



UPLOADER (Backend)

**LLD**

Document Version / Details: Ver. 1.0 / 10-03-2023

WORD DOCUMENT

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Version** | **Date** | **Description** | **ADO ID** | **Owner** |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**VERSION HISTORY**

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| SL.NO | CONTENTS | P.NO |
| 1 | Introduction | 4 |
| 2 | Business Requirements | 4 |
| 3 | Technical Components | 5 |
| 4 | Data Model Design | 8 |
| 5 | Glossary | 9 |
| 6 | Processing of uploader | 12 |
| 6.1 | Pipeline pipeline\_cp\_and\_process\_uploader\_data | 17 |
| 7 | Uploader Loading Process | 24 |
| 7.1 | Load Input File Column Definitions | 29 |
| 8 | Config File and Source File reference | 73 |
| 9 | DQ Validation | 107 |
| 10 | Technical Details | 124 |
| 10.1 | Back-end environment setup | 124 |
| 10.2 | Downloading Excel with 1M Records | 127 |
| 10.3 | Upload Notifications | 128 |
| 10.4 | Concurrency protection | 130 |
| 10.5 | Database reference Entity diagram | 134 |
| 10.6 | Metadata tables | 135 |
| 10.7 | Generic objects | 159 |
| 10.8 | Procedures and functions | 163 |
| 10.9 | Procedures for metadata maintenance | 201 |
| 10.10 | Views | 219 |
| 10.11 | Data Factory reference | 234 |
| 10.12 | Database users, roles, schemas | 281 |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

# INTRODUCTION

* Uploader Tool is a web-based application, specifically created for CNOS & GC for maintaining fact data/manual inputs.
* The tool enables maintenance/update of manual inputs via download and upload functionality. Download functionality enables the user to export input files as an excel and provide/update inputs by updating the downloaded file. After updating the file, the user will then upload the file using the Upload functionality and will undergo a series of data quality checks. Once the upload passed the validation, data will be submitted and will be further processed by the CNOS & GC Core calculation engine.
* There are 2 main types of input files in the tool, the load file, and the non-load file. Load file is the type of file that is generated by joining data from Rubik (Customer & Product), Shipments, and other corporate dimension data. Non-load files are custom files that are created for a specific purpose and are not connected to any upstream source. Both types of files are pre-configured based on the Business Requirements provided by the Solution Team, however, Ops Team or the admin user can update the configuration or the grid settings of each input file using the Input File Definition functionality.

# BUSINESS REQUIREMENTS

* To Uploader tool is a web application for maintaining KPIs for each region and input files generated from aggregated Geo data, Shipment data, Customer Mapping and Product Mapping data.
* Input SPOCs work on a downloaded excel file which they will fill with their inputs and upload these inputs to the application.
* After the file is uploaded, it will undergo basic data validation and DQ checks. Once the uploaded file passes all basic data validation and DQ checks, it will automatically be submitted.

**A diagram of a diagram

Description automatically generated with medium confidence**

Figure: Go roadmap of Uploader

**A computer screen shot of a diagram

Description automatically generated**

Figure: Business process diagram

# TECHNICAL COMPONENTS

* Azure ADF – To read data and orchestrate complete flow.
* Databricks – To write or apply business logics for CNOS & GC
* Azure Blob – To store data.
* SQL Server – To store and retrieve data.

**Technologies (Azure Components)**

Microsoft Azure resources assigned for Uploader Project are split into 8 Resource groups (2 Resource Groups <processing> and <application> per environment <DEV>, <QA>, <UAT>, <PROD>):

* **DEV**

processing (AZ-RG-CNOSGC-Uploader-Processing-DevOps-01)

application (AZ-RG-CNOSGC-Uploader-App-DevOps-01)

* **QA**

processing (AZ-RG-CNOSGC-Uploader-Processing-QA-01)

application (AZ-RG-CNOSGC-Uploader-App-QA-01)

* **UAT**

processing (AZ-RG-CNOSGC-Uploader-Processing-UAT-01)

application (AZ-RG-CNOSGC-Uploader-App-UAT-01)

* **PROD**

processing (AZ-RG-CNOSGC-Uploader-Processing-PROD-01)

application (AZ-RG-CNOSGC-Uploader-App-PROD-01)

**Azure Databricks Workspaces:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Environment** | **Name** | **Resource Group** | **Subscription** |
| DEV | dbrcnosgcuploadern01 | az-rg-cnosgc-uploader-processing-devops-01 | PG-NA-External-NonProd-04 |
| QA | dbrcnosgcuploaderqan01 | az-rg-cnosgc-uploader-processing-qa-01 | PG-NA-External-NonProd-04 |
| UAT | dbrcnosgcuploaderuatn01 | az-rg-cnosgc-uploader-processing-uat-01 | PG-NA-External-NonProd-04 |
| PROD | dbrcnosgcuploaderp01 | AZ-RG-CNOSGC-Uploader-Processing-PROD-01 | PG-NA-External-Prod-04 |

**Azure Data Factory:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Environment** | **Name** | **Resource Group** | **Subscription** |
| DEV | adf-cnosgc-uploader-n01 | az-rg-cnosgc-uploader-processing-devops-01 | PG-NA-External-NonProd-04 |
| QA | adf-cnosgc-uploader-qa-n01 | az-rg-cnosgc-uploader-processing-qa-01 | PG-NA-External-NonProd-04 |
| UAT | adf-cnosgc-uploader-uat-n01 | az-rg-cnosgc-uploader-processing-uat-01 | PG-NA-External-NonProd-04 |
| PROD | adf-cnosgc-uploader-prod-n01 | AZ-RG-CNOSGC-Uploader-Processing-PROD-01 | PG-NA-External-Prod-04 |

