Logo_PUMA2Detailed Design Specification

**RCM Data Integration - HCA and HCP**

Technical Design Document

January 2016

|  | **Name / Project Role** | **Signature** | **Date *(dd/mmm/yyyy)*** |
| --- | --- | --- | --- |
| **Author** | Biswajit Majumdar / Tanmay Behera  Lead Developer | |  |
| **Reviewer** | Kumar Shangumam  Solution Architect | |  |
| **Reviewer** |  | |  |
| **Approver[[1]](#footnote-1)\*** | Sangeeta Pillai  Integration Hub Application Lead | |  |
| Danny Aranha  IPL Lead | |  |

Distribution List

| **Names & Department** | **Rationale** |
| --- | --- |
|  |  |

Version History

| **Version** | **Date *(dd/mmm/yyyy)*** | **Modification** |
| --- | --- | --- |
| 0.1 | 05-Feb-16 | Initial Draft |
| 0.2 | 16-Feb-16 | Updated with TIBCO processing |
| 1 | 03-Mar-16 | 1st published version for review |
|  |  |  |
|  |  |  |

**TABLE OF CONTENTS**

[1. Document Purpose 4](#_Toc447098066)

[1.1. Glossary 4](#_Toc447098067)

[2. Introduction 4](#_Toc447098068)

[2.1. Background and Overview of Service 4](#_Toc447098069)

[2.2. Technology Stack 5](#_Toc447098070)

[3. Requirement Compliance 5](#_Toc447098071)

[3.1. Functional Requirements 5](#_Toc447098072)

[3.2. Source Details 5](#_Toc447098073)

[4. design decisions 6](#_Toc447098074)

[5. e2e architecture 6](#_Toc447098075)

[6. high level design 7](#_Toc447098076)

[6.1. Inbound Customer Only Flow 7](#_Toc447098077)

[6.2. Inbound Sales Data Flow 9](#_Toc447098078)

[6.3. Data flow from Database 11](#_Toc447098079)

[6.4. File Transfer From SAP File Server 13](#_Toc447098080)

[6.5. Email Attachment 13](#_Toc447098081)

[6.6. Real Time API Call 14](#_Toc447098082)

[7. Detailed Technical Design 15](#_Toc447098083)

[7.1. Job Automation Process Flow 16](#_Toc447098084)

[7.2. INFA BDE Environment 17](#_Toc447098085)

[7.3. TIBCO Environment 29](#_Toc447098086)

[7.3.1. DATABASE AS SOURCE 29](#_Toc447098087)

[7.3.2. SAP FILE SERVER AS SOURCE 29](#_Toc447098088)

[7.3.3. EMAIL AS SOURCE 30](#_Toc447098089)

[8. Standard and Rules 31](#_Toc447098090)

[10. Technical Component Level specification: 34](#_Toc447098091)

[10.1 : Customer data file processing for AS-IS passthrough files 34](#_Toc447098092)

[10.2: Customer data file processing for transaction data with customer data 35](#_Toc447098093)

[10.3: Customer data file processing for transaction data and customer data 35](#_Toc447098094)

# Document Purpose

The purpose of the Detailed Specification Document is to provide a complete and detailed description of the **RCM Data Integration** application within the target technical environment: software components, file structure, databases, interfaces.

## Glossary

| **Term or Acronym** | **Definition** |
| --- | --- |
| Quantum | Sanofi-Aventis pharmaceutical operations data warehouse. Comprises of CCT and DSS |
| CCT | Stands for Central Customer Table and is the Pharma Customer Master and stores source records and the Commonized CCT records |
| DSS | Decision Support System and is the commercial datawarehouse supporting Sales and Transactions. |
| HCP | Health Care Professionals |
| HCA/O | Health Care Accounts/ Organizations. These are Institutions that contract for, purchase, and/or dispense medications. This includes Group Practices, Hospitals, Departments, Pharmacies, Plans, Payors, Pharmacy Benefit Managers, Group Purchasing Organizations, Long Term Care facilities, and Wholesalers |
| Integration HUB | HDFS data store |
| MFT | Managed File Transfer |
| RCM | Regional Customer master |
| Inbound flow | The inbound flow mentioned here covers the file ingestion from source to Integration hub and delivered to RCM |
| Infa BDE | Informatica Big Data Edition |
| HDFS | Hadoop Distributed File system |

# Introduction

## Background and Overview of Service

Sanofi embarked on a journey to replace its internally developed Customer Master application (CCT) with new cloud based customer master solution Reltio to manage customer (HCP and HCO) profile and related affiliation.

The objective of the Project is to implement an integration processes to exchange customer data with the RCM provider via Integration Hub and support batch and real-time data integration with RCM and the relevant downstream applications.

## Technology Stack

The technology stack document is available in the following link:

[Integration Hub Technology Stack](http://communitiesemea.sanofi.com/communities/Commercial_Capabilities_Program/ODS/_layouts/PowerPoint.aspx?PowerPointView=ReadingView&PresentationId=/communities/Commercial_Capabilities_Program/ODS/Shared%20Documents/50%20-%20Information/51%20ODS%20Documentation/Integration%20Hub%20Technology%20Stack.pptx&Source=http%3A%2F%2Fcommunitiesemea%2Esanofi%2Ecom%2Fcommunities%2FCommercial%5FCapabilities%5FProgram%2FODS%2FShared%2520Documents%2FForms%2FAllItems%2Easpx%3FRootFolder%3D%252Fcommunities%252FCommercial%255FCapabilities%255FProgram%252FODS%252FShared%2520Documents%252F50%2520%252D%2520Information%252F51%2520ODS%2520Documentation&DefaultItemOpen=1&DefaultItemOpen=1)

# Requirement Compliance

## Functional Requirements

HCA Inbound FRS:

<http://communitiesemea.sanofi.com/communities/Commercial_Capabilities_Program/Regional-Customer-Master/_layouts/WordViewer.aspx?id=/communities/Commercial_Capabilities_Program/Regional-Customer-Master/Shared%20Documents/60-%20Execution/20%20Requirements/RCM%20Integration%20-%20Functional%20Specifications%20-%20HCA%20Inbound%2012Feb2016.docx&Source=http%3A%2F%2Fcommunitiesemea%2Esanofi%2Ecom%2Fcommunities%2FCommercial%5FCapabilities%5FProgram%2FRegional%2DCustomer%2DMaster%2FShared%2520Documents%2FForms%2FAllItems%2Easpx%3FRootFolder%3D%252Fcommunities%252FCommercial%255FCapabilities%255FProgram%252FRegional%252DCustomer%252DMaster%252FShared%2520Documents%252F60%252D%2520Execution%252F20%2520Requirements&DefaultItemOpen=1&DefaultItemOpen=1>

HCP Inbound FRS:

<http://communitiesemea.sanofi.com/communities/Commercial_Capabilities_Program/Regional-Customer-Master/_layouts/WordViewer.aspx?id=/communities/Commercial_Capabilities_Program/Regional-Customer-Master/Shared%20Documents/60-%20Execution/20%20Requirements/RCM%20Integration%20-%20Functional%20Specifications%20-%20HCP%20Inbound%2011Feb2016.docx&Source=http%3A%2F%2Fcommunitiesemea%2Esanofi%2Ecom%2Fcommunities%2FCommercial%5FCapabilities%5FProgram%2FRegional%2DCustomer%2DMaster%2FShared%2520Documents%2FForms%2FAllItems%2Easpx%3FRootFolder%3D%252Fcommunities%252FCommercial%255FCapabilities%255FProgram%252FRegional%252DCustomer%252DMaster%252FShared%2520Documents%252F60%252D%2520Execution%252F20%2520Requirements&DefaultItemOpen=1&DefaultItemOpen=1>

## Source Details



# design decisions

|  |  |
| --- | --- |
| **No.** | **Key Design Decisions** |
| 1 | MFT will send all source files into HDFS Raw zone of Integration Hub |
| 2 | MFT will post message in JMS message queue once files are uploaded to Raw zone |
| 3 | TIBCO will use to write files into HDFS Raw zone for the following cases:   1. Source as Relational DB 2. SAP 3. Email attachment 4. Any other future source where MFT will not provide source data |
| 4 | TIBCO will post message in JMS message queue after uploading the files to Raw zone. This will applicable only where TIBCO upload file in HDFS. |
| 5 | All inbound files will be processed in Integration Hub depending on the availability of the source files and customer data will be shared with RCM. |
| 6 | After processing of files(AS-IS / Sales), INFA BDE will copy customer data into For Purpose zone |
| 7 | INFA BDE will use java tranformation to post success and failure message in JMS message queue. |
| 8 | For inbound to RCM, MFT will pull data from For Purpose zone and deliver to MFT gateway server and RCM will pull data from MFT gateway server |
| 9 | All the source files will be compressed and stored in the source specific archive directory on HDFS |
| 10 | All processed data will be cleared after completion of the process in the process zone. |
| 11 | Reject files would be placed in the reject folder in case of any failure of validation check and email communication would be sent to Reltio which in turn communicate to sources |
| 12 | Control total checks will be performed on the files and results to be stored |

# e2e architecture

The following diagram depicts the conceptual flow pertaining to RCM Data Integration through the Integration Hub.

All sources of customer master data will feed data into the Integration Hub. The Integration Hub will maintain a copy of the original feed received from the sources. The source will put their files into MFT location, MFT will push it to HDFS RAW zone. In case of relational database, and few others TIBCO will pull the data from sources and upload the files into HDFS raw zone. All the source files will copy to In-Process zone before execution of any validation or processing.

Upon execution of successful validations, AS-IS passthrough source files is copied from RAW zone to FOR PURPOSE zone in the sourcewise locations and from there MFT will pull the files for distribution to RCM.

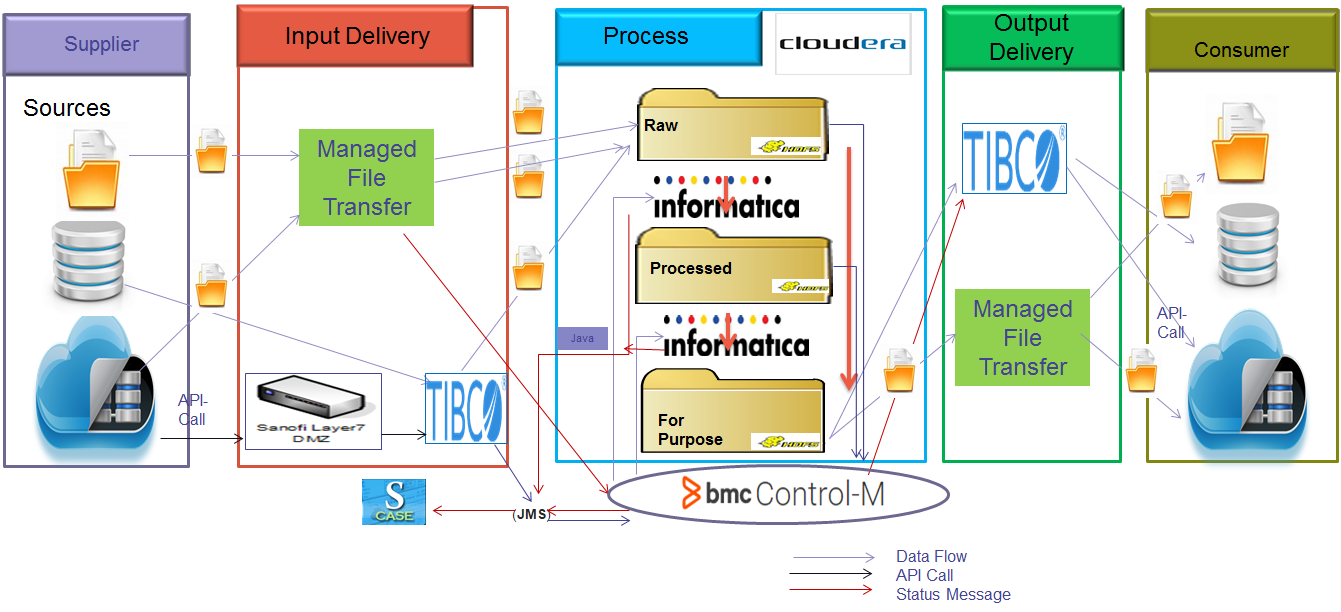
Integration Hub environment also receives two types of Sales files,

* Customer and Transactional information in same file: Customer as well as Transaction Data are available in the same file (Customer data will be extracted).
* Customer and Transactional information in different files: Separate files for customer data and transactional data. These customer files is directly used for further processing.

When Integration Hub receives the customer Data along with the transaction Data in same file, a separate Integration process component segregates the Customer data attributes from the source provided sales file and creates a customer data only file which in turn is sent for further validations before being sent to the RCM. The original source sales file follows the Integration Hub archival strategy.

The customer file created after the segregation process goes through the validation processes in the integration hub. After going through validation processes, Customer data is compared with existing master reference in Integration Hub(x-ref). If any new customer data is identified, system creates a file with only new customer data and message posted to JMS message queue through INFA BDE component. TIBCO consumes this message and picks the new customer file from pre-defined location in HDFS. After getting the new customer file, TIBCO invokes a Real time call to RCM to create the new customer in RCM. After getting the response from RCM, TIBCO updates the golden copies in the HIVE table and also the x-ref of those customers are updated in Integration Hub.

For distribution of Customer data, INFA BDE queries the Hive table to create necessary dataset for downstream. This dataset are copied in HDFS in FOR PURPOSE out zone in the specific out folders. MFT will pull the data from FOR PUPOSE out zone to feed into the respective Target system.

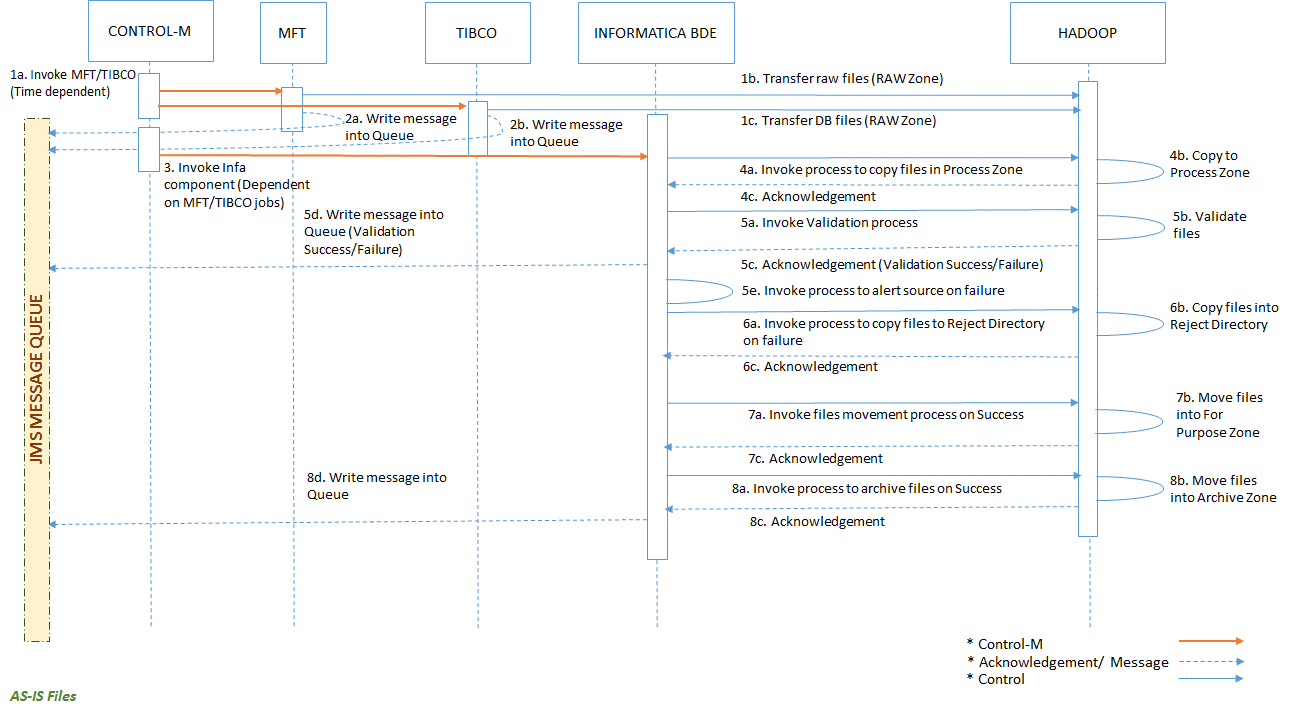


# high level design

## Inbound Customer Only Flow

AS-IS pass through data flow diagram, depicts the scenario where INFA BDE process receives the sourec file in HDFS Raw zone and subsequently it copies the file to PROCESS zone for validation , when validations are successfull , the source files are moved from Raw to FOR PURPOSE zone and then to OUT MFT location. The same copy of files are also moved to Archive zone.

