONX GNC37DDD.SORT.WGS.SUSS.G\*
* Output file : GNCP.NONX.GNC37DDD.GNC3743.WGS.FRPT.G\*

1. STEPP080: GNC3744 EHB request reports file processing.

* Input file : GNCP.NONX.GNC37DDD.GNC9843.WGS.RESP.G\*
* Output file : GNCP.NONX.GNC37DDD.GNC3744.WGS.RRPT.G\*

1. STEPU090: PSUTDUMP route the success report to dispatch.
2. STEPU100: PSUTDUMP route the failure report to dispatch.
3. STEPU110: PSUTDUMP route the request report to dispatch.
4. ABENDCHK: ABEND routine.

**Job GNC37DTD**

Purpose: To send the response file to CHIPS via NDM.

* + - This job executes the PRC GNC37DTD.

STEPU010: DMBATCH send the response file to ACES via NDM.

* Input file : GNCP.NONX.GNC37DDD.GNC9843.WGS.RESP.G\*
* Output file : CHIPS.DVC2097.ODS.BEN.RSPFL(+1)

**NASCO:**

The following are the jobs executed:

* + 1. **GNC37FDD**
    2. **GNC37FTD**

Each is explained in detail below.

**Job GNC37FDD**

**Purpose:** To process the feed file from NASCO. Job will be triggered once the feed file is received in WGS side.

**Input file:** GNCP.NONX.GNC37FDD.NASCO.FEED.G\*

**Output files:** GNCP.NONX.GNC37FDD.GNC9843.WGS.RESP.G\*

GNCP.NONX.GNC37FDD.GNC3742.WGS.SRPT.G\*

GNCP.NONX.GNC37FDD.GNC3743.WGS.FRPT.G\*

GNCP.NONX.GNC37FDD.GNC3744.WGS.RRPT.\*G

* + - This job executes the PRC GNC372DD.

1. STEPP005: GNC9846 benefit feed file validation program.

* Input file : GNCP.NONX.GNC37FDD.NASCO.FEED
* Output file : GNCP.NONX.GNC37DDD.NASCO.CNTLFIL.G\*

1. STEPS010: SORT splitting the feed file to remove the header record.
2. STEPS015: SORT step to merge the new header with feed file.
3. STEPU017: IDCAMS step to set the return code 08.
4. STEPU020: IEBGENER take the back up of benefit feed file.
5. STEPU025: IEFBR14 to allocate the GSAM file.
6. STEPS026: SORT to split the header, detailed and trailer records.
7. STEPS027: SORT the detail records based on start date and coverage plan id.
8. STEPS028: SORT to merger the header sorted detail and trailer records.
9. STEPP030: GNC9843 EHB benefit feed file processing program.

* Input file :

GNCP.NONX.GNC37FDD.WGS.SOUT.G\*

* Output file : GNCP.NONX.GNC37FDD.GNC9843.WGS.RESP.G\*

1. STEPS050: SORT step for sorting the files based on status code.

* Input file : GNCP.NONX.GNC37FDD.GNC9843.WGS.RESP.G\*
* Output files :

GNCP.NONX GNC37FDD.SORT.WGS.SUSS.G\*

GNCP.NONX GNC37FDD.SORT.WGS.FAIL.G\*

1. STEPP060: GNC3742 EHB success reports file processing.

* Input file : GNCP.NONX GNC37FDD.SORT.WGS.SUSS.G\*
* Output file : GNCP.NONX.GNC37FDD.GNC3742.WGS.SRPT.G\*

1. STEPP070: GNC3743 EHB failure reports file processing.

* Input file : GNCP.NONX GNC37FDD.SORT.WGS.SUSS.G\*
* Output file : GNCP.NONX.GNC37FDD.GNC3743.WGS.FRPT.G\*

1. STEPP080: GNC3744 EHB request reports file processing.

* Input file : GNCP.NONX.GNC37FDD.GNC9843.WGS.RESP.G\*
* Output file : GNCP.NONX.GNC37FDD.GNC3744.WGS.RRPT.G\*

1. STEPU090: PSUTDUMP route the success report to dispatch.
2. STEPU100: PSUTDUMP route the failure report to dispatch.
3. STEPU110: PSUTDUMP route the request report to dispatch.
4. ABENDCHK: ABEND routine.

**Job GNC37FTD**

**Purpose**: To send the response file to NASCO via NDM.

* + - This job executes the Proc GNC37FTD

STEPU010: DMBATCH send the response file to NASCO via NDM

* Input file : GNCP.NONX.GNC37FDD.GNC9843.WGS.RESP.G\*
* Output file : CHIPS.DVC2097.ODS.BEN.RSPFL(+1)

**WGS:**

The following are the jobs executed:

* + 1. **GNC37ADD**

**Job GNC37ADD**

**Purpose:** To process the feed file from WGS.

**Input file:** GNCP.NONX.GNC37ADD.WGS.FEED.G\*

**Output files:** GNCP.NONX.GNC37ADD.GNC9843.WGS.RESP.G\*

GNCP.NONX.GNC37ADD.GNC3742.WGS.SRPT.G\*

GNCP.NONX.GNC37ADD.GNC3743.WGS.FRPT.G\*

GNCP.NONX.GNC37ADD.GNC3744.WGS.RRPT.\*G

**Executing Proc GNC370DD**

1. STEPP005: GNC9846 benefit feed file validation program.

* Input file : GNCP.NONX.GNC37ADD.WGS.FEED
* Output file : GNCP.NONX.GNC37ADD.WGS.CNTLFIL.G\*

1. STEPS010: SORT splitting the feed file to remove the header record.
2. STEPS015: SORT step to merge the new header with feed file.
3. STEPU017: IDCAMS step to set the return code 08.
4. STEPU020: IEBGENER take the back up of benefit feed file.
5. STEPU025: IEFBR14 to allocate the GSAM file.
6. STEPS026: SORT to split the header, detailed and trailer records.
7. STEPS027: SORT the detail records based on start date and coverage plan id.
8. STEPS028: SORT to merger the header sorted detail and trailer records.
9. STEPP030: GNC9843 EHB benefit feed file processing program.

* Input file :

GNCP.NONX.GNC37ADD.WGS.SOUT.G\*

* Output file : GNCP.NONX.GNC37ADD.GNC9843.WGS.RESP.G\*

1. STEPS050: SORT step for sorting the files based on status code.

* Input file : GNCP.NONX.GNC37ADD.GNC9843.WGS.RESP.G\*
* Output files :

GNCP.NONX GNC37ADD.SORT.WGS.SUSS.G\*

GNCP.NONX GNC37ADD.SORT.WGS.FAIL.G\*

1. STEPP060: GNC3742 EHB success reports file processing.

* Input file : GNCP.NONX GNC37ADD.SORT.WGS.SUSS.G\*
* Output file : GNCP.NONX.GNC37ADD.GNC3742.WGS.SRPT.G\*

1. STEPP070: GNC3743 EHB failure reports file processing.

* Input file : GNCP.NONX GNC37ADD.SORT.WGS.SUSS.G\*
* Output file : GNCP.NONX.GNC37ADD.GNC3743.WGS.FRPT.G\*

1. STEPP080: GNC3744 EHB request reports file processing.

* Input file : GNCP.NONX.GNC37ADD.GNC9843.WGS.RESP.G\*
* Output file : GNCP.NONX.GNC37ADD.GNC3744.WGS.RRPT.G\*

1. STEPU090: PSUTDUMP route the success report to dispatch.
2. STEPU100: PSUTDUMP route the failure report to dispatch.
3. STEPU110: PSUTDUMP route the request report to dispatch.
4. ABENDCHK: ABEND routine.

**SPIDER:**

The following job is executed: **GNC37HDD**

The job is explained in detail below.

**Job GNC37HDD**

**Purpose:** To process the feed file from SPIDER.

**Input file:** GNCP.NONX.GNC37HDD.WGS.FEED.G\*

**Output files:** GNCP.NONX.GNC37HDD.GNC9843.WGS.RESP.G\*

GNCP.NONX.GNC37HDD.GNC3742.WGS.SRPT.G\*

GNCP.NONX.GNC37HDD.GNC3743.WGS.FRPT.G\*

GNCP.NONX.GNC37HDD.GNC3744.WGS.RRPT.\*G

**Executing Proc GNC370DD**

1. STEPP005: GNC9846 benefit feed file validation program.

* Input file : GNCP.NONX.GNC37HDD.WGS.FEED
* Output file : GNCP.NONX.GNC37HDD.WGS.CNTLFIL.G\*

1. STEPS010: SORT splitting the feed file to remove the header record.
2. STEPS015: SORT step to merge the new header with feed file.
3. STEPU017: IDCAMS step to set the return code 08.
4. STEPU020: IEBGENER take the back up of benefit feed file.
5. STEPU025: IEFBR14 to allocate the GSAM file.
6. STEPS026: SORT to split the header, detailed and trailer records.
7. STEPS027: SORT the detail records based on start date and coverage plan id.
8. STEPS028: SORT to merger the header sorted detail and trailer records.
9. STEPP030: GNC9843 EHB benefit feed file processing program.

* Input file :

GNCP.NONX.GNC37HDD.WGS.SOUT.G\*

* Output file : GNCP.NONX.GNC37HDD.GNC9843.WGS.RESP.G\*

1. STEPS050: SORT step for sorting the files based on status code.

* Input file : GNCP.NONX.GNC37HDD.GNC9843.WGS.RESP.G\*
* Output files :

GNCP.NONX GNC37HDD.SORT.WGS.SUSS.G\*

GNCP.NONX GNC37HDD.SORT.WGS.FAIL.G\*

1. STEPP060: GNC3742 EHB success reports file processing.

* Input file : GNCP.NONX GNC37HDD.SORT.WGS.SUSS.G\*
* Output file : GNCP.NONX.GNC37HDD.GNC3742.WGS.SRPT.G\*

1. STEPP070: GNC3743 EHB failure reports file processing.

* Input file : GNCP.NONX GNC37HDD.SORT.WGS.SUSS.G\*
* Output file : GNCP.NONX.GNC37HDD.GNC3743.WGS.FRPT.G\*

1. STEPP080: GNC3744 EHB request reports file processing.

* Input file : GNCP.NONX.GNC37HDD.GNC9843.WGS.RESP.G\*
* Output file : GNCP.NONX.GNC37HDD.GNC3744.WGS.RRPT.G\*

1. STEPU090: PSUTDUMP route the success report to dispatch.
2. STEPU100: PSUTDUMP route the failure report to dispatch.
3. STEPU110: PSUTDUMP route the request report to dispatch.
4. ABENDCHK: ABEND routine.

**FTP/NDM:**

All the Legacy system should send and receive the feed/acknowledgement in EHB format.

Each Legacy system should share the time/frequency and days on which feed files will be sent. Based on this data WGS claims system will schedule the jobs.

For CS90 and Health link, the feed file is received and the response file is send to the legacy system via FTP. For CHIPS, ACES**,** FACETS and NASCO, the feed file is received and the response file is send to the legacy system via NDM. This is not applicable for WGS and SPIDER.

**Benefit feed load and reports job scheduling flow**



****

**EyeMed Synch Process**

The scope of this system is to keep the accumulators in sync in ODS DB for EyeMed system. Eyemed System receives the VIS claims from different Eligibility systems lik ISG, CS90, ACES, etc., Eyemed sends the claims details to WGS to store the accum details in ODS. WGS receives the EyeMed VIS Claims accum data in request file, performs validations, updates ODS DB and send the response of the accum update in ODS to EyeMed system. The updated Accum report is sent to the corresponding Eligibility systems using Eligibility Src code by the Backward Sync process.

The step by step process for EyeMed synch is as follows,

* Receive the EYEMED request file and FTP to WGS system in a standard EBS Layout.
* Calls the Source module GNC9420 to read the request file and perform the below functions using EBSRS routine.
* GNCEBSRS will validate the input data fields by calling GNCEBS4S module; validate the contract data fields by calling the Benefit translational table (Contr\_Lev\_info Table) using GNCEBS5S module.
* For processing the records with valid contract, GNCOD03S module is called for processing the Accum.
* GNCOD03S calls Benefit translational table (Accum\_Lev\_Info Table) to derive the actual accum name, corresponding Indicators, and Accum’s Benefit Max Amount.
* The Derived Actual Accum name (Primary Accum) is validated against GNCSACMT table base to decide the accum is IND or FAM.
* For Individual Primary Accum its corresponding Family Accum will also be treated as a Primary Accum and will be updated to ODS.
* The Benefit Max Amount for the Accum is decided based on the Embedded/Non Embedded indicator in the Contr\_Lev\_info table.
  + For an Embedded contract, the individual accumulators need to be updated till the individual maximum amount is met to set the over-apply indicator**;** and Family accumulators are to be updated till their family maximum amount before the over apply indicator is set.
  + For a non-embedded contract, both the individual and family accumulators are to be updated till their family accumulator maximum amount before the over-apply indicator is set.
* Update the Primary accum in Accum ODS DB with Prod Type ‘VIS’.
* When Adult Indicator is set in the request and Adult OOP Exclude Indicator is set in the Benefit translational table, ODS DB will the updated with zero amount, and with accum reason text as **'ODS/WGS RECEIVED $ XX.00 BUT UPDATE BYPASSED DUE TO ADULT OOP EXCLSN. $ 0.00 APPLIED FOR TRACKING ONLY.'**
* Special Functionalities,
  + For CS90, NASCO, Health Link, Facets – Patient Member code is prefixed with 1 before inserting to ODS DB.

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | **EYEMED** | |
|  | **Request Member code** | **TOTL** | **DTL** |
| **IND** | **00** | **100** | **100** |
| **FAM** | **00** | **00** | **100** |

Eg:

* + For ACES, FACETS- Spaces will be populated to coverage plan ID while inserting into DB.
* When the LQCO indicator is set for the contract, LQ and CO accums for the primary accums are updated when the service date falls in the last quarter of the benefit period.
* **Secondary Accum update for Primary Accums.**
  + When **DED\_IN\_OOP** indicator is set for the contract, DED amount is updated to OOP accum

|  |  |
| --- | --- |
| **Type of Accum** | **Valid DB values** |
| Ded-IN-OOP-IND | A - Preferred PAR, PAR and NON PAR DED in OOP. B - Both PAR and NON PAR DED in OOP P - Only PAR DED included in PAR OOP N - Only NON PAR DED included in NON PAR OOP S - Only Preferred PAR DED included in OOP. T - Only Preferred PAR and Regular PAR DED included in OOP. U - Only Preferred PAR and NON PAR DED included in OOP. X - DED is not included in OOP. |

* + When **Cross Apply** Indicator is set for the contract, PAR accums is crossapplied to NONPAR, or NONPAR accum crossapplied to PAR based on the indicator.

|  |  |
| --- | --- |
| **Type of Accum** | **Valid DB values** |
| DED-IN-to-OUT-IND | Y - PAR DED applies to NON PAR S - PAR DED applies to Preferred PAR B - PAR DED applies to Preferred both PAR and NON PAR X - PAR DED does not cross apply |
| DED-OUT-to-IN-IND | Y - NON PAR DED applies to PAR S - NON PAR DED applies to Preferred PAR B - NON PAR DED applies to Preferred PAR and PAR X - NON PAR DED does not cross apply |
| OOP-IN-to-OUT-IND | Y - PAR OOP applies to NON PAR S - PAR OOP applies to Preferred PAR B - PAR OOP applies to both Preferred PAR and NON PAR X - PAR OOP does not cross apply |
| OOP-OUT-to-IN-IND | Y - NON PAR OOP applies to PAR S - NON PAR OOP applies to Preferred PAR B - NON PAR OOP applies to both Preferred PAR and PAR X - NON PAR does not cross apply |

* + When **Preferred PAR Cross Applied** Indicator is set for the contract, the accums are crossapplied based on the indicator. Valid Indicator Values are as below.

|  |  |
| --- | --- |
| **Type of Accum** | **Valid DB values** |
| Preferred Deductible Accum | P - Preferred PAR DED applies to Regular PAR |
| N - Preferred PAR DED applies to NON PAR |
| B - Preferred PAR DED applies to both Regular PAR and NON PAR |
| X - Preferred PAR DED does not cross apply |
| Preferred Out of Pocket Accum | P - Preferred PAR OOP applies to Regular PAR |
| N - Preferred PAR OOP applies to NON PAR |
| B - Preferred PAR OOP applies to both Regular PAR and NON PAR |
| X - Preferred PAR OOP does not cross apply |

* + When Dual update indicator is set for the primary accums, Dual update is only carried out for DED/OOP accums. It is handled only for PPO contract and PPO accums of POS contract.
* **All the secondary Accums will be updated in the Accum ODS DB with the Prod Type ‘VIS’.**
* When the LQCO indicator is set for the contract, LQ and CO accums for the secondary accums are updated when the service date falls in the last quarter of the benefit period.
* The response file with status return code is created based on the ODS Update status.
* The response file is send back to EyeMed system as FTP file.

**Component Details:**

**GNC9420 SRC:-**

* Receives Eyemed file in EBS layout.
* Populate header fields and trailer fields.
* If claim source and Elig Src code is received as alphanumeric, it is converted to corresponding numeric Src code using GNCSRCDT tablebase.
* Checks for mandatory filed for the below fields. Else return the corresponding status codes and ends transaction
* Claim Src code – 893 (returns status code 32 for Source code invalid)
* Prod type – ‘VIS’ (returns status code 33 for product type invalid)
* Retrieval function-0004 (returns status code 34 for invalid function)
* Calls GNCEBSRS for detail record processing
* Writes the response layout to EBS Response Area.

**GNCEBSRS:-**

* GNCEBS4S is called for data validation and mandatory check validation (returns status code 22 for Missing data, 23 for data exception)



* GNCEBS5S is called for Contract validation by passing below key values to GNCBNIOS (return status code 25 for contract not found)
  + IN-ELIG-SRC-CD
  + IN-SRVC-STRT-DT
  + IN-SRVC-END-DT
  + IN-CVRG-PLAN-ID
  + IN-MBR-TIER-LVL
* GNCOD03S is called for below Eyemed accum update process.

**GNCOD03S:-**

**Primary Accum processing**

* Derives the Primary accum to be updated to ODS by calling GNCEBS5S by passing below key values (return status code 91 –accum invalid if accum not found)
  + IN-ELIG-SRC-CD
  + IN-SRVC-STRT-DT
  + IN-SRVC-END-DT
  + IN-CVRG-PLAN-ID
  + IN-MBR-TIER-LVL
  + IN-LMTN-ACM-NM – (Valid Values – ‘DED’ or ‘OOP’)
  + IN-ACM-TYPE
  + IN-NTWK-CD

If the accum is invalid, response is populated with call status and transaction is ended.

* When the Primary accum is an Individual accum, its corresponding Family accum is also updated along with the secondary accum of the family accum.

**Embedded and Non-Embedded contracts**

* The Benefit Max Amount for the Accum is decided based on the Embedded/Non Embedded indicator in the Contr\_Lev\_info table.
  + For an Embedded contract, the individual accumulators need to be updated till the individual maximum amount is met to set the over-apply indicator**;** and Family accumulators are to be updated till their family maximum amount before the over apply indicator is set.
  + For a non-embedded contract, both the individual and family accumulators are to be updated till their family accumulator maximum amount before the over-apply indicator is set.

**Date Derivation Logic**

* Benefit effective date and Benefit End date to be updated to ODS is derived based on Service Date and the accum level Benefit Effective date from the request file.
  + If Accum level Benefit Start Date is CCYY-01-01 format is considered as CY.
  + If Accum level Benefit Start Date is in CCYY-MM-DD format it is considered as BY.
  + MMDD from the Accum level Benefit start date and the CCYY of Service date used to update the Benefit effective date and Benefit End date in the ODS DB.

**Duplicate Check**

* + No partial update is supported for Eyemed.
  + **Scenario 1:**
* First record is duplicate of Second record, First record is inserted to ODS DB with Call-status 01(return Code 01). Second Record is rolled back with call-status 36(return code 36-Duplicate record).
  + **Scenario 2:**
* First record has 3 accums array.
  + If third accum is duplicate of first accum, all the inserted accums for the record are rolled back with overall call-status 36 (return code 36-Duplicate record).

If the Duplicate record is found, response is populated with call status and transaction is ended.

**Over Apply check**

* Over-apply is checked for primary accums for amount, days and occurs if values are greater than zeros in benefit translational table.
  + If the total amount + In-request amount > Benefit amount, Positive Over apply is set for the accum. (Accum Level return Code – 05, Overall call status - 31)
  + If total amount + In-request amount < Zeroes, Negative Over apply is set for the accum.(Accum Level return code – 06, Overall call status - 31)
  + If Maximum Benefit Occurs > zeroes,
    - Total Occurs + In-request occurs > Max Benefit Occurs, Positive Over apply is set for the accum. (Accum Level return Code – 05, Overall call status - 31)
  + If Maximum Benefit Days > zeroes,
    - Total Days + In-request Days > Max Benefit Days, Positive Over apply is set for the accum. (Accum Level return Code – 05, Overall call status - 31)

**Update Primary Accum**

* Primary Accums are updated to ODS tables with primary accum.
  + Only if the Accum is VIS Comingled, the accums get inserted to the ODS DB. If not comingled, return code 24-Benefit not found is passed.
  + Special Functionalities,
* For CS90, NASCO, Health Link, Facets – Patient Member code is prefixed with 1 before inserting to ODS DB.

Eg:

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | **EYEMED** | |
|  | **Request Member code** | **TOTL** | **DTL** |
| **IND** | **00** | **100** | **100** |
| **FAM** | **00** | **00** | **100** |

* For ACES, FACETS- Spaces will be populated to coverage plan ID while inserting into DB
* When Adult Indicator is set in the request and Adult OOP Exclude Indicator is set in the Benefit translational table, ODS DB will the updated with zero amount, and with accum reason text as 'ODS/WGS RECEIVED $ XX.00 BUT UPDATE BYPASSED DUE TO ADULT OOP EXCLSN. $ 0.00 APPLIED FOR TRACKING ONLY.'

**Adult OOP Exclusion**

* When Adult Indicator is set in the request and Adult OOP Exclude Indicator is set in the Benefit translational table, ODS DB will the updated with zero amount, and with accum reason text as 'ODS/WGS RECEIVED $ XX.00 BUT UPDATE BYPASSED DUE TO ADULT OOP EXCLSN. $ 0.00 APPLIED FOR TRACKING ONLY.'

**LQCO Logic**

* + LQCO logic is followed for EyeMed.
* Applicable for both BY and CY contracts so no specific checks required.
* Applicable only for DED and OOP regular shared/non-shared accums
  + Does not apply for Special/upfront DED and OOP accums
* LQ and CO accums will be updated in ODS DB along with Regular (primary accum) only if –
  + Service date falls in the last quarter
  + LQCO indicator value from benefit table is any of the below for the contract:
    - B - LQ applicable for both DED and OOP PAR and NON PAR
    - P - LQ applicable for both DED and OOP PAR
    - N - LQ applicable for both DED and OOP NON PAR
    - D - LQ applicable only for DED PAR & NON PAR
    - O - LQ applicable only for OOP PAR & NON PAR
    - E - LQ applicable only for DED PAR
    - F - LQ applicable only for DED NON PAR
    - Q - LQ applicable only for OOP PAR
    - R - LQ applicable only for OOP NON PAR
* LQ accums will be updated for Current Year Benefit period and CO accums will be updated for the next year benefit period
* Over-Apply status indicator:
  + LQ Accum – over-apply indicator value will be same as regular (primary) accum
  + CO Accum – over-apply indicator calculation will be done using the benefit MAX amount retrieved for the accum for the next year benefit period.
* CO Accums will update only when the accum is VIS comingled for the next year benefit period. Else LQ accum will be updated in ODS DB along with regular primary Accum.
* Populate Response for Primary Accum based on the status of the accum update to ODS.
* Rollback and transaction end is performed only for below scenarios.
  + Duplicate accum found
  + Accum found invalid
  + Benefit not found
  + DB unavailable
  + Table base error

**Secondary Accum Processing**

For EyeMed, we have handled secondary accum update.