**BE Components per environment:**

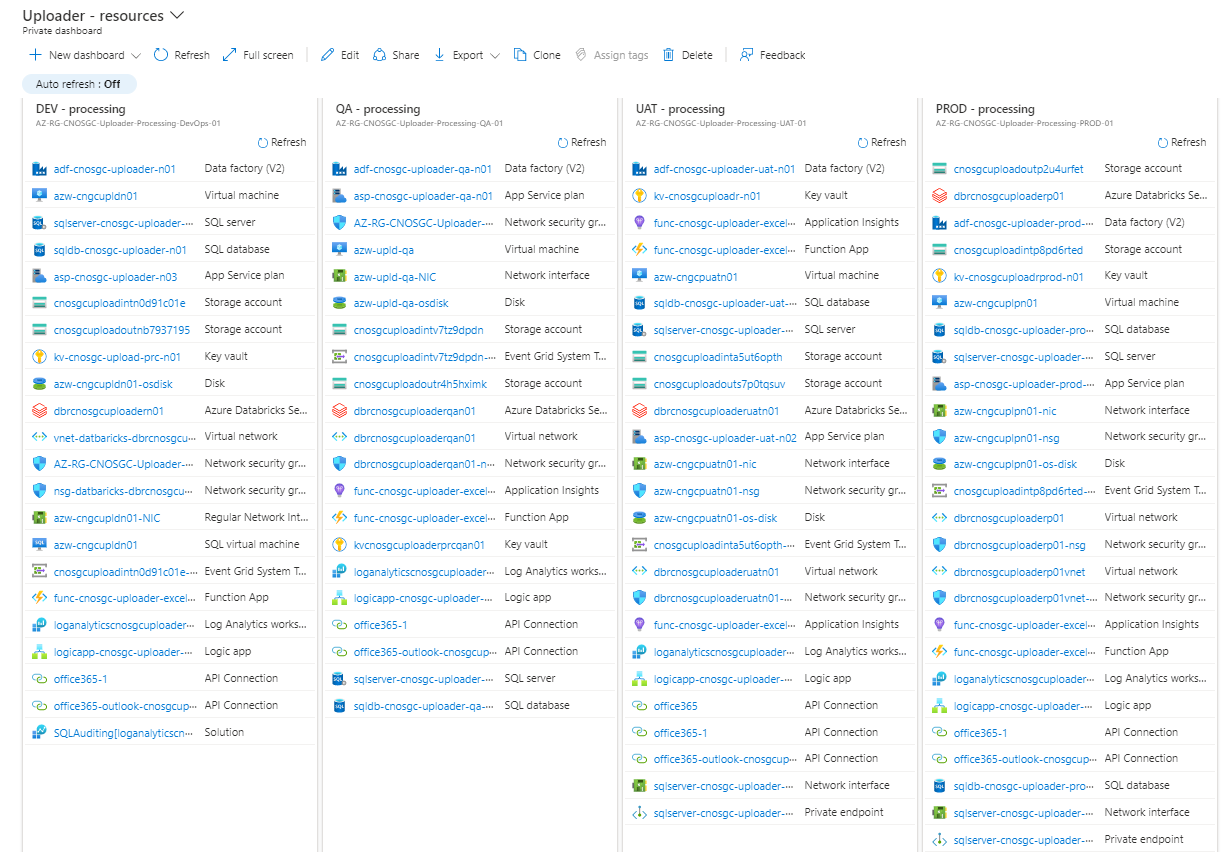


Figure: BE Components per environment

# DATA MODEL DESIGN

**Architecture:**

A screenshot of a computer game

Description automatically generated

Figure: Logical data flow

A diagram of a software process

Description automatically generated

Figure: architecture diagram

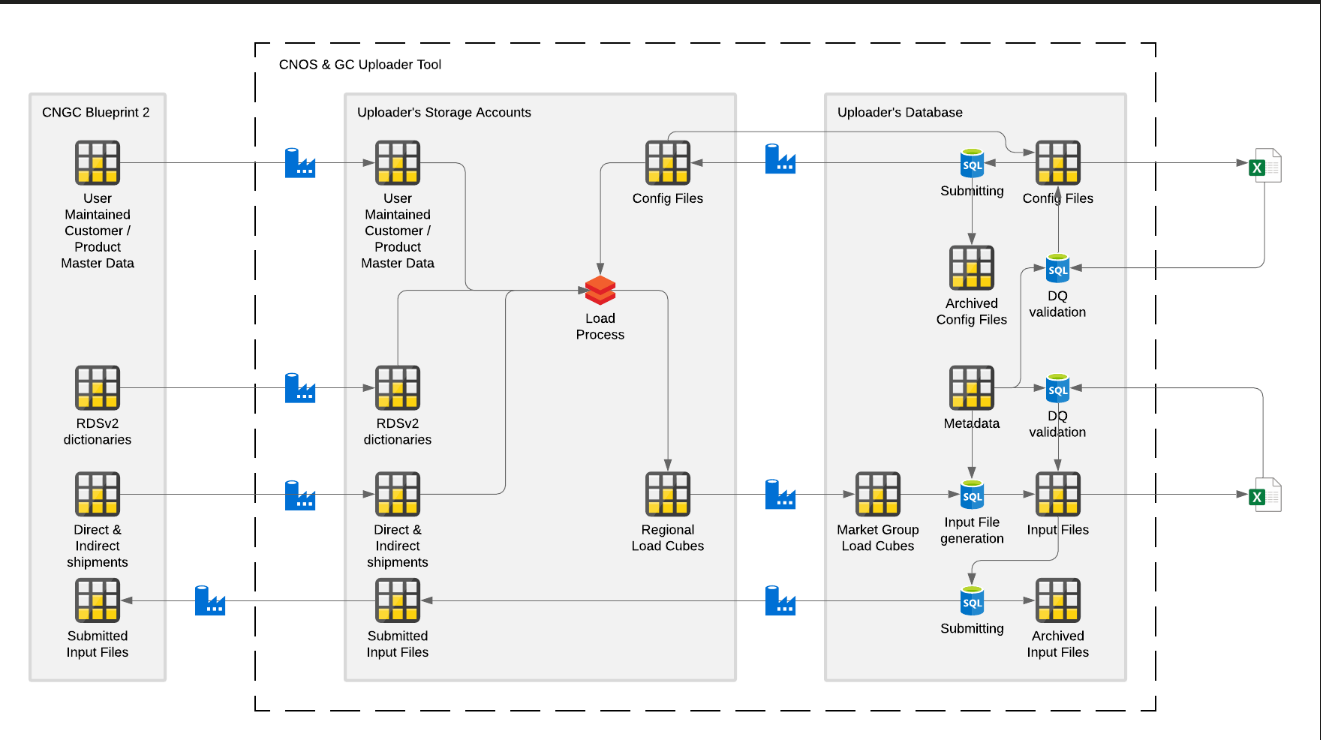


Figure: Data flow

# GLOSSARY

## Actuals Input File

Input File which contains fact data for current and previous Fiscal Year.

## Business Key

Set of Load Columns which forms part of the Load Input File definition and are marked as key columns in Load Cube reference. Used to match rows coming from Load Process with those already existing in Working Table.

## Configuration File (Config File)

Special type of Input File being used in Load Process to generate proper Input Files in certain way.

## Forecast Input File

Input File which contains future fact data for non-historical products.  
The monthly scope includes the current calendar month of the processing and spans till the end of the next Fiscal Year.

## Input File

Named set of data in scope of Market Group to be updated by End Users. Has versioned structure and attributes.

## Input File Version

Snapshot of Input File Definition structure (list of columns and its data types) and attributes

## Input File Definition

List of columns in Input File and its data types along with general Input File attributes like:

* is it Actuals or Forecast Input File
* should it have Invalid Rows mechanism enabled

## Invalid Row

Row which is excluded from submission by manual action of End User who marks it as invalid upon upload. Exact semantics described.

## Invalid Row Indicator (IRI)

Editable (via Upload) technical column indicating if row is [Invalid]

Column which is repopulated from one of Uploader's sources during Load Process.

## Load Input File

Input File which has at least on Load Column (opposite to [Non-Load Input File]

## Load Process

Process of transformation of source data and preparation of Input Files.

## Market

Subdivision of Market Group

## Market Group

Main geography/market dimension level in Uploader. All Input Files (except [Config Files]) are assigned to single Market Group.

## Non-Load Input File

Input File which has no Load Columns (opposite to [Load Input File]

## Obsolete Row

Row being present in Working Table but being out of monthly scope for this Actual/Forecast Input File. Obsolete Row indicator is set only by Load Process. Obsolete Row Indicator (ORI)

Non-editable technical column indicating if row is [Obsolete]

## Region

Higher level of market hierarchy. Currently AMA, Europe or Latin America.

## Row identifier

Technical column with unique identifier of each row in [Input File]

## Scope

Chosen element of Market/File hierarchy describing the action or processing in Uploader.  
Can represent one of levels below:

* global
* regional
* Market Group level
* Input File in Market Group
* version of Input File
* version of Input File limited to single Market

## Submission

Process of exposing Input File for downstream being final consequence of upload of the file by End User. Only rows not being Obsolete and Invalid are being submitted.

## Submit table

Table which contains snapshot of last submission of given Input File.

## System Column

Technical columns in tables/files representing Input Files. Described here.

## Working Table

Table underlying Input File data visualised in UI. Beside data to be submitted it contains also Invalid and Obsolete rows.

# PROCESSING OF UPLOADER

Data processing for uploader consists of several steps:

* Moving input data to working directories
* Classification
* Reading
* Archiving

## Input Data

Input data are provided in parquet format. They are available in following blob storage path: 'cngcbp2/input\_files'. To establish connection with this blob storage we use a SAS token which is stored in following secret 'uploader-source-data-SAS-url'.

## Working directories

The ADF pipeline for uploader moves input data to following directories:

Incoming folder: 'cngc-storage/incoming/uploader\_source\_data'

Processing folder: 'cngc-storage/processing/uploader\_source\_data'

Data is firstly moved to the incoming directory and later to the processing one. During data classification (see the next step) data are moved to subdirectories of the processing folder (for example 'cngc-storage/processing/uploader\_source\_data/TT\_DIR' ).

## Classification

Input data are provided without meaningful names and has to be categorized - assigned to one of following categories:

* TT\_DIR
* TT\_IDR
* TT\_DIR\_FCST
* CMF\_DIR
* CMF\_IDR
* National\_SD\_DIR
* National\_SD\_IDR
* National\_SD\_DIR\_FCST
* Regional\_SD\_DIR
* Regional\_SD\_DIR\_FCST
* TDC\_SU
* FX\_RATES\_FRGN
* FX\_RATES\_TDC
* GEO\_MAPPING
* Profit\_center
* Profit\_center\_LA
* SD\_cust\_dim
* FUND\_SUBSECTOR\_MAP
* ACCT\_ID\_TYPE\_MAP
* CUST\_MANUL\_INPUT
* FM\_RPTNG\_INPUT
* FUND\_SUBSECTOR\_MAP
* SAP\_SHIPT\_MAPNG
* PC\_SBSTR\_NSRD
* SD\_TPR
* VAL\_FUND\_OPT
* LE\_MAPNG
* TDC\_LIMITS
* TANDER
* OPTIMA\_WD\_SCHEDULE
* IPLAN\_AMA\_EDS\_TDC
* FX\_RATES\_ARG
* LA\_NIT
* LA\_LIVE\_ARG
* LA\_LIVE\_BR
* LA\_LIVE\_CHI
* LA\_LIVE\_COL
* LA\_LIVE\_MEX
* LA\_LIVE\_PERU
* LA\_FIXED\_ARG
* LA\_FIXED\_BR
* LA\_FIXED\_CHI
* LA\_FIXED\_COL

Assignment is done using column names in processed parquet file as a criterion. Sometimes a file cannot be assigned to any of these categories, then it will be labelled as 'other'. Parquet files are processed in a loop, one file per one loop iteration. Once the file is processed it is copied to a dedicated subfolder of the 'processing' folder (like 'cngc-storage/processing/uploader\_source\_data/National\_SD\_IDR' - each subfolder is named after category of file).

Following ADB notebook is responsible for data categorization:

'Shared/CNGC\_COMMON/10\_INGESTION/UPLOADER\_PREPROCESSING/100\_UPLOADER\_CATEGORIZATION'.

It is called by the data processing pipeline for uploader (mentioned above).

## Classification algorithm:

* The algorithm works as follows: every parquet file consists of several columns. For every category there is a list of category-specific columns and a counter value with total number of columns per category. Total number of characteristic columns may differ between categories. There is also a list which consists of category names sorted by total number of columns in descending order.
* The initial category for a parquet file is set by default to 'other' (none of categories mentioned above).
* Every column name in parquet file is compared against lists with column names for each category. If the name of processed column is in a list dedicated to specific category, then counter value for this category is decreased by one.
* After every column name in parquet file has been processed counter values for each category are checked. If there is a category for which all of column names coincide with columns names of the parquet file (its counter is equal to zero) then the category for the parquet file is changed to name of this category. If there are multiple categories with their counters equal to zero, then the category with greater number of columns is chosen (indexes for categories in question are compared in a list with category names sorted by total number of columns).