Flow Sequence Diagram



Flow Sequence Description :

|  |  |
| --- | --- |
| **Sequence No.** | **Description** |
| 1a | Control-M job invokes MFT/TIBCO jobs to transfer raw and DB files respectively |
| 1b | MFT job transfers the raw files of different sources into HDFS RAW zone |
| 1c | TIBCO job transfers the database files into HDFS RAW zone |
| 2a | Once MFT jobs complete transfer of file, write the message into JMS Message Queue |
| 2b | Once TIBCO jobs complete transfer of file, write the message into JMS Message Queue |
| 3 | Once job 1b and 1c are completed, Control-M invokes next dependent jobs i.e. Infa BDE Integration jobs of corresponding sources |
| 4a | Job 3 inturn triggers the Infa BDE integration process to copy files into HDFS Process Zone |
| 4b | Control passes to Hadoop and it copies the files into HDFS Process Zone |
| 4c | Hadoop sends acknowledgement to Infa BDE once the copy task is completed |
| 5a | Then Infa BDE invokes validation process (Pattern, Control and threshold check) |
| 5b | Contro passes to Hadoop and Hadoop does all the required validations |
| 5c | Hadoop sends acknowledgements to Infa BDE once the validation tasks are completed irrespective of success/failure of validation tasks |
| 5d | Infa BDE writes the messages about validation outcomes (success/failure) into JMS message queue |
| 5e | In case of validation failure, Infa BDE invokes process (email etc) to alert corresponding sources |
| 6a | In case of validation failure, Infa BDE then triggers process to move files into Reject directory |
| 6b | Control passes to hadoop and it moves the files into Reject directory |
| 6c | Hadoop sends acknowledgement to Infa BDE once the move to Reject dir task is completed |
| 7a | In case of successful validations, Infa BDE triggers process to move files into For-purpose zone |
| 7b | Control passes to hadoop and it moves the files into For-purpose zone |
| 7c | Hadoop sends acknowledgement to Infa BDE once the move to For-purpose zone task is completed |
| 8a | Once job 7c is completed successfully, Infa BDE triggers process to move files into Archive zone |
| 8b | Control passes to hadoop and it moves the files into Archive zone |
| 8c | Hadoop sends acknowledgement to Infa BDE once the move to Archive zone task is completed |
| 8d | Once job 6c or 8c, is completed successfully, Infa BDE writes the messages about process completion status (success/failure) into JMS message queue |

Data Flow Diagram



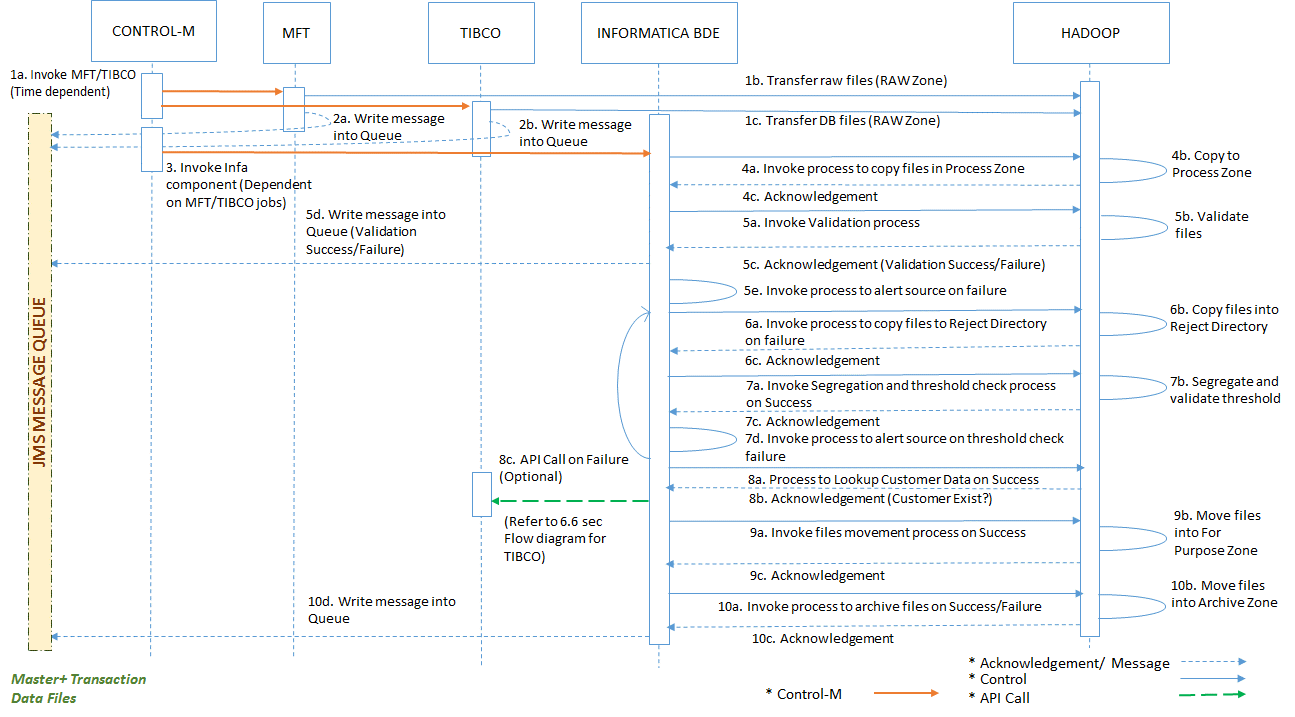
Following sources will follow the AS –IS pattern

|  |  |  |
| --- | --- | --- |
| SL No | Source Name | List of Files |
| 1 | All Decile | All\_decile\_tall\_{QN}{YYYY}.zip |
| 2 | CARS | CARC\_{YYYYMMDD}.dat |
| 3 | GPO | {Source Name} {YYYY} {MM}.txt |
| 4 | HCOS | Sanofi\_CustQtr\_affiliation\_type.M{YYYYNN}.001.GZ |
| 5 | HCOS | Sanofi\_CustQtr\_business.M{YYYYNN}.001.GZ |
| 6 | HCOS | Sanofi\_CustQtr\_business\_detail\_fact.M{YYYYNN}.001.GZ |
| 7 | HCOS | Sanofi\_CustQtr\_contact\_affiliation\_fact.M{YYYYNN}.001.GZ |
| 8 | HCOS | Sanofi\_CustQtr\_professional.M{YYYYNN}.001.GZ |
| 9 | HCOS | Sanofi\_CustQtr\_professional\_title.M{YYYYNN}.001.GZ |
| 10 | HCOS | Sanofi\_CustQtr\_provider\_affiliation\_fact.M{YYYYNN}.001.GZ |
| 11 | HCOS | Sanofi\_CustQtr\_relationship\_fact.M{YYYYNN}.001.GZ |
| 12 | iEI Full | <mmddyyyy>CCTACCOUNT.EXP.gz |
| 13 | iEI Full | <mmddyyyy>CCTACCOUNTADDR.EXP.gz |
| 14 | iEI Full | <mmddyyyy>CCTAFFILIATION.EXP.gz |
| 15 | iEI Full | <mmddyyyy>CCTPHONE.EXP.gz |
| 16 | iEI Full | CCTADDR\_<yyyymmdd>.EXP.Gz |
| 17 | iEI Full | CCTCUST\_<yyyymmdd>.EXP.Gz |
| 18 | iEI Full | CCTEDUC\_<yyyymmdd>.EXP.Gz |
| 19 | iEI Full | CCTPHON\_<yyyymmdd>.EXP.Gz |
| 20 | iEI Full | CCTSOCT\_<yyyymmdd>.EXP.Gz |
| 21 | iEI Full | CCTSTLC\_<yyyymmdd>.EXP.Gz |
| 22 | IMS AMA PDRP | XPPDRP\_{YYYY}{MM}.001.ZIP |
| 23 | IMS DDD Outlet Change Log | DDDC\_{YYYY}{WW}\_001.001.ZIP |
| 24 | IMS DDD Outlet Master | DDDM\_{YYYY}{MM}\_001.001.ZIP |
| 25 | IMS HCP Change Log | CDS\_ZSSR\_CPCHL\_M<NN>\_CSV\_001.ZIP |
| 26 | IMS HCPS Monthly | HCPS.HCP\_AMAMASTER\_<DD-MON-YYYY>.TXT\_001.zip |
| 27 | IMS HCPS Monthly | HCPS.HCP\_RAWDEA\_<DD-MON-YYYY>.TXT.zip |
| 28 | IMS HCPS Weekly | HCPS.HCP\_ADDRESS\_<DD-MON-YYYY>.TXT.zip |
| 29 | IMS HCPS Weekly | HCPS.HCP\_ADDRESS\_DEA\_<DD-MON-YYYY>.TXT.zip |
| 30 | IMS HCPS Weekly | HCPS.HCP\_CONSOL\_<DD-MON-YYYY>.TXT.zip |
| 31 | IMS HCPS Weekly | HCPS.HCP\_IDENTIFIER\_<DD-MON-YYYY>.TXT.zip |
| 32 | IMS HCPS Weekly | HCPS.HCP\_LICENSE\_<DD-MON-YYYY>.TXT.zip |
| 33 | IMS HCPS Weekly | HCPS.HCP\_PROFESSION\_<DD-MON-YYYY>.TXT.zip |
| 34 | IMS HCPS Weekly | HCPS.HCP\_PROFILE\_<DD-MON-YYYY>.TXT.zip |
| 35 | IMS HCPS Weekly | HCPS.HCP\_REGULATION\_<DD-MON-YYYY>.TXT.zip |
| 36 | IMS HCPS Weekly | HCPS.HCP\_SPECIALTY\_<DD-MON-YYYY>.TXT.zip |
| 37 | IMS HCPS Weekly | HCPS.HCP\_SPLIT\_<DD-MON-YYYY>.TXT.zip |
| 38 | IMS Plan Workbook | PLWK\_MAC\_M{MM}.DAT.001.ZIP |
| 39 | IMS Plan Workbook | PLWK\_NEWVIEW\_M{MM}.DAT.001.ZIP |
| 40 | IMS Plan Workbook | PLWK\_PBM\_M{MM}.DAT.001.ZIP |
| 41 | IMS Plan Workbook | XPPY\_M{MM}.DAT.001.ZIP |
| 42 | KTLP (Publicis Healthcare) | CCT<yyyymmdd>.txt |
| 43 | Med'meme | DELTA\_DATA\_DUMP\_Clinical\_Trials\_yyyymmdd.csv |
| 44 | Med'meme | DELTA\_DATA\_DUMP\_Contact\_Information\_yyyymmdd.csv |
| 45 | Med'meme | DELTA\_DATA\_DUMP\_Current\_Role\_yyyymmdd.csv |
| 46 | Med'meme | DELTA\_DATA\_DUMP\_Editorial\_Responsibilities\_yyyymmdd.csv |
| 47 | Med'meme | DELTA\_DATA\_DUMP\_Education\_yyyymmdd.csv |
| 48 | Med'meme | DELTA\_DATA\_DUMP\_KOL\_Name\_yyyymmdd.csv |
| 49 | Med'meme | DELTA\_DATA\_DUMP\_Presentations\_yyyymmdd.csv |
| 50 | Med'meme | DELTA\_DATA\_DUMP\_Previous\_Role\_yyyymmdd.csv |
| 51 | Med'meme | DELTA\_DATA\_DUMP\_Publications\_yyyymmdd.csv |
| 52 | Med'meme | DELTA\_DATA\_DUMP\_Treatment\_Guidelines\_yyyymmdd.csv |
| 53 | NPPES | NPPES\_Data\_Dissemination\_{MMM\_YYYY}.zip |
| 54 | Regeneron | AFRG\_RGN\_Accounts\_<mm-dd-yyyy>\_<hh24mi>.zip |
| 55 | Regeneron | AGRG\_RGN\_Accounts\_<mm-dd-yyyy>\_<hh24mi>.zip |
| 56 | Regeneron | PGRG\_RGN\_Prescribers\_<mm-dd-yyyy>\_<hh24mi>.zip |
| 57 | Regeneron | RGNA\_XREF\_<mm-dd-yyyy>.zip |
| 58 | Regeneron | RGNP\_XREF\_<mm-dd-yyyy>.zip |
| 59 | Sanofi Payer Grouping | HCA Affiliations {YYYY}{MM}.txt |
| 60 | Sanofi Payer Grouping | HCA Health Care Organization {YYYY}{MM}.txt |
| 61 | Sanofi Payer Grouping | Merger and Deletion ID list {YYYY}{MM}.txt |
| 62 | Sanofi Payer Grouping | Sanofi Payer Grouping Master {YYYY}{MM}.txt |
| 63 | SAP | SAPC\_{Timestamp}.dat |
| 64 | Transparency - AADE | File not available |
| 65 | Transparency - AS360 | AGTP\_AS36\_Accounts\_<Timestamp>.txt |
| 66 | Transparency - AS360 | PGTP\_AS36\_Prescribers\_<Timestamp>.txt |
| 67 | Transparency - Clubnet | N/A – Clubnet accounts is not being loaded in Quantum |
| 68 | Transparency - Clubnet | PGCN\_Prescribers\_<Timestamp>.gz |
| 69 | Transparency - DASH (Pasteur) | AGPS\_NonFocus\_Accounts\_<Timestamp>.txt |
| 70 | Transparency - DASH (Pasteur) | PGPS\_NonFocus\_Prescribers\_<Timestamp>.txt |
| 71 | Transparency - Genzyme | AGTP\_GZ\_Accounts\_<Timestamp>.txt |
| 72 | Transparency - Genzyme | PGTP\_GZ\_Prescribers\_<Timestamp>.txt |
| 73 | Transparency - Impact | PGIM\_Prescribers\_<Timestamp>.gz |
| 74 | Transparency - Millenium | PGTP\_Prescriber\_<Timestamp>.gz |
| 75 | Transparency - Pasteur | AGPS\_Accounts\_<Timestamp>.gz |
| 76 | Transparency - Pasteur | PGPS\_Prescribers\_<Timestamp>.gz |
| 77 | Transparency - PRISMA | File not available |
| 78 | Transparency - SalesLogix | GZSLX\_HCA\_<timestamp>.txt |
| 79 | Transparency - SalesLogix | GZSLX\_HCP\_<timestamp>.txt |
| 80 | Transparency - Sanofi Grants | AGGR\_Accounts\_<Timestamp>.txt |
| 81 | Transparency - SAP | AGSA\_Accounts\_<Timestamp>.TXT |
| 82 | Transparency - SAP | PGSA\_Prescriber\_<Timestamp>.TXT |
| 83 | Transparency - TDM | AGTP\_SMS\_Accounts\_<Timestamp>.txt |
| 84 | Transparency - TDM | PGTP\_SMS\_Prescribers\_<timestamp>.txt |
| 85 | Transparency - Trace | AGTR\_Accounts\_<Timestamp>.gz |
| 86 | Transparency - Trace | PGTR\_Prescribers\_<Timestamp>.gz |
| 87 | Unsubscribe portal | HCP\_MKTG\_OPT\_XFR |
| 88 | Unsubscribe portal | HCP\_PDRP\_OPT\_XFR |
| 89 | Relypsa | Yet to be finalized |

## Inbound Sales Data Flow

INFA BDE verifies the Customer Data with reference data set and if there are any non existing Customers, a customer file is created containing the new customer data for realtime RCM consumption through TIBCO process. Getting the file from TIBCO RCM will create the new customer IDs and send back the golden copies of the new customer data to TIBCO. TIBCO will update the new customer data in the Integration Hub HIVE table directly.