Design followed as of now is

* IF DED-IN-OOP indicator is set, GNCOD02S module is called for secondary accum derivation for the corresponding primary accum.

|  |  |
| --- | --- |
| **Type of Accum** | **Valid DB values** |
| Ded-IN-OOP-IND | A - Preferred PAR, PAR and NON PAR DED in OOP. B - Both PAR and NON PAR DED in OOP P - Only PAR DED included in PAR OOP N - Only NON PAR DED included in NON PAR OOP S - Only Preferred PAR DED included in OOP. T - Only Preferred PAR and Regular PAR DED included in OOP. U - Only Preferred PAR and NON PAR DED included in OOP. X - DED is not included in OOP. |

* If Cross apply indicators are set, GNCOD02S module is called for secondary accum derivation for the corresponding primary accum.

|  |  |
| --- | --- |
| **Type of Accum** | **Valid DB values** |
| DED-IN-to-OUT-IND | Y - PAR DED applies to NON PAR S - PAR DED applies to Preferred PAR B - PAR DED applies to Preferred both PAR and NON PAR X - PAR DED does not cross apply |
| DED-OUT-to-IN-IND | Y - NON PAR DED applies to PAR S - NON PAR DED applies to Preferred PAR B - NON PAR DED applies to Preferred PAR and PAR X - NON PAR DED does not cross apply |
| OOP-IN-to-OUT-IND | Y - PAR OOP applies to NON PAR S - PAR OOP applies to Preferred PAR B - PAR OOP applies to both Preferred PAR and NON PAR X - PAR OOP does not cross apply |
| OOP-OUT-to-IN-IND | Y - NON PAR OOP applies to PAR S - NON PAR OOP applies to Preferred PAR B - NON PAR OOP applies to both Preferred PAR and PAR X - NON PAR does not cross apply |

* If Preferred PAR indicators are set, GNCOD02S module is called for secondary accum derivation for the corresponding primary accum.

|  |  |
| --- | --- |
| **Type of Accum** | **Valid DB values** |
| Preferred Deductible Accum | P - Preferred PAR DED applies to Regular PAR |
| N - Preferred PAR DED applies to NON PAR |
| B - Preferred PAR DED applies to both Regular PAR and NON PAR |
| X - Preferred PAR DED does not cross apply |
| Preferred Out of Pocket Accum | P - Preferred PAR OOP applies to Regular PAR |
| N - Preferred PAR OOP applies to NON PAR |
| B - Preferred PAR OOP applies to both Regular PAR and NON PAR |
| X - Preferred PAR OOP does not cross apply |

* The secondary accum is validated in Benefit translational table.
* The derived secondary accums are updated to ODS table.
  + Only if the secondary accum is VIS Comingled, the accums get inserted to the ODS DB.

**Dual Accum Processing**

Logic Followed is:

The Dual Update is handled only for WGS/ISG non-spider member and is determined using the Indicator in the Contract DB (CONTR\_LEV\_INFO) - HIX\_EHB\_IND = ‘Y’

* It is only for DED/OOP accums
* It is only for PPO contract and PPO accums of POS contract.
* If the contract is PPO shared, then regular non-shared accums will be updated based on the Input Claim Network
* If the contract is PPO non-shared, then regular shared accum will be updated irrespective of the network indicator
* If the contract is POS then dual update will be done only when the input claim network indicator is NPAR (as the PAR network would be HMO accums) – same logic as a. and b. would be followed.
* Benefit DB validations are not required for the Dual Update derived accums.

So the over-applied indicator will be always ‘N’. And MAX BNFT amount in the TOTL DB will be ‘0.00’.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Type of contract** | **IN/OUT network** | **Primary Accum** | **Claim Network** | **Secondary accum** |
| PPO | Shared Contract | INDDED | PAR | PINDED |
| PPO | Shared Contract | INDDED | NPAR | NINDED |
| PPO | Shared Contract | FAMDED | PAR | PFMDED |
| PPO | Shared Contract | FAMDED | NPAR | NFMDED |
| PPO | Non-shared contract | PINDED | N/A | INDDED |
| PPO | Non-shared contract | NINDED | N/A | INDDED |
| PPO | Non-shared contract | PFMDED | N/A | FAMDED |
| PPO | Non-shared contract | NFMDED | N/A | FAMDED |
| POS | Shared Contract | INDDED | NPAR | NINDED |
| POS | Shared Contract | FAMDED | NPAR | NFMDED |
| POS | Non-shared contract | NINDED | N/A | INDDED |
| POS | Non-shared contract | NFMDED | N/A | FAMDED |

* Dual Accums are updated to ODS tables with primary accum.
  + Only if the Accum is VIS Comingled, the accums get inserted to the ODS DB.

**GNC945DD JCL:-**

Existing job is modified to execute new program to read the file from Eyemed system and update the ODS DB and send back the response to corresponding system using below step,

Calls program : GNC9420

Input File : EYEMED Request File (with data only in Request Header and Request Area)

Output Files : Response file generated.

Job is modified to execute new Report program GNC9404

Calls program : GNC9404 for report generation

Input File : Response File Generated

Output File : Success Report

Failure Report

**DISPLATCH REPORTS**

Success Report Dispatch ID: GNC94009

Failure Report Dispatch ID : GNC94010

**Process Flow**

****

**ESI – Real time Transaction**

The ESI system should interact with Accum ODS table for accum update and inquiry while doing Pharmacy (RX) claims processing for EHB Contracts. The interaction with ODS will be real-time using the services build by SOA team. The request message from ESI system will be received by the real time services and parses the data to the standard accum update/inquiry format (EBS Layout) for all setups that ESI uses Accum ODS for. This request will be routed to an interface EBS module after setting appropriate function as per the request (Inquiry/Update etc.). The EBS routine will do data validation - mandatory field’s validation, contract validation, process the request and return the call status and other required fields as response. The response code returned from EBS module will be mapped back to the return codes for ESI by the services modules.

Note:

1. ESI sends accum names using 6 bytes numeric Generic name for POST transaction. Please refer ‘Generic Accum Name Mapping’ for how it is derived and loaded in Benefit ODS.
2. Always, the ESI post transaction for Original Claim (EA) will be preceded by ESI Inquiry Transaction to retrieve the valid Generic accum names from the benefit ODS.
3. The ESI post transaction for Reversal Claim (ER) will not be preceded by ESI Inquiry as the reversal process at ODS end will reverse all the accums loaded for Original Claim irrespective of the Accums sent in the request.
4. Duplicate check will be performed for primary accum or the accum sent in the request only when the RESEND counter is greater than ‘0’.

**Benefit Translation Table Lookup Process:-**

The Benefit Translation table consists of a Contract Level Info Table and Accum Level Info Table. The Contract Info table has the contract related details including the contract indicatives. Accum Info table will have the details of all valid accums for a particular contract. The Contract Info Table and Accum Info Tables are linked using foreign key relation to tag the list of valid accums to the corresponding contract code. For benefit translation table lookup the IO routine for Benefit Translation tables will be called setting appropriate function and passing the following fields from input request

* Eligibility Source Code
* Contract Code
* Service Start Date
* Service End Date
* Member Tier Indicator

Using the above fields, the IO routine will check the Benefit details present in Contract Info Table with Benefit Start Date Less than or Equal to the Service Start Date and Benefit End Date Greater than the Service End Date. When a hit is found for this, the corresponding contract details will be fetched from the Contract Info Table. This will include details like embedded indicator, Deductible in OOP indicator, last Quarter indicator etc.

**Inquiry Transaction:**

Transaction Name - WCXGACM

Service Name - getMemberAccum

**Accum Inquiry Request and Response**

The fields required for accum Inquiry request and the fields validation are as given below,

**GNCEBSRS – SRC**

The data and mandatory field validation will be performed for the fields given above. The inquiry request would have a minimum of single accum record. ESI performs inquiry transaction to retrieve the valid accum names and so the request comes from ESI without any accum names populated BUT since the service layer has a validation that we need at least one accum name, ESI populates a default value ‘ACM1’.

For the records which passed data validation, the contract will be validated against the benefit translation tables.

**GNCOD07S - SRC**

Using Source Code, Contract Code, Member Tier Level, Benefit Start and End dates and the Version Number from Contract Info table as Key, the Accum Info Table will be queried for fetching all available Accums for the combination for RX comingled indicator ‘Y’ and the generic accum name not spaces (value should be populated). All the accum related details will be fetched from Accum Info Table.

**Remaining Amount Calculation:-**

With the below key fields EBS will fetch the accum data from ODS table, for each accum retrieved from the Benefit Translation tables only if the RX commingled indicator is ‘Y’ for the accum.

* HCID
* Member Code
* Case ID
* Benefit Effective Date
* Accum Name
* Accum Suffix Name
* Contract Code

The Remaining Amount, Remaining Occurrence and Days will be calculated by subtracting the total benefit amount, occurrence and days fetched from ODS total table for the particular accum from the Accum Limit, Maximum Occurrence and Maximum Days retrieved from Accum Info table for this accum.

**Populating Call Status:-**

The fetched details will be populated into the accum array. When processing of one accum record is complete, the corresponding call status from EBS will be updated in the OUT-TOT-REC-STS field. The valid accum level status code for inquiry request will be as follows,

01 – Call Success

EBS will process all the accum records in the input request and populate the response code accordingly. Once the processing is complete for all the accum records, the over-all call status will be updated in the response area. The following are valid transaction level call status for inquiry request,

|  |  |
| --- | --- |
| Over-all Call status | Description |
| 01 | Call Successful - This status will be returned when the transaction completed Successfully |
| 21 | Database not available - This status will be returned when the status of the DB call is not successful (This is applicable for the Benefit translation tables, Accum ODS Detail and Accum ODS Total Tables) |
| 22 | Missing Data - This status will be returned when the transaction failed due to mandatory inputs missing |
| 23 | Data Exception - This status will be returned when the transaction failed due to data format issues in input or invalid data in the mandatory fields like source code, function code etc. |
| 02 | **Contract not found** - This status will be returned and the transaction will get failed for the below conditions **(internal status code will be '25' but sent to ESI as '02'):** 1. When the Benefit ID (contract code) in the request is not present in benefit translation table. 2. When no record exists in the benefit translation table for the Benefit Effective and Benefit End dates derived based on the Service Start and Service End dates in the request. **Commingled accum not found** – This status will be returned when none of the accums retrieved from the Accum Info Table is commingled with Pharmacy benefit (RX). |

**Populating Response Area**

* XXX-OUT-CLM-NBR, XXX-OUT-CLM-SRC-CD, XXX-OUT-MBR-HCID, XXX-OUT-PAT-MBR-CD, XXX-OUT-CASE-NBR, XXX-OUT-GRP-NBR, XXX-OUT-CVRG-PLAN-ID – populated from its corresponding input/request fields
* XXX-OUT-NUM-OF-TOTAL-RECS – number of accums returned in response
* XXX-RSP-MORE-DATA-FLAG – will be set to ‘Y’ if more than 12 accums are present under the input/requested contract
* XXX-OUT-LMTN-ACM-NM – will be generic accum name retrieved from GNC.ACCUM\_LEV\_INFO (GEN\_ACCUM\_NM field)
* XXX-OUT-BNFT-EFCTV-DT – will be used this from the input transaction request - XXX-IN-BNFT-EFCTV-DT (1) - 1st occurrence as this is mandatory field
* XXX-OUT-BNFT-TRMNTN-DT – will be used this from the input transaction request - XXX-IN-BNFT-TRMNTN-DT (1) - 1st occurrence as this is mandatory field
* XXX-OUT-TOT-BNFT-AMT, XXX-OUT-TOT-BNFT-OCC, XXX-OUT-TOT-BNFT-DAY – will be populated using the TOT values from GNC.ACCUMR\_TOTL DB
* XXX-OUT-TOT-REMAN-BNFT-AMT, XXX-OUT-TOT-REMAN-BNFT-OCC, XXX-OUT-TOT-REMAN-BNFT-DAY – will be derived as below using TOT values retrieved from TOTL DB and the Benefit MAX values retrieved from ACCUM\_LEV\_INFO table.
* XXX-OUT-TOT-REMAN-BNFT-AMT = GNCBNIOS-ACM-LMT-AMT – XXX-OUT-TOT-BNFT-AMT
* XXX-OUT-TOT-REMAN-BNFT-OCC = GNCBNIOS-ACM-OCR-CNT – XXX-OUT-TOT-BNFT-OCC
* XXX-OUT-TOT-REMAN-BNFT-DAY = GNCBNIOS-ACM-DYS-CNT – XXX-OUT-TOT-BNFT-DAY
* XXX-OUT-TOT-ACM-TYPE – will be populated using the GENERIC accum name from the ACCUM\_LEV\_INFO table.
* If the last two bytes of the generic accum name (which defines the type of accum) is ‘01’ then the accum type will be updated as Individual and if it is ’02’ then family will be updated.
* XXX-OUT-TOT-INN-OON-CD – will be populated only when received in input/request (XXX-IN-NTWK-CD)
* XXX-OUT-TOT-REC-STATUS – will be always ‘01’ as the accum response will be populated only from the ACCUM\_LEV\_INFO table for RX comingled indicator set to Y.
* XXX-OUT-ACM-SHR-STAT – will be always set to ‘Y’ for all the accums in the response as the only RX comingled accum will be retrieved from ACCUM\_LEV\_INFO table.

**Update Transaction:**

Transaction Name - WCXAACM

Service Name - addMemberAccum

**Accum Update Request and Response**

The fields required for accum Inquiry request and the fields validation are as given below,

**GNCEBSRS – SRC**

The data and mandatory field validation will be performed for the fields given above. The update request should have the total number of accum records for update in the field IN-TOT-RECS in the request area. Also the accum array in the request area would not have the actual accum name (as stored in the ODS tables) and hence need to be derived as per the doc in the reference section.

For the records which passed data validation, the contract will be validated against the benefit translation tables

**GNCOD06S – SRC**

Then with the same Eligibility Source Code, Contract Code, Member Tier Level, Version Number, Benefit Start and End dates from Contract Info table as Foreign Key and the Accum Name from Input request as Primary Key, the Accum Info Table will be queried for Accum Entry. If a hit is found, the accum in the input request is valid for the particular contract during the service date period.

**Updating Accums and Populating Call Status:-**

If all validations are success, EBS will call IO routine for ODS to add accum data to ODS tables, passing all the fields from input request. The record will be inserted in accum detail table under the product type ‘RX’. After accum detail table insert, the accum total table will be inserted/updated with accum data based on key fields with product type ‘RX’ and ‘TOT’ for the accum name in the input request.

While updating the ODS total table, the total accumulated amount under product type ‘TOT’ will be compared against the accum limit fetched from benefit translation table lookup. If the total accumulated amount exceeds the accum limit, the over-applied indicator in ODS DTL and TOTL table will be set to ‘Y’.

After accum update, the details will be populated into the accum array. When processing of one accum record is complete, the corresponding call status from EBS will be updated in the OUT-TOT-REC-STS field. The valid accum level status code for add/update request will be as follows,

01 - Accum update Success

05 - Accum update Success with Over-Apply

06 – Success with negative amount

36 – Duplicate accum update (accum level) and 01 in overall transaction level

91 - Accum update failed as Accum is invalid – This status will be returned when the accum retrieved from the Accum Info Table is not commingled with Pharmacy benefit (RX).

|  |  |
| --- | --- |
| **Over-all/transaction Call status** | **Description** |
| 01 | **Call Successful** - This status will be returned when the transaction completed Successfully **Duplicate record found** (Accum level status will be '36' but sent to ESI as success call '01') -This status will be returned and the transactions will get failed when the values in fields like HCID, Member code, Case Number, Benefit start and end date, Coverage plan ID, Claim Number, Service Start and Claim Source Code, End dates, Product Type code, Accum Name, Accum amount/occurs/days, Accum Suffix Name, Accum Reason code and Network Code are already available in the Accum ODS. ODS executes the Duplicate check only when resend counter > 0 |
| 02 | If the original Claim is not found for the Reversal process (when the ACM RSN code is ER or RV) |
| 05 | Accum update Success with Over-Apply |
| 06 | Success with negative amount |
| 21 | Database not available - This status will be returned when the status of the DB call is not successful |
| 22 | Missing Data - This status will be returned when the transaction failed due to mandatory inputs missing |
| 23 | Data Exception - This status will be returned when the transaction failed due to data format issues in input or invalid data in the Mandatory fields. |
| 25 | Contract not found - This status will be returned and the transaction will get failed for the below conditions and none of the accums will be loaded in the ODS. 1. When the Benefit ID (contract code) in the request is not present in benefit translation table. 2. When no record exists in the benefit translation table for the Benefit Effective and Benefit End dates derived based on the Service Start and Service End dates in the request. |
| 91 | Accum is invalid – This status will be returned and the transaction will get failed for the below conditions and none of the accums will be loaded in the ODS. 1. When any of the accum that received in the POST request is not available in Benefit table for that specific Benefit ID that comes in the request 2. When any of the accum that received in the POST request is not commingled with Pharmacy benefit (RX) |

EBS will process all the accum records in the input request and populate the response code accordingly. The following are valid transaction level call status for update request

**Note: If any of the accumulators in the request is not valid (based on RX commingled indicator), all the previous updates for other accums in the same request will be rolled back. No partial updates will be done.**

**Populating Response Area**

* XXX-OUT-CLM-NBR, XXX-OUT-CLM-SRC-CD, XXX-OUT-MBR-HCID, XXX-OUT-PAT-MBR-CD, XXX-OUT-CASE-NBR, XXX-OUT-GRP-NBR, XXX-OUT-CVRG-PLAN-ID – populated from its corresponding input/request fields
* XXX-OUT-NUM-OF-TOTAL-RECS – number of accums should be same as input count - XXX-IN-TOT-RECS
* XXX-RSP-MORE-DATA-FLAG – will be always ‘N’ as more than 12 accums are not expected in the request for update transaction.
* XXX-OUT-LMTN-ACM-NM – will be generic accum name from the input/claim request - XXX-IN-LMTN-ACM-NM
* XXX-OUT-BNFT-EFCTV-DT – will be used this from the input transaction request for each accum - XXX-IN-BNFT-EFCTV-DT
* XXX-OUT-BNFT-TRMNTN-DT – will be used this from the input transaction request for each accum - XXX-IN-BNFT-TRMNTN-DT
* XXX-OUT-TOT-BNFT-AMT, XXX-OUT-TOT-BNFT-OCC, XXX-OUT-TOT-BNFT-DAY – will be populated from input XXX-IN-BNFT-DTL-AMT, XXX-IN-BNFT-DTL-OCC and XXX-IN-BNFT-DTL-DAY
* XXX-OUT-TOT-REMAN-BNFT-AMT, XXX-OUT-TOT-REMAN-BNFT-OCC, XXX-OUT-TOT-REMAN-BNFT-DAY – will be derived as below using TOT values retrieved from TOTL DB, DTL amount (claim amount) from the request and the Benefit MAX values retrieved from ACCUM\_LEV\_INFO table.
* XXX-OUT-TOT-REMAN-BNFT-AMT = GNCBNIOS-AC-LMT-AMT – (TOTL\_BNFT\_AMT + XXX-OUT-TOT-BNFT-AMT)
* XXX-OUT-TOT-REMAN-BNFT-OCC = GNCBNIOS-AC-LMT-OCC – (TOTL\_BNFT\_OCR\_CNT + XXX-OUT-TOT-BNFT-OCC)
* XXX-OUT-TOT-REMAN-BNFT-DAY = GNCBNIOS-AC-LMT-DAY – (TOTL\_BNFT\_DAY\_CNT + XXX-OUT-TOT-BNFT-DAY)
* XXX-OUT-TOT-ACM-TYPE – will be populated from input/claim request (XXX-IN-ACM-TYPE)
* XXX-OUT-TOT-INN-OON-CD – will be populated from input/request (XXX-IN-NTWK-CD)
* XXX-OUT-TOT-REC-STATUS – will be populated based on different scenarios
  + 01 - Accum update Success
  + 05 - Accum update Success with Over-Apply
  + 06 – Success with negative amount
  + 36 – Duplicate accum update
  + 91 - Accum update failed as Accum is invalid – This status will be returned when the accum retrieved from the Accum Info Table is not commingled with Pharmacy benefit (RX).
* XXX-OUT-ACM-SHR-STAT – will be always set to ‘Y’ for all the accums in the response as the only RX comingled accum will be processed from ACCUM\_LEV\_INFO table and even if one accum is not present in BENEFIT table or RX not comingled then the entire transaction will be rejected.

**Accumulator Reversal Process**

When a reversal request (Reason code - XXX-IN-ACM-RSN-CD – **‘ER’** or **‘RV’**) for an accum comes in from ESI, the reversal is done for all the accums that were already updated to the ODS DB for the Claim without considering the accumulators or the network type in the ER request.

The following steps will be done.

* Check the existence of records for the same HCID, Member Code, Claim Number, Service start and end date in the ODS DTL DB – GNC.ACCUMR\_DTL.
* All the existing records for the above criteria are extracted from ODS DTL DB.
* These records are grouped based on accum and the contract - and its amounts are added.
* For example, if the DED accum PINDED has 4 records with amounts 80$, 20$, 100$ and 20$, then adding all the amounts gives a value $120 for PINDED
* These grouped records are reversed by negating the total amount for each accum (which would be ‘-$120’ for the above example) and inserted into the ODS DB. In the above example, an ER/RV record with -$120 will be inserted into the accum Detail DB so that the net amount becomes zero. The reason code and the user id used to update in ODS DB depends on the below scenarios
* An accum name has only a single entry with 80$ - We insert records in ODS DB with amount -80$, and this amount -80$ is passed back in the response. Accum reason code updated for the new record is ‘ER’. User id updated in DB is ESIERRQT.
* An accum name has multiple entries (say 80$,-20$) – We insert records in ODS DB with amount ‘-60$’, and this amount (-60$) is passed back in the response. Accum reason code updated for the new record is ‘ERA’, with REASON text ‘FOUND MORE THAN ONE CLAIM, AMOUNT SYSTEMATICALLY ADJUSTED’. User id is ESIERRQT.
* Note: in both the above scenarios, if the reason code is ‘RV’, then RV will be used to update in ODS DB. The above reason code ‘ER’ and ‘ERA’ is used only for input reason code ‘ER’.
* For an accum, if the recalculated amount from ODS is 0$, we do not insert records in ODS DB. Overall successful response (status code – 01) is sent back to ESI. Reason text is not populated here in the response area of the request.
* Used the new DB field TOT-MAX-AMT from the TOTL DB to calculate the over-applied indicator for these reversal accums.
* If the all retrieved accums amount is zero, then it will be considered ‘Duplicate reversal request’ and no action will be taken. The transaction status will be success ‘01’ and error text will be ‘DUPLICATE REQ FOUND’.
* If no existing records are retrieved for the requested claim number, then the reversal records are inserted with accum names as ‘FAILDED’, ‘FAILOOP’, ‘FAILCOIN’, ‘FAILCOPAY’, ‘FAILOTHER’ and ‘FAILLFTME’ for the input Deductible, Out of Pocket, Co-insurance, Co-pay, Other types (like FERTLMAX etc.) and Lifetime accums respectively. The type of accum is determined using GNCSACMT tablebase call (before reading GNCSACMT tablebase, generic accum name will be converted to Actual accum name by reading Benefit table ACCUM\_LEV\_INFO. The ODS Detail Table is also inserted with the message in the Reason Text field - ‘REVERSAL CLAIM WITHOUT ORIGINAL.SO, TRANSACTION WAS REJECTED’ and user id will be ‘ESIERXEA’.

The response at the transaction level is set to ‘02’ (Accum not found)

**Embedded and Non-Embedded contracts**

For inquiry request for embedded and Non-Embedded contracts, the remaining amounts for accumulators are determined as below:

For a single member; the remaining amount is calculated from individual accum maximum for both individual and family accums in the request.

For a family member; the remaining amount is calculated from individual accum maximum for individual accums and family accum maximum for family accums in the request.

For update request, the accums will be updated as below:

For an Embedded contract, the individual accumulators need to be updated till the individual maximum amount and family accumulators are to be updated till their family maximum amount respectively.

For a non-embedded contract, both the individual and family accumulators are to be updated till their family accumulator maximum amount before the over-apply indicator is set.

**Preferred Cross-Apply – Secondary Accum Update**

For update transaction, Preferred cross-apply is applicable for ESI system. Currently, preferred Cross-apply contracts are expected only from ACES.