## Data reading

* Once data have been classified, they read and used to create table which are part of the 'stage\_uploader' schema:

| **Table name** | **Category** |
| --- | --- |
| stage\_uploader.tdc\_su | TDC SU |
| stage\_uploader.tt\_dir | TT\_DIR |
| stage\_uploader.tt\_idr | TT\_IDR |
| stage\_uploader.tt\_dir\_fcst | TT\_DIR\_FCST |
| stage\_uploader.cmf\_dir | CMF\_DIR |
| stage\_uploader.cmf\_idr | CMF\_IDR |
| stage\_uploader.national\_sd\_dir | National\_SD\_DIR |
| stage\_uploader.national\_sd\_idr | National\_SD\_IDR |
| stage\_uploader.national\_sd\_dir\_fcst | Regional\_SD\_DIR\_FCST |
| stage\_uploader.regional\_sd\_dir | Regional\_SD\_DIR |
| stage\_uploader.regional\_sd\_dir\_fcst | Regional\_SD\_DIR\_FCST |
| stage\_uploader.fx\_rates\_frgn | FX\_RATES\_FRGN |
| stage\_uploader.fx\_rates\_tdc | FX\_RATES\_TDC |
| stage\_uploader.geo\_mapping | GEO\_MAPPING |
| stage\_uploader.profit\_center | Profit\_center |
| stage\_uploader.profit\_center\_la | Profit\_center\_LA |
| stage\_uploader.sd\_cust | SD\_cust\_dim |
| stage\_uploader.acct\_id\_type\_map | ACCT\_ID\_TYPE\_MAP |
| stage\_uploader.cust\_manul\_input\_la | CMI\_LA |
| stage\_uploader.cust\_manul\_input\_ladmar | CMI\_LADMAR |
| stage\_uploader.fm\_rptng\_input | FM\_RPTNG\_INPUT |
| stage\_uploader.fund\_subsector\_map | FUND\_SUBSECTOR\_MAP |
| stage\_uploader.sap\_shipt\_mapng | SAP\_SHIPT\_MAPNG |
| stage\_uploader.pc\_sbstr\_nsrd | PC\_SBSTR\_NSRD |
| stage\_uploader.sd\_tpr | SD\_TPR |
| stage\_uploader.val\_fund\_opt | VAL\_FUND\_OPT |
| stage\_uploader.le\_mapng | LE\_MAPNG |
| stage\_uploader.tdc\_limits | TDC\_LIMITS |
| stage\_uploader.tander | TANDER |
| stage\_uploader.fx\_rates\_argentina | FX\_RATES\_ARG |
| stage\_uploader.optima\_wd\_schedule | OPTIMA\_WD\_SCHEDULE |
| stage\_uploader.sd\_live\_rates\_la\_argentina | LA\_LIVE\_ARG |
| stage\_uploader.sd\_fixed\_budget\_la\_argentina | LA\_FIXED \_ARG |
| stage\_uploader.la\_nit | LA\_NIT |

Data are read by following ADB notebook:

'Shared/CNGC\_COMMON/10\_INGESTION/UPLOADER\_PREPROCESSING/110\_UPLOADER\_INGESTION'

The notebook is called by data processing pipeline for uploader (mentioned at the beginning of the article).

## Archiving

Once data processing is complete all input files are archived. They are moved to subdirectories of following folder:

Archiving folder: 'cngc-storage/uploader\_archive'

In this folder new subfolders are generated to store recent data. They are named using following convention: "UPLOADER\_ARCHIVE\_" + (current date in YYYYMMDD format) + "\_" + (current time)

Data are archived by following ADB notebook:

'Shared/CNGC\_COMMON/10\_INGESTION/UPLOADER\_PREPROCESSING/120\_UPLOADER\_ARCHIVE'

## s

### Basic information about pipeline:

|  |  |
| --- | --- |
| Name | pipeline\_cp\_and\_process\_uploader\_data |
| Folder | 55\_uploader |
| Number of steps | 7 |
| Description | This pipeline copies input data from a blob storage for uploader and launches ADB notebooks to split it into several categories, read it and archive it. |

### Pipeline Schema

A screenshot of a computer

Description automatically generated

### Pipeline parameters

Pipeline does not have any parameters.

### Variables

There are no variables defined for this pipeline.

### Settings

The pipeline has a trigger:   
  
A screenshot of a computer

Description automatically generated

### Steps

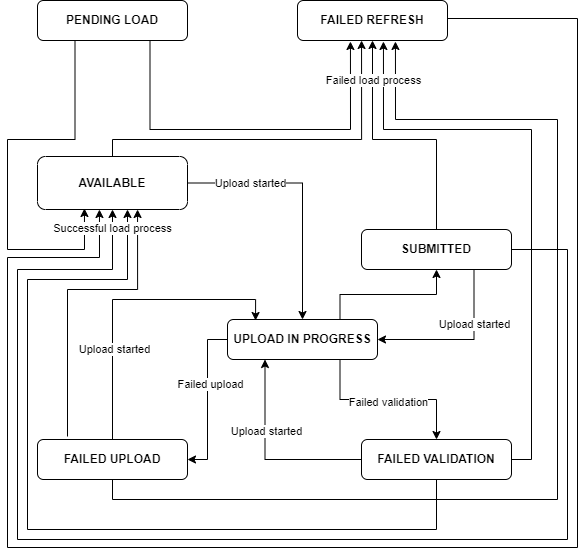
* The first four steps concentrate on copying and transferring data. Input data are copied to the ‘incoming’ folder and then to the ‘processing’ one, but firstly aforementioned folders are being emptied.
* In the last three steps ADB notebooks are called. The first ADB notebook splits data into several categories. The second one reads data and creates tables from it, the last notebook archives data.

1. **delete\_incoming\_data**  
   In this step every parquet file in the ‘incoming’ is being deleted:  
    *cngc-storage/incoming/uploader\_source\_data*
2. **copy\_to\_incoming**   
   In this step input data are copied to the ‘incoming’ folder:  
   source dataset: *cngcbp2/input\_files*  
   destination: *cngc-storage/incoming/uploader\_source\_data*
3. **delete\_processing\_data**   
   In this step every parquet file in the ‘processing’ folder is being deleted:  
    *cngc-storage/processing/uploader\_source\_data*
4. **copy\_to\_processing**  
   In this step data are copied from the ‘incoming’ folder to the ‘processing’ one:  
   source dataset: *cngc-storage/incoming/uploader\_source\_data*  
   destination: *cngc-storage/processing/uploader\_source\_data*
5. **Uploader\_data\_categorization**   
   In this step following ADB notebook is called: *“/Shared/CNGC\_COMMON/10\_INGESTION/UPLOADER\_PREPROCESSING/100\_UPLOADER\_CATEGORIZATION*”  
   This notebook divides data into several categories like ‘FX\_RATES\_TDC’ or ‘National\_SD\_DIR’. For those categories dedicated subfolders of the ‘processing’ exist (like cngc-storage/processing/uploader\_source\_data/FX\_Rates\_TDC) where data are moved.
6. **Uploader\_ingestion**  
   In this step following ADB notebook is called: *“/Shared/CNGC\_COMMON/10\_INGESTION/UPLOADER\_PREPROCESSING/110\_UPLOADER\_INGESTION”*In this notebook tables are created directly from parquet files (before new table will be created the old one has to be dropped down).
7. **Uploader\_archivization**   
   In this step following ADB notebook is called: */Shared/CNGC\_COMMON/10\_INGESTION/UPLOADER\_PREPROCESSING/120\_UPLOADER\_ARCHIVE”*In this notebook data are being archived – moved to dedicated subfolders of directory *cngc-storage/uploader\_archive/*

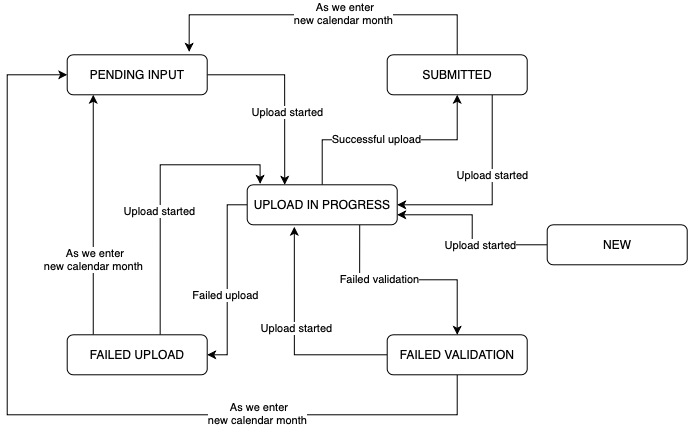
* **Column data type**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Data type name** | **Description** | **Display format** | **Sample values when displayed** | **Excel file format** | **Sample values in Excel file** | **DB data type** | **Comment** |
| BOOLEAN | Boolean values | Checkbox | Checkbox | Y/N | Y/N | CHAR(1) |  |
| DATE | Date without time | MMM-DD-YYYY | Apr-01-2021 | MM/DD/YYYY | 4/1/2021 | DATETIME2(8) |  |
| INTEGER | Integer values | 'xxxx', '-x' | '99371', '-8' | '99371', '-8' | '99371', '-8' | INT |  |
| MONTH | Month including year | MMM-YY | Apr-21 | MM/DD/YYYY | 4/1/2021 | DATETIME2(8) | Example from current MMR input |
| NUMBER | Number with given precision and scale | xxxx.xx | 99371.01 | xxxx.xx | 99371.01 | NUMERIC |  |
| PERCENT | Numeric shown in form of percentage | 'x.xx%', '-x.xx%' | '0.50%', '-4.00%' | 'x.xxx', '-x.xx' | '0.005', '-0.04' | NUMERIC | "1" for 100% |
| TEXT | Text string values | 'Text' | 'Text' | 'Text' | 'Text' | VARCHAR |  |