Flow Sequence Diagram

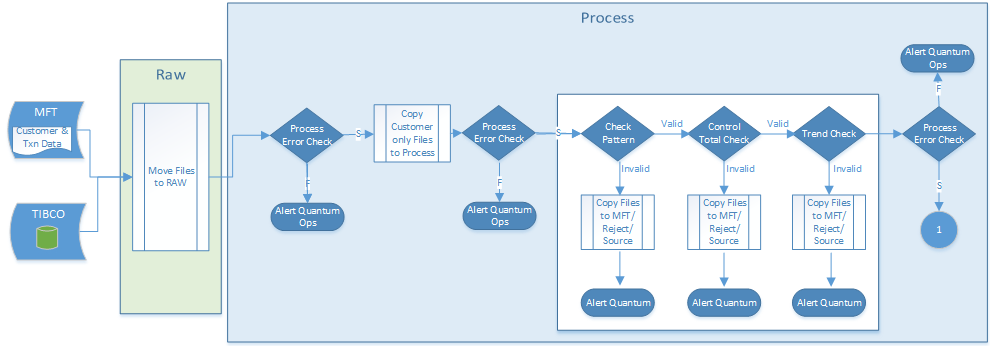


Flow Sequence Description :

|  |  |
| --- | --- |
| **Sequence No.** | **Description** |
| 1a | Control-M job invokes MFT/TIBCO jobs to transfer raw and DB files respectively |
| 1b | MFT job transfers the raw files of different sources into HDFS RAW zone |
| 1c | TIBCO job transfers the database files into HDFS RAW zone |
| 2a | Once MFT jobs complete transfer of file, write the message into JMS Message Queue |
| 2b | Once TIBCO jobs complete transfer of file, write the message into JMS Message Queue |
|  |  |
| 3 | Once job 1b and 1c are completed, Control-M invokes next dependent jobs i.e. Infa BDE Integration jobs of corresponding sources |
| 4a | Job 3 in-turn triggers the Infa BDE integration process to copy files into HDFS Process Zone |
| 4b | Control passes to Hadoop and it copies the files into HDFS Process Zone |
| 4c | Hadoop sends acknowledgement to Infa BDE once the copy task is completed |
| 5a | Then Infa BDE invokes validation process (Pattern, Control check) |
| 5b | Control passes to Hadoop and Hadoop does all the required validations |
| 5c | Hadoop sends acknowledgements to Infa BDE once the validation tasks are completed irrespective of success/failure of validation tasks |
| 5d | Infa BDE writes the messages about validation outcomes (success/failure) into JMS message queue |
| 5e | In case of validation failure, Infa BDE invokes process (email etc ) to alert corresponding sources |
| 6a | In case of validation failure, Infa BDE then triggers process to move files into Reject directory |
| 6b | Control passes to hadoop and it moves the files into Reject directory |
| 6c | Hadoop sends acknowledgement to Infa BDE once the move to Reject dir task is completed |
|  |  |
| 7a | In case of successful validations, Infa BDE triggers process to segregate customer records from sales files and then threshold validation on extracted customer data set |
| 7b | Control passes to hadoop and it segregates customer records from sales files based on details present in configuration files and then validate the threshold |
| 7c | Hadoop sends acknowledgement to Infa BDE once the segregation and threshold validation tasks is completed irrespective of success/failure of validation tasks |
| 7d | In case of threshold validation failure, Infa BDE invokes process (email etc ) to alert corresponding sources and again Infa BDE invokes jobs 6a, 6b and 6c to move the files into Reject directory |
|  |  |
| 8a | In case of successful threshold validations, Infa BDE invokes process to lookup customer details against Universe and X-ref and control passes to Hadoop |
| 8b | Hadoop acknowledges whether the customers exist and based on that Infa BDE creates non-existing customer files (These files are pickup by TIBCO) |
|  |  |
| 9a | Then Infa BDE triggers process to move files into For-purpose zone |
| 9b | Control passes to hadoop and it moves the files into For-purpose zone |
| 9c | Hadoop sends acknowledgement to Infa BDE once the move to For-purpose zone task is completed |
|  |  |
| 10a | Once job 9c is completed successfully, Infa BDE triggers process to move files into Archive zone |
| 10b | Control passes to hadoop and it moves the files into Archive zone |
| 10c | Hadoop sends acknowledgement to Infa BDE once the move to Archive zone task is completed |
| 10d | Once job 10c or 6c,is completed successfully, Infa BDE writes the messages about process completion status (success/failure) into JMS message queue |

Data Flow Diagram

* Transactional Source (Customer Data in separate file)

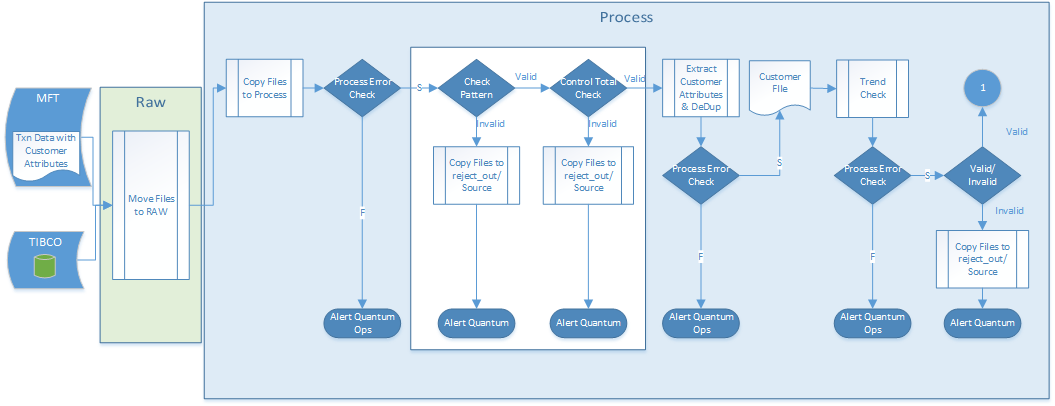


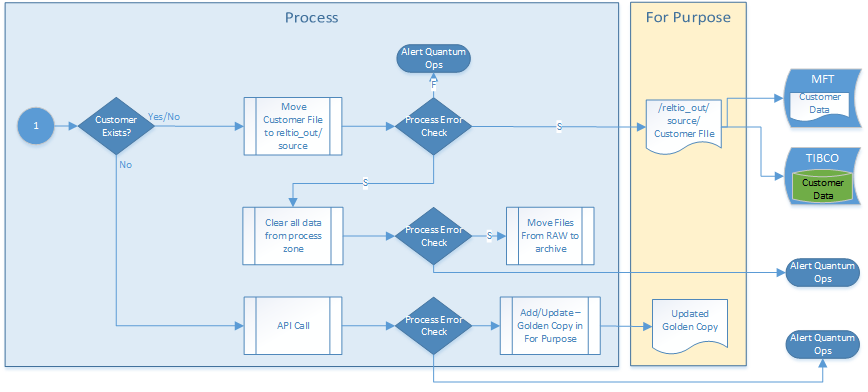


Following sources will follow the Sales Files pattern(Customer Data in separate file)

|  |  |  |
| --- | --- | --- |
| SL No | Source Name | List of Files |
| 1 | IMS WxDM | NEW.PROD.RW{REPORTWEEKNUMBER}.CLI{CLIENTNUMBER}.{WEEKNUMBER}.SPL{MARKETNUMBER}.FACT.D.001.ZIP |
| 2 | IMS WxDM | NEW.PROD.RW{REPORTWEEKNUMBER}.CLI{CLIENTNUMBER}.{WEEKNUMBER}.SUMFIXED.001.ZIP |
| 3 | IMS WxDM | NEW.PROD.RW{REPORTWEEKNUMBER}.CLI{CLIENTNUMBER}.{WEEKNUMBER}.SUMFIXFH.001.ZIP |
| 4 | IMS WxDM | NEW.PROD.RW{REPORTWEEKNUMBER}.WK{WEEKNUMBER}.LINK{0NN}.BGORIG.001.ZIP |
| 5 | IMS WxDM | NEW.PROD.RW{REPORTWEEKNUMBER}.WK{WEEKNUMBER}.LINK{0NN}.BGPATENT.001.ZIP |
| 6 | IMS WxDM | NEW.PROD.RW{REPORTWEEKNUMBER}.WK{WEEKNUMBER}.LINK{0NN}.BGSOURCE.001.ZIP |
| 7 | IMS WxDM | NEW.PROD.RW{REPORTWEEKNUMBER}.WK{WEEKNUMBER}.LINK{0NN}.BGTRADE.001.ZIP |
| 8 | IMS WxDM | NEW.PROD.RW{REPORTWEEKNUMBER}.WK{WEEKNUMBER}.LINK{0NN}.CMFTYPE.001.ZIP |
| 9 | IMS WxDM | NEW.PROD.RW{REPORTWEEKNUMBER}.WK{WEEKNUMBER}.LINK{0NN}.CORP.001.ZIP |
| 10 | IMS WxDM | NEW.PROD.RW{REPORTWEEKNUMBER}.WK{WEEKNUMBER}.LINK{0NN}.DELSYS.001.ZIP |
| 11 | IMS WxDM | NEW.PROD.RW{REPORTWEEKNUMBER}.WK{WEEKNUMBER}.LINK{0NN}.ETHPROP.001.ZIP |
| 12 | IMS WxDM | NEW.PROD.RW{REPORTWEEKNUMBER}.WK{WEEKNUMBER}.LINK{0NN}.FORM.001.ZIP |
| 13 | IMS WxDM | NEW.PROD.RW{REPORTWEEKNUMBER}.WK{WEEKNUMBER}.LINK{0NN}.MFR.001.ZIP |
| 14 | IMS WxDM | NEW.PROD.RW{REPORTWEEKNUMBER}.WK{WEEKNUMBER}.LINK{0NN}.PERIOD.001.ZIP |
| 15 | IMS WxDM | NEW.PROD.RW{REPORTWEEKNUMBER}.WK{WEEKNUMBER}.LINK{0NN}.PGN.001.ZIP |
| 16 | IMS WxDM | NEW.PROD.RW{REPORTWEEKNUMBER}.WK{WEEKNUMBER}.LINK{0NN}.PRDPGN.001.ZIP |
| 17 | IMS WxDM | NEW.PROD.RW{REPORTWEEKNUMBER}.WK{WEEKNUMBER}.LINK{0NN}.PRODUCT.001.ZIP |
| 18 | IMS WxDM | NEW.PROD.RW{REPORTWEEKNUMBER}.WK{WEEKNUMBER}.LINK{0NN}.RXSTATUS.001.ZIP |
| 19 | IMS WxDM | NEW.PROD.RW{REPORTWEEKNUMBER}.WK{WEEKNUMBER}.LINK{0NN}.SRCESUPP.001.ZIP |
| 20 | IMS WxDM | NEW.PROD.RW{REPORTWEEKNUMBER}.WK{WEEKNUMBER}.LINK{0NN}.TLC.001.ZIP |
| 21 | IMS WxDM | NEW.PROD.RW{REPORTWEEKNUMBER}.WK{WEEKNUMBER}.LINK{0NN}.USC.001.ZIP |
| 22 | SAP | SAPS\_{Timestamp}.dat |
| 23 | IMS DDD - Weekly | CTRL\_DDW\_{CLIENT NUMBER}\_{dollars}\_{WW}{YYYY}.001.ZIP |
| 24 | IMS DDD - Monthly | CTRL\_DDD\_{CLIENT NUMBER}\_M{MM}.001.ZIP |
| 25 | IMS DDD - Weekly Parallel | CTRL\_DGW\_{CLIENT NUMBER}\_{dollars}\_{WW}{YYYY}.001.ZIP |
| 26 | CARS | CARS\_{YYYYMMDD}.dat |
| 27 | IMS WxDM - Parallel | PRLL.PROD.RW{REPORTWEEKNUMBER}.CLI{CLIENTNUMBER}.{WEEKNUMBER}.SPL{MARKETNUMBER}.FACT.D.001.ZIP |
| 28 | IMS WxDM - Parallel | PRLL.PROD.RW{REPORTWEEKNUMBER}.CLI{CLIENTNUMBER}.{WEEKNUMBER}.SUMFIXED.001.ZIP |
| 29 | IMS WxDM - Parallel | PRLL.PROD.RW{REPORTWEEKNUMBER}.CLI{CLIENTNUMBER}.{WEEKNUMBER}.SUMFIXFH.001.ZIP |
| 30 | IMS WxDM - Parallel | NEW.PROD.RW{REPORTWEEKNUMBER}.WK{WEEKNUMBER}.LINK{0NN}.BGORIG.001.ZIP |
| 31 | IMS WxDM - Parallel | NEW.PROD.RW{REPORTWEEKNUMBER}.WK{WEEKNUMBER}.LINK{0NN}.BGPATENT.001.ZIP |
| 32 | IMS WxDM - Parallel | NEW.PROD.RW{REPORTWEEKNUMBER}.WK{WEEKNUMBER}.LINK{0NN}.BGSOURCE.001.ZIP |
| 33 | IMS WxDM - Parallel | NEW.PROD.RW{REPORTWEEKNUMBER}.WK{WEEKNUMBER}.LINK{0NN}.BGTRADE.001.ZIP |
| 34 | IMS WxDM - Parallel | NEW.PROD.RW{REPORTWEEKNUMBER}.WK{WEEKNUMBER}.LINK{0NN}.CMFTYPE.001.ZIP |
| 35 | IMS WxDM - Parallel | NEW.PROD.RW{REPORTWEEKNUMBER}.WK{WEEKNUMBER}.LINK{0NN}.CORP.001.ZIP |
| 36 | IMS WxDM - Parallel | NEW.PROD.RW{REPORTWEEKNUMBER}.WK{WEEKNUMBER}.LINK{0NN}.DELSYS.001.ZIP |
| 37 | IMS WxDM - Parallel | NEW.PROD.RW{REPORTWEEKNUMBER}.WK{WEEKNUMBER}.LINK{0NN}.ETHPROP.001.ZIP |
| 38 | IMS WxDM - Parallel | NEW.PROD.RW{REPORTWEEKNUMBER}.WK{WEEKNUMBER}.LINK{0NN}.FORM.001.ZIP |
| 39 | IMS WxDM - Parallel | NEW.PROD.RW{REPORTWEEKNUMBER}.WK{WEEKNUMBER}.LINK{0NN}.MFR.001.ZIP |
| 40 | IMS WxDM - Parallel | NEW.PROD.RW{REPORTWEEKNUMBER}.WK{WEEKNUMBER}.LINK{0NN}.PERIOD.001.ZIP |
| 41 | IMS WxDM - Parallel | NEW.PROD.RW{REPORTWEEKNUMBER}.WK{WEEKNUMBER}.LINK{0NN}.PGN.001.ZIP |
| 42 | IMS WxDM - Parallel | NEW.PROD.RW{REPORTWEEKNUMBER}.WK{WEEKNUMBER}.LINK{0NN}.PRDPGN.001.ZIP |
| 43 | IMS WxDM - Parallel | NEW.PROD.RW{REPORTWEEKNUMBER}.WK{WEEKNUMBER}.LINK{0NN}.PRODUCT.001.ZIP |
| 44 | IMS WxDM - Parallel | NEW.PROD.RW{REPORTWEEKNUMBER}.WK{WEEKNUMBER}.LINK{0NN}.RXSTATUS.001.ZIP |
| 45 | IMS WxDM - Parallel | NEW.PROD.RW{REPORTWEEKNUMBER}.WK{WEEKNUMBER}.LINK{0NN}.SRCESUPP.001.ZIP |
| 46 | IMS WxDM - Parallel | NEW.PROD.RW{REPORTWEEKNUMBER}.WK{WEEKNUMBER}.LINK{0NN}.TLC.001.ZIP |
| 47 | IMS WxDM - Parallel | NEW.PROD.RW{REPORTWEEKNUMBER}.WK{WEEKNUMBER}.LINK{0NN}.USC.001.ZIP |

* Transactional Source (Customer data in same file)



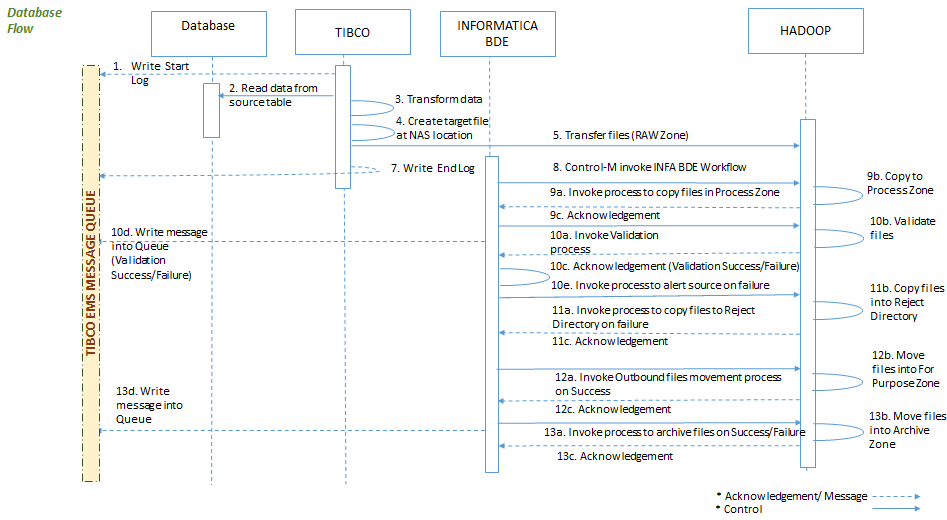


Following sources will follow the Sales Files pattern(Customer data in same file)

