|  |  |
| --- | --- |
| **Table** | **Jobs details** |
| **Job Set 1** | **RMBAPP\_PRD\_TPW\_PROCESS** |
|  | RMBAPP\_TPW\_RESTART1 |
|  | RMBAPP\_TPW\_RESTART2 |
| **Job Set 2** | **RMBAPP\_PRD\_TFM\_DISAGREEGATION** |
|  | RMBAPP\_IDENTIFY\_AFFCT\_TRANS |
|  | RMBAPP\_PROCESS\_NON\_AGG\_TRANS |
|  | RMBAPP\_DEAGG\_CLEANUP |
|  | RMBAPP\_UPDATE\_DEAGG\_REQUEST\_STATUS |
|  | RMBAPP\_DISC\_ARR\_DEAGG |
| **Job Set 3** | **RMBAPP\_PRD\_TFM\_AGREEGATION** |
|  | RMBAPP\_ROLLBACK |
|  | RMBAPP\_FLUSH\_CACHE |
|  | RMBAPP\_HEADER\_VALIDATION |
|  | RMBAPP\_VALIDATE\_PRODUCT\_ASSIGN |
|  | RMBAPP\_PRODUCT\_PRICING\_VERFY |
|  | RMBAPP\_UPDATE\_STATUS |
|  | RMBAPP\_RATING\_BATCH\_PRODUCTIZED |
|  | RMBAPP\_MARK\_COMPLETION |
|  | RMBAPP\_AGG\_CLEANUP |
| **Job Set 4** | **RMBAPP\_PRD\_DISC\_ARR** |
|  | RMBAPP\_ACCUM\_DISC\_ARR |
|  | RMBAPP\_CREATE\_BILL\_CHG\_DISC |
| **Job Set 5** | **RMBAPP\_PRD\_CSADM\_UPD** |
|  | RMBAPP\_CISADM\_STAT\_UPD |
| **Job Set 6** | **RMBAPP\_PRD\_DB\_STUPD** |
|  | RMBREP\_REP\_STAT\_UPD |
|  | RMBHUB\_HUB\_STAT\_UPD |
| **Job Set 7** | **RMBAPP\_PRD\_COLLECTIONS** |
|  | RMBAPP\_PROMISE\_TO\_PAY\_MONITOR |
|  | RMBAPP\_COLLECTIONS\_OVERDUE\_MONITOR |
|  | RMBAPP\_COLLECTIONS\_OVERDUE\_CUTEVENT |
| **Job Set 8** | **RMBAPP\_PRD\_BILLING** |
|  | RMBAPP\_ACTIVATE\_CONTRACTS |
|  | RMBAPP\_CREATE\_BILLS |
|  | RMBAPP\_WRITEOFF\_ADJ\_SRA\_CUST |
| **Job Set 10** | **RMBREP\_PRD\_DLY** |
|  | ORMB\_DLY\_PEND\_DOC |

**TFM AGG Batch Jobs:**

**Initial Product Derivation Batch (C1-TXNIP) :**

This will copy data from staging table to CI\_TXN\_DTL. Invoke the defined rules. Check the CI\_TXN\_DTL\_PRITM table if transactions are processed and inserted into this table.

**Price Assignment Batch (C1- TXNVP):**

This batch derives price assignment for each of the Account Id – Product combination. Successful execution of this batch should populate the CI\_TXN\_DTL\_PRITM\_SUMMARY table.

**Update Error Batch (C1-TXNEX):**

This is error handling batch which marks the transactions in error or ignore status if the pricing is not available or ignore switch is Y respectively. Successful execution of this batch should mark all the transactions in CI\_TXN\_DETAIL table in EROR or IGNR status for which pricing could not be derived or ignore switch flag is Y respectively.

**SQ Calculation Batch (C1-TXNSQ):**

Successful completion of this batch would create the distinct billable charges and populate the CI\_BILL\_CHG. It will update billable charge id in CI\_TXN\_DTL\_PRITM table against corresponding transactions.

**Finalize Transaction Batch (C1-TXNCM):**

This batch marks the transaction either into COMP (i.e. COMPLETE) or EROR (i.e. ERROR) status, if the transaction has been processed successfully by all the batches.

**Transaction Clean up Batch (C1-TXNCU):**

This batch is used to clean up or recalculate the billable charges if the transaction is marked in EROR while running the C1-TXNCM batch. Purges the CI\_TXN\_DTL\_PRITM\_SUMMARY table.

## **Invoice Generation Process in Detail:**

       ORMB produces an invoice extract to facilitate the generation of the invoice summary and supporting reports which are sent out to WellPoint customers.  The invoice extract contains data specific to the bill generated, as well as the list of reporting packages that apply to the customer.

       Business Objects, to generate the supporting reports based on the reporting packages applicable to the customer. ORMB includes the bill id on the invoice extract.

       A link is available in the system to allow users to view the invoice summary and supporting reports send out to the customer



       The above sequence of events triggered by the Control M batch job (DAILY PEND DOC).

       The below logic is implemented in the JAVA file RMBREPORTVIEWER-0.0.1-SNAPSHOT.jar

**Step 1:**

  Daily Pending Doc job is triggered to generate invoices and reports. Two kinds of reports are generated via. Internal Reports and External Reports. External reports are used by customers while internal reports are used by Anthem.

  BPE (Bill Print Extract) table contains the information ready for pending invoice generation. BPE table is joined with customer setup tables to generate reports. All kinds of reports generated for different accounts might not be used where sometimes the generated Invoice might have zero dollars, therefore such unused reports are subject to ORMB suppression rules.

  The CI\_BILL table which loads data into BPE table contains all invoices and they are classified into Pending invoices and Finalized invoices.

**Step 2:**

  Reports are generated in chunks from the data in BPE and customer setup tables. A chunk of 12 reports are generated at a time and sent to the BO server to generate the output XML invoices. These invoices are then ftp’d to the UNIX server for storage. The invoices are also generated in the form of pdf files.

  **Suppression Logic:** This Logic suppress the $0 invoice reports so that the client does not see a blank report. Query generates the output which has account ID, invoice ID, RS\_CD, Price Item code. Suppression logic is handled in the JAVA code.

**Step 3:**

  URL is generated for the user in the RMB application for the pending Invoices which directs user to reporting UI.

**Step 5:**

  The generated pdf pending invoices are accessed by the business team through a UI. The invoices are review, finalized and then dispatched to the customers. In case of errors, the invoice is discarded and a request is placed by the business team to generate the correct invoice which is accomplished by the RMB DISAGG process.