* **Allowed file statuses:**
* AVAILABLE: the most recent action for the file was refresh made by Load Process (only Input Files having Load Columns
* FAILED UPLOAD: the most recent action was upload which had some critical validation failed
* FAILED VALIDATION: the most recent action was upload which succeeded, but because of some critical validation failure the file was not submitted
* UPLOAD IN PROGRESS: upload started and not yet finished (includes whole validation and process of exposing file to the downstream)
* NEW: newly created non-load-based Non-Load Input File
* PENDING INPUT: Non-Load Input File not yet Uploaded in this calendar month
* PENDING LOAD: newly created source-based Load Input File (Input Files having Load Columns)
* SUBMITTED: file successfully uploaded
* FAILED REFRESH: Load Input File failed file refresh made by Load Process
* **Diagram for Load Input Files**



* **Diagram for Non-Load Input Files**



* **CNOS & GC Uploader vs Master Minder GC**

| **Example** | **Names used in Master Minder for GC** | **Unified naming for Uploader** | **Remarks** |
| --- | --- | --- | --- |
| AMA | Region | **Region** | Sometimes equals to Market Group (LA) |
| Europe - BNL | Geo group | **Market group** |  |
| BE LUX | "Country" | **Market** |  |
| SD Customer Direct | Template | **Input File Definition** | Set of characteristics defining the data. In MMR being set up both in Back-End (list of allowed columns) and in Grid Settings customisation (data types, labels, etc). In Uploader Input File name, list of columns, data format are customisable at Market Group level. |
| SD Customer Direct for DENMARK | Template | **Input File (for given Market or Market Group)** | Actual set of data created according to its definition |
| Geo Mapping, Profit Center and Market Mapping | Technical template | **Config File** | Set of data which controls transformation of data during Load Process. Also having precise **Config File Definition** |
|  | Template submission and publication (separate steps) | **Submission of the Input File** | The consent of exposing data to downstream. In Uploader we prefer ***Submit*** over *Publish* and *Commit* |

# UPLOADER LOADING PROCESS

**The Loading Process for Uploader consists of 2 distinct steps:**

1. **Ingestion**
2. **Trigger Loading Pipelines**

Ingestion includes Trigger the following pipelines.

1. in-CopyShipments
2. in-CopyRDSDictionaries
3. in-CopyCustomerMapping
4. in-CopyProductMapping

Parameters all four pipelines are similar:

Graphical user interface, text, application

Description automatically generated

in\_parnt\_comp\_exctn\_id = 1

in\_user\_name = ‘Auto Load’

All four pipelines can be triggered simultaneously, as their process is mutually exclusive of one another.

Loading process for uploader consists for two different scenarios

## Global Load

(triggered via ld-LoadMainGlobal)

Parameter required

Table

Description automatically generated

in\_parnt\_comp\_exctn\_id = 1

in\_user\_name = ‘Auto Load’

For All Global Load last 3 parameters must always be set as ‘Y’

## Selective Load

(Triggered via ld-LoadMainSelective)

This process is carried out when there is a requirement to process only certain markets.

Parameters

Table

Description automatically generated

in\_parnt\_comp\_exctn\_id = 1

in\_user\_name = ‘Auto Load’

For the Last Parameter (in\_json\_txt) we have multiple cases.

Case 1: Only Single Markets needs to be processed, Provide parameter as:

[{"in\_mkt\_grp\_name":"AMA","in\_cust\_prcss\_ind":"Y","in\_prod\_prcss\_ind":"Y","in\_shpmt\_prcss\_ind":"Y"}]

Case 2: More than one Market needs to be processed, provide parameters as:

[

{"in\_mkt\_grp\_name":"AMA","in\_cust\_prcss\_ind":"Y","in\_prod\_prcss\_ind":"Y","in\_shpmt\_prcss\_ind":"Y"},

{"in\_mkt\_grp\_name":"LA","in\_cust\_prcss\_ind":"Y","in\_prod\_prcss\_ind":"Y","in\_shpmt\_prcss\_ind":"Y"},

{"in\_mkt\_grp\_name":"Europe - EE","in\_cust\_prcss\_ind":"Y","in\_prod\_prcss\_ind":"Y","in\_shpmt\_prcss\_ind":"Y"}

]

During the selective load process, even though only single input has changed, i.e., consider only (Customer Mapping has changed), care should be taken that remaining 2 parameters including in\_prod\_prcss\_ind, in\_shpmt\_prcss\_ind must be also set to ‘Y’.

This means that all 3 inputs must be processed together in order to avoid any join condition failure while refreshing the worktables in Azure SQL Server.

**Exception:**

For Market Europe – ALL, this procedure should not be followed as Europe – ALL will be automatically refreshed when there is any change in Europe related market.

* LA SMO owns Product mapping maintenance in MDM, and GC engine extracts the Product mapping from LA Data hub and prepares final version of Product Mapping.
* Product mapping from MDM is maintained and ingested at Brand level, following the 5005-product hierarchy.
* Product mapping is segregated into two separate sets of code, one for Japan, and the other for all non-Japan SMOs.

## LOAD PROCESS

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | |  | Step | ADB Component | Process | Output | | Loading of Customer Mapping | 1 | 210\_ingst\_cngc\_cust\_mapng | To load ingested Customer Mapping csv files as a Databricks Table | t210\_ingst\_cngc\_cust\_mapng\_la t210\_ingst\_cngc\_cust\_mapng\_ama t210\_ingst\_cngc\_cust\_mapng\_eu | | Loading of Product Mapping | 1 | 220\_ingst\_cngc\_prod\_mapng | To load ingested Product Mapping csv files as a Databricks Table | t220\_ingst\_cngc\_prod\_mapng\_jpn\_excdd t220\_ingst\_cngc\_prod\_mapng\_jpn\_only | | Loading of RDS | 1 | 230\_ingst\_cngc\_rds\_dict | To load ingested RDS parquet as a Databricks Table | t230\_ingst\_cngc\_rds\_dict\_prod\_hier\_dim t230\_ingst\_cngc\_rds\_dict\_cust\_hier\_dim t230\_ingst\_cngc\_rds\_dict\_geo\_hier\_dim t230\_ingst\_cngc\_rds\_dict\_proft\_ctr\_hier\_dim t230\_ingst\_cngc\_rds\_dict\_prod\_life\_cycle\_dim | | Loading of Shipments | 1 | 240\_ingst\_cngc\_shpmt\_dirct | To load ingested Direct Shipments parquet as a Databricks Table | t240\_ingst\_cngc\_shpmt\_dirct | | Loading of Shipments | 1 | 250\_ingst\_cngc\_shpmt\_indir\_ama | To load ingested Indirect Shipments AMA parquet as a Databricks Table | t250\_ingst\_cngc\_shpmt\_indir\_ama | | Loading of Shipments | 1 | 260\_ingst\_cngc\_shpmt\_indir\_eu | To load ingested Indirect Shipments EU parquet as a Databricks Table | t260\_ingst\_cngc\_shpmt\_indir\_eu | | Loading of Geo Mapping | 1 | 270\_ingst\_upldr\_geo\_mapng | To load geo mapping submit parquet as a Databricks Table | t270\_ingst\_upldr\_geo\_mapng | | Loading of Profit Center & Market Mapping | 1 | 280\_ingst\_upldr\_proft\_ctr\_mkt\_mapng | To load profit center & market mapping submit parquet as a Databricks Table | t280\_ingst\_upldr\_proft\_ctr\_mkt\_mapng | | Creating RDS Dictionary | 2 | 310\_dict\_rds | Create RDS Dictionaries that contains needed data for the load process | t310\_dict\_rds\_p5005 t310\_dict\_rds\_p5801 t310\_dict\_rds\_c898 t310\_dict\_rds\_g705 t310\_dict\_rds\_g707 t310\_dict\_rds\_plc | | Creating Time Dimension | 2 | 320\_dict\_time | Generate time dimension table containing months under PFY, CFY and FFY based on the date of execution | t320\_dict\_time\_mkt\_grp\_load\_cube | | Processing of Shipments | 3 | 410\_shpmt\_union | To combine ingested direct and indirect shipments data into one dataset | t410\_shpmt\_union | | Processing of Shipments | 3 | 420\_shpmt\_time | Filter shipments through months that exist in time dimension table and are actuals | t420\_shpmt\_time | | Processing of Shipments | 3 | 430\_shpmt\_filtr | Filter fact data using sql conditions defined in Profit Center and Market Mapping config file | t430\_shpmt\_filtr | | Processing of Shipments | 3 | 440\_shpmt\_geo\_mapng | Filter shipments based on profit center id and geo id in Geo Mapping | t440\_shpmt\_geo\_mapng | | Processing of Shipments | 3 | 450\_shpmt\_cache | filter shipments by market group based on the process indicator having 'Y' defined in the market group process configuration | t450\_shpmt\_cache | | Processing of Products | 3 | 510\_prod\_geo\_mapng | Cross join products with geo mapping | t510\_prod\_geo\_mapng | | Processing of Products | 3 | 520\_prod\_union | To combine ingested Excluded Japan and Only Japan Product Mapping data into one dataset. | t520\_prod\_union | | Processing of Products | 3 | 530\_prod\_prod\_mapng | filter products base on defined product exclusion in product mapping | t530\_prod\_prod\_mapng | | Processing of Products | 3 | 540\_prod\_prod\_life\_cycle | filter products based on prod skid and iso country code in product life cycle | t540\_prod\_prod\_life\_cycle | | Processing of Products | 3 | 550\_prod\_cache | filter products by market group based on the process indicator having 'Y' defined in the market group process configuration | t550\_prod\_cache | | Processing of Customers | 4 | 610\_cust\_union | To combine ingested LA, AMA and EU Customer Mapping data into one dataset. | t610\_cust\_union | | Processing of Customers | 4 | 620\_cust\_geo\_mapng | Filter customers based on geo id in Geo Mapping | t620\_cust\_geo\_mapng | | Processing of Customers | 4 | 630\_cust\_cache | filter customers by market group based on the process indicator having 'Y' defined in the market group process configuration | t630\_cust\_cache | | Cross Filter of Shipments and Products | 5 | 710\_shpmt\_filtr\_by\_prod | filter shipments by products fpc id | t710\_shpmt\_filtr\_by\_prod | | Cross Filter of Shipments and Products | 6 | 720\_shpmt\_filtr\_by\_prod\_cust | filter shipments by customers | t720\_shpmt\_filtr\_by\_prod\_cust | | Cross Filter of Shipments and Products | 7 | 730\_prod\_filtr\_by\_shpmt\_prod\_cust | filter products by shipments brand | t730\_prod\_filtr\_by\_shpmt\_prod\_cust | | Cross Filter of Shipments and Products | 8 | 740\_prod\_filtr\_by\_shpmt\_prod | filter products by shipments fpc id | t740\_prod\_filtr\_by\_shpmt\_prod | | Generation of Cubes | 9 | 810\_cust\_prod\_brand | cross join products with customers | t810\_cust\_prod\_brand | | Generation of Cubes | 10 | 820\_cust\_prod\_brand\_shpmt | join customers with shipments | t820\_cust\_prod\_brand\_shpmt | | Generation of Cubes | 11 | 830\_prod\_fpc\_shpmt | join products with shipments | t830\_prod\_fpc\_shpmt | |