If all other Legacy systems are not applicable for Preferred Cross-apply, then the DB indicator values MUST be stored as ‘X’ – which means ‘doesn’t Cross apply’

Below are the valid DB values

|  |  |
| --- | --- |
| **Type of Accum** | **Valid DB values** |
|  | P - Preferred PAR DED applies to Regular PAR N - Preferred PAR DED applies to NON PAR B - Preferred PAR DED applies to both Regular PAR and NON PAR X - Preferred PAR DED does not cross apply |
| Preferred Deductible Accum |
| Preferred Out of Pocket Accum | P - Preferred PAR OOP applies to Regular PAR N - Preferred PAR OOP applies to NON PAR B - Preferred PAR OOP applies to both Regular PAR and NON PAR X - Preferred PAR OOP does not cross apply |
|  |
| Preferred Co-Pay Accum | P - Preferred PAR Copay applies to Regular PAR N - Preferred PAR Copay applies to NON PAR B - Preferred PAR Copay applies to both Regular PAR and NON PAR X - Preferred PAR Copay does not cross apply |
|  |
| Preferred Co-insurance Accum | P - Preferred PAR Coins applies to Regular PAR N - Preferred PAR Coins applies to NON PAR B - Preferred PAR Coins applies to Regular PAR and NON PAR X - Preferred PAR Coins does not cross apply |
|  |

But based on the generic accum name derivation, contract other than regular PPO could have only preferred accum for PAR Network and Regular PPO accums for NPAR.

Came up the below mapping as we don’t have the capability to identify whether the accum is of shared/non-shared.

The below mapping is when the Preferred Deductible Accum = ‘P’ in benefit ODS. Either ‘**Derived Acm1 (shared)’** or ‘**Derived Acm2 (Non-Shared)’** will be updated in ODS.

|  |  |  |  |
| --- | --- | --- | --- |
| **Primary Acm** | **Indicator Value - P** | **Derived Acm1** | **Derived Acm2** |
| BASINDED | PRF\_PAR\_IND\_P | INDDED | PINDED |
| BASFMDED | PRF\_PAR\_IND\_P | FAMDED | PFMDED |
| EPOIDED | PRF\_PAR\_IND\_P | INDDED | PINDED |
| EPOFDED | PRF\_PAR\_IND\_P | FAMDED | PFMDED |
| SPMMIDED | PRF\_PAR\_IND\_P | INDDED | PINDED |
| SPMMFDED | PRF\_PAR\_IND\_P | FAMDED | PFMDED |
| HMINDOOP | PRF\_PAR\_IND\_P | INDOOP | IINOOP |
| HMFAMOOP | PRF\_PAR\_IND\_P | FAMOOP | FOOPPBOC |
| EPOIOOP | PRF\_PAR\_IND\_P | INDOOP | IINOOP |
| EPOFOOP | PRF\_PAR\_IND\_P | FAMOOP | FOOPPBOC |
| SPMMISL | PRF\_PAR\_IND\_P | INDOOP | IINOOP |
| SPMMFSL | PRF\_PAR\_IND\_P | FAMOOP | FOOPPBOC |
| HINDCPMX | PRF\_PAR\_IND\_P | SINDCPY | PINDCPY |
| HFAMCPMX | PRF\_PAR\_IND\_P | SFAMCPY | PFAMCPY |
| EPOICPY | PRF\_PAR\_IND\_P | SINDCPY | PINDCPY |
| EPOFCPY | PRF\_PAR\_IND\_P | SFAMCPY | PFAMCPY |
| SPMMICPY | PRF\_PAR\_IND\_P | SINDCPY | PINDCPY |
| SPMMFCPY | PRF\_PAR\_IND\_P | SFAMCPY | PFAMCPY |
| HINDCOIN | PRF\_PAR\_IND\_P | SINDCOIN | PINDCOIN |
| HFAMCOIN | PRF\_PAR\_IND\_P | SFAMCOIN | PFAMCOIN |
| EPOICOIN | PRF\_PAR\_IND\_P | SINDCOIN | PINDCOIN |
| EPOFCOIN | PRF\_PAR\_IND\_P | SFAMCOIN | PFAMCOIN |
| SPMMICOI | PRF\_PAR\_IND\_P | SINDCOIN | PINDCOIN |
| SPMMFCOI | PRF\_PAR\_IND\_P | SFAMCOIN | PFAMCOIN |

The below mapping is when the Preferred Deductible Accum = ‘N’ in benefit ODS. Either ‘**Derived Acm1 (shared)’** or ‘**Derived Acm2 (Non-Shared)’** will be updated in ODS.

|  |  |  |  |
| --- | --- | --- | --- |
| **Primary Acm** | **Indicator Value - N** | **Derived Acm1** | **Derived Acm2** |
| BASINDED | PRF\_PAR\_IND\_N | INDDED | NINDED |
| BASFMDED | PRF\_PAR\_IND\_N | FAMDED | NFMDED |
| EPOIDED | PRF\_PAR\_IND\_N | INDDED | NINDED |
| EPOFDED | PRF\_PAR\_IND\_N | FAMDED | NFMDED |
| SPMMIDED | PRF\_PAR\_IND\_N | INDDED | NINDED |
| SPMMFDED | PRF\_PAR\_IND\_N | FAMDED | NFMDED |
| HMINDOOP | PRF\_PAR\_IND\_N | INDOOP | OINOOP |
| HMFAMOOP | PRF\_PAR\_IND\_N | FAMOOP | FOOPNPOC |
| EPOIOOP | PRF\_PAR\_IND\_N | INDOOP | OINOOP |
| EPOFOOP | PRF\_PAR\_IND\_N | FAMOOP | FOOPNPOC |
| SPMMISL | PRF\_PAR\_IND\_N | INDOOP | OINOOP |
| SPMMFSL | PRF\_PAR\_IND\_N | FAMOOP | FOOPNPOC |
| HINDCPMX | PRF\_PAR\_IND\_N | SINDCPY | NINDCPY |
| HFAMCPMX | PRF\_PAR\_IND\_N | SFAMCPY | NFAMCPY |
| EPOICPY | PRF\_PAR\_IND\_N | SINDCPY | NINDCPY |
| EPOFCPY | PRF\_PAR\_IND\_N | SFAMCPY | NFAMCPY |
| SPMMICPY | PRF\_PAR\_IND\_N | SINDCPY | NINDCPY |
| SPMMFCPY | PRF\_PAR\_IND\_N | SFAMCPY | NFAMCPY |
| HINDCOIN | PRF\_PAR\_IND\_N | SINDCOIN | NINDCOIN |
| HFAMCOIN | PRF\_PAR\_IND\_N | SFAMCOIN | NFAMCOIN |
| EPOICOIN | PRF\_PAR\_IND\_N | SINDCOIN | NINDCOIN |
| EPOFCOIN | PRF\_PAR\_IND\_N | SFAMCOIN | NFAMCOIN |
| SPMMICOI | PRF\_PAR\_IND\_N | SINDCOIN | NINDCOIN |
| SPMMFCOI | PRF\_PAR\_IND\_N | SFAMCOIN | NFAMCOIN |

The below mapping is when the Preferred Deductible Accum = ‘B’ in benefit ODS. Either ‘**Derived Acm1 (shared)’** or ‘**Derived Acm2 (Non-Shared - PAR) and Derived Acm3 (Non-Shared – NPAR)’** will be updated in ODS.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Primary Acm** | **Indicator Value - B** | **Derived Acm1** | **Derived Acm2** | **Derived Acm3** |
| BASINDED | PRF\_PAR\_IND\_B | INDDED | PINDED | NINDED |
| BASFMDED | PRF\_PAR\_IND\_B | FAMDED | PFMDED | NFMDED |
| EPOIDED | PRF\_PAR\_IND\_B | INDDED | PINDED | NINDED |
| EPOFDED | PRF\_PAR\_IND\_B | FAMDED | PFMDED | NFMDED |
| SPMMIDED | PRF\_PAR\_IND\_B | INDDED | PINDED | NINDED |
| SPMMFDED | PRF\_PAR\_IND\_B | FAMDED | PFMDED | NFMDED |
| HMINDOOP | PRF\_PAR\_IND\_B | INDOOP | IINOOP | OINOOP |
| HMFAMOOP | PRF\_PAR\_IND\_B | FAMOOP | FOOPPBOC | FOOPNPOC |
| EPOIOOP | PRF\_PAR\_IND\_B | INDOOP | IINOOP | OINOOP |
| EPOFOOP | PRF\_PAR\_IND\_B | FAMOOP | FOOPPBOC | FOOPNPOC |
| SPMMISL | PRF\_PAR\_IND\_B | INDOOP | IINOOP | OINOOP |
| SPMMFSL | PRF\_PAR\_IND\_B | FAMOOP | FOOPPBOC | FOOPNPOC |
| HINDCPMX | PRF\_PAR\_IND\_B | SINDCPY | PINDCPY | NINDCPY |
| HFAMCPMX | PRF\_PAR\_IND\_B | SFAMCPY | PFAMCPY | NFAMCPY |
| EPOICPY | PRF\_PAR\_IND\_B | SINDCPY | PINDCPY | NINDCPY |
| EPOFCPY | PRF\_PAR\_IND\_B | SFAMCPY | PFAMCPY | NFAMCPY |
| SPMMICPY | PRF\_PAR\_IND\_B | SINDCPY | PINDCPY | NINDCPY |
| SPMMFCPY | PRF\_PAR\_IND\_B | SFAMCPY | PFAMCPY | NFAMCPY |
| HINDCOIN | PRF\_PAR\_IND\_B | SINDCOIN | PINDCOIN | NINDCOIN |
| HFAMCOIN | PRF\_PAR\_IND\_B | SFAMCOIN | PFAMCOIN | NFAMCOIN |
| EPOICOIN | PRF\_PAR\_IND\_B | SINDCOIN | PINDCOIN | NINDCOIN |
| EPOFCOIN | PRF\_PAR\_IND\_B | SFAMCOIN | PFAMCOIN | NFAMCOIN |
| SPMMICOI | PRF\_PAR\_IND\_B | SINDCOIN | PINDCOIN | NINDCOIN |
| SPMMFCOI | PRF\_PAR\_IND\_B | SFAMCOIN | PFAMCOIN | NFAMCOIN |

**Note:**

The accums should be validated against ACCUM\_LEV\_INFO table before starting with the process. It’s RX comingled indicator should also be ‘Y’.

* If the RX comingled indicator is ‘N’ or if the derived (secondary) accum is not found, then the cross-apply update would be skipped. But as still the primary accum was processed successfully, it would be considered success and next accum would be processed.
* If in case of any fail in processing secondary accum would not be considered as failure.
* Only in case of DB unavailable, all the updates will be rolled back.
* Irrespective of the type of accum, the claim (input/request) network is used to update the cross-applied accums (secondary accums) ODS DB.
* Embedded/Non-embedded logic is applied for the cross-applied accums (secondary accums) too.

A tablebase GNCOD02T is created for all secondary accum derivation. For preferred cross-apply, the above mapping will be updated in the tablebase.

E.g.: If the input accum is BASINDED and if the network is NPAR, then tablebase will be called using BASINDED and DED\_PRF\_PAR\_IND-N as key and retrieve both shared and non-shared accum. As there is no way of finding whether the contract is shared or not, we will check the derived accums one by one against benefit table and only one of the accum will be found and it will be updated in ODS DB. (As the benefit table could have either shared or non-shared accum, not both)

**Dual Update – Secondary Accum**

The Dual Update is handled only for WGS/ISG non-spider member and is determined using the Indicator in the Contract DB (CONTR\_LEV\_INFO) - HIX\_EHB\_IND = ‘Y’

* It is only for DED/OOP accums
* It is only for PPO contract and PPO accums of POS contract.
* If the contract is PPO shared, then regular non-shared accums will be updated based on the Input Claim Network
* If the contract is PPO non-shared, then regular shared accum will be updated irrespective of the network indicator
* If the contract is POS then dual update will be done only when the input claim network indicator is NPAR (as the PAR network would be HMO accums) – same logic as a. and b. would be followed.

Benefit DB validations are not required for the Dual Update derived accums (secondary accum). So, the over-applied indicator will be always ‘N’. And MAX BNFT amount in the TOTL DB will be ‘0.00’.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Type of contract** | **IN/OUT network** | **Primary Accum** | **Claim Network** | **Secondary accum** |
| PPO | Shared Contract | INDDED | PAR | PINDED |
| PPO | Shared Contract | INDDED | NPAR | NINDED |
| PPO | Shared Contract | FAMDED | PAR | PFMDED |
| PPO | Shared Contract | FAMDED | NPAR | NFMDED |
| PPO | Non-shared contract | PINDED | N/A | INDDED |
| PPO | Non-shared contract | NINDED | N/A | INDDED |
| PPO | Non-shared contract | PFMDED | N/A | FAMDED |
| PPO | Non-shared contract | NFMDED | N/A | FAMDED |
| POS | Shared Contract | INDDED | NPAR | NINDED |
| POS | Shared Contract | FAMDED | NPAR | NFMDED |
| POS | Non-shared contract | NINDED | N/A | INDDED |
| POS | Non-shared contract | NFMDED | N/A | FAMDED |

**Handling HRA1**

**ESI Inquiry**

Before updating retrieved accums from Benefit Table in the response, input request is searched for HRA1. If any of the accum entry has HRA1, then HRA1 accum and its related details from request will updated as the first entry in the Response area and following the remaining accums from the Benefit table.

**ESI Update**

If the requested Accum is HRA1, the accum level validation and ODS DB update process is skipped. The response will be populated from the input and the status will be set as success call ‘01’.

**Generic Accum Name Mapping:**

Since ESI sends only generic accum names in the request, there is a process to map the PAR/NONPAR/SHARED accum names to the generic accum names in the benefit feed file load process. Please find below the details:

1. A new table base GNCGACMT has been created that maps the actual accum names to the corresponding generic accum names. The key fields to this table base are the accum name and the contract type.
2. The program GNC9843 has been modified to call this table base and fetch the generic accum name for each of the standard accums that comes in the benefit feed file.
3. The Benefit Translation tables (GNC. ACCUM\_LEV\_INFO and GNC.ACCUM\_LEV\_INFO\_HIST) have an additional column GEN\_ACCUM\_NM which will store the value fetched from the GNCGACMT table base.
4. The following jobs load the benefits feed files to the benefit translation tables.
   1. GNC37ADD to GNC37EDD
   2. GNC37GDD
   3. GNC37HDD

Please find below the generic accum name mapping for PPO accums. The first two bytes denote the type of accum (DED/OOP), the next two bytes denote whether the accum is PAR/NPAR/Shared and the last two bytes denote whether the accum is Individual or Family.

|  |  |  |
| --- | --- | --- |
| **Accum Name** | **Contract Type** | Generic Name |
| INDDED | P | 010301 |
| PINDED | P | 010101 |
| NINDED | P | 010201 |
| INDOOP | P | 020301 |
| IINOOP | P | 020101 |
| OINOOP | P | 020201 |
| FAMDED | P | 010302 |
| PFMDED | P | 010102 |
| NFMDED | P | 010202 |
| FAMOOP | P | 020302 |
| FOOPPBOC | P | 020102 |
| FOOPNPOC | P | 020202 |

**P2P Process – ISG to ISGHIX/ISGEHB Transfer**

The below process is for accum portability, when a member moves from an ISG/HIX product to a HIX plan (both on exchange and off exchange). The P2P transfer of accums have the following major process involved,

* P2P Trigger Process
* P2P accum update process
* Run-Out Processing
* P2P Inquiry Screen

The P2P change feed file from ISG and WGS membership teams are created in below jobs:

IYM984ID job – This contains the feed for ISG to ISG daily change

IYM985ID job – This contains the feed for ISG to ISG Renewals

GNC151BD job - WGMP.NONX.WGMP486D.P2PFILE.G\* contains the feed for WGS to WGS daily changes and Renewals

Member other enrollment (MOE) file from membership team (created in IYM601ND job) will be used to trigger the WGS to ISG P2P process.

**P2P Trigger Process**

Following are the major process involved in P2P trigger process.

Input files are

1. ISG daily change file

2. ISG Renewal file

3. WGS daily/ Renewal file and

4. MOE file

* Sort the MOE file to include the valid WGS TO STAR transfers.
* Eliminate the duplicate legacy records and re-format to obtain the records with WGS to STAR transfers.
* Map the WGS to STAR transfers filtered from MOE file to P2P layout, with old and new contract details.
* Map the ISG transfer records from membership feed to P2P layout, with old and new contract details.
* Read the reformatted file in P2P layout
* Check for exclusions in exclusion table. GNCPEXLT tablebase will be used to store the exclusions of members based on the product type, state code etc
* Derive the benefit administrative options for old and new plan based on contract.
  + For legacy contracts, retrieve the details from CONT db.
  + For HIX contracts, retrieve the details from SPIDER tables
  + For EHB contracts, retrieve the details from Benefit tables
* For EHB contracts, check whether it’s case/group/contract code are present in exclusion table bases. If corresponding CASE/GROUP/CONTRACT are present, it’s excluded from transfer process.
* Determine the P2P eligibility using synch rules and date scenarios. The module GNCSYRUS will be reused to identify P2P eligibility too.
  + If the “state overrides apply” indicator is set in the GNCSYRLT tablebase, the WCSSTCDT table will be referred to decide on any state specific overrides.
  + The state table WCSSTCDT will be modified to include a new field which will decide whether to apply credit or not on state code basis.
* All the members who are backward/forward eligible will be stored in MBR\_SYNC table using the IO routine GNCSYIOS.

**P2P Accumulator Update Process**

When a member moves from an ISG/ HIX product to a HIX plan (both on exchange and off exchange), the accumulators should also be moved.

Following are the major functions involved in accumulator update process:

* The P2P eligible members will be received from P2P trigger process
* If old product is ISG/WGS, retrieve the accumulator dollars for the old product from IMS accum DB.
* If old product is HIX/EHB, retrieve the accumulator dollars of the old product from ODS accumulator tables.
* When accumulators are retrieved from old plan, only those claims present within the forward sync dates are moved to the new plan.
* When old member start date is same as old member end date, it’s considered as never effective transfer. For such transfers, non-medical accumulators and medical accumulators with reason code other than ‘S-‘, are moved to the new contract. Medical claims directly applied to the old contract are not moved to the new plan. These will be adjusted by the users. Also, we do not establish accum sync for never effective transfers.
* Derive the accumulator names for the new contract from GNCSPPAT tablebase. Based on the benefit option of the new contract, move necessary dollars to the new plan.
* The new accumulator name derived is validated against SPIDER tables for HIX/Benefit tables for EHB. If the new accumulator name is not present for the new contract, we will not transfer the accumulator. It will be routed to error report.
* If the old product is medical + Rx and new product is Medical Alone, then the medical dollars will be ported to the new accumulator, Rx will be moved as hidden segment with a different product type-PRX. Similarly, we have PDN and PVS for dental and vision accumulators.
* Once accum name is validated, update these accumulators to ODS accumulator tables(ACCUMR\_DTL and ACCUMR\_TOTL) with reason code ‘PP’
* Accumulator Records will be created for each of the comingled system and updated to the ODS database.
* The contract maximum will be checked against the accumulator amount on update and over applied indicator will be set if the accum amount reaches or exceeds contract maximum.
* To handle multiple transfer scenarios, accumulator details like DCN, service date, monies, occurs and days will be updated in ODS accum detail table as part of P2P process.
* Successful P2P accumulator transfers are stored in P2P\_ACCUM table. These will be visible in P2P inquiry screen.
* If the old plan does not have accumulators, accumulator sync is established to facilitate forward and backward sync through run-out process.

**Run-Out Processing**

This involves sync up of run out claims on a P2P transferred member. The claims for old plan will be updated to new plan accums and vise versa if the claims fall within the sync eligibility dates.

* The ODS trigger DB will be swept to extract out all the transactions happened on ODS, (Including ESI, EyeMED, DCare etc.) for the particular day.
* Each transaction will be validated against the P2P member Sync table to determine any prior P2P transfer for the member contract
* If the member is found to have entry in the P2P member table, all the entries corresponding to the member case combination will be retrieved from the P2P table
* Each record will be evaluated for forward sync and backward sync eligibility by validating service dates against forward/backward sync effective and end dates.
* If a particular contract record is found eligible for backward or forward sync the ODS will be updated with the respective amount in the i/p record.
* This process will be repeated for all the eligible contracts for the member in the current benefit period.
* Forward sync claims are updated with reason code ‘FS’ and backward sync claims are updated with reason code ‘BS’ in ODS accum detail table.

The above process will be applicable for all the transactions including Medical, ESI, EyeMED, DCare etc.

**P2P Inquiry/Update Screens**

This involves changes to display/update all the P2P transactions for all ISG and WGS transfers. User will be able to view the details based on queue code or member details.

The P2P inquiry screens (HT - ODS - P2P TRANSFERS) will be available using the menu option XX – ODS MENU from the main profiles like SYSTEMS. The P2P inquiry screen can be retrieved from the below option in the new menu.

* P2P inquiry screen - HI - HIX INQUIRY
* The user input ( either queue or member details) will be retrieved in the driver module using router call
* The P2P member sync table and P2P accum table will be called to retrieve the P2P data record will and will be displayed on to the screen.
* User will be able to retrieve P2P records using below combinations
  + - Queue code
    - Old cert + new cert
    - Old HCID + New HCID
    - Old HCID or new HCID
    - Old Cert or new Cert
* PF09 function key will allow user to view next record in a queue or Cert/HCID query .
* PF10 function key will allow user to view previous record in a queue or Cert/HCID query

The P2P update screen will be available using the menu option HT – HIX TRANSFERS from the XX-ODS menu under main profiles like SYSTEMS. Option HU is available for HIX P2P insert/update.

**Add a P2P Entry**

* Mandatory fields cert id , HCID, member code, group number, accum details and case number for old and new member for adding an entry
* The above fields will be validated and appropriate error messages will be thrown on the screen
* A PF04 request after entering valid entry, will validate the record against membership, contract and p2P eligibility
* “PLEASE ENTER TO CONFIRM” message will be thrown if the record is found eligible for a P2P.
* A PF12 request on getting the above message will confirm the transaction with a message “ACCUM UPDATE SUCCESSFUL” on successful completion.

Update a P2P Entry

* Existing records to be updated can be pulled using queue code or cert/HCID combination.
* PF06 will be available to open up the fields to be updated
* The required fields can be entered for update and a PF12 request will confirm the request.

**Components:**

**GNC151BD JOB/PROC:-**This JOB is to read the MOE file and P2P membership feed file and process members. The following process will be done,

* Sort MOE file and retrieve records with WGS to HIX transfer
* Unload the P2P exclusion table to file
* Read ISG membership feed file by calling GNC1113 module and reformat the file to GNCISGMC copybook layout
* Read MOE file by calling GNC1114 module and reformat the file to GNCISGMC copybook layout
* Read the reformatted files by calling driver module GNC1115.
* Calls the processing module GNCGP2PS for below steps
  + Read the P2P exclusion table unload file and look for exclusions for the member

The below is the layout for the P2P table

|  |
| --- |
| CONTRACT CODE |
| SYSTEM |
| MBU-CODE |
| RULE-ID |
| GROUP NUMBER |
| CASE NUMBER |
| COMPANY CODE |
| EXCLUSION TYPE |
| BUSINESS |

* + Read contract DB to retrieve contract details.
  + The P2P eligibility for the members is identified using the P2P rules and date scenarios. The module GNCSYRUS will be modified to check for renewal scenarios
* Store the P2P eligible members in SYNC\_MBR table. The below is the latest structure of DB2 table

****

* Create an output file with the eligibility information

**GNC161BD JOB/PROC:-**This JOB reads the P2P eligible information and update ODS accumulator tables.

* Execute the new driver module GNC1116 to read P2P eligible records from trigger process.
* Call the new sub-module GNCP2PHS to retrieve the accumulator and contract details of the old product
* Call the new sub-module GNCP2PUS to derive the accumulator names for the new product from SPIDER DB. Accumulators to be ported are determined based on the benefit administrative options and updated to the ODS accumulator tables, ACCUMR\_TOT and ACCUMR\_DTL
* Call the new sub-module GNCPAIOS to store the ported accumulator details

**GNC1115 SRC: -** The purpose of this module is to map the membership details from the membership file to P2P layout. This program does the following,

* Perform membership validations call Group DB, Case DB and Contract DB and validate the details and fetch the WGS/ISG details.
* Determine P2P exclusion using GNCPEXLT table
* For other members without exclusion determine case , cert or contract switches
* If Old Case = New case and Old cert = New cert and Old contract Not= New contract the plan is Contract switch
* After the plan switch is identified the module GNCSYRUS is called to determine P2P eligibility
* Backward/Forward eligible records are inserted to Synch DB by calling GNCSYIOS

**GNCGP2PS SRC:-** The purpose of this module is to process the member and retrieve the contract details by calling contract DB for WGS, Spider table call for HIX and ODS benefit tables for EHB transfers. The module performs the below functions.