|  |  |  |  |
| --- | --- | --- | --- |
| SL No | Source Name | List of Files | Segregated Customer Files |
| 1 | Davita | Sanofi\_binder\_fill\_data\_with additional fields\_{YYYY(MM+1)01}.xlsx | Sanofi\_binder\_fill\_data\_with additional fields\_{YYYY(MM+1)01}\_Cust\_Davita\_{MMDDYYYY}.txt |
| 2 | Fresenius | <YYYY-(MM+1)-DD> Sanofi Prescriber Data.xlsx | <YYYY-(MM+1)-DD> Sanofi Prescriber Data\_Cust\_Fresenius\_{MMDDYYYY}.txt |
| 3 | Caremark | Caremark {YYYYMM}.txt | Caremark {MMMYY}\_Cust\_Caremark\_{MMDDYYYY}.txt |
| 4 | OptumRx | Genzyme\_Data\_OptumRx\_{MMDDYYYY}.txt | Genzyme\_Data\_OptumRx\_{MMDDYYYY}\_Cust\_Optumrx\_{MMDDYYYY}.txt |
| 5 | SPP Synvisc | \_Master\_SPP\_{Mth}.xlsx | \_MASTER\_SPP\_{MTH}\_Cust\_SPP\_{MMDDYYYY}.txt |
| 6 | SPP Synvisc | \_Master\_Optum\_{Mth}.xlsx | \_MASTER\_OPTUM\_{MTH}\_Cust\_Optum\_{MMDDYYYY}.txt |
| 7 | IMS DDD - Weekly | {DDOD}\_weekly\_{MARKET}\_{dollars}\_{WW}{YYYY}\_{CLIENT NUMBER}{REPORT NUMBER}.001.ZIP | DDD\_weekly\_cust\_{MMDDYYYY}.txt |
| 8 | IMS DDD - Weekly | {DDOU}\_weekly\_{MARKET}\_{unit1}\_{WW}{YYYY}\_{CLIENT NUMBER}{REPORT NUMBER}.001.ZIP |
| 9 | IMS DDD - Weekly | {DDOU}\_weekly\_{MARKET}\_{unit2}\_{WW}{YYYY}\_{CLIENT NUMBER}{REPORT NUMBER}.001.ZIP |
| 10 | IMS DDD - Weekly | {DDZD}\_weekly\_{MARKET}\_{dollars}\_{WW}{YYYY}\_{CLIENT NUMBER}{REPORT NUMBER}.001.ZIP |
| 11 | IMS DDD - Weekly | {DDZU}\_weekly\_{MARKET}\_{UNIT1}\_{WW}{YYYY}\_{CLIENT NUMBER}{REPORT NUMBER}.001.ZIP |
| 12 | IMS DDD - Weekly | {DDZU}\_weekly\_{MARKET}\_{unit2}\_{WW}{YYYY}\_{CLIENT NUMBER}{REPORT NUMBER}.001.ZIP |
| 13 | IMS DDD - Monthly | DDDO\_M{MM}\_{REPORT NUMBER}.001.ZIP | Not available |
| 14 | IMS DDD - Weekly Parallel | {DGOD}\_weekly\_{MARKET}\_{dollars}\_{WW}{YYYY}\_{CLIENT NUMBER}{REPORT NUMBER}.001.ZIP | DDD\_weekly\_cust\_parallel\_{MMDDYYYY}.txt |
| 15 | IMS DDD - Weekly Parallel | {DGOU}\_weekly\_{MARKET}\_{unit1}\_{WW}{YYYY}\_{CLIENT NUMBER}{REPORT NUMBER}.001.ZIP |
| 16 | IMS DDD - Weekly Parallel | {DGOU}\_weekly\_{MARKET}\_{unit2}\_{WW}{YYYY}\_{CLIENT NUMBER}{REPORT NUMBER}.001.ZIP |
| 17 | IMS DDD - Monthly Parallel | DDOG\_M{MM}\_{REPORT NUMBER}.001.ZIP | Not available |
| 18 | Integrichain | DNA Investigator Template - Channel Sales - DNA Pharmacy (All POCs).csv | Integrichain\_cust\_{MMDDYYYY}.txt |
| 19 | CMOP | Sanofi Monthly Standard Report\_{MMM}{YY}.txt | Sanofi Monthly Standard Report\_Cust\_CMOP\_{MMDDYYYY}.txt |
| 20 | Prime Vendor | Sanofi PV Monthly data Report.xls | Sanofi PV Monthly data Report\_Cust\_PV\_{MMDDYYYY}.txt |
| 21 | Relypsa | Yet to be finalized | {Relypsa Sales Filename} - file name to be finalized |

## Data flow from Database

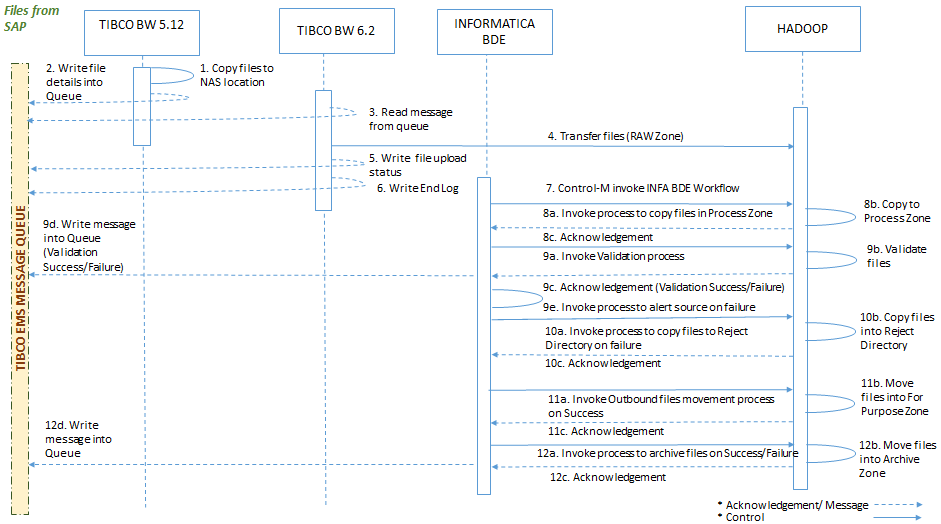
In this scenario, TIBCO BW will pull the data from source tables then, the data is transformed as per requirement and output file are created. This output file are transferred to the HDFS RAW Zone using the HDFS Operation activity of the TIBCO BW Big Data plugin.More detail steps are provided In TIBCO Environment section of the document.



|  |  |  |
| --- | --- | --- |
| SL No | Source Name | List of Files |
| 1 | Unsubscribe Portal | HCP\_PDRP\_OPT\_XFR |
| 2 | Unsubscribe Portal | HCP\_MKTG\_OPT\_XFR |

## File Transfer From SAP File Server

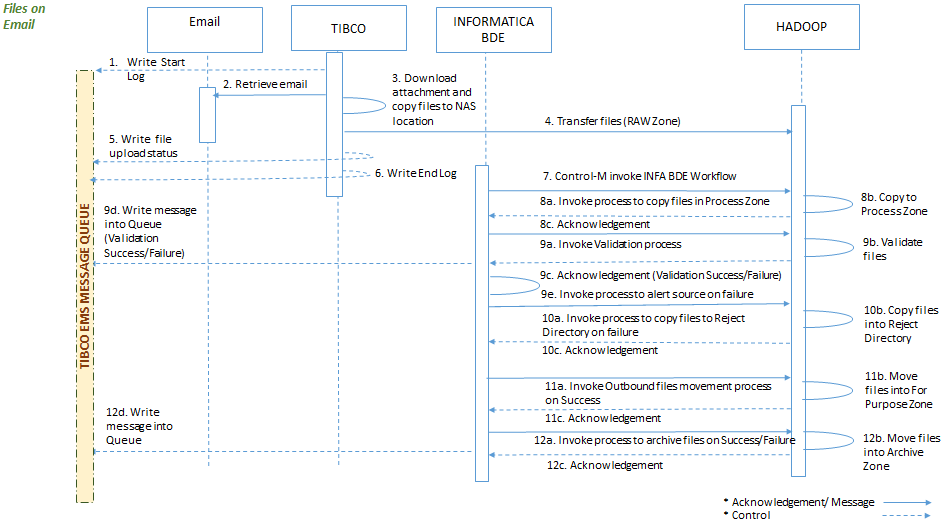
Currently, TIBCO extract the data from SAP by using SAP adapter and place the file in Quantum server location. The existing process are enhanced to create a parallel process which will copy all the SAP in HDFS raw zone



|  |  |  |
| --- | --- | --- |
| SL No | Source Name | List of Files |
| 1 | SAP | SAPC\_{Timestamp}.dat |
| 2 | SAP | CTRL\_SAPC\_{Timestamp}.dat |

## Email Attachment

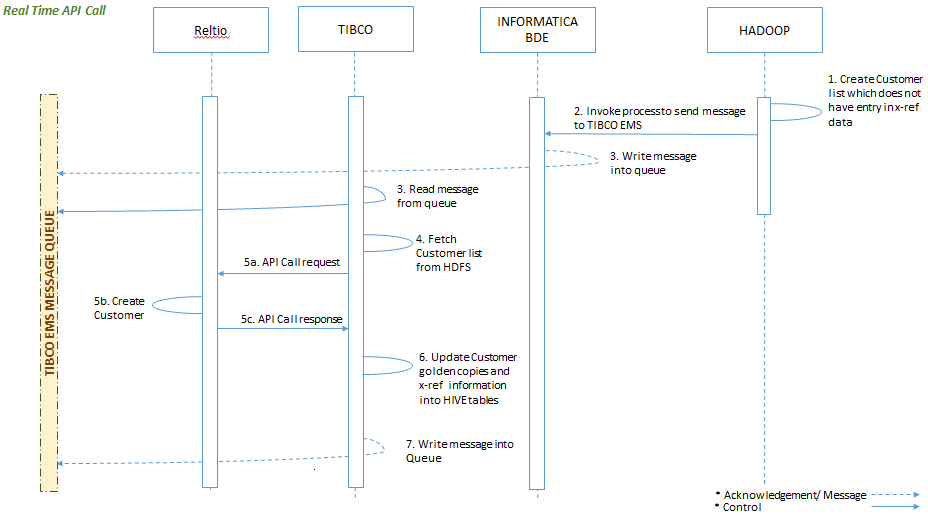
In this scenario, TIBCO BW on receiving an email will download the attachment and copy it to the local NAS storage. The file is then transferred to the HDFS Raw Zone.



|  |  |  |
| --- | --- | --- |
| SL No | Source Name | List of Files |
|  |  |  |
|  |  |  |
|  |  |  |

## Real Time API Call

In the below scenario, INFA BDE is processing a Sales datafile containing transaction information for which customer data is not present in x-ref data. The INTEGRATION HUB creates a list of these customers and writes a message into JMS MESSAGE queue.TIBCO reads the message in JMS MESSAGE queue and pulls the created customer list from HDFS and loop through it based on the bulk api call limit to make a web API call to reltio/RCM with the source provided data.Reltio sends a response containing the customer data to TIBCO. TIBCO will then write the customer golden copy into HIVE tables.



## TIBCO Message Component

INFA BDE communicates with JMS message queue after process completion. INFA BDE will send message to JMS queue in below format.

<?xml version="1.0" encoding="UTF-8"?>

<ns0:MCASE\_LOG xmlns:ns0="http://www.tibco.com/schemas/MCASEMonitor/Shared Schemas/Schema.xsd">

<ns0:STATUS\_ID>0</ns0:STATUS\_ID>

<ns0:EXTENDED\_TEXT>Successfully Updated E-Comm Database with the Product Information, TransactionID = PRODUCT\_E-IDOC-S-0000000176614075-20140922203016</ns0:EXTENDED\_TEXT>

<ns0:APP\_CD>SDWEB</ns0:APP\_CD>

<ns0:COMP\_CD>PRODUCT\_E</ns0:COMP\_CD>

<ns0:STAGE\_NAME>A00DVDAMSCOS010</ns0:STAGE\_NAME>

<ns0:KEY1\_VALUE>PRODUCT\_E-IDOC-S-0000000176614075-20140922203016</ns0:KEY1\_VALUE>

<ns0:KEY2\_VALUE>0000000176614075</ns0:KEY2\_VALUE>

<ns0:KEY3\_VALUE>000000000050069885</ns0:KEY3\_VALUE>

<ns0:START\_TIME>2014-09-22 20:30:17</ns0:START\_TIME>

<ns0:END\_TIME>2014-09-22 20:30:17</ns0:END\_TIME>

<ns0:BUSINESS\_UNIT>SDWEB</ns0:BUSINESS\_UNIT>

<ns0:ENV\_SOURCE>DKF</ns0:ENV\_SOURCE>

<ns0:ENV\_TYPE>DEV</ns0:ENV\_TYPE>

</ns0:MCASE\_LOG>

# Detailed Technical Design

Below is the detailed level Design for each type of source systems processing through INFA BDE environment.

1. AS-IS Customer data file processing: INFA BDE performs required validation checks and after successful validation customer data file passes through to make it available for RCM to consume.
2. Sales files (Customer and Transaction data in same file): INFA BDE performs required validation checks and after successful validation it will segregate the customer data from transaction data and makes customer data available for RCM to consume.
3. Sales files (Customer and Transaction data in different file): INFA BDE will performs required validation checks on customer file and after successful validation it will copy customer files to transfer to RCM

In case of last 2 scenarios, source customer data will reference the cross reference data available from RCM to determine the requirement of real time customer creation. If any customer exists in source file but not available in the RCM cross reference data(x-ref), INFA BDE will generate a new customer data file in the PROCESS ZONE and put it in pre-defined HDFS location from where TIBCO writes message into TIBCO EMS topic and consumes the file and call RCM API in real time to create the customer. RCM creates the customer in real time and returns golden copy of customer with cross reference information. TIBCO writes back customer information to Hive table.

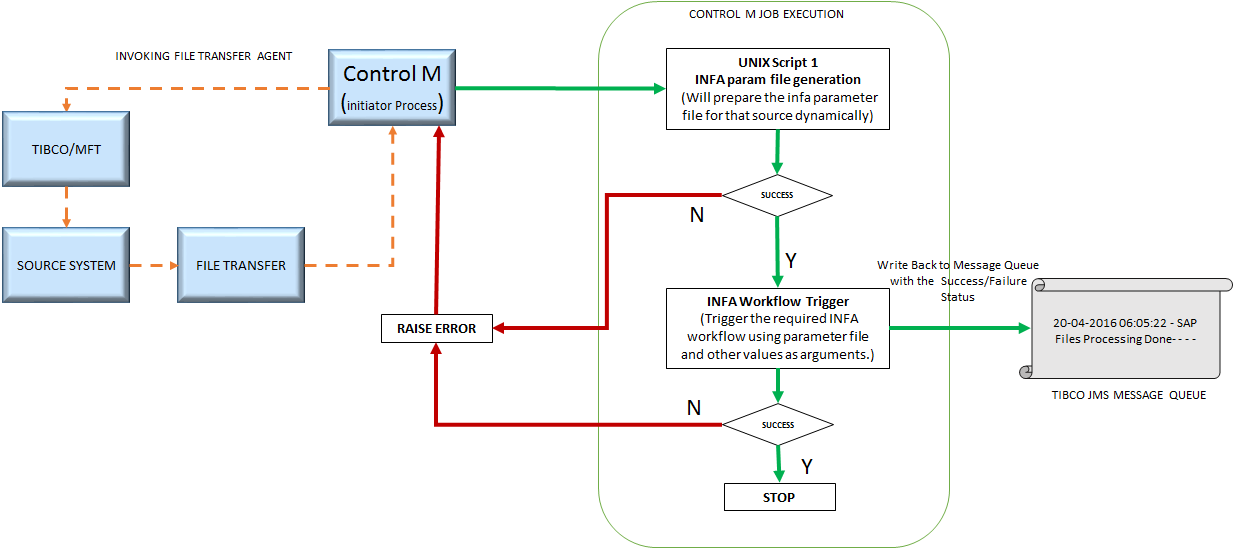
Every component in this RCM integration scheduled by Control M. The result of each processing step will be logged in JMS message queue.

.

**Process Initiation**

* + Control M process Scheduling : Control M triggers the Initiator Process to copy file to HDFS raw zone.
  + Control M triggering infa application: After placing the file, Control M triggers the Unix/INFA components to process the files further.
* Creates an Informatica parameter file taking the details from config file for the specific source system.
* Triggers the specific Informatica application for that source using the above infa parameter file.

## Job Automation Process Flow

The process automation is done with the help of Informatica BDE, Control M, File Transfer tool (MFT/TIBCO) and Unix Shell scripts. A Process is invoked through a time based trigger (Control M triggering the file transfer component). The following events takes place once any source system file is available to be processed:

Event 1: Control M initiates a process which triggers the file transfer agent to copy file from respective source systems and place it in the raw zone of HDFS in source specific directories.

Event 2: After placing the file, Control M will triggers the Unix/INFA components to process the files further.

Event 3: At first Control M triggers a shell script stored at a specific location in the batch server. This script dynamically creates a Informatica parameter file with all required information for this source system. Any kind of failure and error encountered while executing this job, is handled and notified by Control M.

Event 4: After successful creation of infa parameter file, application/workflow corresponding to that source system is triggered using the parameter file generated in event 3. Any kind of failure and error encountered while executing this job is handled and notified by Control M.

Event 5: Informatica is responsible for producing message to TIBCO JMS message queue for successful/failed status of the file processing.