## Load Input File Column Definitions

### TDC\_SU

Market Groups:

* AMA
* JAPAN
* Europe - BNL
* Europe - CCAR
* Europe - CE
* Europe - DACH
* Europe - EE
* Europe - FRANCE
* Europe - IBERIA
* Europe - ITALY
* Europe - NORDICS
* Europe - SEE
* Europe - TURKEY
* Europe - UKI
* LA

Input File Definitions:

| **COLUMN TYPE** | **COLUMN NAME** | **COLUMN LABEL** | **DATA TYPE** | **LENGTH** | **PRECISION** |
| --- | --- | --- | --- | --- | --- |
| SYSTEM | sys\_row\_id | Row ID | INTEGER | NULL | NULL |
| SYSTEM | sys\_invld\_ind | Invalid Row Indicator | BOOLEAN | NULL | NULL |
| SYSTEM | sys\_obslt\_ind | Obsolete Row Indicator | BOOLEAN | NULL | NULL |
| SYSTEM | sys\_init\_actn\_id | Initial Action ID | INTEGER | NULL | NULL |
| SYSTEM | sys\_last\_uplod\_actn\_id | Last Upload Action ID | INTEGER | NULL | NULL |
| SYSTEM | sys\_mkt\_id | Market ID | INTEGER | NULL | NULL |
| SYSTEM | sys\_last\_mdfd\_datetm | Last Modification Date | DATE | NULL | NULL |
| SYSTEM | sys\_last\_mdfd\_user\_name | User who uploaded as the last | TEXT | 50 | NULL |
| LOAD | MKT\_GEO\_ID | Market Geo ID | TEXT | 50 | NULL |
| LOAD | MKT\_NAME | Market | TEXT | 50 | NULL |
| LOAD | SBSTR\_NAME | Subsector Name | TEXT | 50 | NULL |
| LOAD | FY\_CODE | FY | TEXT | 50 | NULL |
| LOAD | FPC\_ID | FPC | INTEGER | NULL | NULL |
| LOAD | PROFT\_CTR\_ID | Profit Center ID | INTEGER | NULL | NULL |
| LOAD | PROFT\_CTR\_NAME | Profit Center Name | TEXT | 50 | NULL |
| LOAD | CUSTM\_SMO\_NAME | Custom SMO | TEXT | 50 | NULL |
| LOAD | SBSTR\_ID | Subsector ID | INTEGER | NULL | NULL |
| LOAD | CATEG\_ID | Category ID | INTEGER | NULL | NULL |
| LOAD | CATEG\_NAME | Category Name | TEXT | 50 | NULL |
| LOAD | BRAND\_ID | Brand ID | INTEGER | NULL | NULL |
| LOAD | BRAND\_NAME | Brand Name | TEXT | 50 | NULL |
| LOAD | MTH\_NUM | Month | MONTH | NULL | NULL |
| LOAD | CUSTM\_REGN\_NAME | Custom Region | TEXT | 50 | NULL |
| LOAD | CUSTM\_CLSTR\_NAME | Custom Cluster | TEXT | 50 | NULL |
| LOAD | MKT\_GRP\_NAME | Market Group | TEXT | 50 | NULL |
| LOAD | actl\_ind | Is month in range of Actual Input File | BOOLEAN | 1 | NULL |
| CUSTOM | BRAND\_FORM\_ID | Brand Form ID | TEXT | 50 | NULL |
| CUSTOM | BRAND\_FORM\_NAME | Brand Form Name | TEXT | 500 | NULL |
| CUSTOM | FPC\_NAME | FPC Name | TEXT | 500 | NULL |
| CUSTOM | TDC\_USD\_SU | TDC USD/SU | NUMBER | 22 | 8 |
| CUSTOM | TDC\_CUSD\_SU | TDC\_CUSD\_SU | NUMBER | 38 | 8 |
| CUSTOM | TDC\_RUSD\_SU | TDC\_RUSD\_SU | NUMBER | 38 | 8 |
| CUSTOM | TDC\_LC\_SU | TDC\_LC\_SU | NUMBER | 38 | 8 |
| CUSTOM | LOCAL\_CRNCY\_CODE | Local Currency | TEXT | 3 | NULL |
| CUSTOM | VOL\_SU | VOL\_SU | NUMBER | 22 | 8 |
| CUSTOM | ATTR\_1\_VAL | ATTR\_1\_VAL | TEXT | 500 | NULL |
| CUSTOM | ATTR\_2\_VAL | ATTR\_2\_VAL | TEXT | 500 | NULL |
| CUSTOM | ATTR\_3\_VAL | ATTR\_3\_VAL | TEXT | 500 | NULL |
| CUSTOM | ATTR\_4\_VAL | ATTR\_4\_VAL | TEXT | 500 | NULL |
| CUSTOM | ATTR\_5\_VAL | ATTR\_5\_VAL | TEXT | 500 | NULL |

### SD Cust

Market Groups:

* AMA
* JAPAN

Input File Definitions:

| **COLUMN TYPE** | **COLUMN NAME** | **COLUMN LABEL** | **DATA TYPE** | **LENGTH** | **PRECISION** |
| --- | --- | --- | --- | --- | --- |
| SYSTEM | sys\_row\_id | Row surrogate key | INTEGER | NULL | NULL |
| SYSTEM | sys\_invld\_ind | Is row marked as Invalid | BOOLEAN | NULL | NULL |
| SYSTEM | sys\_obslt\_ind | Is row marked as Obsolete | BOOLEAN | NULL | NULL |
| SYSTEM | sys\_init\_actn\_id | Load/Upload Action which added this row | INTEGER | NULL | NULL |
| SYSTEM | sys\_last\_uplod\_actn\_id | Identifier of most recent Upload Action of this ro | INTEGER | NULL | NULL |
| SYSTEM | sys\_mkt\_id | Identifier of Market, added if mkt\_col\_id in Input | INTEGER | NULL | NULL |
| SYSTEM | sys\_last\_mdfd\_datetm | Date of last modification including Load Process t | DATE | NULL | NULL |
| SYSTEM | sys\_last\_mdfd\_user\_name | "User who uploaded as the last or ""AUTO\_LOAD"" if th" | TEXT | 50 | NULL |
| LOAD | MKT\_GRP\_NAME | Market Group | TEXT | 50 | NULL |
| LOAD | CUSTM\_REGN\_NAME | Custom Region | TEXT | 50 | NULL |
| LOAD | CUSTM\_SMO\_NAME | Custom SMO | TEXT | 50 | NULL |
| LOAD | CUSTM\_CLSTR\_NAME | Custom Cluster | TEXT | 50 | NULL |
| LOAD | MKT\_GEO\_ID | Market Geo ID | TEXT | 50 | NULL |
| LOAD | MKT\_NAME | Market | TEXT | 50 | NULL |
| LOAD | RPTNG\_CUST\_NAME | Reporting Customer Name | TEXT | 50 | NULL |
| LOAD | PROFT\_CTR\_ID | Profit Center ID | INTEGER | NULL | NULL |
| LOAD | PROFT\_CTR\_NAME | Profit Center Name | TEXT | 50 | NULL |
| LOAD | SBSTR\_ID | Subsector ID | INTEGER | NULL | NULL |
| LOAD | SBSTR\_NAME | Subsector Name | TEXT | 50 | NULL |
| LOAD | FY\_CODE | FY | TEXT | 50 | NULL |
| LOAD | MTH\_NUM | Month | MONTH | NULL | NULL |
| LOAD | CATEG\_ID | Category ID | INTEGER | NULL | NULL |
| LOAD | CATEG\_NAME | Category Name | TEXT | 50 | NULL |
| CUSTOM | ATTR\_1\_VAL | SD (%NIV) | PERCENT | 38 | 8 |
| CUSTOM | ATTR\_2\_VAL | ATTR\_2\_VAL | TEXT | 500 | NULL |
| CUSTOM | ATTR\_3\_VAL | ATTR\_3\_VAL | TEXT | 500 | NULL |
| CUSTOM | ATTR\_4\_VAL | ATTR\_4\_VAL | TEXT | 500 | NULL |
| CUSTOM | SD\_GIV\_PCT | SD\_GIV\_PCT | NUMBER | 22 | 8 |
| CUSTOM | ATTR\_5\_VAL | ATTR\_5\_VAL | TEXT | 500 | NULL |