* Open exclusion table unload file.
* Get current date from WGMDATES module, write error report header and do the checkpoint logic
* Determine P2P exclusions by using GNCPEXLT table
* Read the GNCPEXLT unload file and move to array.
* Search the array with Old plan Case, Cert and Contract and set exclusion flag if present in the table.
* Perform the same with New case, Cert and contract code
* Call contract DB by passing the old contract details. Populate the LQC and other fields
* Calculate the accum date using the contract type (Benefit year/Calendar year).
* Call contract DB by passing the new contract details. Populate the LQC and other fields
* Calculate the accum date using the contract type (Benefit year/Calendar year).
* Determine the product switch for contract switch/case switch/
* Determine the P2P eligibility of the member using the synch rules and by calling GNCSYRUS module.
* Map the forward/backward eligible member fields and store them in Synch DB by calling the IO routine GNCSYIOS

**GNCSYRUS SRC : -** This module identifies the P2P forward/backward eligibility. The below steps are done for this process.

* The module GNCSYRUS will be modified to include additional path for P2P process. For P2P the eligibility will be done also based on the State code.
* The rule table GNCSYRLT will be called for eligibility check. The table will be of below format

|  |  |
| --- | --- |
| RULE-ID (Key) | A unique rule id for each rule |
| SYSTEM-IND | System indicator |
| OLD- STATE-CDE | Old State code |
| OLD-XCHANG   (ON/OFF) | Old exchange indicator (On/Off) |
| NEW-XCHANG   (ON/OFF) | New exchange indicator (On/Off) |
| OLD-BUSINESS(IN/SG/LG) | Old product type (SG/IN) |
| NEW-BUSINESS(IN/SG/LG) | New product type (SG/IN) |
| ST-SPECFIED-APPL | State code override applies (Y/N) |
| CREDIT-INDICATOR | Apply Credit or not |
| NEW-STATE-CDE | New State code |
| DED-APPLY-IND | Deductible apply indicator |
| OOP-APPLY-IND | OOP apply indicator |
| LQCO-APPLY-IND | Last quarter carryover apply indicator |
| BENFT-APPLY-IND | Benefit apply indicator |
| OLD-CARRIER | Old carrier |
| NEW-CARRIER | New carrier |

* The prior plan and new plan details from the input file will be searched against the P2P rule table for a matching entry. The credit indicator for the matching record will be used to decide on whether to apply P2P credit or not.
* If the “state overrides apply” indicator is set in the above table, the WCSSTCDT table will be referred to decide on any state specific overrides.
* The state table WCSSTCDT will be modified to include a new field which will decide whether to apply credit or not on state code basis.
* The prior and current contract effective/ end dates and member effective dates will then be validated to determine P2P forward sync and backward sync eligibility/duration using the below date checks

Below document have details on date calculation (forward sync and backward sync duration) for different type of switches.



**GNC1116 SRC: -** This module is the new driver module for P2P accumulator update process. Input file is the P2P eligible records from trigger process.

The module does the following procedures.

* Read the input file and do all the below steps for each record.
* Identify the accumulator effective date segments based on the type of P2P transfer.
* For CY transfers, effective date will be calculated as 0101CCYY and end date will be 1231CCYYY.
* For BY transfers, membership DB is called to get anniversary date in MMDD. Old contract year is appended to get benefit year from date. WGMDTCLS module is then called to obtain benefit year through date.
* Depending on the new/old effective dates, we will have four types of date changes
  + CY to CY
  + CY to BY
  + BY to CY
  + BY to BY
* Depending on the type of old contract, set either “GET ALL ACCUMS ISG”/ “GET ALL ACCUMS WGS” flag or “GET ALL ACCUMS ODS” flag and call new sub-module, GNCP2PHS.
* In GNCP2PHS module, we retrieve the accumulator data for old product and benefit administrative options will be checked to validate the accumulator value. Accumulator data will be passed back to the main module in a two dimensional array as shown below. The array can handle up to 50 accumulators and each accumulator can hold up to 500 accumulator details.
* Call new sub-module, GNCP2PUS, to update the ODS accumulators for the new contract.
* Call new sub-module, GNCPAIOS, to update the ported accumulator details to P2P\_ACCUM DB2 table
* Go back to step 1 until end of file is reached.

**GNCP2PHS SRC:-** The new sub-module does the following functions.

* Identify whether the old product is ISG, WGS, EHB or HIX based on the switch set from the driver module
* For ISG/WGS plans, call the accumulator IMS DB through GNCACIOS call and obtain the root/total segments.
* Do the below processes for all the total segments
  + Identify the accumulator name and populate to the array
  + Retrieve all the audit segments for the accumulator.
  + For each audit segment, if it is a medical claim, call the history DB through ICLHIIOS/GNCHIIOS call and retrieve the claim service date.
  + If the service date falls within the date range of forward sync, the accumulator monies, occurs, days and DCN from audit segment are added to the array
* For HIX/EHB plans, call GNCEBSRS module to retrieve the accumulators from ACCUMR\_DTL and ACCUMR\_TOTL tables. The service start and end date present in the ACCUMR\_DTL table will be used to find out the accumulator deltas updated during the benefit period. Those details that falls within the forward sync dates will only be populated to the array.

**GNCP2PUS SRC:-** The new sub-module does the following functions.

* Receive the two dimensional accumulator array for old product from the driver module
* For each of the accumulator of the old product, access tablebase GNCSPPAT to obtain the new accumulator name. Keys to identify the accumulator name are
  + Old accumulator name
  + From and to network codes
  + From and to contract type(CY/BY)
  + From and to product type(PPO/HMO)
* Once ‘to accumulator’ name is identified from tablebase, the accumulator details are updated to ODS table through GNCEBSRS call.
* If the old product supports Medical, vision, RX and dental and the new product supports medical alone, all accumulators related to vision, RX and dental will be moved to ODS db, but will be kept as hidden records. In ODS db, the records will be stored with a different product type code ‘PRX’, ‘PVS’ or ‘PDN’. These accumulator amounts should not be reflected in the ‘TOT’ record type in ACCUMR\_TOTL table.
* Accumulator dollars will be updated to the ODS even if it over applies but the over applied indicator will be set. This logic is existing in GNCEBSRS module.

**GNCPAIOS SRC: -** This is the new IO module for P2P\_ACCUM table. There are 4 functions introduced as given below:

* INSERT rows to P2P\_ACCUM table
* FETCH rows from P2P\_ACCUM table with old contract details
* FETCH rows from P2P\_ACCUM table with old contract details and old accum name
* FETCH rows from P2P\_ACCUM table with new contract details
* FETCH rows from P2P\_ACCUM table with new contract details and new accum name

**GNCEBSRS SRC: -** Following changes have been made in this module:

* Added logic to skip the validation of secondary accum update during EHB P2P transfers

**GNC91\*ND jobs –** This job executes GNC911ND proc, which further executes GNC1119 module and creates claim extract file

**GNC1119 SRC: -** This module is executed by GNC91\*ND jobs, which is scheduled hourly. It retrieves the non-HIX and non-EHB claims from finalized file to GNC9520C copybook layout

**GNC18\*BD jobs –** This job executes GNC18XBD proc, which further executes GNC1117 module and creates P2P sync eligible records

**GNC1117 SRC: -** This module is executed by GNC18\*BD jobs, which is scheduled hourly. Read Processes the finalized file extract and ODS db unload to check for sync eligibility.

* Populate the claim details into old plan details and query MBR\_SYNC/P2P\_ACCUM tables for any new plans using GNCSYIOS routine.
* If there are new plans and if the forward sync indicator is Y, check whether the claim service date is within the forward sync dates. If yes, populate the new member/plan/source details to sync array.
* Populate the claim details into new plan details and query MBR\_SYNC/P2P\_ACCUM tables for any old plans using GNCSYIOS routine.
* If there are old plans and if the backward sync indicator is Y, check whether the claim service date is within the backward sync dates. If yes, populate the oldmember/plan/source details to sync array.
* Once member is validated with membership tables using GNCMBRSS/GNCMBRWS routines, it prepares an output file in GNC1117D copybook layout.

**GNC1118 SRC: -** This module is executed by GNC181BD job, which is scheduled hourly. Read input file in GNC1117D layout and populate fields to call GNCACOBS routine based on source to be synced up. GNCACOBS module will further call EBS routine/GNCACIOS routine to update ODS/IMS accum db.

**GNCP2PI SRC:-** This is the driver program for P2P inquiry screen. This screen can be used to view all the completed Product to Product transfers. The P2P transfers can be retrieved by giving the Queue code or the New and Old Cert or HCID details. PF09 key can be used to retrieve next member in the specified queue or next member for the subscriber. PF10 can be used to view the previously traversed member.

**GNCP2PQ SRC: -** The driver module that calls the router to display the data on the P2P screen. This module calls the sub program GNCP2PQS to retrieve the data from Member Sync Database and P2P accum Database.

**GNCP2PQS SRC: -** The subprogram called from the driver module GNCP2PQ which fetches the data from Member Sync Database and P2P accum Database by calling the I/O Routine GNCSYIOS.

**GNCP2PI PSB:-** The new PSB that will be used to access the CIP and NAV Databases for populating the new P2P inquiry screen.

**GNCSYIOS SRC: -** The I/O routine to access Member Sync Database and P2P accum Database.

**GNCP2PHM SRC: -** The MFS component for the new P2P inquiry screen.

**GNCP2PU SRC:-** This is the driver program for the new P2P update screen. This screen will be used to add and update a P2P record in the system. This screen will support any P2P transfer from ISG to HIX and HIX to HIX product. The P2P transfers can be retrieved by giving the Queue code or the New and Old Cert or HCID details to be available for update. PF09 key can be used to retrieve next member in the specified queue or next member for the subscriber. PF10 can be used to view the previously traversed member.

**GNCP2PES SRC:-** The module perform all the functionalities related to the the P2P update screen. This module validates the P2P data entered, checks for eligibility and updates the member sycn , ODS and P2P accum table on an ADD request. Module GNCP2PVS is used to validate P2P eligibility and update member sync table. GNCPPUPS is used to update ODS as well as P2P accum table. The update request first clears all the member accum data and then reinserts the screen data.

**GNCP2PVS SRC:-** The module perform all the P2P eligibility check by validating the prior and new member/contract and accum details. This also updates the member sycn table.

**GNCPPUPS SRC: -** The module perform ODS and accum sync table update. Its called from GNCP2PES on confirmation request.

**EHB SCREENS**

EHB screens include screen for accum inquiry & updation, accum selection, Audit trial, Benefit inquiry & updation, and product selection screen.  This document will explain the following:

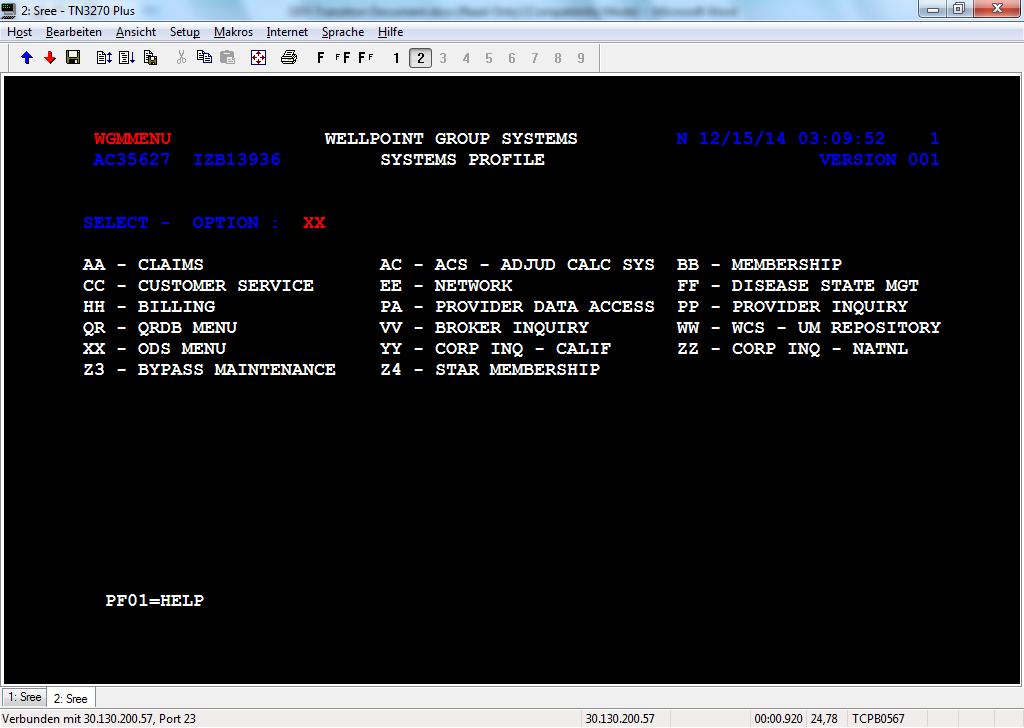
* ACCUM INQUIRY SCREEN  [GNCODAI]
* ACCUM AUDIT TRIAL SCREEN [GNCODAA]
* ACCUM UPDATE SELECTION SCREEN [GNCODAU]
* ACCUM UPDATE SCREEN [GNCOACU]
* BENEFIT INFO INQUIRY SCREEN [GNCOBII]
* BENEFIT SELECTION SCREEN [GNCOBPI]
* BENEFIT UPDATE SCREEN [GNCOBIU]

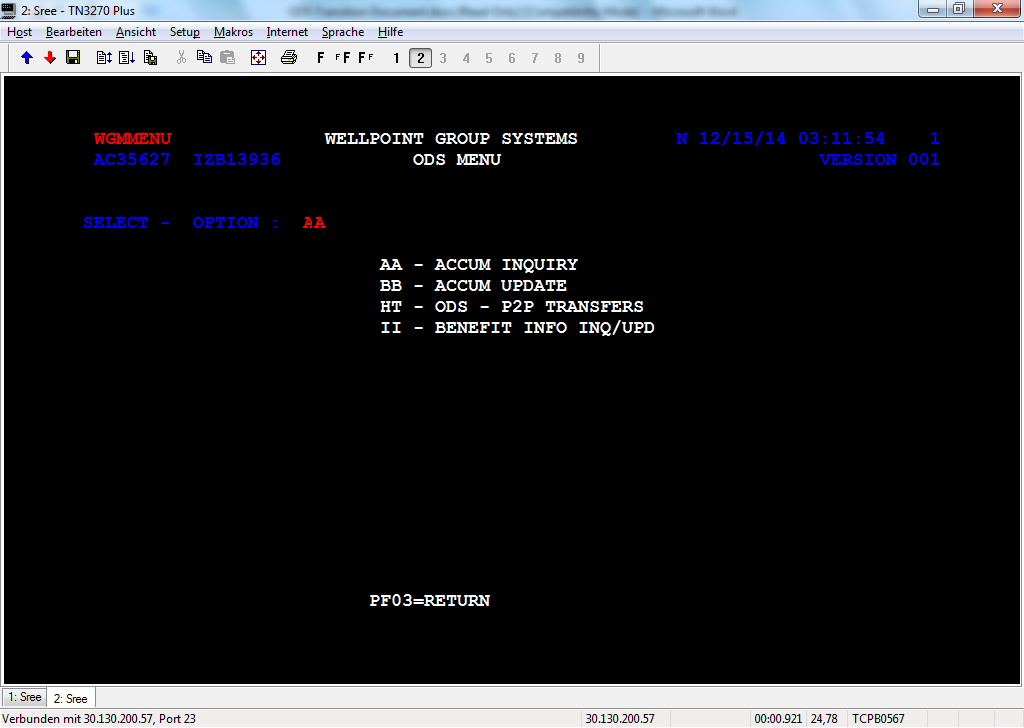
1. **ACCUM INQUIRY SCREEN**

The new inquiry screen, ACCUM INQUIRY SCREEN will be invoked by choosing the new ‘XX’ -> ‘AA’. The screen will display the Accum information that includes Accum name, Accum description, from and through dates, amount, occurs, days, Audit Trail selection, Accum over Applied Indicator, product types, Field to indicate summary or detailed view etc. The DB accessed for getting the accum info is ODS Total table.

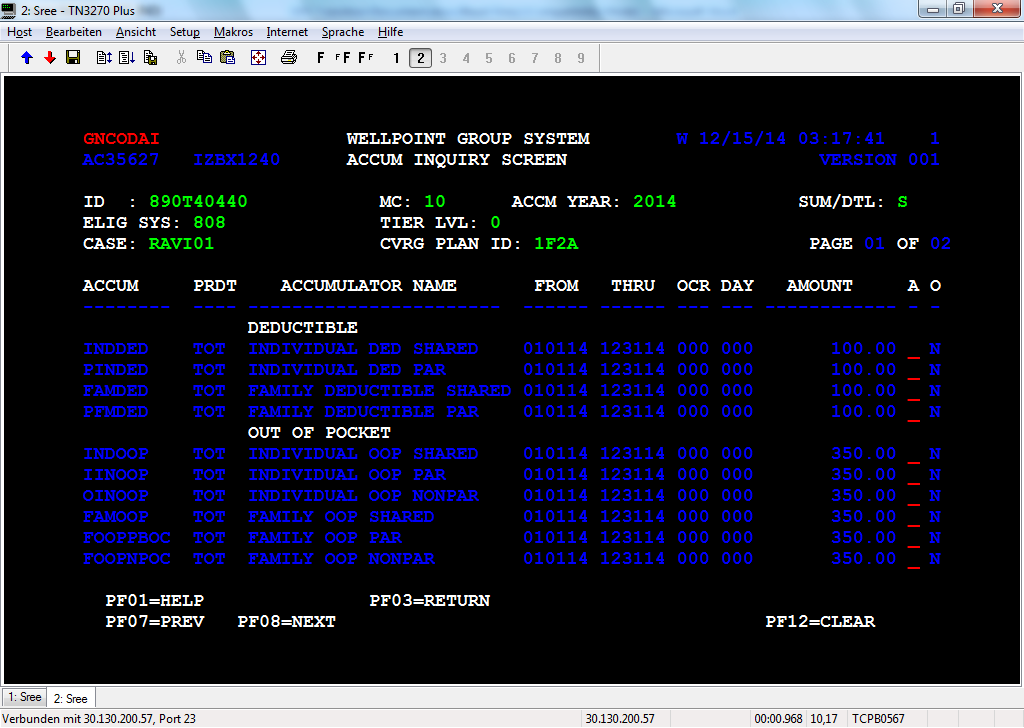
**Initial Navigation to Accum Inquiry Screen:**

**Enter Systems 🡺 XX 🡺 AA**



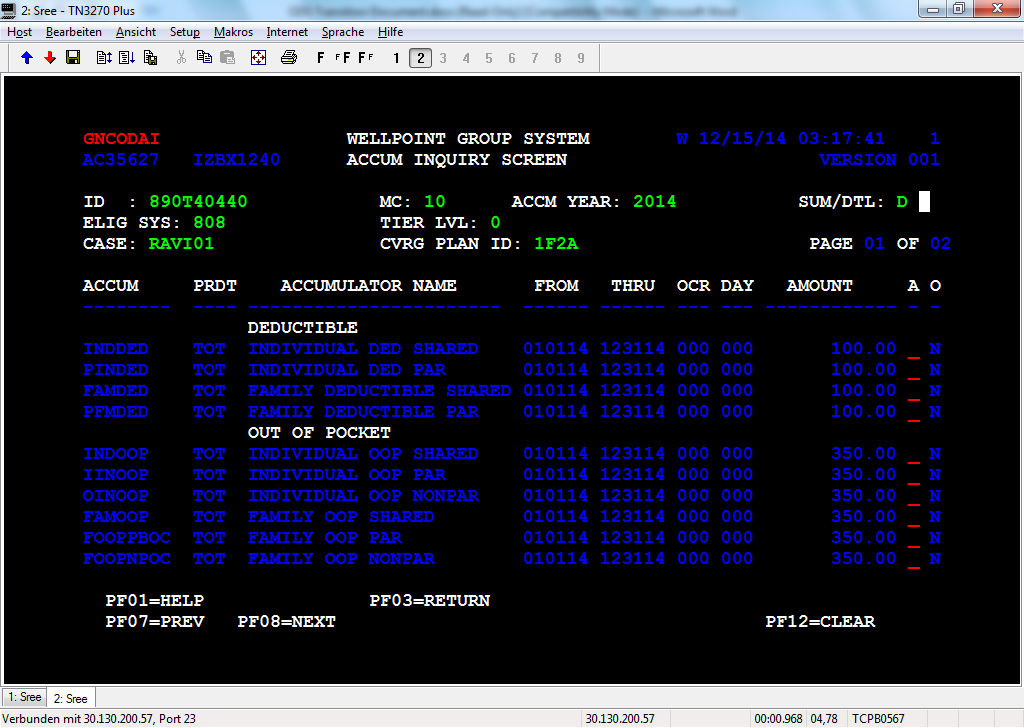


Give HC ID, Member code, Accum year, Eligible System indicator, Tier level, Case number, Coverage plan ID and press enter.

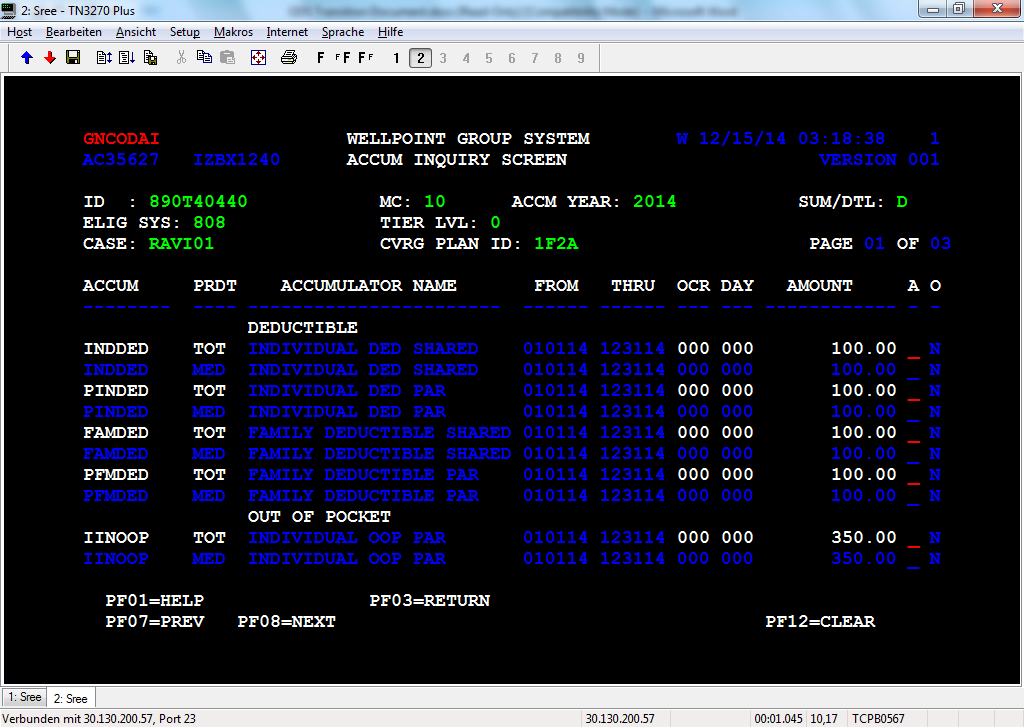


Change Summary/Detail option to switch between summary or detail view.