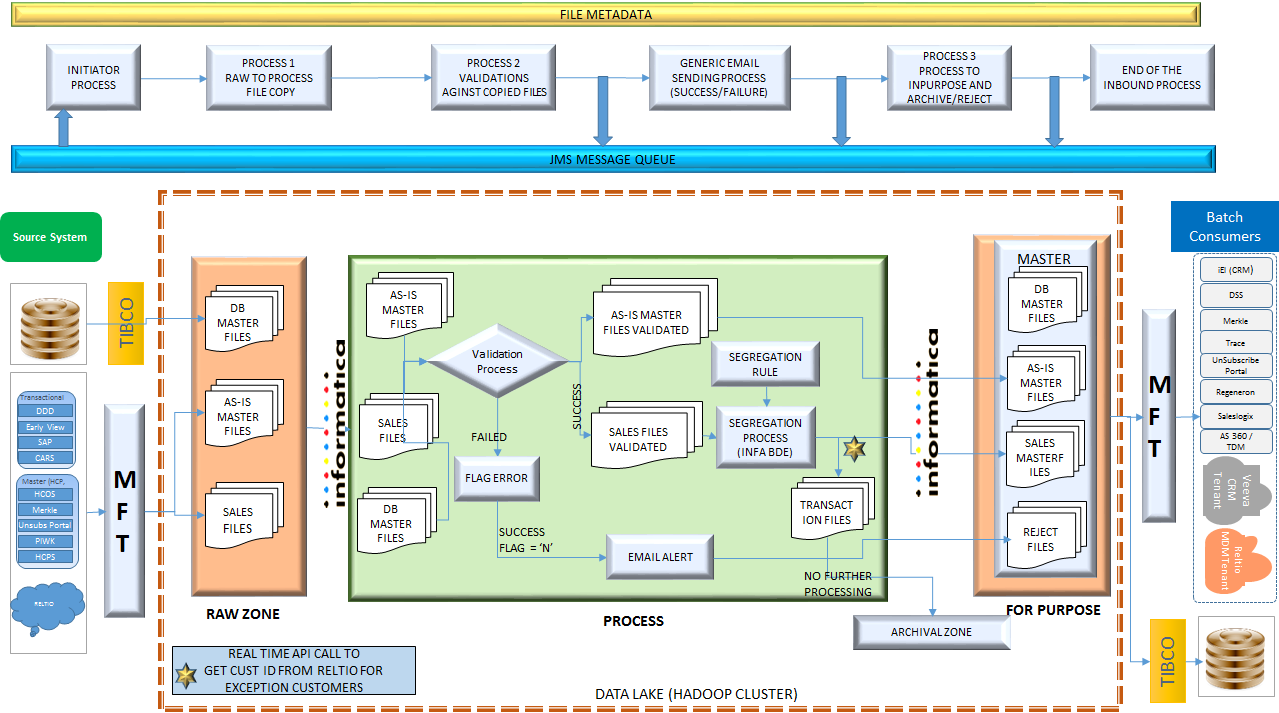
Event 6: On successful completion of the Informatica workflow, the entire process is marked as complete.

## INFA BDE Environment

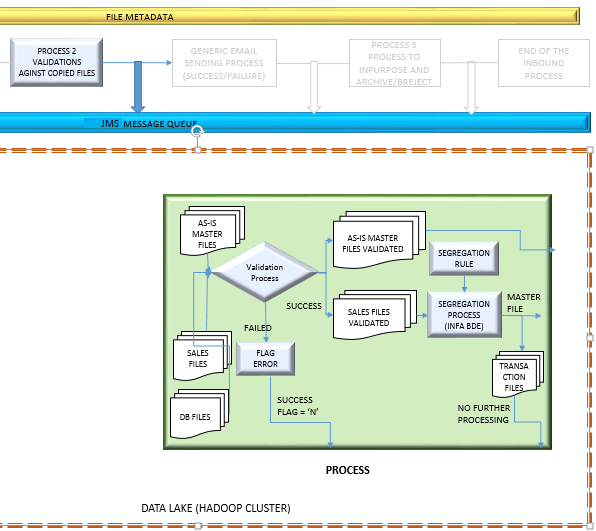
In the previous section, it is discussed how Control M invokes INITIATIOR PROCESS and how workflow for specific sources are getting executed. After invocation of a specific workflow, the process control shifts to INFA BDE component and associated Unix Scripts. For any type of error logging, exception handling and messaging during process, INFA BDE component and associated Unix scripts will handle it.

Level 1 :

Depicts the over all process of INFA BDE component, taking file system and relational database system as source.



Level 2 - A :



**PROCESS2**

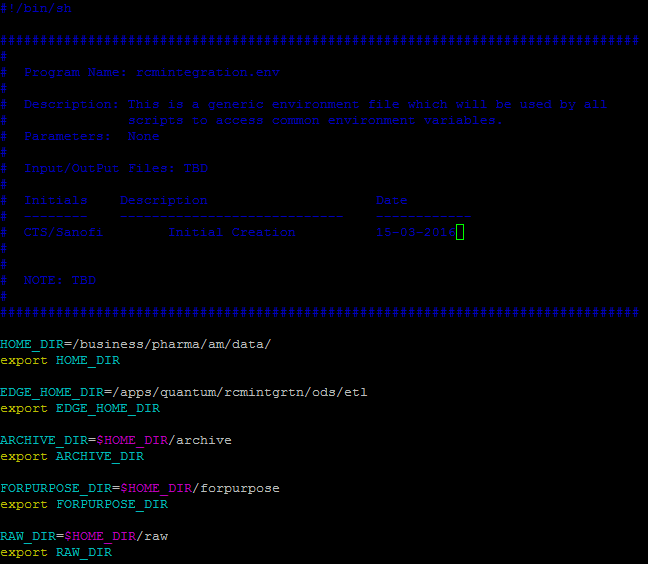
This PROCESS primarily performs the file level validations against the files copied from RAW zone.

File validation rules (CONTROL, PATTERN, THRESHOLD Checks) are fetched from param\_rcmintegration.config(metadata file) and param\_fileformat.config (metadata file) file for respective source systems. File validation process is a reusable function which includes parameter files, source definition taken from functional documents and component level program module which validates the files according to the definition written in parameter files.

Environment file:

**rcmintegration.env** file contains all the required environment definitions. Variables are defined to manage different paths and used those variables in scripts, so that in case of any change in directory path, will have no impact on scripts, only we need to change in the environment file.





Parameter files are:

1. param\_rcmintegration.config
2. param\_fileformat.config

param\_rcmintegration.config

This is a parameter file used to store parameter set for RCM integration for various processes.

param\_rcmintegration.config file contains the below details:

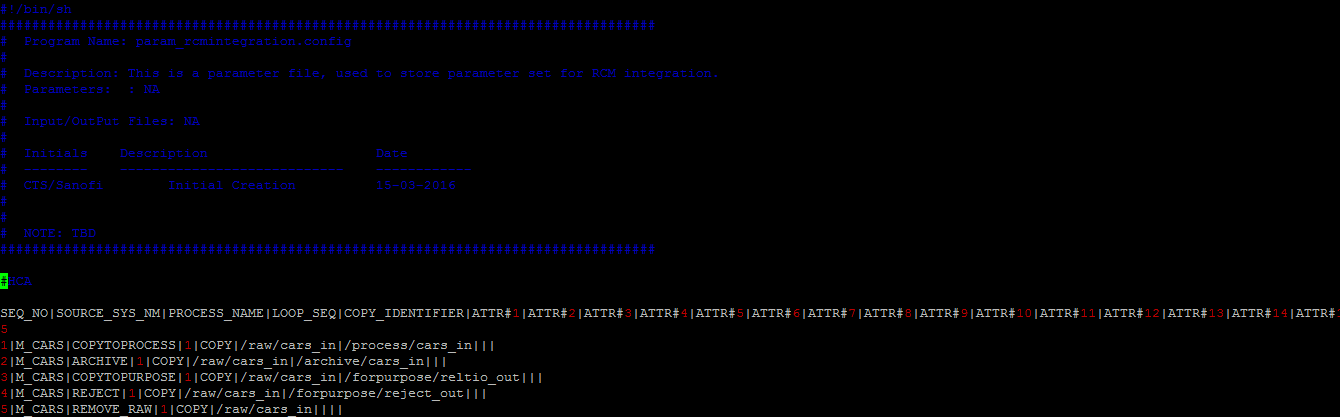
SEQ\_NO|SOURCE\_SYS\_NM|PROCESS\_NAME|LOOP\_SEQ|COPY\_IDENTIFIER|ATTR#1|ATTR#2|ATTR#3|ATTR#4|ATTR#5|ATTR#6|ATTR#7|ATTR#8|ATTR#9|ATTR#10|ATTR#11|ATTR#12|ATTR#13|ATTR#14|ATTR#15

In process\_name, there are entries for different processes like COPYTOPROCESS, ARCHIVE, COPYTOPURPOSE, VALIDATION etc. For each source system, we are making entries in this config file and grain is on source level.

**Attribute defifnition:**

|  |  |
| --- | --- |
| **Attribute Name** | **Definition** |
| SEQ\_NO | Serial number |
| SOURCE\_SYS\_NM | Name of the source system (this wil be same in both the config files) |
| PROCESS\_NAME | name of the process/task e.g COPYTOPROCESS ARCHIVE COPYTOPURPOSE REJECT REMOVE\_RAW REMOVE\_PROCESS VALIDATION |
| LOOP\_SEQ | sequence number within same source entries |
| COPY\_IDENTIFIER | Identifier to check whether files will be copied to one HDFS path to another (For the task of COPYTOPROCESS ARCHIVE COPYTOPURPOSE etc.). Value - COPY |
| Attr# | configurable attributes such as validation flags, directory paths etc. |

param\_rcmintegration.config:



param\_fileformat.config

This is a parameter file used to store parameter set for RCM integration at 2nd Level.

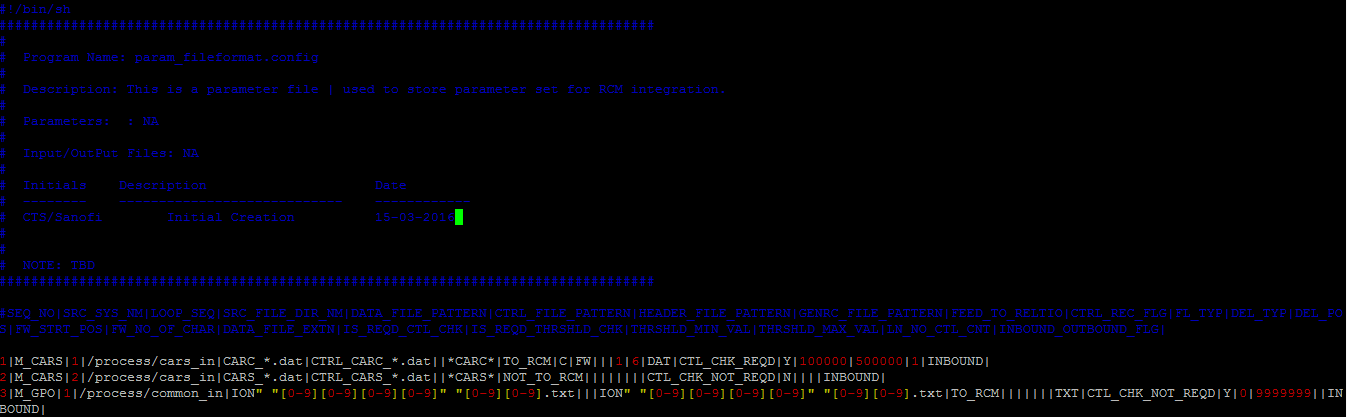
param\_fileformat.config file, we are keeping all the required configuration values in file level. SOURCE\_SYS\_NM would contain the same values in both the files. param\_fileformat.config file contains below details:

SEQ\_NO|SRC\_SYS\_NM|LOOP\_SEQ|SRC\_FILE\_DIR\_NM|DATA\_FILE\_PATTERN|CTRL\_FILE\_PATTERN|HEADER\_FILE\_PATTERN|GENRC\_FILE\_PATTERN|FEED\_TO\_RELTIO|CTRL\_REC\_FLG|FL\_TYP|DEL\_TYP|DEL\_POS|FW\_STRT\_POS|FW\_NO\_OF\_CHAR|DATA\_FILE\_EXTN|IS\_REQD\_CTL\_CHK|IS\_REQD\_THRSHLD\_CHK|THRSHLD\_MIN\_VAL|THRSHLD\_MAX\_VAL|LN\_NO\_CTL\_CNT|INBOUND\_OUTBOUND\_FLG|

**Attribute defifnition:**

|  |  |
| --- | --- |
| **Attribute Name** | **Definition** |
| SEQ\_NO | Serial number |
| SRC\_SYS\_NM | Source system name |
| LOOP\_SEQ | sequence number within same source entries |
| SRC\_FILE\_DIR\_NM | Source file directory path |
| DATA\_FILE\_PATTERN | Data file pattern |
| CTRL\_FILE\_PATTERN | Control file pattern |
| HEADER\_FILE\_PATTERN | Header file pattern |
| GENRC\_FILE\_PATTERN | generic file pattern to pick all the files from same source |
| FEED\_TO\_RELTIO | Flag to identify whether this file is required to pass to Reltio, possible values - TO\_RCM/NOT\_TO\_RCM |
| CTRL\_REC\_FLG | Identifier to check where the control count is available in header/control file. C- Control, H- header |
| FL\_TYP | Type of file where control record is mentioned. FW - Fixed width, DEL – Delimited |
| DEL\_TYP | Delimiter type. e.g. PIPE – ‘|’ |
| DEL\_POS | Position of the delimiter where the control record is present. |
| FW\_STRT\_POS | Contains the starting position of the control record for fixed width file. |
| FW\_NO\_OF\_CHAR | Contains the number of characters in the control count in fixed width file. |
| DATA\_FILE\_EXTN | File extension of data file to identify whether files are Zipped or not. ZIP- Zipped/ Non-ZIP/TXT – text/ XLSX – Excel etc. |
| IS\_REQD\_CTL\_CHK | Flag to identify whether control check is required in file level. |
| IS\_REQD\_THRSHLD\_CHK | Flag to identify whether threshold check is required in file level. |
| THRSHLD\_MIN\_VAL | Min number of expected records in corresponding file. |
| THRSHLD\_MAX\_VAL | Min number of expected records in corresponding file. |
| LN\_NO\_CTL\_CNT | Line number where the control count will be present in control/header file. |
| INBOUND\_OUTBOUND\_FLG | Whether file is coming from source (INBOUND) or INTEGRATION HUB has extracted Customer data and prepared file (OUTBOUND). |
|  |  |
|  |  |

param\_fileformat.config:



**Integration Hub file validation Mechanism:**

Integration Hub performs three different types of validations

* Pattern Check: The process ensures file Set Count Check for a source. System will maintain a set of information in both param\_rcmintegration.config(metadata file) and param\_fileformat.config (metadata file) file which will primarily contain the column as mentioned in both metadata files. These entries are specific to the specific source system and the file having entries for all the source systems on which this check needs to be applied. Checks are done on the total files recieved from the source in the RAW zone for a source system and are compared to the values present in the METADATAFILE for that source system.
* Control Check: This process ensures file count check against respective control files. It is expected that the incoming files are associated with a control file which holds the file name and the corresponding file count in it. This process checks the corresponding file name and it's record count after moving the file to the process zone. In case source file has .zip or .gz extension these are unzipped before initiaiting count verification process.
* Threshold Check: It ensures “Record count checks against threshold value”. This process use above mentioned metadata file to get the threshold value against sourec file and checks the record present in the file. It compares the threshold record count with that of the actual record count in the incoming file.

Validation Flow:

1. Get the message from MFT / TIBCO through java API.
2. Start traversing the exact source folder in RAW zone
3. Start PATTERN CHECK validation
4. Lookup the parameter file for number of files, should come for this source.
5. Assign a variable with the value in step 4
6. Count no. of files for the source.
7. Validate the result of step 4 and step 6.
8. If matched, go to step 9, else go to step **error**
9. Traverse each datafile files for respective control file
10. If data file has its control file then go to step 11 , else skip this data file as valid
11. Compare the control file record count value with actual data file record count.
12. If matched then go to step 13 , else go to step **error**
13. Traverse the parameter file for respective data file.
14. Get the threshold value from parameter file/ hive table.
15. Compare it with the record count for that data file.
16. If there is some high peak noticed beyond a limit go to step error , else go to step 17
17. File is valid

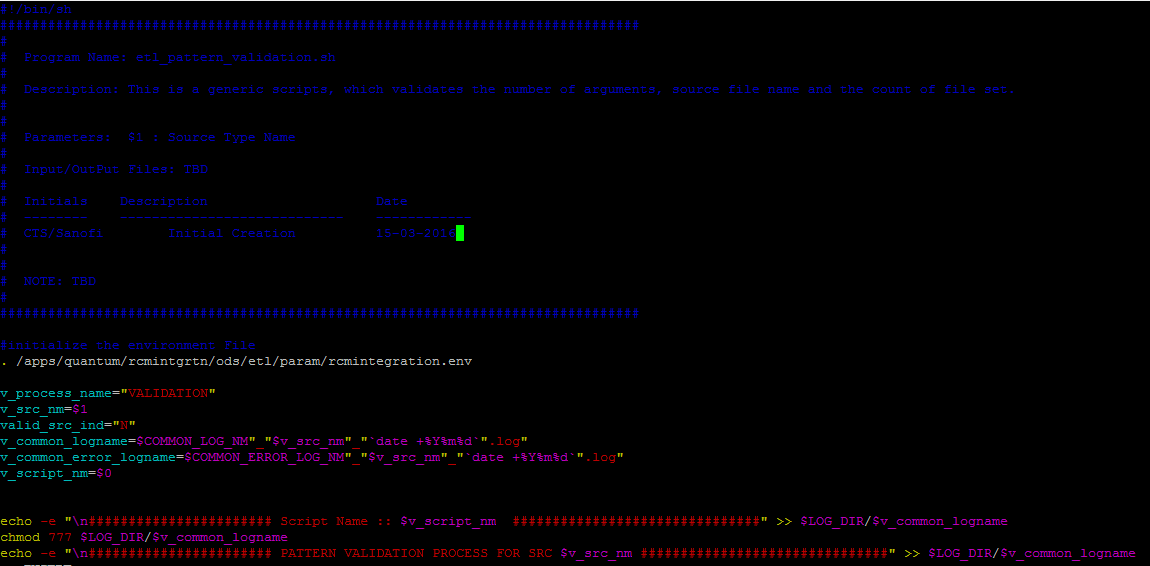
**error :** error handling:

1. Archive the file in reject data location
2. Inform the RCM via mail notification

**Program Component for Validation:**

1. Pattern Validation : For this validation , we have generic **etl\_pattern\_validation.sh** script.

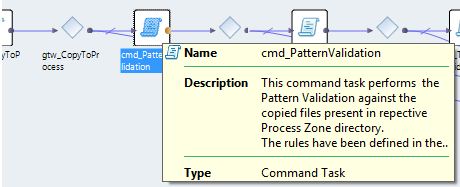




This script accepts source\_system name (As-IS mentioned in config files) and first invokes the environment file for initialization of directory variables. Then search for required pattern check flag in ATTR#5, file patterns, expected number of files for each pattern and corresponding directory path, in the param\_rcmintegration.config file aginst source\_sytem\_nm (comparing with received argument) and process\_name (VALIDATION). If the value of ATTR#5 is PATRN\_CHK\_REQ then script progresses further and checks for the actual file count using file pattern and compares with expected count - successful if both the counts match.