### Customer Manual Input

Market Groups:

* LA

Input File Definitions:

| **COLUMN TYPE** | **COLUMN NAME** | **COLUMN LABEL** | **DATA TYPE** | **LENGTH** | **PRECISION** |
| --- | --- | --- | --- | --- | --- |
| SYSTEM | sys\_row\_id | Row ID | INTEGER | NULL | NULL |
| SYSTEM | sys\_invld\_ind | Invalid Row Indicator | BOOLEAN | NULL | NULL |
| SYSTEM | sys\_obslt\_ind | Obsolete Row Indicator | BOOLEAN | NULL | NULL |
| SYSTEM | sys\_init\_actn\_id | Initial Action ID | INTEGER | NULL | NULL |
| SYSTEM | sys\_last\_uplod\_actn\_id | Last Upload Action ID | INTEGER | NULL | NULL |
| SYSTEM | sys\_mkt\_id | Market ID | INTEGER | NULL | NULL |
| SYSTEM | sys\_last\_mdfd\_datetm | Last Modification Date | DATE | NULL | NULL |
| SYSTEM | sys\_last\_mdfd\_user\_name | User who uploaded as the last | TEXT | 50 | NULL |
| LOAD | CUSTM\_SMO\_NAME | Custom SMO | TEXT | 50 | NULL |
| LOAD | MKT\_GEO\_ID | Market Geo ID | TEXT | 50 | NULL |
| LOAD | MKT\_NAME | Market | TEXT | 50 | NULL |
| LOAD | RPTNG\_CUST\_ID | Reporting Customer ID | INTEGER | NULL | NULL |
| LOAD | RPTNG\_CUST\_NAME | Reporting Customer Name | TEXT | 50 | NULL |
| LOAD | SBSTR\_ID | Subsector ID | INTEGER | NULL | NULL |
| LOAD | SBSTR\_NAME | Subsector Name | TEXT | 50 | NULL |
| LOAD | CATEG\_ID | Category ID | INTEGER | NULL | NULL |
| LOAD | CATEG\_NAME | Category Name | TEXT | 50 | NULL |
| LOAD | FY\_CODE | FY | TEXT | 50 | NULL |
| LOAD | MTH\_NUM | Month | MONTH | NULL | NULL |
| LOAD | MKT\_GRP\_NAME | Market Group | TEXT | 50 | NULL |
| CUSTOM | NSRD\_MANUL\_INPUT\_LC\_VAL | NSRd Manual Input - Base PC | NUMBER | 22 | 8 |
| CUSTOM | SD\_MANUL\_INPUT\_LC\_VAL | SD Manual Input - Base PC | NUMBER | 22 | 8 |
| CUSTOM | ATTR\_1\_VAL | NSRd Manual Input - Export PC | NUMBER | 22 | 8 |
| CUSTOM | ATTR\_2\_VAL | SD Manual Input - Export PC | NUMBER | 22 | 8 |
| CUSTOM | ATTR\_3\_VAL | ATTR\_3\_VAL | TEXT | 500 | NULL |
| CUSTOM | ATTR\_4\_VAL | ATTR\_4\_VAL | TEXT | 500 | NULL |
| CUSTOM | ATTR\_5\_VAL | ATTR\_5\_VAL | TEXT | 500 | NULL |

### TT Direct Non-Standard

Market Groups:

* Europe - BNL
* Europe - CCAR
* Europe - FRANCE
* Europe - TURKEY

Input File Definitions:

| **COLUMN TYPE** | **COLUMN NAME** | **COLUMN LABEL** | **DATA TYPE** | **LENGTH** | **PRECISION** |
| --- | --- | --- | --- | --- | --- |
| SYSTEM | sys\_row\_id | Row surrogate key | INTEGER | NULL | NULL |
| SYSTEM | sys\_invld\_ind | Is row marked as Invalid | BOOLEAN | NULL | NULL |
| SYSTEM | sys\_obslt\_ind | Is row marked as Obsolete | BOOLEAN | NULL | NULL |
| SYSTEM | sys\_init\_actn\_id | Load/Upload Action which added this row | INTEGER | NULL | NULL |
| SYSTEM | sys\_last\_uplod\_actn\_id | Identifier of most recent Upload Action of this ro | INTEGER | NULL | NULL |
| SYSTEM | sys\_mkt\_id | Identifier of Market, added if mkt\_col\_id in Input | INTEGER | NULL | NULL |
| SYSTEM | sys\_last\_mdfd\_datetm | Date of last modification including Load Process t | DATE | NULL | NULL |
| SYSTEM | sys\_last\_mdfd\_user\_name | "User who uploaded as the last or ""AUTO\_LOAD"" if th" | TEXT | 50 | NULL |
| LOAD | MKT\_GEO\_ID | Market Geo ID | TEXT | 50 | NULL |
| LOAD | MKT\_NAME | Market | TEXT | 50 | NULL |
| LOAD | RPTNG\_CUST\_NAME | Reporting Customer Name | TEXT | 50 | NULL |
| LOAD | SBSTR\_ID | Subsector ID | INTEGER | NULL | NULL |
| LOAD | SBSTR\_NAME | Subsector Name | TEXT | 50 | NULL |
| LOAD | CATEG\_ID | Category ID | INTEGER | NULL | NULL |
| LOAD | CATEG\_NAME | Category Name | TEXT | 50 | NULL |
| LOAD | FY\_CODE | FY | TEXT | 50 | NULL |
| LOAD | MTH\_NUM | Month | MONTH | NULL | NULL |
| LOAD | BRAND\_ID | Brand ID | INTEGER | NULL | NULL |
| LOAD | BRAND\_NAME | Brand Name | TEXT | 50 | NULL |
| CUSTOM | ATTR\_4\_VAL | ATTR\_4\_VAL | NUMBER | 38 | 8 |
| CUSTOM | ATTR\_3\_VAL | ATTR\_3\_VAL | NUMBER | 38 | 8 |
| CUSTOM | ATTR\_2\_VAL | ATTR\_2\_VAL | NUMBER | 38 | 8 |
| CUSTOM | ATTR\_1\_VAL | ATTR\_1\_VAL | NUMBER | 38 | 8 |
| CUSTOM | EXCDD\_CASH\_DISC\_PPR\_VAL | EXCDD\_CASH\_DISC\_PPR\_VAL | PERCENT | 38 | 8 |
| CUSTOM | CASH\_DISC\_2\_VAL | TT Cash Discount | NUMBER | 38 | 8 |
| CUSTOM | LGSTC\_DISC\_VAL | TT Logistic Discount | NUMBER | 38 | 8 |
| CUSTOM | EFFCY\_DISC\_VAL | TT Efficiency Discount | NUMBER | 38 | 8 |
| CUSTOM | DSTBR\_DISC\_VAL | TT Distributor Discount | NUMBER | 38 | 8 |
| CUSTOM | OTHER\_VAL | TT Other | NUMBER | 38 | 8 |
| CUSTOM | FLEX\_FOLD\_IN\_VAL | TT Flex Fold In | NUMBER | 38 | 8 |
| CUSTOM | CASH\_DISC\_1\_VAL | CASH\_DISC\_1\_VAL | PERCENT | 38 | 8 |
| CUSTOM | NO\_BRKDW\_VAL | TT No Breakdown | PERCENT | 38 | 8 |
| CUSTOM | CUST\_L4\_ID | CUST\_L4\_ID | TEXT | 50 | NULL |

### SD Customer Direct Non-Standard

Market Groups:

* Europe - BNL

Input File Definitions:

| **COLUMN TYPE** | **COLUMN NAME** | **COLUMN LABEL** | **DATA TYPE** | **LENGTH** | **PRECISION** |
| --- | --- | --- | --- | --- | --- |
| SYSTEM | sys\_row\_id | Row surrogate key | INTEGER | NULL | NULL |
| SYSTEM | sys\_invld\_ind | Is row marked as Invalid | BOOLEAN | NULL | NULL |
| SYSTEM | sys\_obslt\_ind | Is row marked as Obsolete | BOOLEAN | NULL | NULL |
| SYSTEM | sys\_init\_actn\_id | Load/Upload Action which added this row | INTEGER | NULL | NULL |
| SYSTEM | sys\_last\_uplod\_actn\_id | Identifier of most recent Upload Action of this ro | INTEGER | NULL | NULL |
| SYSTEM | sys\_mkt\_id | Identifier of Market, added if mkt\_col\_id in Input | INTEGER | NULL | NULL |
| SYSTEM | sys\_last\_mdfd\_datetm | Date of last modification including Load Process t | DATE | NULL | NULL |
| SYSTEM | sys\_last\_mdfd\_user\_name | "User who uploaded as the last or ""AUTO\_LOAD"" if th" | TEXT | 50 | NULL |
| LOAD | MKT\_GEO\_ID | Market Geo ID | TEXT | 50 | NULL |
| LOAD | MKT\_NAME | Market | TEXT | 50 | NULL |
| LOAD | RPTNG\_CUST\_NAME | Reporting Customer Name | TEXT | 50 | NULL |
| LOAD | SBSTR\_ID | Subsector ID | INTEGER | NULL | NULL |
| LOAD | SBSTR\_NAME | Subsector Name | TEXT | 50 | NULL |
| LOAD | CATEG\_ID | Category ID | INTEGER | NULL | NULL |
| LOAD | CATEG\_NAME | Category Name | TEXT | 50 | NULL |
| LOAD | BRAND\_ID | Brand ID | INTEGER | NULL | NULL |
| LOAD | BRAND\_NAME | Brand Name | TEXT | 50 | NULL |
| LOAD | FY\_CODE | FY | TEXT | 50 | NULL |
| LOAD | MTH\_NUM | Month | MONTH | NULL | NULL |
| CUSTOM | NO\_BRKDW\_VAL | SD No Breakdown | PERCENT | 38 | 8 |
| CUSTOM | ATTR\_4\_VAL | ATTR\_4\_VAL | TEXT | 50 | NULL |
| CUSTOM | ATTR\_3\_VAL | ATTR\_3\_VAL | NUMBER | 38 | 8 |
| CUSTOM | ATTR\_2\_VAL | ATTR\_2\_VAL | NUMBER | 38 | 8 |
| CUSTOM | ATTR\_1\_VAL | ATTR\_1\_VAL | NUMBER | 38 | 8 |
| CUSTOM | CNSMR\_VAL | CNSMR\_VAL | NUMBER | 38 | 8 |
| CUSTOM | CUST\_VAL | CUST\_VAL | NUMBER | 38 | 8 |
| CUSTOM | OTHER\_VAL | OTHER\_VAL | NUMBER | 38 | 8 |
| CUSTOM | LYLTY\_CARD\_VAL | LYLTY\_CARD\_VAL | NUMBER | 38 | 8 |
| CUSTOM | SCNR\_VAL | SCNR\_VAL | NUMBER | 38 | 8 |
| CUSTOM | DIST\_DRIVE\_VAL | DIST\_DRIVE\_VAL | NUMBER | 38 | 8 |
| CUSTOM | MSP\_RSRV\_CUST\_INVC\_VAL | MSP\_RSRV\_CUST\_INVC\_VAL | NUMBER | 38 | 8 |
| CUSTOM | MSP\_RSRV\_CREDT\_NOTE\_VAL | MSP\_RSRV\_CREDT\_NOTE\_VAL | NUMBER | 38 | 8 |
| CUSTOM | QBDF\_CUST\_INVC\_VAL | QBDF\_CUST\_INVC\_VAL | NUMBER | 38 | 8 |
| CUSTOM | QBDF\_CREDT\_NOTE\_VAL | QBDF\_CREDT\_NOTE\_VAL | NUMBER | 38 | 8 |
| CUSTOM | QBDF\_ON\_INVC\_VAL | QBDF\_ON\_INVC\_VAL | NUMBER | 38 | 8 |
| CUSTOM | CORP\_PLAN\_VAL | CORP\_PLAN\_VAL | NUMBER | 38 | 8 |
| CUSTOM | INNVN\_FUND\_VAL | INNVN\_FUND\_VAL | NUMBER | 38 | 8 |
| CUSTOM | ANN\_AGRMT\_VAL | ANN\_AGRMT\_VAL | NUMBER | 38 | 8 |
| CUSTOM | CUST\_L4\_ID | CUST\_L4\_ID | TEXT | 50 | NULL |
| CUSTOM | ATTR\_6\_VAL | ATTR\_6\_VAL | NUMBER | 38 | 8 |
| CUSTOM | ATTR\_12\_VAL | ATTR\_12\_VAL | NUMBER | 38 | 8 |
| CUSTOM | ATTR\_7\_VAL | ATTR\_7\_VAL | NUMBER | 38 | 8 |
| CUSTOM | ATTR\_11\_VAL | ATTR\_11\_VAL | NUMBER | 38 | 8 |
| CUSTOM | ATTR\_9\_VAL | ATTR\_9\_VAL | NUMBER | 38 | 8 |
| CUSTOM | ATTR\_10\_VAL | ATTR\_10\_VAL | NUMBER | 38 | 8 |
| CUSTOM | ATTR\_5\_VAL | ATTR\_5\_VAL | NUMBER | 38 | 8 |
| CUSTOM | ATTR\_8\_VAL | ATTR\_8\_VAL | NUMBER | 38 | 8 |
| CUSTOM | ATTR\_13\_VAL | ATTR\_13\_VAL | NUMBER | 38 | 8 |

### SD National Direct Non-Standard

Market Groups:

* Europe - BNL

Input File Definitions:

| **COLUMN TYPE** | **COLUMN NAME** | **COLUMN LABEL** | **DATA TYPE** | **LENGTH** | **PRECISION** |
| --- | --- | --- | --- | --- | --- |
| SYSTEM | sys\_row\_id | Row surrogate key | INTEGER | NULL | NULL |
| SYSTEM | sys\_invld\_ind | Is row marked as Invalid | BOOLEAN | NULL | NULL |
| SYSTEM | sys\_obslt\_ind | Is row marked as Obsolete | BOOLEAN | NULL | NULL |
| SYSTEM | sys\_init\_actn\_id | Load/Upload Action which added this row | INTEGER | NULL | NULL |
| SYSTEM | sys\_last\_uplod\_actn\_id | Identifier of most recent Upload Action of this ro | INTEGER | NULL | NULL |
| SYSTEM | sys\_mkt\_id | Identifier of Market, added if mkt\_col\_id in Input | INTEGER | NULL | NULL |
| SYSTEM | sys\_last\_mdfd\_datetm | Date of last modification including Load Process t | DATE | NULL | NULL |
| SYSTEM | sys\_last\_mdfd\_user\_name | "User who uploaded as the last or ""AUTO\_LOAD"" if th" | TEXT | 50 | NULL |
| LOAD | MKT\_GEO\_ID | Market Geo ID | TEXT | 50 | NULL |
| LOAD | MKT\_NAME | Market | TEXT | 50 | NULL |
| LOAD | SBSTR\_ID | Subsector ID | INTEGER | NULL | NULL |
| LOAD | SBSTR\_NAME | Subsector Name | TEXT | 50 | NULL |
| LOAD | CATEG\_ID | Category ID | INTEGER | NULL | NULL |
| LOAD | CATEG\_NAME | Category Name | TEXT | 50 | NULL |
| LOAD | FY\_CODE | FY | TEXT | 50 | NULL |
| LOAD | MTH\_NUM | Month | MONTH | NULL | NULL |
| LOAD | BRAND\_ID | Brand ID | INTEGER | NULL | NULL |
| LOAD | BRAND\_NAME | Brand Name | TEXT | 50 | NULL |
| LOAD | RPTNG\_CUST\_NAME | Reporting Customer Name | TEXT | 50 | NULL |
| LOAD | MKT\_GRP\_NAME | Market Group | TEXT | 50 | NULL |
| CUSTOM | CUST\_L4\_ID | CUST\_L4\_ID | TEXT | 50 | NULL |
| CUSTOM | NATL\_SD\_VAL | SD National | PERCENT | 38 | 8 |
| CUSTOM | ATTR\_4\_VAL | ATTR\_4\_VAL | NUMBER | 38 | 8 |
| CUSTOM | ATTR\_3\_VAL | ATTR\_3\_VAL | NUMBER | 38 | 8 |
| CUSTOM | ATTR\_2\_VAL | ATTR\_2\_VAL | NUMBER | 38 | 8 |
| CUSTOM | ATTR\_1\_VAL | ATTR\_1\_VAL | NUMBER | 38 | 8 |

### SD Regional Direct

Market Groups:

* Europe - BNL
* Europe - CCAR
* Europe - CE
* Europe - DACH
* Europe - EE
* Europe - FRANCE
* Europe - IBERIA
* Europe - ITALY
* Europe - NORDICS
* Europe - SEE
* Europe - TURKEY
* Europe – UKI­

Input File Definitions:

| **COLUMN TYPE** | **COLUMN NAME** | **COLUMN LABEL** | **DATA TYPE** | **LENGTH** | **PRECISION** |
| --- | --- | --- | --- | --- | --- |
| SYSTEM | sys\_row\_id | Row surrogate key | INTEGER | NULL | NULL |
| SYSTEM | sys\_invld\_ind | Is row marked as Invalid | BOOLEAN | NULL | NULL |
| SYSTEM | sys\_obslt\_ind | Is row marked as Obsolete | BOOLEAN | NULL | NULL |
| SYSTEM | sys\_init\_actn\_id | Load/Upload Action which added this row | INTEGER | NULL | NULL |
| SYSTEM | sys\_last\_uplod\_actn\_id | Identifier of most recent Upload Action of this ro | INTEGER | NULL | NULL |
| SYSTEM | sys\_mkt\_id | Identifier of Market, added if mkt\_col\_id in Input | INTEGER | NULL | NULL |
| SYSTEM | sys\_last\_mdfd\_datetm | Date of last modification including Load Process t | DATE | NULL | NULL |
| SYSTEM | sys\_last\_mdfd\_user\_name | "User who uploaded as the last or ""AUTO\_LOAD"" if th" | TEXT | 50 | NULL |
| LOAD | MKT\_GEO\_ID | Market Geo ID | TEXT | 50 | NULL |
| LOAD | MKT\_NAME | Market | TEXT | 50 | NULL |
| LOAD | RPTNG\_CUST\_NAME | Reporting Customer Name | TEXT | 50 | NULL |
| LOAD | SBSTR\_ID | Subsector ID | INTEGER | NULL | NULL |
| LOAD | SBSTR\_NAME | Subsector Name | TEXT | 50 | NULL |
| LOAD | CATEG\_ID | Category ID | INTEGER | NULL | NULL |
| LOAD | CATEG\_NAME | Category Name | TEXT | 50 | NULL |
| LOAD | FY\_CODE | FY | TEXT | 50 | NULL |
| LOAD | MTH\_NUM | Month | MONTH | NULL | NULL |
| LOAD | BRAND\_ID | Brand ID | INTEGER | NULL | NULL |
| LOAD | BRAND\_NAME | Brand Name | TEXT | 50 | NULL |
| CUSTOM | REGNL\_SD\_VAL | SD Regional | PERCENT | 38 | 8 |
| CUSTOM | ATTR\_4\_VAL | ATTR\_4\_VAL | NUMBER | 38 | 8 |
| CUSTOM | ATTR\_3\_VAL | ATTR\_3\_VAL | NUMBER | 38 | 8 |
| CUSTOM | ATTR\_2\_VAL | ATTR\_2\_VAL | NUMBER | 38 | 8 |
| CUSTOM | ATTR\_1\_VAL | ATTR\_1\_VAL | NUMBER | 38 | 8 |
| CUSTOM | CUST\_L4\_ID | CUST\_L4\_ID | TEXT | 50 | NULL |