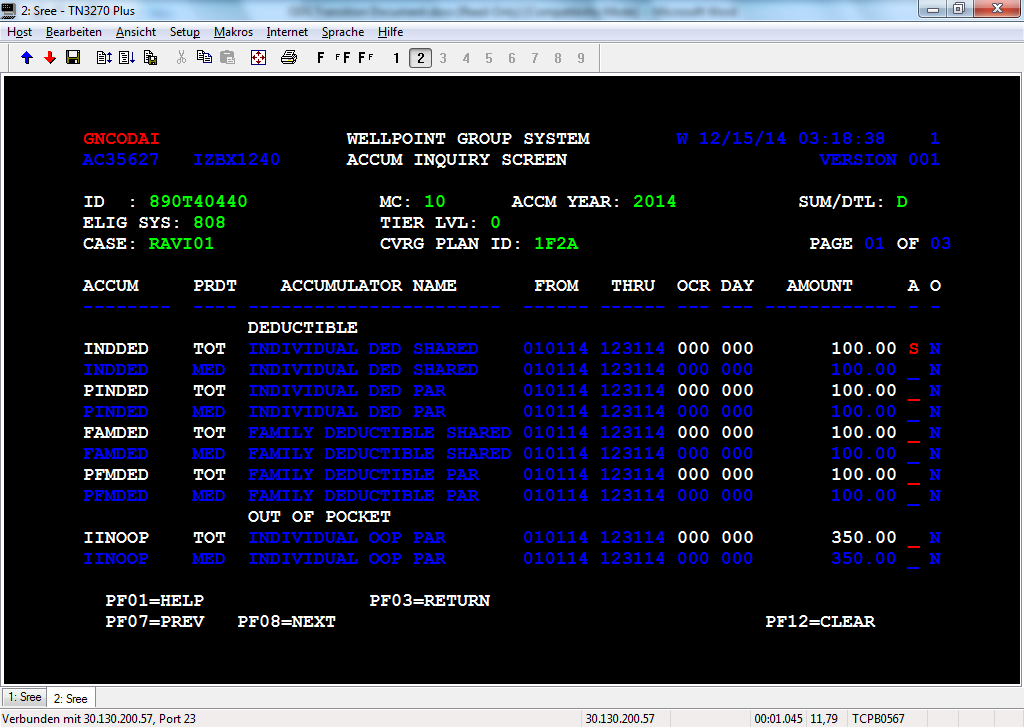
Summary View:-



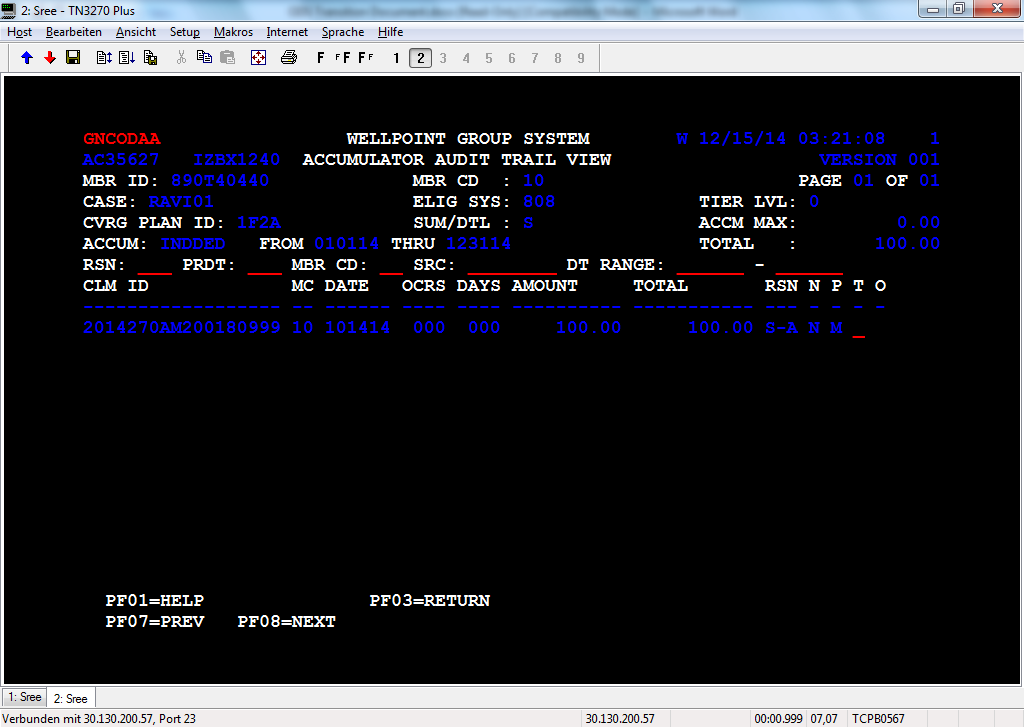
Detail View:-



Give “S” to Audit trial selection ‘A’ – You will navigate to Audit trial screen



Audit trial screen



The editable fields in the screen are as below.

ID: - This should be HCID

MC: - Member Code

ACCM YEAR: - Benefit Year for which Accum details to be displayed

ELIG SYS: - Member Eligibility Source System

TIER LVL: - Contract TIER Level

PROD TYPE: - Product Type (MED, VIS, DEN etc.)

CASE: - CASE ID

CONT: - Contract Code

Mandatory fields that user should give to get the accum details from ODS total table are,

ID: - This should be HCID

MC: - Member Code

ACCM YEAR: - Benefit Year for which Accum details to be displayed

CASE: - CASE ID

CONT: - Contract Code

New function is to be introduced in GNCODIOS to get accum details for all product types (TOT, MED, RX, DEN, VIS etc.). The screen will have the following PF keys

PF01 – Help for all the fields

PF03 – Return to ODS Menu

PF07 – To view the previous page

PF08 – To view the next page

PF12 – Clear the data in the screen

Member code for source systems CS90 and Healthlink are of 3 bytes in the ODS Total table. The user is allowed to give the member code as 2 bytes in the screen. Contract code for source systems ACES and FACETS are spaces for the accum records in ODS Total table.

**Component Details:**

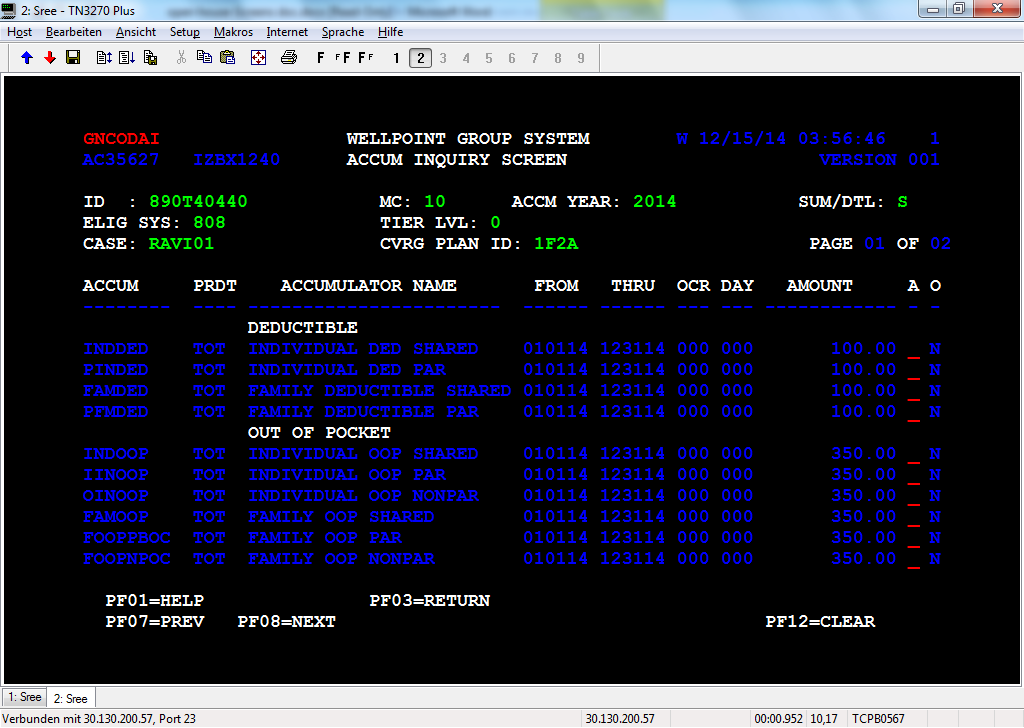
GNCODAI/GNCODAIS: GNCODAI is Accumulator Display program. The screen will display all the accumulators for the given combination of above mentioned mandatory fields. Accums will display on the screen. When the user look for accumulator name using the search criteria for enquiry or update, the system will highlight the accum row for easy identification.

* GNCODAI is using to display all the accumulators for given HCID, Contract, Source system, Contract, Tier level and Year.
* If all the mandatory data are present the new benefit i/O routine is called with these key fields and in order to retrieve “Accum Level Info” based on the benefit details entered in “Benefit Level Info” from CONTR\_LEV\_INFO and ACCUM\_LEV\_INFO DB2 tables
* User will be able to perform upward and downward scrolling if user gives PF07/PF08
* All TOT records will be fetched and filtered based on the ACCM YEAR given by user and displayed in the screen as given in the layout. This will be the summary view of the screen. ‘A’ field in summary view should be open for all the accums.
* When user switches to detail view by giving “D” in SUM/STL field, need to get the accum records with all the product types for the above inputs. A field should be open only for TOT records.
* User will be taken to audit trial screen once selected and depress entered Audit trail screen which will display claims updated for a selected accumulator.
* The other Functions keys used for this screen are
  + PF03 for return
  + PF07 for Previous page
  + PF08 for Next page

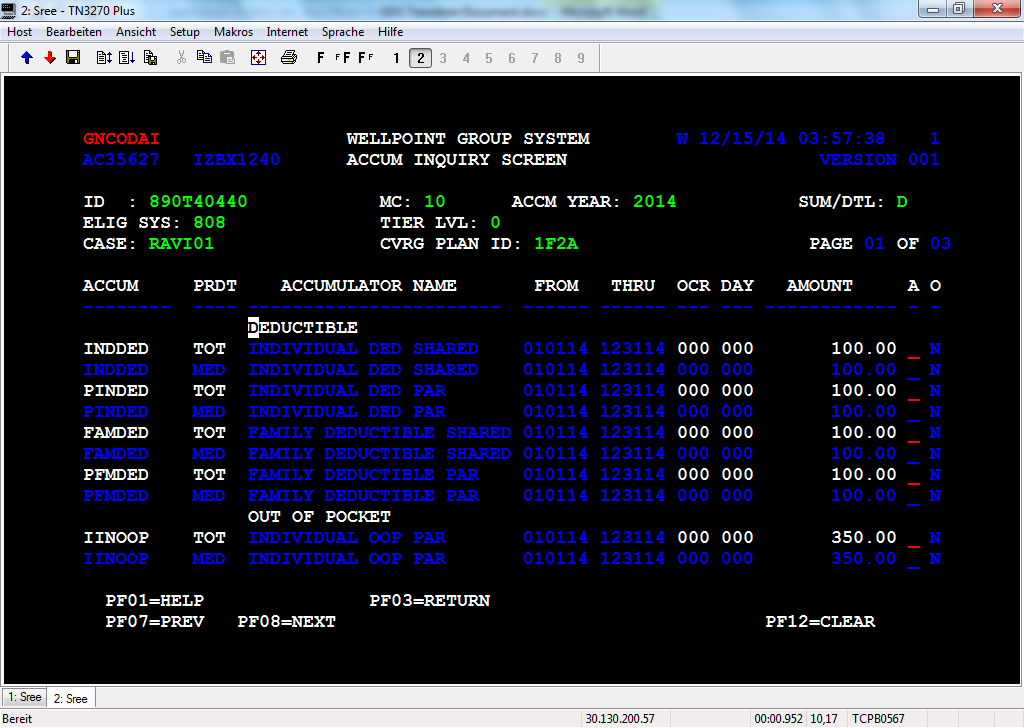
1. **ACCUM AUDIT TRIAL SCREEN**

|  |
| --- |
| From the Accum inquiry screen, if a particular Accum is selected by keying 'S' in A field, the |
| Audit trail summary screen will be displayed. This screen extract and display all the transaction |
| details from Accum ODS Detail table for that particular Accum with its from and thru dates |

Audit Trial Summary View



Audit trial detail view



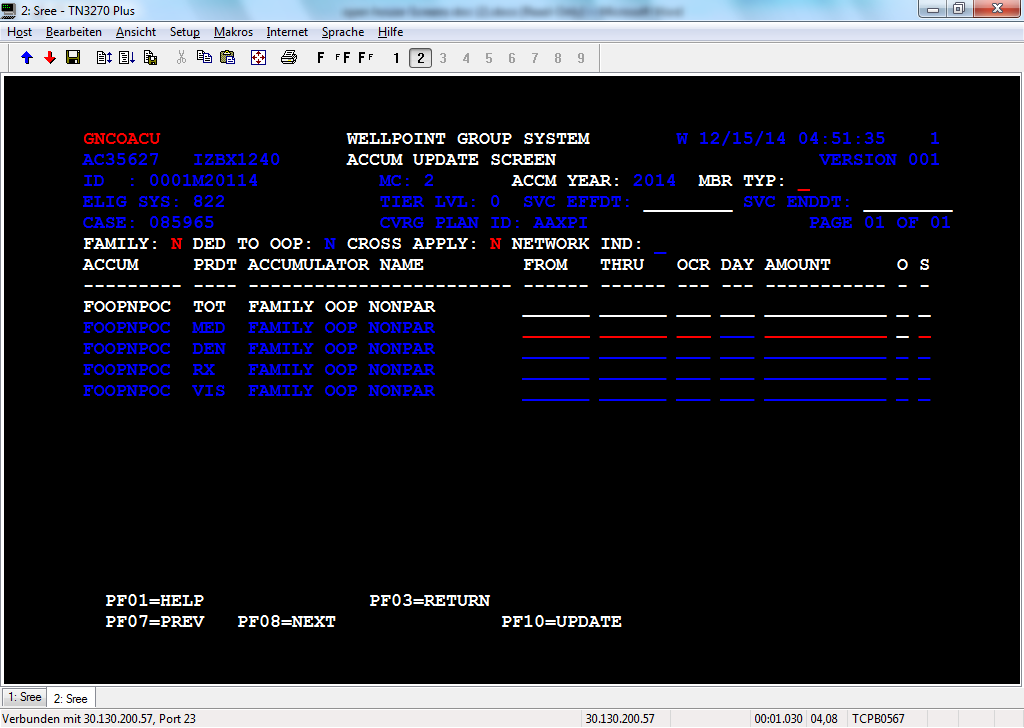
GNCODAA: Accum Audit Trial Screen which displays the details about the selected accumulator.

* Total amount of the selected accumulator displayed on the screen and the total amount can be attained by filter.
* If the value on Summary/Detail field is ‘S’ then the summary will display and if it is ‘D’ then detail will display.
* The ‘\*’ in the TXT field indicate an over apply to the deductible or out of pocket accumulator. If we select the TXT field a Message ‘UPDATE CAUSED BY OVERAPLLY’ will throw.
* The Functions keys used for this screen are
  + PF01 for help
  + PF03 for Return to previous Screen
  + PF07 for Previous page
  + PF08 for Next page

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| The header fields : |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
| |  |  | | --- | --- | | ID : | HCID that user gives in Accum inquiry screen | | MC: | Member code that user gives in Accum Inquiry screen | | ELIG SYS: | user gives in Accum Inquiry screen | | CASE: | user gives in Accum Inquiry screen | | CONT: | user gives in Accum Inquiry screen | | ACCUM: | that user selects in Accum Inquiry screen | | FROM : | Accum from date | | THRU: | Accum Thru date | |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
| Filter fields: - |  |  |  |  |  |  |  |  |  |
| Reason code, Product type, Member code, Source system, Date range |  |  |  |  |  |  |  |  |  |
| Accum Max : Accum max from benefit table |  |  |  |  |  |  |  |  |  |
| Total : Total benefit amount from ODS Total Table |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
| Detail fields:- |  |  |  |  |  |  |  |  |  |
| Claim ID (18 bytes), DATE (Claim processed date), OCRS, DAYS, AMOUNT, TOTAL (running total) |  |  |  |  |  |  |  |  |  |
| RSN (Reason code), N (network code), P (Product type), T(to view text screen) |  |  |  |  |  |  |  |  |  |
| O ( \* if over applied otherwise spaces)   1. **ACCUM UPDATE SELECTION SCREEN**   The new selection screen, ACCUM UPDATE SELECTION SCREEN will be invoked by choosing the new ‘XX’ -> ‘BB’. The screen will display the Accum information that includes Accum name, Accum description, from and through dates, amount, and occurs, days, Accum Over Applied Indicator, product types.    The DB accessed for getting the accumulators is Benefit Translation table (CONTR\_LEV\_INFO and ACCUM\_LEV\_INFO) and the dates, amounts, occurs, days, overapplied indicator should be taken from ODS Total table.  The editable fields in the screen are as below.  ID: - This should be HCID  MC: - Member Code  ACCM YEAR: - Benefit Year for which Accum details to be displayed  ELIG SYS: - Member Eligibility Source System  TIER LVL: - Contract TIER Level  CASE: - CASE ID  CONT: - Contract Code  The screen will have the following PF keys  PF01 – Help for all the fields  PF03 – Return to ODS Menu  PF07 – To view the previous page  PF08 – To view the next page  PF12 – Clear the data in the screen  GNCODAU/GNCODAUS:- In Accum update selection screen, the user should be able to input the data for which he should be able to view the details fetched from ODS tables as well as Benefit translation tables.  The inputs given by user should be validated and if they are valid GNCBNIOS is called to get the accums that are valid for the member and GNCEBSRS is called to get the corresponding details for each of these accums.  GNCBNIOS S is called with corresponding function to fetch data based on the below key fields.  CONT: - This would be the coverage plan id.  ELIG SYS: - The source system code.  TIER LVL: - Member tier  ACCM YEAR: - Benefit Year for which Accum details to be displayed  GNCEBSRS is called with corresponding function to fetch data based on the below key fields.  ID: - This should be HCID  MC: - Member Code  ACCM YEAR: - Benefit Year for which Accum details to be displayed  CASE: - CASE ID  CONT: - Contract Code  The user should be able to select the accum that he wants to update. At a time only accum can be selected for update.  The details retrieved from ODS are checked against the accumulators fetched from Benefit Translation Table and details are populated in the screen. Accums for which we don’t have any details from ODS Total table will have spaces for the below fields in the screen.  FROM THRU OCR DAY AMOUNT O   * The user can enter the actual system name in ELIG SYS field. However while passing inputs to I/O routine the code value corresponding to source system name is to be fetched from the new tablebase. * Member code for source systems CS90 and Healthlink are of 3 bytes in the ODS Total table. The user is allowed to give the member code as 2 bytes in the screen. Contract code for source systems ACES and FACETS are spaces for the accum records in ODS Total table. So even if the user gives the contract code or not, it should be kept as spaces before calling GNCEBSRS for ELIG SYS being ACES and FACETS. * So all the editable fields in the screen are mandatory fields that need user’s inputs. * The user will be able to select the accum that he wants to update by giving “S’ in the select field. The user will be able to select only one accum to update at a time. |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |

1. **ACCUM UPDATE SCREEN**

When user selects an accum from ACCUM SELECTION SCREEN, the new screen ACCUM UPDATE is thrown with the details of the accum that user would like to update along with its ODS Total Table details. In this screen all the product type (TOT, MED, RX etc.) records for that particular accum is displayed in the screen.



The header fields would be protected and the values for these fields are auto populated from the previous screen that is ‘ACCUM SELECTION’ screen.

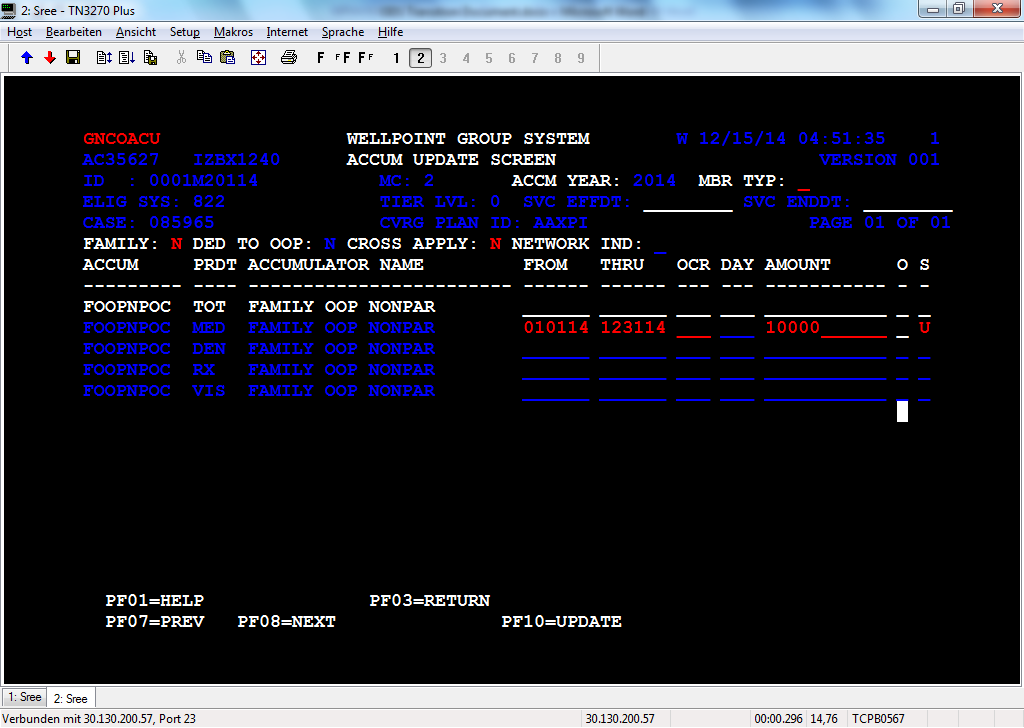
MBR TYP is an editable field in the screen. For non-embedded contract user needs to update the MBR TYP with ‘S’ or ‘F’ based on which the contract maximum is determined. If user gives ‘S’ and updates individual Accum, the amount given is to be checked with the accum’s contract maximum. If user gives ‘F’ and updates individual Accum, the amount given is to be checked with the corresponding family accum’s contract maximum. Based on this over applied indicator will get set.

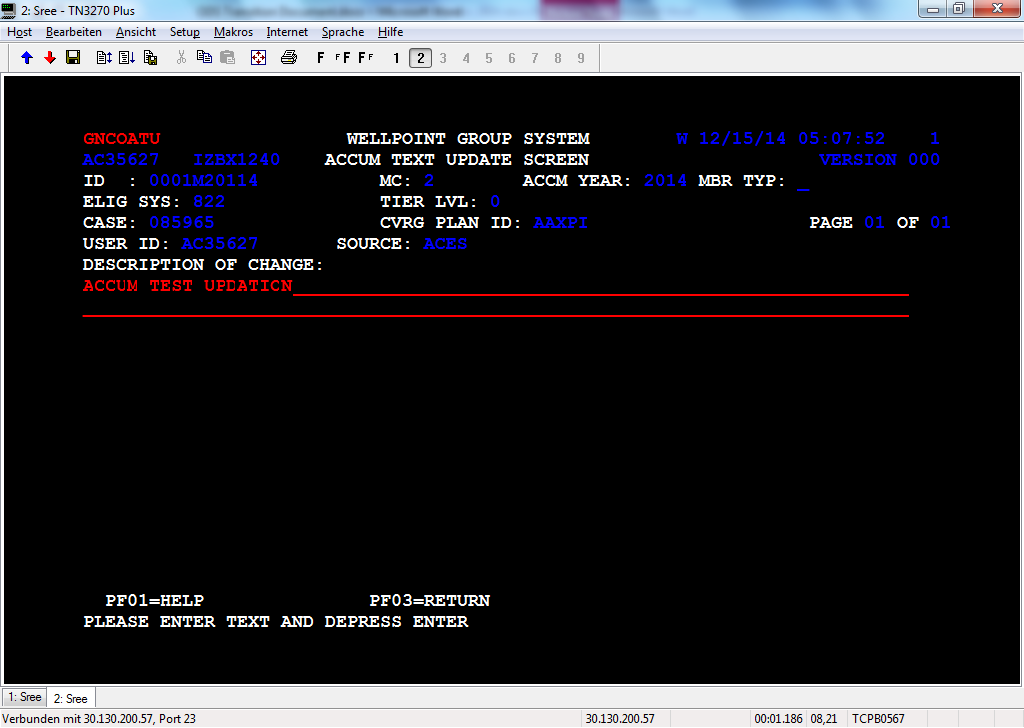
The below details will be editable fields.