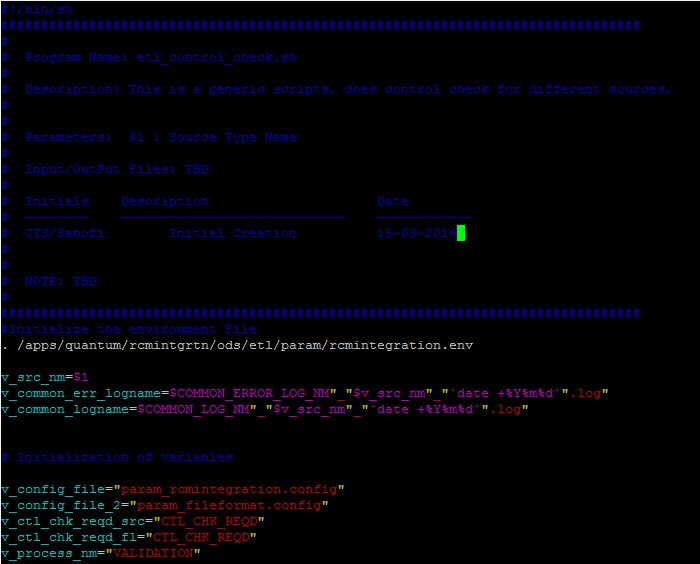
This scripts is called from “cmd\_PatternValidation” command task in “wf\_Integration Hub\_asis\_inbound\_file\_processing” workflow from Informatica BDE.

This workflow invoked by Control M when required.



1. Control Validation: For this validation , we have generic **etl\_control\_validation.sh** script.

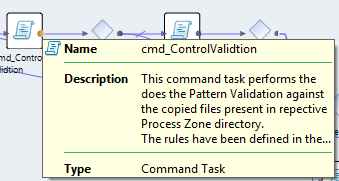




This script accepts source\_system name (As-IS mentioned in config files) and first invokes the environment file for initialization of directory variables. Then search for required Control check flag in ATTR#3 in param\_rcmintegration.config. If the value of ATTR#3 is CTL\_CHK\_REQD then only script progresses further otherwise no control check would be done for that source. If it is mentioned as 'CTL\_CHK\_REQD' in param\_rcmintegration.config, then it checks in param\_fileformat.config for individual control check req. flag (IS\_REQD\_CTL\_CHK ) to identify whether for each file pattern mentioned there control check is needed or not. If yes then it checks what are the required control, header and data file pattern and all other required information and carry out the control check process. It picks the record count of actual file and control count from Control/header file as mentioned in param\_fileformat.config and compares - successful if both the counts match.

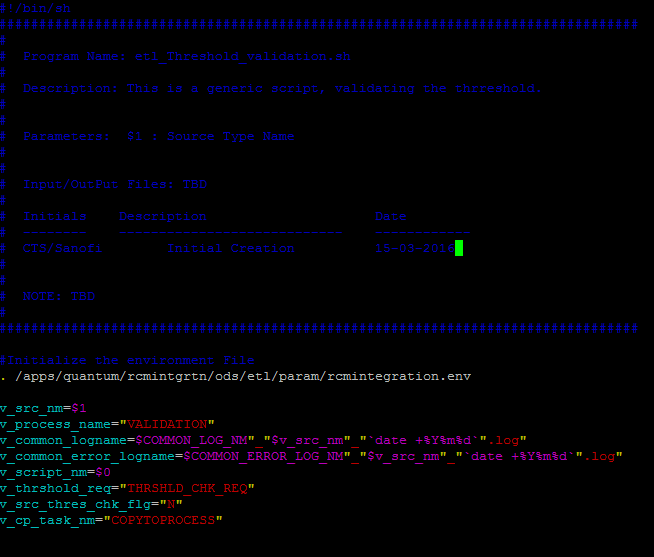
This scripts is called from “cmd\_ControlValidtion” command task in “wf\_Integration Hub\_asis\_inbound\_file\_processing” workflow from Informatica BDE.

This workflow invoked by Control M when required.



1. Threshold Validation: For this validation , we have generic **etl\_threshold\_validation.sh** script.

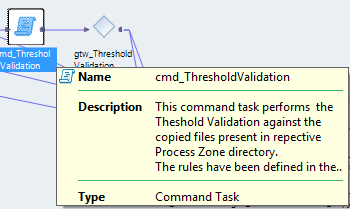




This script accepts source\_system name (As-IS mentioned in config files) and first invokes the environment file for initialization of directory variables. Then search for required Control check flag in ATTR#4 in param\_rcmintegration.config. If the value of ATTR#4 is ‘THRSHLD\_CHK\_REQD’ then only script progresses further otherwise no threshold check would be done for that source system. If it is mentioned as 'THRSHLD\_CHK\_REQD' in param\_rcmintegration.config, then it will check in param\_fileformat.config for individual threshold check req. flag (IS\_REQD\_THRSHLD\_CHK) to identify whether for each file pattern mentioned there threshold check is needed or not. If yes then it checks what are the required data file pattern (DATA\_FILE\_PATTERN), minimum (THRSHLD\_MIN\_VAL) and maximum (THRSHLD\_MAX\_VAL) values and all other required information and carry out the threshold check process. It picks the record count of actual file and checks whether that is in between the threshold values.

This scripts is called from “cmd\_ThresholdValidation” command task in “wf\_Integration Hub\_asis\_inbound\_file\_processing” workflow from Informatica BDE.

This workflow invoked by Control M when required.



This PROCESS holds flags (SUCCESS/FAILURE) for each validation done on the files for any source system.

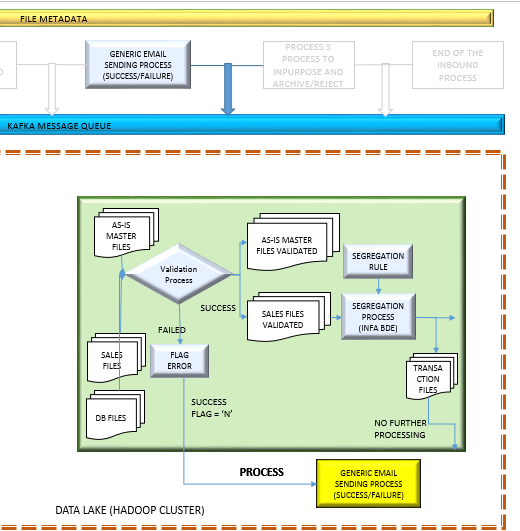
All master as well as transaction files undergo the above validations. Customer files from transactional sources goes through THRESHOLD Checks only.

After successful completion of the validations, it writes into the message queue.

The message structure is mentioned in “**GENERIC MESSAGE SENDING PROCESS**” section.

Message may contain Source name, File name, file count with timestamp. Message format might be like below:

Components Involved: Informatica BDE,UNIX Shell Scripts, Metadata Files , Message Queue.

Level 2 - B:

**GENERIC EMAIL SENDING PROCESS**

This is a generic email task which sends custom emails to the intended recipient for respective source system. Once the file validation FAILS this PROCESS is invoked and an email is sent to the respective Quantum Ops Team requesting to resend the files. It intimates the validation failure and the reason for the same.

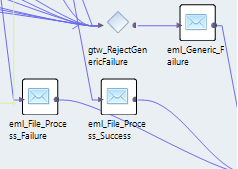
Components Involved: Informatica BDE,UNIX Shell Scripts, Metadata Files.

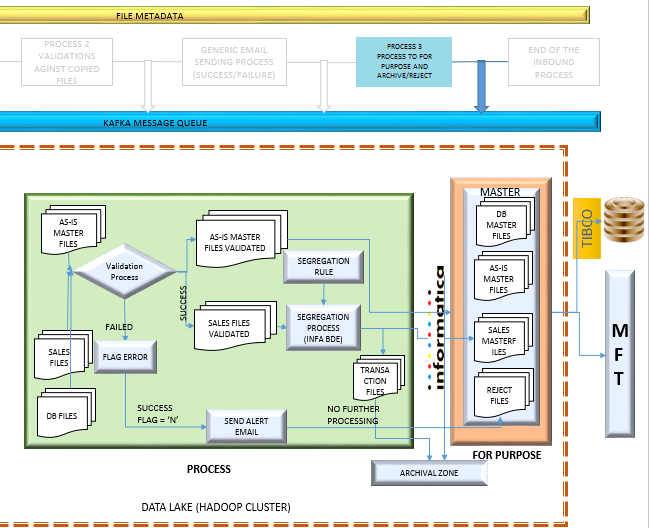
Infa BDE component has developed few email task under workflow “wf\_Integration Hub\_asis\_inbound\_file\_processing”. This are very generic email task and can be called from anywhere withing as and when required.

eml\_Generic\_Failure : For any kind of generic workflow/ process failure send mails to intended recipients.

eml\_File\_Process\_Success : For send mails to intended recipients after success.

eml\_File\_Process\_Failure : For send mails to intended recipients after failure.



Level 2 - D:

**PROCESS 3**

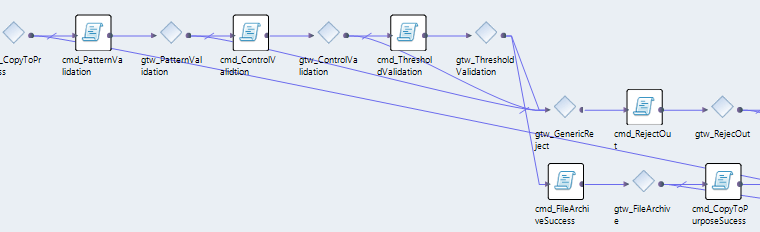
On SUCCESSFUL Validation: File is copied to FOR PURPOSE zone in respective source folder (Eg . Reltio\_out). In case of AS-IS source, file from Raw zone is copied to FOR PURPOSE zone.

Also source files from Raw zone is moved to ARCHIVAL zone in respective folder in compressed format. .

On Validation FAILURE: Source files is moved to reject\_out in the FOR PURPOSE zone. Pulling the file from RAW zone for this activity. For all these activity , process is following METADATA files definition and function accordingly.

Components Involved: Informatica BDE,UNIX Shell Scripts, Metadata Files.

Infa BDE component has developed workflow “wf\_Integration Hub\_asis\_inbound\_file\_processing”.

*Workflow level diagram:* (detailed sequence of task is described in section 9)

After successful completion of the validations, this component writes into the message queue. Message will contain file name and source name, target system name with timestamp.

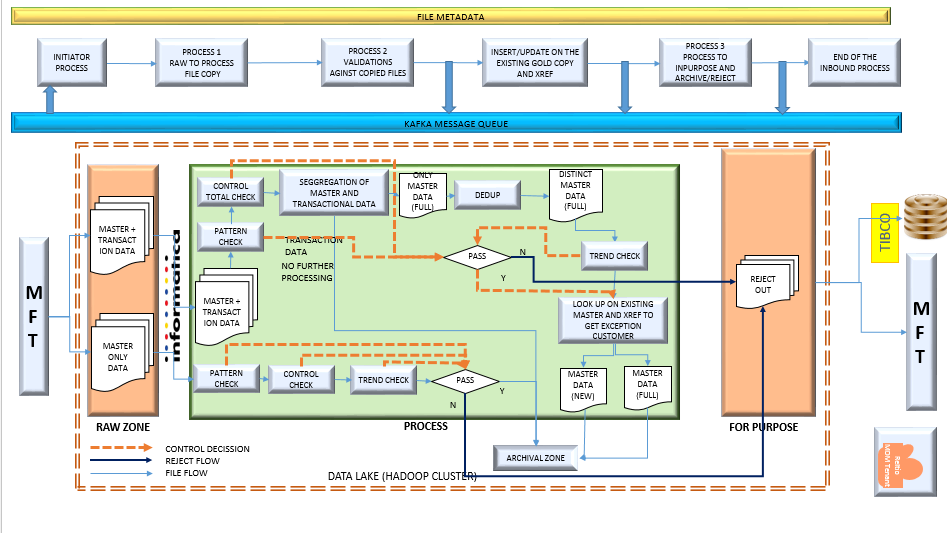
Archival Strategy :

**Master only file**:

1. Validation passed:
2. Copy source files to For Purpose Zone (on corresponding source dir. path) from Raw Zone.
3. Copy source files to Archive from Raw Zone.
4. Remove source files from Process Zone.
5. Remove source files from Raw Zone.
6. Validation failed:
7. Flag the failure (email) to Quantum Ops Team and write message in JMS message topic.
8. Copy source files to Reject\_out from Raw Zone.
9. Remove source files from Process Zone.
10. Remove source files from Raw Zone.

**Master with Transaction data**:

1. Validation passed:
2. Copy source files to For Purpose Zone (on corresponding source dir. path) from Raw Zone.
3. Copy Master Data (New) and Master Data (Full) files to Archive from Process Zone.
4. Copy source files to Archive from Raw Zone.
5. Copy Transaction Data file to Archive from Process Zone.
6. Remove Master Data (New) and Master Data (Full) files from Process Zone.
7. Remove source files from Raw Zone.
8. Validation failed:
9. Flag the failure (email) to Source and write message in JMS message topic.
10. Copy source files to Reject\_out from Raw Zone.
11. Copy Master Data (Full) file (if avialable) to Reject\_out from Process Zone.
12. Remove Master Data (Full) file from Process Zone.
13. Remove source files from Process Zone.
14. Remove source files from Raw Zone.



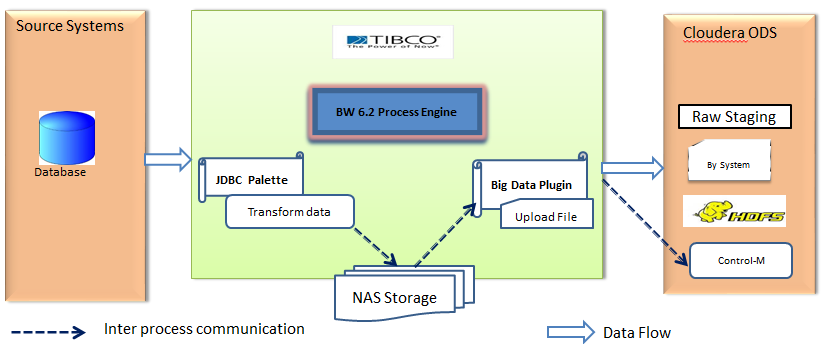
**GENERIC MESSAGE SENDING PROCESS** (TBD)

INFA BDE has a component which will generate message after completion of certain process. That component will use the below .xsd format to write message in the JMS queue.



## TIBCO Environment

### Database As Source



1. TIBCO BW 6.2 process will start either based on timer or message based event.

2.The BW JDBC Palette will query the data present in the HCP, DSS/TDM source tables.

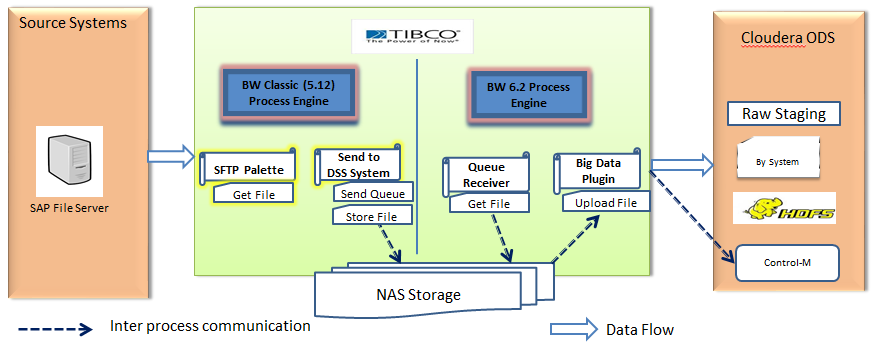
3.The fetched data will be transformed as per requirement and output file is created at local NAS storage location.

4.BW process will perform Kerberos authentication to connect to Cloudera HDFS file system using the big data plugin.

5.Process will upload the output file to Cloudera HDFS file system using the HDFS PUT Operation activity of the big data plugin.

6.A TIBCO EMS control message is published with meta data information.

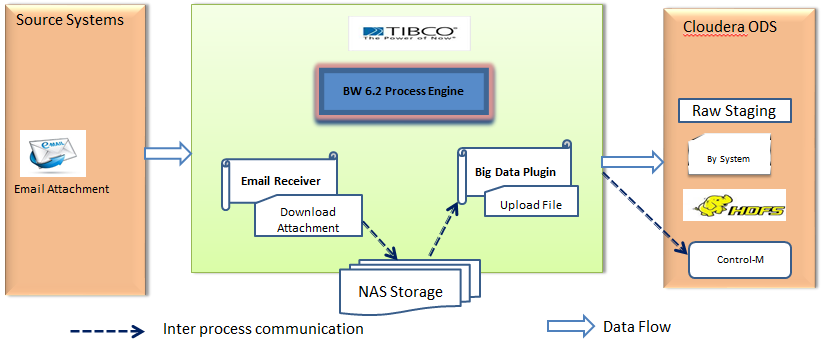
### Sap File Server As Source



1. Existing TIBCO BW Classic will process SAP files from STP location and perform the required validations and notifications ASIS and transfer the files to DSS system.
2. Changes will be done to existing process to store a copy of the success file transferred to NAS share location and send a EMS queue message with details.
3. TIBCO BW 6.2 process will start based on queue message event.
4. BW process will perform Kerberos authentication to connect to Cloudera HDFS file system using the big data plugin.
5. Process will upload the output file to Cloudera HDFS file system using the HDFS Operation activity of big data plugin.
6. A TIBCO EMS control message is published with meta data information.

Assumption: Current TIBCO BW process which processes file to DSS wont retire after MDM implementation.

### Email As Source



1. TIBCO BW 6.2 process will start on receiving the email message and download the attachment to local NAS location.
2. BW process will perform Kerberos authentication to connect to Cloudera HDFS file system using the big data plugin.
3. Process will upload the output file to Cloudera HDFS file system using the HDFS Operation activity of the big data plugin.
4. A TIBCO EMS control message is published with meta data information.

# 8. Standard and Rules

Naming Standards

8.1. Inbound Data File Naming Standards:

Control M initiates the end to end process for each of the source systems and first, makes the source files availbale in the RAW zone and subsequently triggers the workflows and unix scripts for processing the source files.

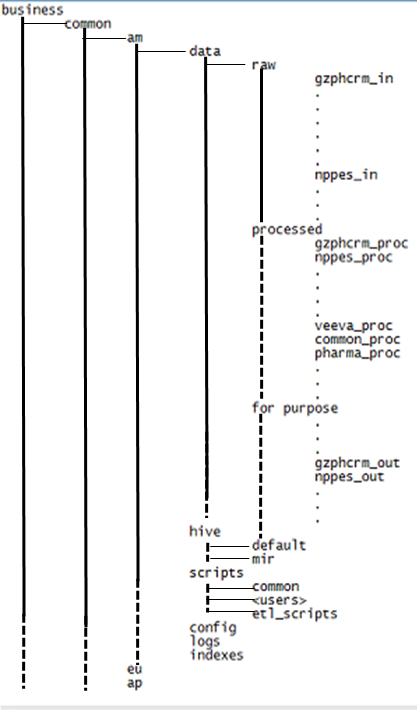
PFB the details of the incoming files in the attached document:



8.2. Build Environment Standards

O/S File System Folder Structure:

The sample folder structure is shown below for the data section:



8.3. O/S Scripts Naming pattern and coding conventions:

INFA BDE operating System Environment follows the below described coding convetions:

1. An environment file which will hold all the definition of folder structure of specified O/S environment. The name of the file “etl\_run\_env.sh”(if it is bash shell )
2. The above file contains, programmer defined directory definition on integration hub environment.
3. If any password file needed that will be defined later.
4. Any program scripts or definition script should follow the below pattern of wrtting code :-
   1. Definition of shell on which it executes(i.e. #!/usr/bin/sh), if the environment is in bash shell
   2. Definition of Scripts, which includes
      1. Program Name
      2. Description
      3. Parameters(if any)
      4. Input/Output Files
      5. Initial, Desciption, Dates
      6. Purpose
      7. Code Body
      8. Parameter Acceptance Section
      9. Environment files calling section
      10. Code section
      11. Exception handling
      12. Audit / Logging section
5. Naming pattern of Scripts : **etl\_<functionality>\_<datalake area>.sh**
6. Naming pattern of parameter file **: param\_<functionality>\_<datalake area>.sh**
7. Naming pattern of log file : **log\_<functionality>\_<datalake area>.sh**

8.5. Informatica Object Naming Standards in ETL Layer:



Below are the naming conventions that are followd as a part of development activity.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| SL No | SUBJECT | OBJECT | NAMING CONVENSION | EXAMPLE | COMMENTS |
| 1 | INFA BDE | Mapping | m\_<layer\_name>\_<Source\_name>\_<attribute type> | m\_inb\_dss\_cust | inb:= inbound |
| m\_stg\_dss\_cust | stg:=staging (Raw) |
| m\_prc\_dss\_cust | prc:= process |
| m\_out\_dss\_cust | out := outbound |
| 2 | INFA BDE | Workflow | wf\_<layer\_name>\_<Source\_name>\_<attribute type> | wf\_inb\_dss\_cust | inb:= inbound |
| wf\_stg\_dss\_cust | stg:=staging (Raw) |
| wf\_prc\_dss\_cust | prc:= process |
| wf\_out\_dss\_cust | out := outbound |
| 3 | INFA BDE | Application | app\_<layer\_name>\_<Source\_name>\_<attribute type> | app\_inb\_dss\_cust | inb:= inbound |
| app\_stg\_dss\_cust | stg:=staging (Raw) |
| app\_prc\_dss\_cust | prc:= process |
| app\_out\_dss\_cust | out := outbound |
| 4 | INFA BDE | Inbound Table | Table\_name | c\_cust\_master |  |
| 5 | INFA BDE | Expression | exp\_<function> | exp\_passthrough |  |
| exp\_uppercase |
| 6 | INFA BDE | Sequence | seq\_<function> | seq\_cust\_id |  |
| 7 | INFA BDE | Filter | flt\_<function> |  |  |
| 8 | INFA BDE | Join | jnr\_<Purpose> | jnr\_cust\_class\_pop |  |
| 9 | INFA BDE | Sorter | srt\_<sort\_column>\_<uniq(if checked)> | srt\_cust\_id/ srt\_cust\_id\_uniq |  |
| 10 | INFA BDE | Router | rtr\_<coulumn\_name> | rtr\_cust\_class |  |
| 11 | INFA BDE | Target File | write\_tgt\_<src\_nm>\_<attrubute\_type> | write\_tgt\_dss\_cust |  |
| 12 | INFA BDE | Mapplet | mplt\_{Description\_of\_Process} | mplt\_lookup\_cust |  |
| 12 | INFA BDE | Mapping Variable | mvar\_<column\_name> | $mvar\_src\_sys\_nm |  |
| $mvar\_sys\_timestamp |
| 13 | INFA BDE | Expression Variable | v\_<column\_name> | v\_cust\_nm |  |
| v\_src\_sys\_id |
| 14 | INFA BDE | Input port | in\_<column\_name> | in\_cust\_nm | For Input only ports |
| in\_src\_sys\_id |
| 15 | INFA BDE | Output port | o\_<column\_name> | o\_cust\_nm | For output only ports |
| o\_src\_sys\_id |

8.6. Technical Design Rules

N/A

# Technical Component Level specification for processing files:

Component Description for AS-IS files:

|  |  |  |  |
| --- | --- | --- | --- |
| SL NO | INFA Component Name | Underlying Unix Scripts Name | Description |
|
| 1 | cmd\_copytoprocess | etl\_cp\_mv\_common\_process.sh | Copy from HDFS raw zone folder to process zone. |
| 2 | cmd\_PatternValidation | etl\_pattern\_validation.sh | Generic Pattern Validations Script |
| 3 | cmd\_ControlValidtion | etl\_control\_validation.sh | Generic control Validations Script |
| 4 | cmd\_ThresholdValidation | etl\_threshold\_validation.sh | Generic Threshold Validations Script |
| 5 | cmd\_RejectOut | etl\_cp\_mv\_common\_process.sh | Generic Common File Move process Script |
| 6 | cmd\_RemoveRawFailure | etl\_cp\_mv\_common\_process.sh | Generic Common File Move process Script |
| 7 | cmd\_RemoveProcessFailure | etl\_cp\_mv\_common\_process.sh | Generic Common File Move process Script |
| 8 | cmd\_FileArchiveSuccess | etl\_cp\_mv\_common\_process.sh | Generic Common File Move process Script |
| 9 | cmd\_CopyToPurposeSucess | etl\_cp\_mv\_common\_process.sh | Generic Common File Move process Script |
| 10 | cmd\_RemoveRawSuccess | etl\_cp\_mv\_common\_process.sh | Generic Common File Move process Script |
| 11 | cmd\_RemoveProcessSuccess | etl\_cp\_mv\_common\_process.sh | Generic Common File Move process Script |
| 12 | eml\_File\_Process\_Success | NA | Generic email notification on success |
| 13 | eml\_Generic\_Failure | NA | Generic email notification on failure |
| 14 | eml\_FileValidation\_Failure | NA | Generic email notification on failure |
| 15 | m\_Java\_JMS\_Producer\_Success | NA | Task for message queueing for success |
| 16 | m\_Java\_JMS\_Producer\_Failure | NA | Task for message queueing for failure |

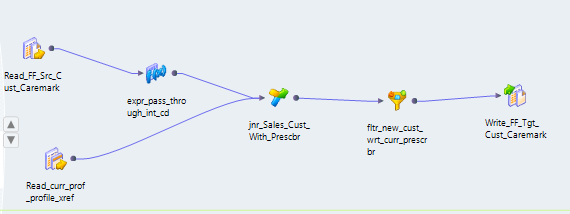
Component Description for SALES files:

|  |  |  |  |
| --- | --- | --- | --- |
| SL NO | INFA Component Name | Underlying Unix Scripts Name | Description |
|
| 1 | cmd\_copytoprocess | etl\_cp\_mv\_common\_process.sh | Copy from HDFS raw zone folder to process zone. |
| 2 | cmd\_PatternValidation | etl\_pattern\_validation.sh | Generic Pattern Validations Script |
| 3 | cmd\_ControlValidtion | etl\_control\_validation.sh | Generic control Validations Script |
| 4 | cmd\_Cust\_Data\_Seg | etl\_mstr\_extract\_process.sh | Generic script to extract master data |
| 5 | cmd\_ThresholdValidation | etl\_threshold\_validation.sh | Generic Threshold Validations Script |
| 6 | cmd\_RejectOut | etl\_cp\_mv\_common\_process.sh | Generic Common File Move process Script |
| 7 | cmd\_RemoveRawFailure | etl\_cp\_mv\_common\_process.sh | Generic Common File Move process Script |
| 8 | cmd\_RemoveProcessFailure | etl\_cp\_mv\_common\_process.sh | Generic Common File Move process Script |
| 9 | m\_Task\_New\_Cust\_Extract\_<sourcename> | NA | Reference current customers and new customer file generation.[Described in Mapping Section] |
| 10 | cmd\_FileArchiveSuccess | etl\_cp\_mv\_common\_process.sh | Generic Common File Move process Script |
| 11 | cmd\_CopyToPurposeSucess | etl\_cp\_mv\_common\_process.sh | Generic Common File Move process Script |
| 12 | cmd\_RemoveRawSuccess | etl\_cp\_mv\_common\_process.sh | Generic Common File Move process Script |
| 13 | cmd\_RemoveProcessSuccess | etl\_cp\_mv\_common\_process.sh | Generic Common File Move process Script |
| 14 | eml\_File\_Process\_Success | NA | Generic email notification on success |
| 15 | eml\_Generic\_Failure | NA | Generic email notification on failure |
| 16 | eml\_FileValidation\_Failure | NA | Generic email notification on failure |
| 17 | m\_Java\_JMS\_Producer\_Success | NA | Task for message queueing for success |
| 18 | m\_Java\_JMS\_Producer\_Failure | NA | Task for message queueing for failure |

**Mapping**

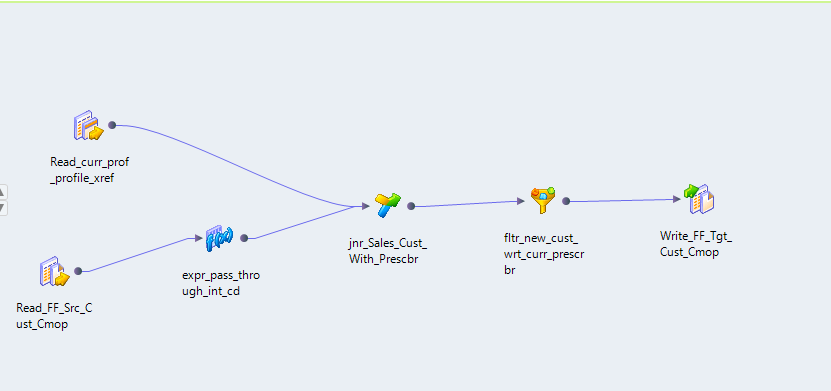
m\_Task\_New\_Cust\_Extract\_caremark

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **SL NO** | **Mapping component Name** | **Description** | **Previous Component** | **Next Component** |
| 1 | Read\_FF\_Src\_Cust\_ caremark | Defined as Source in Informatca Mapping | NA | expr\_pass\_through\_int\_cd |
| 2 | Read\_curr\_prof\_profile\_xref | Defined as x-ref Source in Informatca Mapping | NA |  |
| 3 | expr\_pass\_through\_int\_cd | In expression, the INTERFACE CODE is added as ‘CMRK’ with Source Data | Read\_FF\_Src\_Cust\_ caremark | jnr\_Sales\_Cust\_With\_Prescbr |
| 4 | jnr\_Sales\_Cust\_With\_Prescbr | Source data (Master) is joined with x\_ref on Master.SUPLIER\_NUM=Detail.indiv\_ssk And Master.INTERFACE\_CODE\_SRC= Detail.interface\_code\_xref | 1.       Read\_FF\_Src\_Cust\_ caremark 2.       Read\_curr\_prof\_profile\_xref | fltr\_new\_cust\_wrt\_curr\_prescrbr |
|
| 5 | fltr\_new\_cust\_wrt\_curr\_prescrbr | Verifying whether indiv\_ssk and interface\_code\_xref both are NULL | jnr\_Sales\_Cust\_With\_Prescbr | Write\_FF\_Tgt\_Cust\_Caremark |
| 6 | Write\_FF\_Tgt\_Cust\_Caremark | Generating new customer file under “/business/common/am/data/processed/caremark\_in” , name of the file is “New\_Cust\_M\_CAREMARK\_P.txt” | fltr\_new\_cust\_wrt\_curr\_prescrbr | NA |



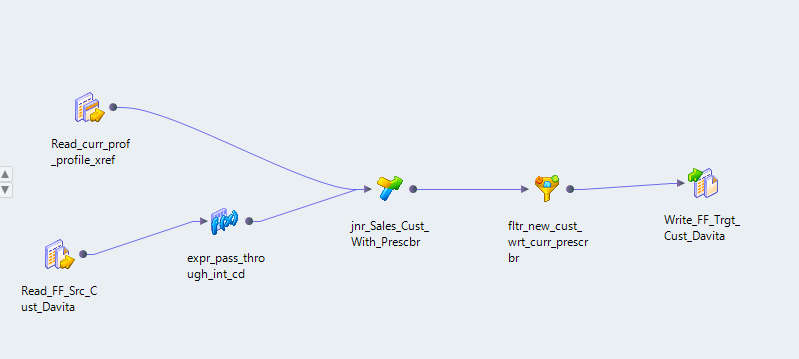
m\_Task\_New\_Cust\_Extract\_Cmop

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **SL NO** | **Mapping component Name** | **Description** | **Previous Component** | **Next Component** |
| 1 | Read\_FF\_Src\_Cust\_Cmop | Defined as Source in Informatca Mapping | NA | expr\_pass\_through\_int\_cd |
| 2 | Read\_curr\_prof\_profile\_xref | Defined as x-ref Source in Informatca Mapping | NA | jnr\_Sales\_Cust\_With\_Prescbr |
| 3 | expr\_pass\_through\_int\_cd | In expression, the INTERFACE CODE is added as ‘CMOP’ with Source Data | Read\_FF\_Src\_Cust\_Cmop | jnr\_Sales\_Cust\_With\_Prescbr |
| 4 | jnr\_Sales\_Cust\_With\_Prescbr | Source data (Master) is joined with x\_ref on Master.CMOP=Detail.indiv\_ssk And Master.INTERFACE\_CODE= Detail.interface\_code\_xref | 1.Read\_curr\_prof\_profile\_xref 2expr\_pass\_through\_int\_cd | fltr\_new\_cust\_wrt\_curr\_prescrbr |
| 5 | fltr\_new\_cust\_wrt\_curr\_prescrbr | Verifying whether indiv\_ssk and interface\_code\_xref both are NULL | jnr\_Sales\_Cust\_With\_Prescbr | Write\_FF\_Tgt\_Cust\_Cmop |
| 6 | Write\_FF\_Tgt\_Cust\_Cmop | Generating new customer file under “/business/common/am/data/processed/cmop\_in” , name of the file is “New\_Cust\_M\_CMOP\_A.txt” | fltr\_new\_cust\_wrt\_curr\_prescrbr | NA |