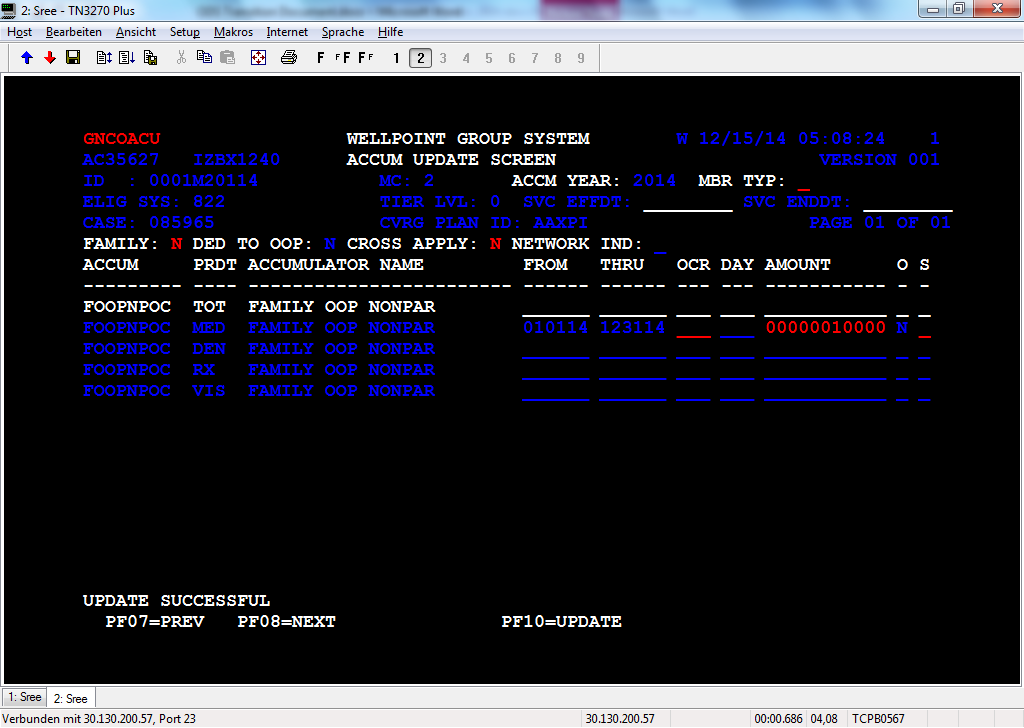
FAMILY, DED TO OOP, CROSS APPLY, NETWORK IND

The TOT record for the Accum will be highlighted but protected. For the other product types, the below fields will be editable so that user can give appropriate values and press PF10 to update the details.

OCR DAY AMOUNT O S







GNCOACU/GNCOACUS: - Accum update programs, which updates the value added to Accum for a particular product type, the same value is added to TOT record also in the ODS Total table.

* A new functions is to be introduced in GNCODIOS to get Accum details for all product types (TOT, MED, RX, DEN, VIS etc.) and
* The over apply field will be open only when the value is 'Y' and will accept only the value 'A' on manual update. Value 'A' indicates that the Accum should not be considered in over apply report creation, even if it got over-applied.
* The user will be able to give below values for S Field.

U- UPDATE IN WGS/ODS; SEND DELTA TO LEGACY

O- UPDATE IN WGS/ODS; DON'T SEND TO LEGACY

* If the user gives U, the Accum reason code value should be updated with MNU and if user gives O the Accum reason code value should be updated with MNO.
* The below details will be editable fields.

FAMILY, DED TO OOP, CROSS APPLY, NETWORK IND

* The TOT record for the Accum will be highlighted but protected. For the other product types, the below fields will be editable so that user can give appropriate values and press PF10 to update the details.

OCR DAY AMOUNT O S

* Whatever value is added to Accum for a particular product type, the same value is added to TOT record also in the ODS Total table.
* A new functions is to be introduced in GNCODIOS to get Accum details for all product types (TOT, MED, RX, DEN, VIS etc.) and
* The over apply field will be open only when the value is 'Y' and will accept only the value 'A' on manual update. Value 'A' indicates that the Accum should not be considered in over apply report creation, even if it got over-applied.
* The user will be able to give below values for S Field.

U- UPDATE IN WGS/ODS; SEND DELTA TO LEGACY

O- UPDATE IN WGS/ODS; DON'T SEND TO LEGACY

* If the user gives U, the Accum reason code value should be updated with MNU and if user gives O the Accum reason code value should be updated with MNO.

GNCEBSRS is called with below details by setting WS-ADD-MEMBER-ACCUM to update the details in ODS Total and Detail tables.

HCID – User input from Accum update selection screen

Member code – User input from Accum update selection screen

Effective date – Date derived from Accum update selection screen

End date – Date derived from Accum update selection screen

Case number – User input from Accum update selection screen

Contract – User input from Accum update selection screen

User ID – User id of the user

Product type code – Taken from screen

Accum Reason code – MNU when U is given in S field, MNO when O is given in S field.

Accum name- Accumulator that user selected from Accum update selection screen

Accum amount – User input

Accum Occurs – User input

Accum Days – User input

Over apply indicator – User input

Accum Text – User input in Accum Text Update Screen

Accum Type – Family or Individual based on the accum

Network code – Par/Npar

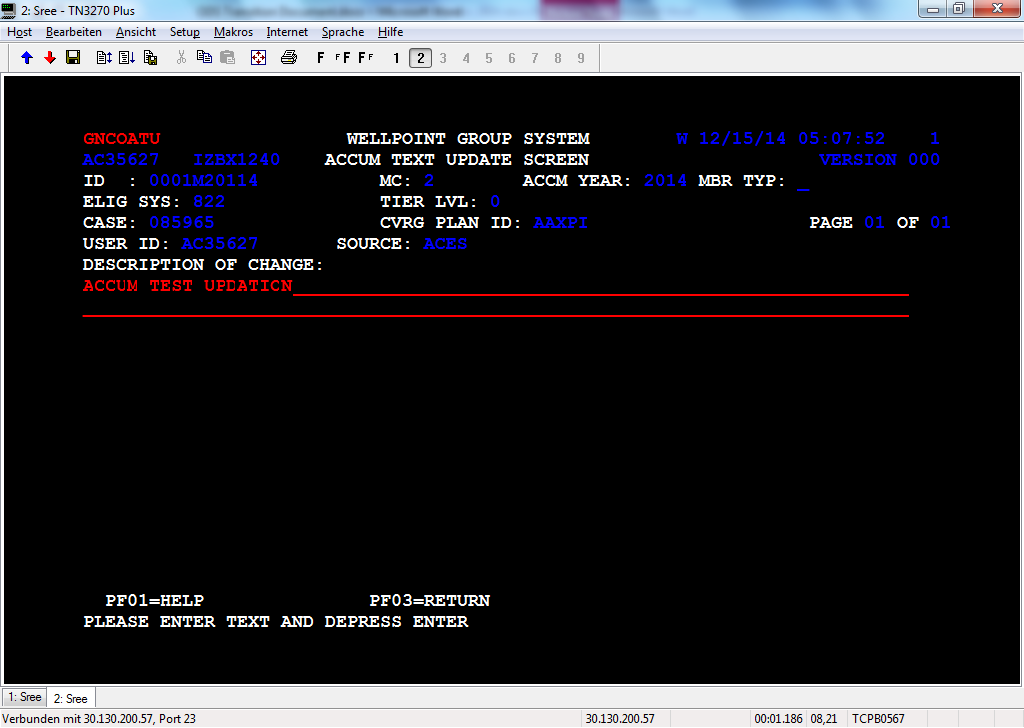
Eligible source system - User input from Accum update selection screen

* For this manual update, Dummy claim is to be generated and this should get loaded in DDCDB. The details need to be updated in ODS Detail table by calling GNCEBSRS using WS-ADD-MEMBER-ACCUM function. The same process for HIX is to be followed here.
* For CDHP claim information message as would be shown in the update screen.

**Accum Text Update Screen - GNCOATU**

When user gives PF10, it should throw ACCUM TEXT UPDATE SCREEN

The header fields will be auto populated from the previous screen. All the header fields are protected. The user will be allowed to enter the text description on the reason for which the accum is updated. The characters given should be of at least 10 bytes. After entering the text depress enter and the control will be back in ACCUM UPDATE SCREEN where “UPDATE SUCCESSFUL” message is populated. If user gives PF03 without giving text, update should not be done.



USERID field is populated with the user’s id and SOURCE field is updated code value name of source give in ELIG SYS field (ELIG SYS is user input which can be either code value or code value name.

1. **BENEFIT INFO INQUIRY SCREEN**

The new inquiry screen, BENEFIT INFO INQUIRY will be invoked by choosing the new ‘XX’ -> ‘II’. When the BENEFIT ACCUM INQUIRY SCREEN is invoked by entering all the mandatory fields like BENFT SRC SYS, CVRG PLAN ID, PRDCT TYP, START DATE, END DATE etc. The screen will be populated with corresponding Accum as well as contract details.

If all the mandatory fields were entered and the PRDCT TYP filed is left blank then it will invoke a new screen, PRODUCT TYPE INQUIRY SCREEN which will auto populate the screen with all the valid product types for that particular CVRG PLAN ID. The DB accessed for getting the Accum info is ODS Total table.

**Initial Navigation to Benefit info Inquiry Screen:**

**Enter Systems 🡺 XX 🡺 II**

|  |
| --- |
| WGMMENU WELLPOINT GROUP SYSTEMS 2 08/14/14 06:48:53 1 |
| AC35627 IZAY0291 SYSTEMS PROFILE VERSION 001 |
|  |
|  |
| SELECT - OPTION : XX |
|  |
| AA - CLAIMS AC - ACS - ADJUD CALC SYS BB - MEMBERSHIP |
| CC - CUSTOMER SERVICE EE - NETWORK FF - DISEASE STATE MGT |
| HH - BILLING HT - HIX TRANSFERS PA - PROVIDER DATA ACCESS |
| PP - PROVIDER INQUIRY QR - QRDB MENU VV - BROKER INQUIRY |
| WW - WCS - UM REPOSITORY XX - ODS MENU YY - CORP INQ - CALIF |
| ZZ - CORP INQ - NATNL Z3 - BYPASS MAINTENANCE Z4 - STAR MEMBERSHIP |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

XX -> II

|  |
| --- |
| WGMMENU WELLPOINT GROUP SYSTEMS 2 08/14/14 06:49:28 1 |
| AC35627 IZAY0291 ODS MENU VERSION 001 |
|  |
|  |
| SELECT - OPTION : II |
|  |
| AA - ACCUM INQUIRY |
| BB - ACCUM UPDATE |
| II - BENEFIT INFO INQ/UPD |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
| PF03=RETURN |

BENEFIT INFO INQUIRY SCREEN WILL BE INVOKED

|  |
| --- |
| GNCOBII WELLPOINT GROUP SYSTEM 2 08/30/14 07:00:22 1 |
| AC35627 IZAX1482 BENEFIT INFO INQUIRY SCREEN VERSION 001 |
|  |
| --------------------------BENEFIT LEVEL INFO ------------------------------- |
| ELIG SYS : \_\_\_\_\_\_\_\_\_\_ CVRG PLAN ID : \_\_\_\_\_\_\_\_\_\_\_\_\_\_ TIER LVL : \_ |
| START DATE : \_\_\_\_\_\_\_\_ END DATE : \_\_\_\_\_\_\_\_ CNTRCT TYP : \_\_\_\_ |
| VERSION NBR : \_\_\_\_\_ CDHP VENDR : \_\_\_\_\_ CDHP PROD TYP: \_\_\_\_ |
| EMBDED : \_ LAST QUARTER : \_ CRY OVR RESET: \_ |
| DED IN OOP  : \_ DED IN TO OUT : \_ DED OUT TO IN: \_ |
| DED PRF PAR : \_ OOP IN TO OUT : \_ OOP OUT TO IN: \_ |
| OOP PRF PAR : \_ CPY IN OOP : \_ CPY IN TO OUT: \_ |
| CPY OUT TO IN : \_ CPY PRF PAR : \_ CONS IN OOP : \_ |
| CONS IN TO OUT: \_ CONS OUT TO IN: \_ CONS PRF PAR : \_ |
| PRDCT TYP : \_\_\_\_ ADLT OOP EXCLD: \_ DUAL UPD IND : \_ |
| LST UPDT TS : \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| --------------------------ACCUM LEVEL INFO --------------------------------- |
| ACCUM T S LIMIT AMT OCCRS DAYS M R V D GEN ACCM |
| \_\_\_\_\_\_\_\_ \_ \_ \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_ \_\_\_\_\_\_ \_ \_ \_ \_ \_\_\_\_\_\_\_\_ |
| \_\_\_\_\_\_\_\_ \_ \_ \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_ \_\_\_\_\_\_ \_ \_ \_ \_ \_\_\_\_\_\_\_\_ |
| \_\_\_\_\_\_\_\_ \_ \_ \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_ \_\_\_\_\_\_ \_ \_ \_ \_ \_\_\_\_\_\_\_\_ |
| \_\_\_\_\_\_\_\_ \_ \_ \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_ \_\_\_\_\_\_ \_ \_ \_ \_ \_\_\_\_\_\_\_\_ |
|  |
| PF01=HELP PF02=ADD PF03=RETURN PF04=BNFTSL PF05=BNFTPR PF06=BNFTNX |
| PF07=PREV PF08=NEXT PF09=CHG/DEL PF12=CLEAR |

User can enter the valid inputs in the editable fields - SRC, PLAN ID, TIER LVL, START and END DATES.Then the screen will be populated with benefits and accum benefit details .

|  |
| --- |
| GNCOBII WELLPOINT GROUP SYSTEM 2 08/30/14 07:03:55 1 |
| AC35627 IZAX1482 BENEFIT INFO INQUIRY SCREEN VERSION 001 |
|  |
| --------------------------BENEFIT LEVEL INFO ------------------------------- |
| ELIG SYS : 808 CVRG PLAN ID : AAAA12345 TIER LVL : 1 |
| START DATE : 01012013 END DATE : 12312013 CNTRCT TYP : P |
| VERSION NBR : 00001 CDHP VENDR : LITES CDHP PROD TYP: HRA |
| EMBDED : X LAST QUARTER : X CRY OVR RESET: X |
| DED IN OOP  : X DED IN TO OUT : X DED OUT TO IN: Y |
| DED PRF PAR : X OOP IN TO OUT : Y OOP OUT TO IN: X |
| OOP PRF PAR : X CPY IN OOP : X CPY IN TO OUT: X |
| CPY OUT TO IN : X CPY PRF PAR : X CONS IN OOP : X |
| CONS IN TO OUT: X CONS OUT TO IN: X CONS PRF PAR : X |
| PRDCT TYP : MED ADLT OOP EXCLD: X DUAL UPD IND : N |
| LST UPDT TS : 2014-08-21-10.06.03 |
| --------------------------ACCUM LEVEL INFO --------------------------------- |
| ACCUM T S LIMIT AMT OCCRS DAYS M R V D GEN ACCM |
| IINOOP I A 000000200.00 \_\_\_\_\_\_ \_\_\_\_\_\_ Y N N N 020101 |
| INDDED I A 000000200.00 \_\_\_\_\_\_ \_\_\_\_\_\_ Y N N N 010201 |
|  |
|  |
|  |
| PF01=HELP PF02=ADD PF03=RETURN PF04=BNFTSL PF05=BNFTPR PF06=BNFTNX |
| PF07=PREV PF08=NEXT PF09=CHG/DEL PF12=CLEAR |

PF Key functionalities:

Press PF6 to view the next version of Benefit

Press PF5 to view the previous version of Benefit

Press PF8 to go to next screen to view the rest of the accum details for same benefit details

Press PF8 to go to next screen to view the rest of the accum details for same benefit details

Press PF12 to clear the screen

User can enter the valid inputs in the editable fields - SRC, PLAN ID, TIER LVL, START and END DATES and press PF4 to go to BENEFIT SELECTION POPUP screen to view all the benefit details like PRDCT-TYPE,COVERAGE PERIOD,TIER LVL

Screen functionalities:

Benefit Period Selection Popup will get invoked during the following scenarios

If the user provides only the below data as input

• Benefit Source System

• Coverage Plan id

• Coverage start and end date

If User doesn’t provide the Coverage Start and End date however provides the below data as input

• Benefit Source System

• Coverage Plan id

• Member Tier

If User doesn’t provide the Member Tier, Coverage Start and End date however provides the below data as input or Press PF4

• Source System

• Plan id

The screen which will contain all product type, coverage dates, and member tier combinations for the plan id will get thrown and user can select the appropriate record. Once the user selects a coverage date, the benefits should be shown in BENEFIT INFO INQUIRY SCREEN.

if there is only one set of product type, TIER LVL and coverage dates combination or if the user is giving VERSION number also along with all the inputs Benefit Source System, Coverage Plan id and Coverage start and end date, it should not throw the benefit period screen, but it should show the benefit and Accum details for that coverage date, TIER LVL and product type in Benefit Info Inquiry Screen itself.

**Component Details:**

GNCOBII: GNCOBII is the driver module for Benefit info inquiry screen. The cip-phases and PF key functionalities are handled in the driver module.

GNCOBIIS: GNCOBIIS is the subroutine module for Benefit info inquiry screen. All the DB interaction functionalities are handled in this module

**Below are transactions and PSBs defined for these screens?**

|  |  |  |
| --- | --- | --- |
| Screen | Program | PSB |
| Benefit Accum Inquiry Screen | GNCOBII | GNCOBII |
| Benefit Period Inquiry Screen | GNCOBPI | GNCOBPI |

**Below are the functions called in subroutine(GNCOBIIS) to fetch the data from Benefit Translation Tables**

User is expected to provide the below data as key to the new screen “BENEFIT INFO INQUIRY SCREEN”

Mandatory Key data:

* + - * Source System (Eg: WGS, CS90 etc.)
      * Plan id

Optional Key data:

* + - * Member Tier
      * Coverage start and end date
      * Version number

**Functionality-1:**

If the user provides the below data as input

* Source System (Eg: WGS, CS90 etc.)
* Plan id
* Member Tier
* Coverage start and end date
* Version number

Then query below tables

* CONTR\_LEV\_INFO
* ACCUM\_LEV\_INFO
* CONTR\_LEV\_INFO\_HIST
* ACCUM\_LEV\_INFO\_HIST

If record not found then display ‘THERE ARE NO RECORDS FOUND FOR THE GIVEN CRITERIA’

Else display the benefit and Accum info in the “BENEFIT INFO INQUIRY SCREEN”

**Functionality-2:**

If the user provides the below data as input

* Source System (Eg: WGS, CS90 etc.)
* Plan id
* Member Tier
* Coverage start and end date

Then query below tables

* CONTR\_LEV\_INFO
* ACCUM\_LEV\_INFO

If record not found then display ‘THERE ARE NO RECORDS FOUND FOR THE GIVEN CRITERIA’

Else display the benefit and Accum info in the “BENEFIT INFO INQUIRY SCREEN”

**Functionality-3:**

If User doesn’t provide the Coverage Start and End date however provides the below data as input

* Source System (Eg: WGS, CS90 etc.)
* Plan id
* Member Tier

Then query below tables

* CONTR\_LEV\_INFO

If record found from above tables then display all benefit periods info (i.e. Product type, Coverage start and end dates) in the “BENEFIT PERIOD SELECTION POPUP”

Else display ‘THERE ARE NO RECORDS FOUND FOR THE GIVEN CRITERIA’

After user selects a benefit period then query the below tables

* CONTR\_LEV\_INFO
* ACCUM\_LEV\_INFO

Then display the Benefit and Accum info in the “BENEFIT INFO INQUIRY SCREEN”

**Functionality-4:**

If the user provides the below data as input

· Source System (Eg: WGS, CS90 etc.)

· Plan id

· Coverage start and end date

Then query below tables

CONTR\_LEV\_INFO

ACCUM\_LEV\_INFO

If record not found then display ‘THERE ARE NO RECORDS FOUND FOR THE GIVEN CRITERIA’

Else display the benefit and Accum info in the “BENEFIT INFO INQUIRY SCREEN”

**Functionality-5:**

If User doesn’t provide the Member Tier, Coverage Start and End date however provides the below data as input

Source System (Eg: WGS, CS90 etc.)

Plan id

Then query below tables

CONTR\_LEV\_INFO

ACCUM\_LEV\_INFO

If record not found then display ‘THERE ARE NO RECORDS FOUND FOR THE GIVEN CRITERIA’

Else display the benefit and Accum info in the “BENEFIT INFO INQUIRY SCREEN”

**Functionality-6:**

Once the user successfully inquire a benefit record, the user can view the history versions by manipulating the version number in the “BENEFIT INFO INQUIRY SCREEN” and depress enter then Functionality-1 will be invoked [PF04].

User should enter

Source System (Eg: WGS, CS90 etc.)

Plan id

Then query below tables

CONTR\_LEV\_INFO

ACCUM\_LEV\_INFO

1. **BENEFIT UPDATE SCREEN**

Authorized user should be able to Add/Update/Delete benefit records in benefit DB2 tables using Benefit update screen

If the user would like to add a new benefit record, he/she can depress PF02 in the Benefit info inquiry screen and this will invoke BENEFIT UPDATE SCREEN where all the fields that are present in the inquiry screen will be editable, except version number and last update timestamp. User can fill all the editable with valid values and depress PF10 to update the benefit record.

The benefit update screen will be thrown with all the fields as editable except version number and last update timestamp

|  |
| --- |
| GNCOBIU WELLPOINT GROUP SYSTEM N 11/25/14 01:48:03 1 |
| SNMSSM6 IZA19020 BENEFIT UPDATE SCREEN VERSION 001 |
|  |
| --------------------------BENEFIT LEVEL INFO ------------------------------- |
| ELIG SYS : \_\_\_\_\_\_\_\_\_\_ CVRG PLAN ID : \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ TIER LVL : \_ |
| START DATE : \_\_\_\_\_\_\_\_ END DATE : \_\_\_\_\_\_\_\_ CNTRCT TYP : \_\_\_\_ |
| VERSION NBR : \_\_\_\_\_ CDHP VENDR : \_\_\_\_\_ CDHP PROD TYP: \_\_\_\_ |
| EMBDED : \_ LAST QUARTER : \_ CRY OVR RESET: \_ |
| DED IN OOP  : \_ DED IN TO OUT : \_ DED OUT TO IN: \_ |
| DED PRF PAR : \_ OOP IN TO OUT : \_ OOP OUT TO IN: \_ |
| OOP PRF PAR : \_ CPY IN OOP : \_ CPY IN TO OUT: \_ |
| CPY OUT TO IN : \_ CPY PRF PAR : \_ CONS IN OOP : \_ |
| CONS IN TO OUT: \_ CONS OUT TO IN: \_ CONS PRF PAR : \_ |
| PRDCT TYP : \_\_\_\_ ADLT OOP EXCLD: \_ DUAL UPD IND : \_ |
| LST UPDT TS : \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| --------------------------ACCUM LEVEL INFO --------------------------------- |
| ACCUM T S LIMIT AMT OCCRS DAYS M R V D GEN ACCM |
| \_\_\_\_\_\_\_\_ \_ \_ \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_ \_\_\_\_\_\_ \_ \_ \_ \_ \_\_\_\_\_\_ |
| \_\_\_\_\_\_\_\_ \_ \_ \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_ \_\_\_\_\_\_ \_ \_ \_ \_ \_\_\_\_\_\_ |
| \_\_\_\_\_\_\_\_ \_ \_ \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_ \_\_\_\_\_\_ \_ \_ \_ \_ \_\_\_\_\_\_ |
| \_\_\_\_\_\_\_\_ \_ \_ \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_ \_\_\_\_\_\_ \_ \_ \_ \_ \_\_\_\_\_\_ |
|  |
| PF01=HELP PF03=RETURN |
| PF07=PREV PF08=NEXT PF10=UPDATE PF11=DELETE PF12=CLEAR |

After giving all the values, validation for all the fields happen during ENTER/PF10. If all the fields are having valid values user can depress PF10

|  |
| --- |
| GNCOBIU WELLPOINT GROUP SYSTEM N 11/25/14 01:48:03 1 |
| SNMSSM6 IZA19020 BENEFIT UPDATE SCREEN VERSION 001 |
|  |
| --------------------------BENEFIT LEVEL INFO ------------------------------- |
| ELIG SYS : 822 CVRG PLAN ID : ACES1234 TIER LVL : 1 |
| START DATE : 01012014 END DATE : 12312014 CNTRCT TYP : P |
| VERSION NBR : \_\_\_\_\_ CDHP VENDR : LITES CDHP PROD TYP: H |
| EMBDED : X LAST QUARTER : X CRY OVR RESET: X |
| DED IN OOP  : X DED IN TO OUT : X DED OUT TO IN: X |
| DED PRF PAR : X OOP IN TO OUT : X OOP OUT TO IN: X |
| OOP PRF PAR : X CPY IN OOP : X CPY IN TO OUT: X |
| CPY OUT TO IN : X CPY PRF PAR : X CONS IN OOP : X |
| CONS IN TO OUT: X CONS OUT TO IN: X CONS PRF PAR : X |
| PRDCT TYP : RX ADLT OOP EXCLD: X DUAL UPD IND : X |
| LST UPDT TS : \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| --------------------------ACCUM LEVEL INFO --------------------------------- |
| ACCUM T S LIMIT AMT OCCRS DAYS M R V D GEN ACCM |
| FAMDED F A 000000000100 000001 000000 Y N Y N 010302 |
| INDDED I A 000000002000 000000 000000 N Y N Y 010301 |
| PFMDED F A 000000000100 000001 000000 Y N Y N 010302 |
| PINDED I A 000000002000 000000 000000 N Y N Y 010301 |
|  |
| PF01=HELP PF03=RETURN |
| PF07=PREV PF08=NEXT PF10=UPDATE PF11=DELETE PF12=CLEAR |

If user needs to give additional Accums, he/she can depress PF08 to insert additional Accums. On PF08, next page will show up as below where user can give the additional Accums.