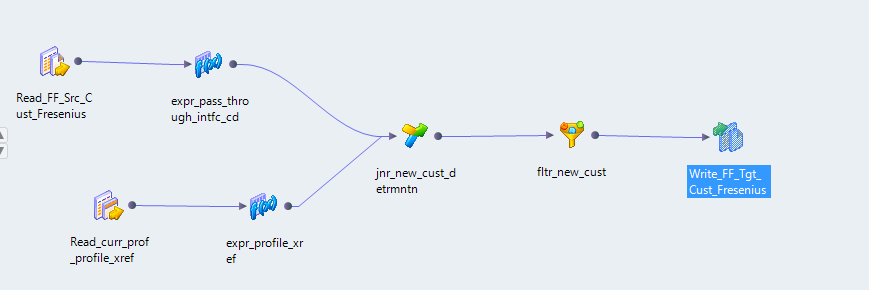
m\_Task\_New\_Cust\_Extract\_Davita

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **SL NO** | **Mapping component Name** | **Description** | **Previous Component** | **Next Component** |
| 1 | Read\_FF\_Src\_Cust\_Davita | Defined as Source in Informatca Mapping | NA | jnr\_Sales\_Cust\_With\_Prescbr |
| 2 | Read\_curr\_prof\_profile\_xref | Defined as x-ref Source in Informatca Mapping | NA | expr\_pass\_through\_int\_cd |
| 3 | expr\_pass\_through\_int\_cd | In expression, the INTERFACE CODE is added as ‘DVTA’ with Source Data | Read\_FF\_Src\_Cust\_Davita | jnr\_Sales\_Cust\_With\_Prescbr |
| 4 | jnr\_Sales\_Cust\_With\_Prescbr | Source data (Master) is joined with x\_ref on Master.NPI=Detail.indiv\_ssk And Master.INTERFACE\_CODE\_= Detail.interface\_code\_xref | 1.Read\_curr\_prof\_profile\_xref 2expr\_pass\_through\_int\_cd | fltr\_new\_cust\_wrt\_curr\_prescrbr |
| 5 | fltr\_new\_cust\_wrt\_curr\_prescrbr | Verifying whether indiv\_ssk and interface\_code\_xref both are NULL | jnr\_Sales\_Cust\_With\_Prescbr | Write\_FF\_Trgt\_Cust\_Davita |
| 6 | Write\_FF\_Trgt\_Cust\_Davita | Generating new customer file under “/business/common/am/data/processed/davita\_in” , name of the file is “New\_Cust\_M\_DAVITA\_P.txt” | fltr\_new\_cust\_wrt\_curr\_prescrbr | NA |



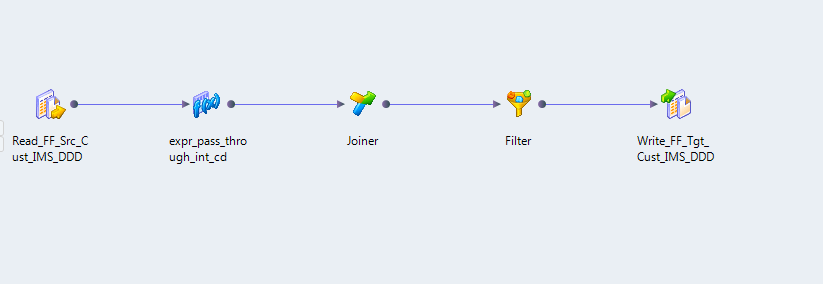
m\_New\_Cust\_Extract\_Fresenius

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **SL NO** | **Mapping component Name** | **Description** | **Previous Component** | **Next Component** |
| 1 | Read\_FF\_Src\_Cust\_Fresenius | Defined as Source in Informatca Mapping | NA | expr\_pass\_through\_intfc\_cd |
| 2 | Read\_curr\_prof\_profile\_xref | Defined as x-ref Source in Informatca Mapping | NA | expr\_profile\_xref |
| 3 | expr\_pass\_through\_intfc\_cd | In expression, the INTERFACE CODE is added as ‘FRNS’ with Source Data | Read\_FF\_Src\_Cust\_Davita | jnr\_new\_cust\_detrmntn |
| 4 | expr\_profile\_xref | In expression, the INTERFACE CODE is added as ‘interface\_code’ with Source Data | Read\_curr\_prof\_profile\_xref | jnr\_new\_cust\_detrmntn |
| 5 | jnr\_new\_cust\_detrmntn | Source data (Master) is joined with x\_ref on Master.NATIONAL\_PROVIDER\_IDENTIFIER\_NPI =Detail.indiv\_ssk And Master.INTERFACE\_CODE\_= Detail.interface\_code\_xref | 1.expr\_pass\_through\_intfc\_cd 2expr\_profile\_xref | fltr\_new\_cust |
| 6 | fltr\_new\_cust | Verifying whether indiv\_ssk and interface\_code\_xref both are NULL | jnr\_new\_cust\_detrmntn | Write\_FF\_Tgt\_Cust\_Fresenius |
| 7 | Write\_FF\_Tgt\_Cust\_Fresenius | Generating new customer file under “/business/common/am/data/processed/fresenius\_in” , name of the file is “New\_Cust\_M\_FRESENIUS\_P.txt” | fltr\_new\_cust\_wrt\_curr\_prescrbr | NA |



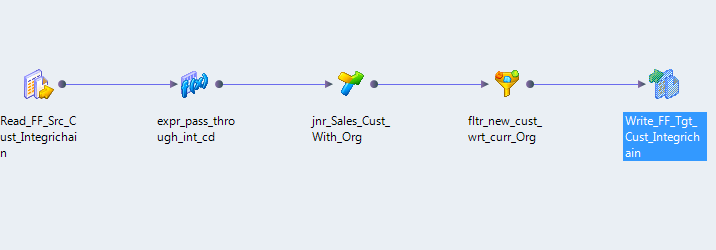
m\_New\_Cust\_Extract\_IMS\_DDD

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **SL NO** | **Mapping component Name** | **Description** | **Previous Component** | **Next Component** |
| 1 | Read\_FF\_Src\_Cust\_IMS\_DDD | Defined as Source in Informatca Mapping | NA | expr\_pass\_through\_int\_cd |
| 2 | expr\_pass\_through\_int\_cd | In expression, the INTERFACE CODE is added as ‘DDDL’ with Source Data | Read\_FF\_Src\_Cust\_IMS\_DDD | Joiner |
| 3 | Joiner | Source data (Master) is joined on Master.DDD\_Outlet\_Number =Detail.org\_ssk And Master.INTERFACE\_CODE\_SRC= Detail.interface\_code\_xref | expr\_pass\_through\_int\_cd | Filter |
| 4 | Filter | Verifying whether indiv\_ssk and interface\_code\_xref both are NULL | Joiner | Write\_FF\_Tgt\_Cust\_IMS\_DDD |
| 5 | Write\_FF\_Tgt\_Cust\_IMS\_DDD | Generating new customer file under “/business/common/am/data/processed/ddd\_in” , name of the file is “New\_Cust\_W\_IMS\_DDD\_A.txt” | Filter | NA |



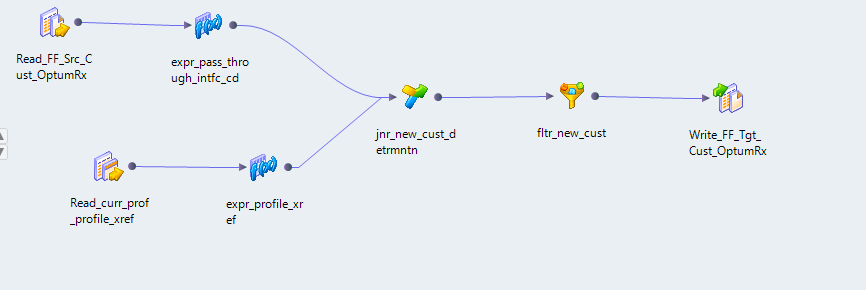
m\_New\_Cust\_Extract\_Integrichain

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **SL NO** | **Mapping component Name** | **Description** | **Previous Component** | **Next Component** |
| 1 | Read\_FF\_Src\_Cust\_Integrichain | Defined as Source in Informatca Mapping | NA | expr\_pass\_through\_int\_cd |
| 2 | expr\_pass\_through\_int\_cd | In expression, the INTERFACE CODE is added as ‘AGIC’ with Source Data | Read\_FF\_Src\_Cust\_IMS\_DDD | jnr\_Sales\_Cust\_With\_Org |
| 3 | jnr\_Sales\_Cust\_With\_Org | Source data (Master) is joined on Master SHIP\_TO\_POC\_ID =Detail.org\_ssk And Master.INTERFACE\_CODE\_SRC= Detail.interface\_code\_xref | expr\_pass\_through\_int\_cd | fltr\_new\_cust\_wrt\_curr\_Org |
| 4 | fltr\_new\_cust\_wrt\_curr\_Org | Verifying whether indiv\_ssk and interface\_code\_xref both are NULL | jnr\_Sales\_Cust\_With\_Org | Write\_FF\_Tgt\_Cust\_Integrichain |
| 5 | Write\_FF\_Tgt\_Cust\_Integrichain | Generating new customer file under “/business/common/am/data/processed/Integrichain\_in” , name of the file is “New\_Cust\_W\_IMS\_INTEGRICHAIN\_A.txt” | fltr\_new\_cust\_wrt\_curr\_Org | NA |



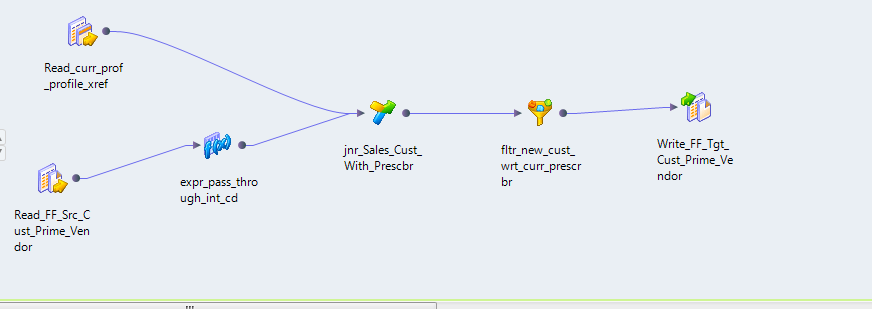
m\_New\_Cust\_Extract\_OptumRx

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **SL NO** | **Mapping component Name** | **Description** | **Previous Component** | **Next Component** |
| 1 | Read\_FF\_Src\_Cust\_OptumRx | Defined as Source in Informatca Mapping | NA | expr\_pass\_through\_intfc\_cd |
| 2 | Read\_curr\_prof\_profile\_xref | Defined as x-ref Source in Informatca Mapping | NA | expr\_profile\_xref |
| 3 | expr\_pass\_through\_intfc\_cd | In expression, the INTERFACE CODE is added as ‘OPTR’ with Source Data | Read\_FF\_Src\_Cust\_OptumRx | jnr\_new\_cust\_detrmntn |
| 4 | expr\_profile\_xref | In expression, the INTERFACE CODE is added as ‘interface\_code’ with Source Data | Read\_curr\_prof\_profile\_xref | jnr\_new\_cust\_detrmntn |
| 5 | jnr\_new\_cust\_detrmntn | Source data (Master) is joined with x\_ref on Master.BP\_ID =Detail.indiv\_ssk And Master.INTERFACE\_CODE\_SRC= Detail.interface\_code\_xref | 1.expr\_pass\_through\_intfc\_cd 2.expr\_profile\_xref | fltr\_new\_cust |
| 6 | fltr\_new\_cust | Verifying whether indiv\_ssk and interface\_code\_xref both are NULL | jnr\_new\_cust\_detrmntn | Write\_FF\_Tgt\_Cust\_OptumRx |
| 7 | Write\_FF\_Tgt\_Cust\_OptumRx | Generating new customer file under “/business/common/am/data/processed/optumrx\_in” , name of the file is “New\_Cust\_M\_OPTUMRX\_P.txt” | fltr\_new\_cust | NA |

m\_New\_Cust\_Extract\_Prime\_Vendor

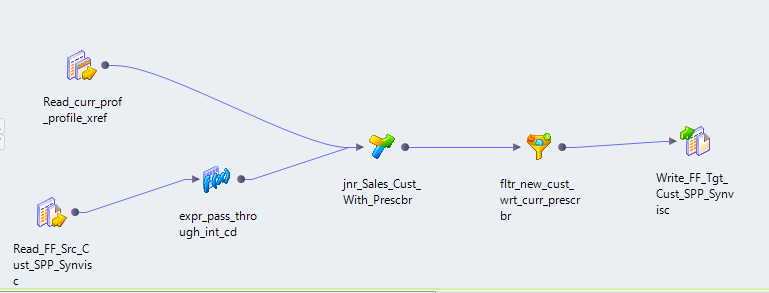
m\_New\_Cust\_Extract\_Prime\_Vendor

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **SL NO** | **Mapping component Name** | **Description** | **Previous Component** | **Next Component** |
| 1 | Read\_FF\_Src\_Cust\_Prime\_Vendor | Defined as Source in Informatca Mapping | NA | expr\_pass\_through\_int\_cd |
| 2 | Read\_curr\_prof\_profile\_xref | Defined as x-ref Source in Informatca Mapping | NA | jnr\_Sales\_Cust\_With\_Prescbr |
| 3 | expr\_pass\_through\_int\_cd | In expression, the INTERFACE CODE is added as ‘PRVN’ with Source Data | Read\_FF\_Src\_Cust\_Prime\_Vendor | jnr\_Sales\_Cust\_With\_Prescbr |
| 4 | jnr\_Sales\_Cust\_With\_Prescbr | Source data (Master) is joined with x\_ref on Master.MCK\_CUST=Detail.indiv\_ssk And Master.INTERFACE\_CODE= Detail.interface\_code\_xref | 1.expr\_pass\_through\_int\_cd | fltr\_new\_cust\_wrt\_curr\_prescrbr |
| 5 | fltr\_new\_cust\_wrt\_curr\_prescrbr | Verifying whether indiv\_ssk and interface\_code\_xref both are NULL | jnr\_Sales\_Cust\_With\_Prescbr | Write\_FF\_Tgt\_Cust\_Prime\_Vendor |
| 6 | Write\_FF\_Tgt\_Cust\_Prime\_Vendor | Generating new customer file under “/business/common/am/data/processed/primevendor\_in” , name of the file is “New\_Cust\_M\_PRIME\_VENDOR\_P.txt” | fltr\_new\_cust\_wrt\_curr\_prescrbr | NA |



m\_New\_Cust\_Extract\_SPP\_synvisc

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **SL NO** | **Mapping component Name** | **Description** | **Previous Component** | **Next Component** |
| 1 | Read\_FF\_Src\_Cust\_SPP\_Synvisc | Defined as Source in Informatca Mapping | NA | expr\_pass\_through\_int\_cd |
| 2 | Read\_curr\_prof\_profile\_xref | Defined as x-ref Source in Informatca Mapping | NA | jnr\_Sales\_Cust\_With\_Prescbr |
| 3 | expr\_pass\_through\_int\_cd | In expression, the INTERFACE CODE is added as ‘SYNVC’ with Source Data | Read\_FF\_Src\_Cust\_SPP\_Synvisc | jnr\_Sales\_Cust\_With\_Prescbr |
| 4 | jnr\_Sales\_Cust\_With\_Prescbr | Source data (Master) is joined with x\_ref on Master.INTERFACE\_CODE =Detail.indiv\_ssk And Master.INTERFACE\_CODE= Detail.interface\_code\_xref INTERFACE\_CODE = interface\_code\_xref | expr\_pass\_through\_int\_cd | fltr\_new\_cust\_wrt\_curr\_prescrbr |
| 5 | fltr\_new\_cust\_wrt\_curr\_prescrbr | Verifying whether indiv\_ssk and interface\_code\_xref both are NULL | jnr\_Sales\_Cust\_With\_Prescbr | Write\_FF\_Tgt\_Cust\_SPP\_Synvisc |
| 6 | Write\_FF\_Tgt\_Cust\_SPP\_Synvisc | Generating new customer file under “/business/common/am/data/processed/synvisc\_in” , name of the file is “New\_Cust\_M\_SPP\_SYNVISC\_P.txt” | fltr\_new\_cust\_wrt\_curr\_prescrbr | NA |



1. \* *Signatures above indicate approvers agree with the content of the document and ensure it will be applicable within their domain of expertise.* [↑](#footnote-ref-1)