If the user would like to change an existing benefit record, he/she can depress PF09 after viewing the details in inquiry screen and this will invoke BENEFIT UPDATE SCREEN where all the fields that are present in the inquiry screen will be editable, except key fields source system, coverage plan id, member tier, benefit effective date, version number and last update timestamp. User can change any of the editable fields with valid values and depress PF10 to update the benefit record.

User needs to view the details of the benefit record that he would like to change in inquiry screen and depress PF09 to view the update screen

|  |
| --- |
| GNCOBII WELLPOINT GROUP SYSTEM N 11/25/14 01:48:03 1 |
| SNMSSM6 IZA19020 BENEFIT UPDATE SCREEN VERSION 001 |
|  |
| --------------------------BENEFIT LEVEL INFO ------------------------------- |
| ELIG SYS : 822 CVRG PLAN ID : ACES1234 TIER LVL : 1 |
| START DATE : 01012014 END DATE : 12312014 CNTRCT TYP : P |
| VERSION NBR : 00001 CDHP VENDR : LITES CDHP PROD TYP: H |
| EMBDED : X LAST QUARTER : X CRY OVR RESET: X |
| DED IN OOP  : X DED IN TO OUT : X DED OUT TO IN: X |
| DED PRF PAR : X OOP IN TO OUT : X OOP OUT TO IN: X |
| OOP PRF PAR : X CPY IN OOP : X CPY IN TO OUT: X |
| CPY OUT TO IN : X CPY PRF PAR : X CONS IN OOP : X |
| CONS IN TO OUT: X CONS OUT TO IN: X CONS PRF PAR : X |
| PRDCT TYP : RX ADLT OOP EXCLD: X DUAL UPD IND : X |
| LST UPDT TS : 2014-11-15-21.43.03 |
| --------------------------ACCUM LEVEL INFO --------------------------------- |
| ACCUM T S LIMIT AMT OCCRS DAYS M R V D GEN ACCM |
| FAMDED F A 000000000100 000001 000000 Y N Y N 010302 |
| INDDED I A 000000002000 000000 000000 N Y N Y 010301 |
| PFMDED F A 000000000100 000001 000000 Y N Y N 010302 |
| PINDED I A 000000002000 000000 000000 N Y N Y 010301 |
|  |
| PF01=HELP PF02=ADD PF03=RETURN PF04=BNFTSL PF05=BNFTPR PF06=BNFTNX |
| PF07=PREV PF08=NEXT PF09=CHG/DEL PF12=CLEAR |

User can change value in any of the editable fields in benefit update screen and press PF10

If the user would like to delete an existing benefit record, he/she can depress PF09 from Benefit inquiry screen and this will invoke BENEFIT UPDATE SCREEN where all the fields that are present in the inquiry screen will be editable, except key field’s source system, coverage plan id, member tier, benefit effective date, version number and last update timestamp. User can depress PF11 for delete.

User can depress Pf11 if he/she wants to delete the benefit record.

**Component Details:**

GNCOBIU: GNCOBIU is the driver module for Benefit update screen. The cip-phases and PF key functionalities are handled in the driver module.

GNCOBIUS: GNCOBIUS is the subroutine module for update screen. All the DB interaction functionalities are handled in this module

GNCBNIOS is the I/O routines that performs the insert/update and delete operations to DB

**Below are transactions and PSBs defined for these screens?**

|  |  |  |
| --- | --- | --- |
| Screen | Program | PSB |
| Benefit Update Screen | GNCOBUI | GNCOBUI |

PF Key functionalities:

PF10 to update the benefit record.

PF08 to scroll to next page

PF11 to delete a benefit record

**Forward Sync and backward sync process**

**Forward sync**

**In Scope**

Scope of this process is to keep the WGS ODS DB accumulators in sync with Legacy systems such as CHIPS, CS90, NASCO, FACETS. WGS system receives all Legacy files from different Legacy systems via Legacy NDM.

Following are the different process for WGS systems:

* Do the balancing for the Legacy request file to avoid the incorrect/duplicate request.
* Once the balancing is success, convert the EHB request file to EBS layout to call the EBS subroutine.
* All basic validations in Input EHB request (EBS layout) file like Mandatory fields validation(GNCEBS4S) and Contract validation(GNCEBS5S) are performed and if successful, then GNCEBSRS will call the GNCOD01S module to update the details in Accum ODS DB(DB2 tables which are holding the accumulator information).
* Benefit translation table (DB2 table) will be used to validate the accum names in the input file and to get the benefit maximum amounts.
* Contract and accumulator validation is done using Benefit translation tables and not using spider tables.
* Last Quarter and Carry Over accumulator (LQCO) update is also performed.
* Added logic to handle accums with ‘MCO’ product type code.
* After processing the input request file, the response file will be sent back to source systems via WGS NDM
* Success and Failure reports are created by reading the response file and will be routed to dispatch.

**Out of Scope**

* Over apply status reports for EHB are not in the scope of this project.
* Recycle record processing for EHB are not in the scope of this project.
* Send legacy forward synch Accums (any non WGS/ISG Accums) to LITES for CDHP contracts are not in the scope of this project.
* SPIDER DB2 routines are not in scope of this project.

**Backward Sync Process**

**In Scope**

The scope of this process is to keep the WGS accumulators to be in sync in Legacy accumulator databases (Legacy systems - CHIPS, CS90, FACETS and NASCO). In Backward sync process all the EHB eligible claims that are not processed in Forward sync process for the Legacy Systems (CHIPS, CS90, FACETS and NASCO) will be sending to the corresponding legacy systems. Here the claims will be filtered out from the ODS Delta tables based on the claim source code.

Following are the different process for ODS WGS systems:

* Taking the Dump of ODS Delta table (GNC.ACCUM TRIGGER2) .Here the records will be filtered out based on the Claims source code. Only records with claim source code other than the legacy systems (CS90, CHIPS, FACETS and NASCO) will be processed for backward sync process. GNC.ACCUMR\_TRIGGER2 is a delta table and the table is loaded with the data from ODS DTL (GNC.ACCUMR\_DTL) table on a daily basis.
* Then the Dump of ODS TOTAL trigger table (GNC.ACCUMR\_TOTL\_TRIGGER) is also taken for filtering the EHB eligible accumulators. GNC.ACCUMR\_TOTL\_TRIGGER is a delta table and which is loaded with the data from ODS TOTL (GNC.ACCUMR\_TOTL) table. Only the records with Product type code as TOT and Eligibility source code as Legacy system (CS90, CHIPS, FACETS and NASCO) are consider for Backward Sync process.
* For unloading both ACCUMR\_TRIGGER2 and ACCUMR\_TOTL\_TRIGGER table same timestamp is used in the Select condition and the Time stamp file is usually creates from the ACCUMR\_TRIGGER2 unload file(Highest timestamp in the file)
* The Backward Sync process reads the records from the DTL unload file and get the Matching TOT segment record from the TOTL unload file, then creates the Backward Sync Request file.
* Based on the Legacy system indicator (Eligibility Source system from the ODS total table (GNC.ACCUMR\_TOTL), the backward Sync process classifies EHB request for the different legacy system in different files.
* Request files are sent to legacy systems via WGS NDM.
* Legacy Systems gets the request file and updates the accums and send back the responses file to ODS via Legacy NDM.
* The Balancing Process in WGS side performs the basic balancing using the EHB Response file from the legacy systems to avoid the incorrect/duplicate response.
* If balancing is success, creates Backward Sync response success and failure reports in specified report format and place the same in dispatch.

**Out of Scope**

* Membership validation and Membership information population are not in the scope of this project.
* ODS SYNC Databases, Benefit translation Databases are not in the scope.

**Component Details**

## Forward Sync Process

**GNCEHBRC (Copy book)**

* GNCEHBRC is the new Legacy EHB Layout.
* The GNCEHBRC record length is 4500.

****

**GNC5810 (Input request trailer validation)**

* Read the EHB file.
* Validate the EHB request file.
* The Trailer amount is compared with the total of the amount of Detail records
* The Trailer count is compared with the total number of Detail record count.
* The previous request date and time is validated against the current request date and time
* If the count or amount mismatches or if the previous and current requests date and time matches then the return code is set to 8 and job will stop execution else it will continue executing.

**GNC5800 (Driver Module)**

* It will read the EHB layout detail record one by one.
* If the SENDER-ID, RECEIVER-ID, IN-ELIG-SRC-CD, IN-CLM-SRC-CD has source code value name, then a conversion is done from source code name to source code value by calling the table base GNCSRCDT table.
* Map the EHB records to EBS layout for calling the EBS routine.
* Call EBS routine using the EBS layout request to perform Mandatory field’s validation(GNCEBS4S), and Contract validation(GNCEBS5S)..
* EHB can hold maximum of 20 accums in the request, but EBS request layout can hold only 12 Accums in a single request. If the EHB request have more than 12 accums, the driver module will do the below process.
* Create the EBS request record using the first 12 accums from the EHB request record
* Call EBS routine using the EBS request and OD01S to Insert/Update the EHB request to ODS DBs
* EBS sends the response back to the EHB driver module and the Driver module convert the response back to EHB response layout.
* The EHB driver module creates the second EBS request using the remaining Accums (from 13 to 20 Accums in the EHB request).
* Call EBS routine using the 2nd EBS request and OD01S to Insert/Update the EHB request to ODS DBs
* EBS sends the 2nd response back to the EHB driver module and the Driver module map all the remaining Accums from EBS response file to already created EHB response(as part of the first EHB response)
* If the first call is success, 12 accums will be updated in ODS .If any of the accum updating failed in the second request it will ROLLBACK all the updates for the previous request.
* If both the request for accum updating is successful it will COMMIT the transaction.
* The response for each accum update request in EBS layout will be stored in temporary records.
* The driver module receives the response file from EBS routine in EBS layout then converts back to EHB layout and sends back the response file to legacy system using WGS NDM process.

**GNCOD01S (ODS insert or Update)**

* Receive the Add accum request from GNCEBSRS for EHB contracts.
* Process Each accum one by one in a particular request .
* Check whether all the accums present in the input request are valid by calling GNCEBS5S module. If any of the accums is not present in the Benefit Translation Table then the transaction will get failed with status code ‘94’ and message Transaction failed – Accum is Invalid .
* Check whether the details like HCID,MBR code/Member sequence number, Case number, Accum Name, Accum Suffix Name, Product type code, Coverage plan ID, Claim number, Source Code, Accum Reason code, Accum amount, Accum occur count, Accum days, Network Code, Benefit Start date, Benefit End date, Service start date, service end date for the same record are already available in the accum ODS. If its already present then the Transaction will get failed with the status code ‘97’ and message Transaction failed - Duplicate record) .
* Then check for over apply indicator by checking the sum of input benefit detail amount or visit or days and ODS TOTAL table amount or visit or days is greater than the maximum benefit translation amount, or visit or days.
* Calculate remaining amount/occurrence/days count using the retrieved MAX amount/occurrence/days count of the accum (present in ACCUM\_LEV\_INFO table for each accum)
* For an Embedded contract, the individual accumulators need to be updated till the individual maximum amount and family accumulators are to be updated till their family maximum amount respectively.
* For a non-embedded contract, both the individual and family accumulators are to be updated till their family accumulator maximum amount before the over-apply indicator is set.
* GNCOD02T table has the mapping for FAM to IND conversion for all the required accums
* Pass the input accum and FAM\_IND as key to the table base and get the individual accum
* Get the individual accums maximum by reading the ACCUM\_LEV\_INFO table using GNCEBS5S which inturn calls GNCBNIOS module –
* If the derived accum is not found in the Accum Benefit table, then processing will be terminated
* Subtract TOT amount/days count/occurrence from the Max fields – to calculate the remaining amount.
* Check for Cross apply indicators for valid value .If present call GNCOD02S module for obtaining the corresponding secondary Derived accums.
* ODS is handling only cross-apply for CS90 forward sync process. That only uses the indicators

|  |  |  |
| --- | --- | --- |
| 10 XXX-DED-IN-TO-OUT-IND | INN Deductible in OON Deductible | Y - INN DED applies to OON |
| X - INN DED does not cross apply |
| 10 XXX-DED-OUT-TO-IN-IND | OON Deductible in INN Deductible | Y - OON DED applies to INN |
| X - OON DED does not cross apply |
| 10 XXX-OOP-IN-TO-OUT-IND | INN OOP in OON OOP | Y - INN OOP applies to OON |
| X - INN OOP does not cross apply |
| 10 XXX-OOP-OUT-TO-IN-IND | OON OOP in INN OOP | Y - OON OOP applies to INN |
| X - OON does not cross apply |

.

* Call GNCODIOS for ODS DB (DTL DB and TOTL DB) updation.
* CS90, FACETS or NASCO systems needs to prefix ‘1’ to the member code field (IN-PAT-MBR-CD) while storing in ODS DB. The prefix will be included in Driver module before calling GNCEBSRS.
* Include GNCSACMT look-up to check whether the particular accum in IND or FAM.
* Pad ‘1’ to the Member code field if the Accum is IND .
* Padding is applicable only for IND accums and not for FAM accums.
* Both the primary and secondary accums are updated in the ODS DB.
* The response file will be generated based on the status of the primary accums only.
* Partial updating is not done. Roll back the transactions if any Accum is invalid in the particular request
* Request with Product type code ‘MCO’ will also be updated in to ODS TOTL and DTL table.

**LQCO Update**

We will receive the accum name from calling module. It is validated Benefit translation table and is updated to ODS DB. Below steps should be executed for LQCO update. We will be calculating LQCO accums for claims with reason code

SA, MA, RV, MU, BF

**Date derivation and validation**

1. Calculate the Accum year-end date.

* Get the end date

End date will Benefit Effective termination date in the request.(-IN-BNFT-TRMNTN-DT)

* Identify the LQ dates (3 months ahead of benefit end date)
  + LQ CO effective end date - same as IN-BNFT-TRMNTN-DT
  + LQCO effective date - will be calculated by calling WGMDTCLS

Call WGMDTCLS module with the - WGMDTCLS-SUB-MONTHS function / WGMDTCLS-NBR-MOS function

1. Check whether the service start date falls within LQ dates.

WS-IN-SRVC-START-DT >= WS-LQCO-EFF-DATE and

WS-IN-SRVC-START-DT <= WS-LQCO-END-DATE

Then set the LQ switch to true.

1. Check the LQCO eligibility indicator from Benefit translation table.

|  |  |
| --- | --- |
| LST\_QTR\_IND | PIC X(001). |

Since LQ indicator has possible values

B - LQ applicable for both DED and OOP PAR and NON PAR  
P - LQ applicable for both DED and OOP PAR  
N - LQ applicable for both DED and OOP NON PAR

D - LQ applicable only for DED PAR & NON PAR  
O - LQ applicable only for OOP PAR & NON PAR  
E - LQ applicable only for DED PAR  
F - LQ applicable only for DED NON PAR  
Q - LQ applicable only for OOP PAR  
R - LQ applicable only for OOP NON PAR  
X - LQ not applicable for DED or OOP.

**LQ Accum derivation**

If eligible for LQCO, will call the table for corresponding LQ accums.

* We are using the existing table GNCSLQAT for getting the LQ accumulator names.

**ODS DB updating**

Find the applicable LQ accum name. Perform below steps for the applicable LQ accum name.

* 1. If eligible for ODS update , Populate fields for ODS update
  2. Set the over apply indicator for LQ accum
* Keep the LQ over apply indicator, same as the value for regular accum
  1. Update LQ accum name in total and detail tables for input product type.
  2. If LQ is successfully updated, calculate next benefit period.
     + Call WGMDTCLS module and calculate the next benefit period by adding 12 months to the ben-effective date
  3. Set the over apply indicator for carryover accumulator.
     + Current year benefit maximum amount will be used.
     + Call the TOTL table with the CO effective and End dates and check benefit amount; visit or days in the benefit table and add that with the input request whether it is greater than the amount or visit or occurs in the Benefit table .Based on this over apply indicator will be set.
  4. Insert/Update regular accumulator to total table under product type, TOT, for next benefit period. Update carryover accumulator to ODS tables under new product type MCO for Med, VCO for Vis, RCO for RX and DCO for Dental in total and detail tables.

In General ODS DB update will be as below for LQ and CO accums

|  |  |  |  |
| --- | --- | --- | --- |
| Accum Name | Effective year | Product type code |  |
| INDDED | 2013 | MED,TOT | Current Year accumulator |
| INDDEDLQ | 2013 | MED,TOT | Current Year LQ accumulator |
| INDDED | 2014 | MCO,TOT | Next year Carryover accumulator |

**GNCOD02S (Secondary Accum Derivation)**

* The OD01S driver module will set the corresponding Cross Apply indicators and call GNCOD02S module and set appropriate functions to true.
* Functions are defined inside GNCOD02S module Cross Apply functionality.
* Accums names passed from the ODS driver module will be received, and corresponding key text will be passed based on the triggered function.
* Calls GNCOD02T table for Derived accums based on Accum name and Key text.

**GNC5820 (REPORT)**

* EHB response file generated through the above process are used for creating the reports.
* Report process validates the response and corresponding Success and Failure reports are created.
* Since the response file has numeric value in Claim source code field, conversion from source code value to source system name is performed inside the report program by calling the source table GNCSRCDT.
* Success and failure reports are to be created based on the Accum Transaction status and the error text.
* If the response record’s transaction status is 00(success) and error text is spaces or the transaction status is 05 or 06, these records are routed to Success report
* All other records are routed to Failure reports.
* Both the Success and Failure Reports routed to dispatch.

**Job details to process the request file from CS90**

JOB: **GNC582DD**

* Execute PROC GNC580DD

PROC: GNC580DD

* Step1 - executes GNC5810 module(does the trailer count validation based on the number of records and the total amount)
* Input file : GNCP.NONX.GNC582DD.CS90.FWD.RQST (Legacy Request)

: GNCP.NONX.GNC582DD.GNC5810.CS90.DTE (+0) (Time stamp)

* Output : GNCP.NONX.GNC582DD.GNC5810.CS90.DTE (+1)
* Step 2 - executes GNC5800 module (Driver module which converts the EHB layout to EBS layout and call the EBS routine)
* Input file : GNCP.NONX.GNC582DD.CS90.FWD.RQST (Legacy Request)
* Output : GNCP.NONX.GNC582DD.GNC5800.CS90.RES
* Step 3 - execute GNC5820(which processes the response records and generates Success/ Failure Reports)
* Input file : GNCP.NONX.GNC582DD.GNC5800.CS90.RES
* Output

Success Report: GNCP.NONX.GNC582DD.GNC5820.CS90.SUC

Failure Report : GNCP.NONX.GNC582DD.GNC5820.CS90.ERR

**Job details to process the request file from FACETS**

JOB: **GNC583DD**

* Execute PROC GNC583DD

PROC: GNC583DD

* Step1 - Infogix balancing step.
* Input file : GNCP.NONX.GNC583DD.FACETS.FWD.RQST (Legacy Request)
* Step 2 - executes GNC5800 module (Driver module which converts the EHB layout to EBS layout and call the EBS routine)
* Input file : GNCP.NONX.GNC583DD.FACETS.FWD.RQST (Legacy Request)
* Output : GNCP.NONX.GNC583DD.GNC5800.FCTS.RES
* Step 3 - execute GNC5820(which processes the response records and generates Success/ Failure Reports)
* Input file : GNCP.NONX.GNC583DD.GNC5800.FCTS.RES
* Output

Success Report: GNCP.NONX.GNC583DD.GNC5820.FCTS.SUC

Failure Report : GNCP.NONX.GNC583DD.GNC5820.FCTS.ERR

**Job details to process the request file from CHIPS**

JOB: **GNC584DD**

* Execute PROC GNC580DD

PROC: GNC580DD

* Step1 - executes GNC5810 module(does the trailer count validation based on the number of records and the total amount)
* Input file : GNCP.NONX.GNC584DD.CHIPS.FWD.RQST (Legacy Request)

: GNCP.NONX.GNC584DD.GNC5810.CHPS.DTE (+0) (Time stamp)

* Output : GNCP.NONX.GNC584DD.GNC5810.CHPS.DTE (+1)
* Step 2 - executes GNC5800 module (Driver module which converts the EHB layout to EBS layout and call the EBS routine)
* Input file : GNCP.NONX.GNC584DD.CHIPS.FWD.RQST (Legacy Request)
* Output : GNCP.NONX.GNC584DD.GNC5800.CHPS.RES
* Step 3 - execute GNC5820(which processes the response records and generates Success/ Failure Reports)
* Input file : GNCP.NONX.GNC584DD.GNC5800.CHPS.RES
* Output

Success Report: GNCP.NONX.GNC584DD.GNC5820.CHPS.SUC

Failure Report : GNCP.NONX.GNC584DD.GNC5820.CHPS.ERR

**NDM –Forward Sync EHB Response files to legacy**

**CS90**

JOB: **GNC582TD**

* Calls the PROC : GNC582TD

PROC: GNC582TD

* Receive EHB Legacy request, GNCP.NONX.GNC582DD.GNC5800.CS90.RES from CS90 legacy system.
* Control card used is: GNC582TD

**FACETS**

JOB: **GNC583TD**

* Calls the PROC : GNC583TD

PROC: GNC583TD

* Receive EHB Legacy request, GNCP.NONX.GNC583DD.GNC5800.FCTS.RES from FACETS legacy system.
* Control card used is, GNC583TD

**CHIPS**

JOB: **GNC584TD**

* Calls the PROC : GNC584TD

PROC: GNC584TD

* Receive EHB Legacy request, GNCP.NONX.GNC584DD.GNC5800.CHPS.RES from CHIPS legacy system.
* Control card used is: GNC584TD

**One Time Accum Load Process**

* Legacy Systems will not need separate file name or job for initial accum load and forward synch file.
* The same process/same job for forward synch is only required to handle initial accum load. Also the same dispatch or report id can be used.
* Whenever initial accum load file is NDM’ed to ODS. Forward synch jobs will get triggered and update details in ODS DB.

## Backward Sync Process

**GNC9510 (Unload the detail (GNC.ACCUMR\_TRIGGER2 table) DB**

* GNC9510 is an existing ODS DTL delta table unload query generating module for HIX ODS Project can be reused for the EHB process as well.
* GNC.ACCUMR\_TRIGGER2 DB2 table is the Accum Delta table for ODS DTL(GNC.ACCUMR\_DTL) table. Whenever there is an insert in ODS DTL table, it executes the respective trigger to insert the records in Delta tables as well.
* There is an existing process to unload the Delta table on an hourly basis. This was introduced as part of HIX ODS project.
* The same unload data file can be used for Accum Delta table, with additional two fields of Benefit Occurrence count and benefit Day count. The corresponding query to unload the ODS DTL Delta table is modified to update this.
* The Unload process also keep the timestamp information for the last unload records to make sure that the next unload process is unloading the Delta table records, having time stamp greater than this time stamp.
* This is an existing program which is used for the current HIX ODS and P2P projects.

New fields: DTL\_BNFT\_OCR\_CNT , DTL\_BNFT\_DAY\_CNT, ACCUMR\_RSN\_CD

**GNC5910 (Unload the Total (GNC.ACCUMR\_TOTL\_TRIGGER table) DB**

* GNC.ACCUMR\_TOTL\_TRIGGER DB2 table is the Accum Delta table for ODS TOTL(GNC.ACCUMR\_TOTL) table. Whenever there is an insert in ODS TOTL table, it executes the respective trigger to insert the records in Delta tables as well.
* A new process required to unload the Delta table based on the Backward Sync request file.
* The unload process will take the dump of the entire TOTL delta table based on the latest timestamp.
* The Unload process also use the timestamp information for the last unload records(ODS DTL) to make sure that the next unload process is unloading the Delta table records, having time stamp greater than this time stamp.

**GNC5920 (Backward Sync Request Creation Process)**

* Input : GNC.ACCUMR\_TRIGGER2 unload file , GNC.ACCUMR\_TOTL\_TRIGGER unload file and the time stamp file
* Read each record from the DTL unload file and process the only records with Claim source code not as Legacy system (CS90, CHIPS, FACETS and NASCO).
* Read each record from the TOTL unload file and validate whether the key fields are matching in both records.
* Total Accum amount, Total Occurrence Count, Total Day count, Max Accum amount, Max Occurrence Count, Max Day count and Eligibility Source Code are populated from ODS DB by taking the dump of ODS TOTL Delta table.
* The Accum type is retrieved from GNCSACMT table base for each of the Accumulators.
* LQ Accums and Accums with Accum Reason Code ‘MNO’ (Manual Update through Screens) would not be considered while processing the Backward Sync Request Files.
* To convert the system source names to code values or vice versa, GNCSRCDT table base is used.
* Move all the mandatory fields to the EHB layout (request) from Delta table unload files and created the EHB Backward Sync request file.
* After populating all values in EHB layout the request file, for each system (based on the Eligibility source code from the total table) creates separate Legacy Backward sync requests.

**GNC5930 (EHB response trailer validation)**

* Read the EHB response file from the legacy system via Legacy NDM.
* The Trailer amount is compared with the total of the amount of Detail records
* The Trailer count is compared with the total number of Detail record count.
* The previous response date and time is validated against the current response date and time
* If the count or amount mismatches or the previous and current response date and time matches, then the return code is set to 8 and job will stop execution else it will continue execution.

**GNC5940 (Report Generation)**

* Input for the report program will be the response file in EHB layout after the successful balancing.
* Validate the response and corresponding Success and Failure reports are created.
* Success and failure reports are created based on the Accum Transaction status and the error text.
* Since the response file has numeric value in Claim source code field, conversion from source code value to source system name is performed inside the report program by calling the source table GNCSRCDT.
* If the transaction status is success and error text is spaces or error text is having value as

‘DOLLAR WILL NOT GET UPDATE TO ODS’ (LEGACY-NON-COMING-ERR), those records are written into Success file.

* All other records write to Failure reports.
* Both the Success and Failure Reports will be routed to dispatch.

**Job details for Total table unload file creation**

JOB: **GNC59\*DD** (\* 🡺 A to H)

* Execute PROC GNC591DD

PROC: GNC591DD

* Step1: - executes program GNC5910 module to dynamically prepare the SQL query
* Step 2 - executes the SQL query and get data from accumulator delta table.
* Step 3- Create timestamp file with highest time stamp.
* Step 4- This will copy timestamp from previous version if time stamp file generated is empty.

Output :- GNCP.NONX.GNC591DD.UNLD.TOTL.TGR

CTC: **GNC59101**

* Sort card to get the lowest time stamp from detail DB

**EHB backward sync request creation process.**

JOB: **GNC59\*ND** (\* 🡺 A to H)

* Execute PROC GNC591ND

PROC: GNC591ND.

* Step 1 : Concatenate the previous generation of unload file for EHB backward sync process from the job GNC951DD.
* Step 2 : Sort the detail unload file
* Step 3 : Sort the total unload file
* Step 4: IEFBR14. Create new versions for the 4 legacy request files such as CHIPS, CS90, FACETS and NASCO
* Step 5: Execute Program GNC5920 to create the request file for the 4 Legacy systems.

Input file: Sorted DB unload files from Step2 and Step 3

Output file: GNCP.NONX.GNC591ND.GNC5920.CS90.RQT

GNCP.NONX.GNC591ND.GNC5920.FCTS.RQT

GNCP.NONX.GNC591ND.GNC5920.CHPS.RQT

GNCP.NONX.GNC591ND.GNC5920.NSCO.RQT

GNCP.NONX.GNC591ND.GNC5920.ERR.FILE

* Step 6 : Abend validation

CTC: **GNC59201**

* Sort card to Sort the detail unload records and extract only the records with claim src code other than legacy systems (CS90,CHIPS,FACETS,NASCO)

CTC: **GNC59202**

* Sort card to Sort the total unload records and extract only the records with product type code TOT and Eligible source code legacy systems (CS90,CHIPS,FACETS,NASCO)

JOB: **GNC59IND**

* Execute PROC GNC59IND

PROC: **GNC59IND**.

* **Step 1:** This step will copy the CS90 file to a backup file
* **Step 2:** Writes the header into sort1 temporary file
* **Step 3:** Writes the detail records into sort2 temporary file
* **Step 4:**Writes the trailer records into sort3 temporary file
* **Step 5:** Concatenate the header, detail and trailer records into the next version of the CS90 request file
* **Step 6:** Take the backup of the entire CS90 request file processed on that day.
* **Step 7:** Delete all previous generation of CS90 request file.

CTC: **GNC59I30**

* Sort card to include the records with ‘DTL’ from the input file and also sort the Accum Reason Code Field in descending order, so as to send the Original Claims before its Reversal Claims.

CTC: **GNC59I20**

* Sort card to include header records for the CS90 file.

CTC: **GNC59I40**

* Sort card to include trailer records for the CS90 file

**NDM - Backward Sync EHB Request files to legacy**

**CS90**

JOB: **GNC592TD**

* Calls the PROC: GNC592TD.

PROC: **GNC592TD**

* Send EHB Legacy request, GNCP.NONX.GNC592DD.GNC5920.CS90.RQT to CS90 legacy system.
* Control card used is GNC592TD

CTC : **GNC592TD**

* Control card for CS90 NDM Process

**FACETS**

JOB: **GNC593TD**

* Calls the PROC : GNC593TD

PROC: **GNC593TD**

* Send EHB Legacy request, GNCP.NONX.GNC592DD.GNC5920.FCTS.RQT to FACETS legacy system.
* Control card used is: GNC593TD

CTC : **GNC593TD**

* Control card for FACETS NDM Process

CTC : **GNC593S1**

* Sort card for Timestamp population for FACETS REQ file name

**CHIPS**

JOB: **GNC594TD**

* Calls the PROC : GNC594TD

PROC: GNC594TD

* Send EHB Legacy request, GNCP.NONX.GNC592DD.GNC5920.CHPS.RQT to CHIPS legacy system.
* Control card used is: GNC594TD

CTC : **GNC594TD**

* Control card for CHIPS NDM Process

**Job details to process the response file from CS90- Balancing and Report Generation**

JOB: **GNC592ND**

* Execute PROC GNC592ND

PROC: GNC592ND

* Step1 - executes GNC5930 module (does the trailer count validation based on the number of records and the total amount).

Input file: GNCP.NONX.GNC592ND.CS90.BKWD.RES (Legacy Response)

: GNCP.NONX.GNC592ND.CS90.DTE (+0) (Time stamp)

Output : GNCP.NONX.GNC592ND.CS90.DTE(+1)

* Step 2 - execute GNC5940, which processes the records and generates Success/ Failure Reports.

Input file : GNCP.NONX.GNC592ND.CS90.BKWD.RES

Output

Success Report: GNCP.NONX.GNC592ND.GNC5940.CS90.SRT

Failure Report : GNCP.NONX.GNC592ND.GNC5940.CS90.FRT

**Job details to process the response file from FACETS- Balancing and Report Generation**

JOB: **GNC593ND**

* Execute PROC GNC593ND

PROC: GNC593ND

* Step1 - executes GNC5930 module (does the trailer count validation based on the number of records and the total amount).

Input file: GNCP.NONX.GNC593ND.FACETS.BKWD.RES (Legacy Response)

* Step 2 - execute GNC5940, which processes the records and generates Success/ Failure Reports.

Input file : GNCP.NONX.GNC593ND.FACETS.BKWD.RES

Output

Success Report: GNCP.NONX.GNC593ND.GNC5940.FCTS.SRT

Failure Report : GNCP.NONX.GNC593ND.GNC5940.FCTS.FRT

**Job details to process the response file from CHIPS- Balancing and Report Generation**

JOB: **GNC594ND**

* Execute PROC GNC594ND

PROC: GNC594ND

* Step1 - executes GNC5930 module (does the trailer count validation based on the number of records and the total amount).

Input file: GNCP.NONX.GNC594ND.CHIPS.BKWD.RES (Legacy Response)

: GNCP.NONX.GNC594ND.CHIPS.DTE (+0) (Time stamp)

Output : GNCP.NONX.GNC594ND.CHIPS.DTE(+1)

* Step 2 - execute GNC5940, which processes the records and generates Success/ Failure Reports.

Input file : GNCP.NONX.GNC594ND.CHIPS.BKWD.RES

Output

Success Report: GNCP.NONX.GNC594ND.GNC5940.CHPS.SRT

Failure Report : GNCP.NONX.GNC594ND.GNC5940.CHPS.FRT

**Sunday only Job to support backward sync: GNC591NW**

* The total unload is taken using the timestamp generated from the detail trigger unload 3 hours file from the job GNC59HDD. Since the job to take detail unload does not runs on Sundays a new job GNC591NW is created. This job is used to create the timestamp for unloading the total trigger table on Sundays.
* Scheduled only on Sunday
* This job will create a blank version for ACCUMR\_TRIGGER2 unload 3 hour file for EHB backward sync processing.

**Monday only job to support backward sync: GNC592NW**

* On Sundays 2 detail trigger unload job are executing before GNC59HDD job runs. Since the job for creating the backward request is not running on Sundays there are chances that we miss to create the backward sync request for these claims. So the new job GNC592NW is introduced. This will create a blank version for backward sync request creation detail unload file. So when the next job GNC59ADD runs the timestamp created in the previous job will be taken for creating the 3 hour file and thus we will not miss any claims from creating backward sync request.
* Scheduled only on Monday
* This job will create a blank version for accumr\_trigger2 unload 3 hour file for EHB backward sync processing.

**ESI feed file creation**

* Removed the step which unload the ACCUMR TOTL TRIGGER table for ESI daily feed file creations. Instead the file unloaded in the job GNC59\*DD will be used as the input.
* Included a step to take a daily backup of all the generations of db unload file
* Added a sort card to include only the records with Product type code as ‘TOT’
* Added a sort card to omit the record with member code as ‘000’ or ‘ ‘
* Included a step to delete all the generations of db unload file

JOB: **GNC947BD**

* Execute PROC GNC947BD

Proc GNC947BD

* Step 1 – IEBGENER, which takes the daily backup of unload file
* Step 2 – Sort the daily unload file and include on the records with product type code as ‘TOT’
* Step 3 – Sort the file from above step and omit the records with member code as ‘000’ and ‘ ‘
* Step 9 - IEFBR14, which deletes all the generation of db unload file\

Rest of the process is not altered.

CTC : **GNC947S1**

* Sort card to include only the records with product type code as 'TOT' and no duplicate records

CTC : **GNC947S4**

* Sort card to omit the records with member code as '000' and ' '

JOB: **GNC630DD**

* Execute PROC GNC630DD

Proc GNC630DD

* Step 1 – IEBGENER, which takes the daily backup of unload file
* Step 2 – Sort the daily unload file and include on the records with product type code as ‘TOT’
* Step 3 – Sort the file from above step and omit the records with member code as ‘000’ and ‘ ‘

Rest of the process is not altered.

CTC : **GNC630S1**

* Sort card to omit the records with member code as '000' and ' '

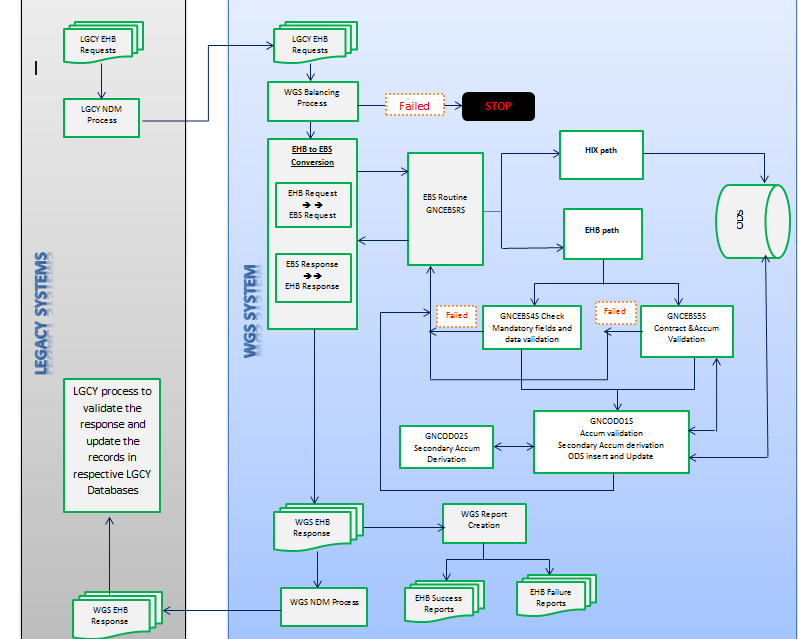
**NDM file Details**

<https://kubera.ust-global.com/accounts/wpclaims/WELL-0928-71-00/Work%20in%20progress/Reference%20Documents/EHB-LEGACY/EHB_Legacy/LG_EHB-Accum_ODS_Legacy_Accum_Exchange_Details.xlsx>

**Process Flow**

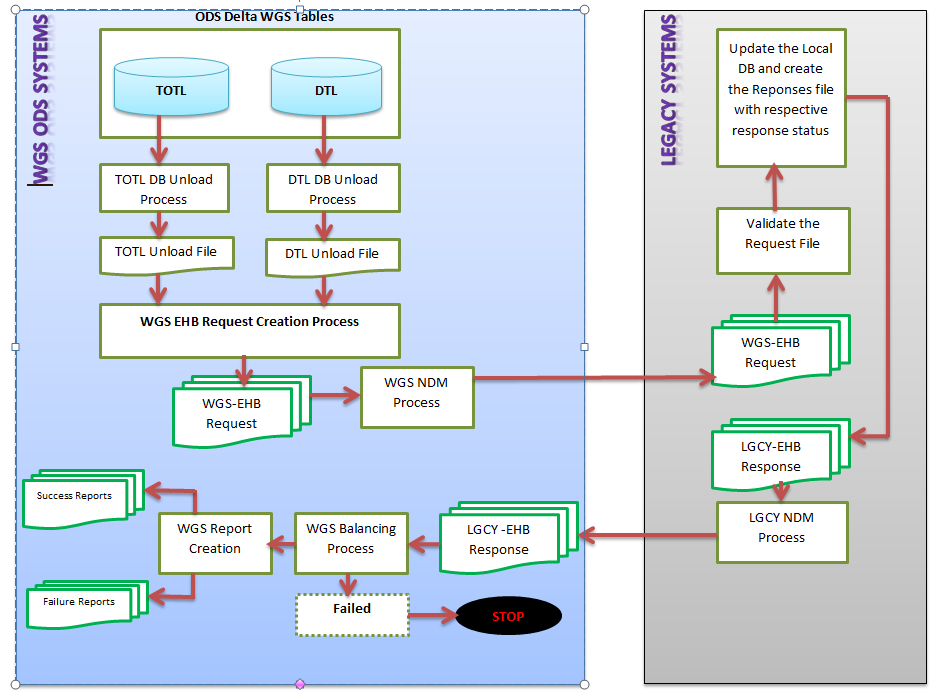
Forward Sync Process

Flow Diagram

****

Backward Sync Process

**Flow Diagram**



**Backward sync job scheduling flow**



**Report Details**

The below reports will be created as part of EHB Legacy Forward and Backward sync process.

|  |  |  |  |
| --- | --- | --- | --- |
| **Process** | **JOB Name** | **Report Description** | **Dispatch ID** |
|  | GNC582DD | Legacy success report /Failure report generation for CS90 | Suc:GNC58201 Fail: GNC58202 |
| Forward Synch | GNC583DD | Legacy success report /Failure report generation for FACETS | Suc:GNC58301 Fail: GNC58302 |
|  | GNC584DD | Legacy success report /Failure report generation for CHIPS | Suc:GNC58401 Fail: GNC58402 |
|  | GNC585DD | Legacy success report /Failure report generation for NASCO | Suc:GNC58501 Fail: GNC58502 |
|  | GNC592ND | Legacy success report /Failure report generation for CS90 | Suc: GNC59201 Fail: GNC59202 |
|  | GNC593ND | Legacy success report /Failure report generation for FACETS | Suc: GNC59301 Fail: GNC59302 |
| Backward Synch | GNC594ND | Legacy success report /Failure report generation for CHIPS | Suc: GNC59401 Fail: GNC59402 |
|  | GNC595ND | Legacy success report /Failure report generation for NASCO | Suc: GNC59501 Fail: GNC59502 |

**Over Applied Report Generation.**

**In Scope**

Scope is limited to the generation of over applied report. Over applied report is generated or the below systems:

* HIX
* PEX
* ACES
* FACETS
* CHIPS
* CS90
* NASCO
* HEALTH LINK

Following is the process flow:

Generate the dump file for Accum Trigger Table

Run the program GNC9665 to generate the over apply reports for other systems.

Run the program GNC9666 to split the over applied report for ACES, FACETS, NASCO, CS90, CHIP and Health Link and routing the reports to dispatch

Run the program GNC9661 to route the HIX and PEX over apply reports to DISPATCH and send the email content to stake holders.

Run the program GNC9660 to generate the over applied report for HIX and PEX

* Dump file will be created with the records form GNC.ACCUMR\_TRIGGER (This is the trigger table for the db2 table, GNC.ACCUMR\_DTL).
* GNC9660 program will be run to validate the Member/contract details for the records in dump file for which over applied indicator is set.
* Generate the Over applied report for HIX and PEX and error file for the records that failed in the validation
* Run the program GNC9661 to route the reports to dispatch and email will be sent to stake holders thru GNP8383.
* GNC9665 program will be run to validate the contract/accum details from benefit tables/accum ODS tables for the legacy records.
* Generate the over applied entries for legacy systems and error files which failed in validation.
* Run the program GNC9666 to split the over applied reports to different systems and route the reports to dispatch.

**Out of Scope**

* Member validation for legacy over applied entries.

**Component Details**

**GNC9660C (Copy book)**

* GNC9660C is the copy book for the report lay out for HIX and PEX.
* Record length is 245 for the over applied report.

**GNC9660 (SRC)**

* This program will read the accum trigger table dump file to generate the over applied report for HIX and PEX.
* Validate the member/contract details thru membership and contract DB call.
* Generate error report for the records that fails in validation.
* If the SRC SYS IND is ISG then write the valid records to HIX over applied report.
* If the SRC SYS IND is WGS then write the valid records to PEX over applied report.

**GNC9661 (SRC)**

* This program is to convert the over applied report in report layout.
* Message template will be created for the email to be sent to stake holders after sending the over applied report.

**GNC9665 (SRC)**

* This program will read the accum trigger table dump file to generate the over applied report for legacy systems.
* Validate the contract/accum details thru I/O routines GNCBNIOS and GNCODIOS.
* Generate the error report for the records that fails in validation.

**GNC9666 (SRC)**

* This program split the legacy over applied report for different systems. Over applied reports for the following systems will be generated:
  + ACES
  + FACETS
  + NASCO
  + CS90
  + HEALTH LINK
  + CHIPS
* Over applied reports will be routed to DISPACTH.

**Job details to generate the Over applied Report.**

Job GNC966ID will be run to generate the over applied report for HIX, PEX and Legacy Systems.

**Input file** - NA (Dump file will be generated for the DB2 table GNC.ACCUMR\_TRIGGER (This is the trigger table for the db2 table, GNC.ACCUMR\_DTL).

**Output files** - GNCP.NONX.GNC966ID.DUMP.ACCMTRGR.G\*

GNCP.NONX.GNC966ID.GNC9660.RPT.G\*

GNCP.NONX.GNC966ID.GNC9660.PEX.RPT.G\*

GNCP.NONX.GNC966ID.GNC9660.ERRFILE.G\*

GNCP.NONX.GNC966ID.GNC9665.RPT.G\*

GNCP.NONX.GNC966ID.GNC9665.ERRFILE.G\*

This job executes the PRC GNC966ID:

* + - 1. STEPU010: Generates the dump file for ACCUM TRIGGER TABLE.
         1. Input - GNC.ACCUMR\_TRIGGER
         2. Output - GNCP.NONX.GNC966ID.DUMP.ACCMTRGR.G\*
      2. GNC96601: Runs the program GNC9660 to generate the over applied report for HIX and PEX.
         1. Input - GNCP.NONX.GNC966ID.DUMP.ACCMTRGR(0)
         2. Output - GNCP.NONX.GNC966ID.GNC9660.RPT.G\*

GNCP.NONX.GNC966ID.GNC9660.PEX.RPT.G\*

GNCP.NONX.GNC966ID.GNC9660.ERRFILE.G\*

* + - 1. GNC96602: Sort the HIX over applied report in the descending order of amount.
      2. GNC96603: Runs the program GNC9661 to process and route the HIX over applied report to DISPATCH.
      3. STEP0010: Sort the PEX over applied report in the descending order of amount.
      4. STEP0020: Run the program GNC9661 to process and route the PEX over applied report to Dispatch.
      5. STEPP030: Runs the program GNC9665 to generate the over applied report for legacy systems (ACES, FACETS, CS90, CHIPS, NASCO and HEALTHLINK).
         1. Input - GNCP.NONX.GNC966ID.DUMP.ACCMTRGR(0)
         2. Output - GNCP.NONX.GNC966ID.GNC9665.RPT.G\*

GNCP.NONX.GNC966ID.GNC9665.ERRFILE.G\*

* + - 1. STEPS040: Sort the legacy over applied report in the descending order of amount and age.
      2. STEPP050: Runs the program GNC9666 to process the legacy over applied report and split the over applied report for different systems (ACES, FACETS, CS90, CHIPS, NASCO and HEALTH LINK).
         1. Input - GNCP.NONX.GNC966ID.GNC9665.RPT.G\*
         2. Output - over applied report for legacy systems and routed to DISPATCH.
      3. GNC96604: Sort to include date in the email template.
      4. GNC96605: Run the program GNP8383 to send the email notification for HIX and PEX report.

**Over applied reports**

Following table have the details of over applied reports for different systems and their report IDs:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **System name** | **Job Name** | **Proc** | **Report ID** | **Remarks** |
| HIX | GNC966ID | GNC966ID | GNC96601 | Over applied report for HIX |
| PEX | GNC96602 | Over applied report for PEX |
| FACETS | GNC96606 | Over applied report for FACETS |
| ACES | GNC96605 | Over applied report for ACES |
| CHIPS | GNC96607 | Over applied report for CHIPS |
| CS90 | GNC96608 | Over applied report for CS90 |
| NSACO | GNC96609 | Over applied report for NASCO |
| HEALTH LINK | GNC96610 | Over applied report for HEALTH LINK |