### SD Customer Indirect Non-Standard

Market Groups:

* Europe - CCAR

Input File Definitions:

| **COLUMN TYPE** | **COLUMN NAME** | **COLUMN LABEL** | **DATA TYPE** | **LENGTH** | **PRECISION** |
| --- | --- | --- | --- | --- | --- |
| SYSTEM | sys\_row\_id | Row surrogate key | INTEGER | NULL | NULL |
| SYSTEM | sys\_invld\_ind | Is row marked as Invalid | BOOLEAN | NULL | NULL |
| SYSTEM | sys\_obslt\_ind | Is row marked as Obsolete | BOOLEAN | NULL | NULL |
| SYSTEM | sys\_init\_actn\_id | Load/Upload Action which added this row | INTEGER | NULL | NULL |
| SYSTEM | sys\_last\_uplod\_actn\_id | Identifier of most recent Upload Action of this ro | INTEGER | NULL | NULL |
| SYSTEM | sys\_mkt\_id | Identifier of Market, added if mkt\_col\_id in Input | INTEGER | NULL | NULL |
| SYSTEM | sys\_last\_mdfd\_datetm | Date of last modification including Load Process t | DATE | NULL | NULL |
| SYSTEM | sys\_last\_mdfd\_user\_name | "User who uploaded as the last or ""AUTO\_LOAD"" if th" | TEXT | 50 | NULL |
| LOAD | MKT\_GEO\_ID | Market Geo ID | TEXT | 50 | NULL |
| LOAD | MKT\_NAME | Market | TEXT | 50 | NULL |
| LOAD | RPTNG\_CUST\_NAME | Reporting Customer Name | TEXT | 50 | NULL |
| LOAD | SBSTR\_ID | Subsector ID | INTEGER | NULL | NULL |
| LOAD | SBSTR\_NAME | Subsector Name | TEXT | 50 | NULL |
| LOAD | CATEG\_ID | Category ID | INTEGER | NULL | NULL |
| LOAD | CATEG\_NAME | Category Name | TEXT | 50 | NULL |
| LOAD | BRAND\_ID | Brand ID | INTEGER | NULL | NULL |
| LOAD | BRAND\_NAME | Brand Name | TEXT | 50 | NULL |
| LOAD | FY\_CODE | FY | TEXT | 50 | NULL |
| LOAD | MTH\_NUM | Month | MONTH | NULL | NULL |
| CUSTOM | NO\_BRKDW\_VAL | SD No Breakdown | PERCENT | 38 | 8 |
| CUSTOM | ATTR\_4\_VAL | ATTR\_4\_VAL | NUMBER | 38 | 8 |
| CUSTOM | ATTR\_3\_VAL | ATTR\_3\_VAL | NUMBER | 38 | 8 |
| CUSTOM | ATTR\_2\_VAL | ATTR\_2\_VAL | NUMBER | 38 | 8 |
| CUSTOM | ATTR\_1\_VAL | ATTR\_1\_VAL | NUMBER | 38 | 8 |
| CUSTOM | CNSMR\_VAL | CNSMR\_VAL | NUMBER | 38 | 8 |
| CUSTOM | CUST\_VAL | CUST\_VAL | NUMBER | 38 | 8 |
| CUSTOM | OTHER\_VAL | OTHER\_VAL | NUMBER | 38 | 8 |
| CUSTOM | LYLTY\_CARD\_VAL | LYLTY\_CARD\_VAL | NUMBER | 38 | 8 |
| CUSTOM | SCNR\_VAL | SCNR\_VAL | NUMBER | 38 | 8 |
| CUSTOM | DIST\_DRIVE\_VAL | DIST\_DRIVE\_VAL | NUMBER | 38 | 8 |
| CUSTOM | MSP\_RSRV\_CUST\_INVC\_VAL | MSP\_RSRV\_CUST\_INVC\_VAL | NUMBER | 38 | 8 |
| CUSTOM | MSP\_RSRV\_CREDT\_NOTE\_VAL | MSP\_RSRV\_CREDT\_NOTE\_VAL | NUMBER | 38 | 8 |
| CUSTOM | QBDF\_CUST\_INVC\_VAL | QBDF\_CUST\_INVC\_VAL | NUMBER | 38 | 8 |
| CUSTOM | QBDF\_CREDT\_NOTE\_VAL | QBDF\_CREDT\_NOTE\_VAL | NUMBER | 38 | 8 |
| CUSTOM | QBDF\_ON\_INVC\_VAL | QBDF\_ON\_INVC\_VAL | NUMBER | 38 | 8 |
| CUSTOM | CORP\_PLAN\_VAL | CORP\_PLAN\_VAL | NUMBER | 38 | 8 |
| CUSTOM | INNVN\_FUND\_VAL | INNVN\_FUND\_VAL | NUMBER | 38 | 8 |
| CUSTOM | ANN\_AGRMT\_VAL | ANN\_AGRMT\_VAL | NUMBER | 38 | 8 |
| CUSTOM | CUST\_L4\_ID | CUST\_L4\_ID | TEXT | 50 | NULL |
| CUSTOM | ATTR\_6\_VAL | ATTR\_6\_VAL | NUMBER | 38 | 8 |
| CUSTOM | ATTR\_12\_VAL | ATTR\_12\_VAL | NUMBER | 38 | 8 |
| CUSTOM | ATTR\_7\_VAL | ATTR\_7\_VAL | NUMBER | 38 | 8 |
| CUSTOM | ATTR\_11\_VAL | ATTR\_11\_VAL | NUMBER | 38 | 8 |
| CUSTOM | ATTR\_9\_VAL | ATTR\_9\_VAL | NUMBER | 38 | 8 |
| CUSTOM | ATTR\_10\_VAL | ATTR\_10\_VAL | NUMBER | 38 | 8 |
| CUSTOM | ATTR\_5\_VAL | ATTR\_5\_VAL | NUMBER | 38 | 8 |
| CUSTOM | ATTR\_8\_VAL | ATTR\_8\_VAL | NUMBER | 38 | 8 |
| CUSTOM | ATTR\_13\_VAL | ATTR\_13\_VAL | NUMBER | 38 | 8 |

### SD Customer Direct

Market Groups:

* Europe - CCAR
* Europe - CE
* Europe - DACH
* Europe - EE
* Europe - FRANCE
* Europe - IBERIA
* Europe - ITALY
* Europe - NORDICS
* Europe - SEE
* Europe - TURKEY
* Europe - UKI

Input File Definitions:

| **COLUMN TYPE** | **COLUMN NAME** | **COLUMN LABEL** | **DATA TYPE** | **LENGTH** | **PRECISION** |
| --- | --- | --- | --- | --- | --- |
| SYSTEM | sys\_row\_id | Row surrogate key | INTEGER | NULL | NULL |
| SYSTEM | sys\_invld\_ind | Is row marked as Invalid | BOOLEAN | NULL | NULL |
| SYSTEM | sys\_obslt\_ind | Is row marked as Obsolete | BOOLEAN | NULL | NULL |
| SYSTEM | sys\_init\_actn\_id | Load/Upload Action which added this row | INTEGER | NULL | NULL |
| SYSTEM | sys\_last\_uplod\_actn\_id | Identifier of most recent Upload Action of this ro | INTEGER | NULL | NULL |
| SYSTEM | sys\_mkt\_id | Identifier of Market, added if mkt\_col\_id in Input | INTEGER | NULL | NULL |
| SYSTEM | sys\_last\_mdfd\_datetm | Date of last modification including Load Process t | DATE | NULL | NULL |
| SYSTEM | sys\_last\_mdfd\_user\_name | "User who uploaded as the last or ""AUTO\_LOAD"" if th" | TEXT | 50 | NULL |
| LOAD | MKT\_GEO\_ID | Market Geo ID | TEXT | 50 | NULL |
| LOAD | MKT\_NAME | Market | TEXT | 50 | NULL |
| LOAD | RPTNG\_CUST\_NAME | Reporting Customer Name | TEXT | 50 | NULL |
| LOAD | SBSTR\_ID | Subsector ID | INTEGER | NULL | NULL |
| LOAD | SBSTR\_NAME | Subsector Name | TEXT | 50 | NULL |
| LOAD | CATEG\_ID | Category ID | INTEGER | NULL | NULL |
| LOAD | CATEG\_NAME | Category Name | TEXT | 50 | NULL |
| LOAD | BRAND\_ID | Brand ID | INTEGER | NULL | NULL |
| LOAD | BRAND\_NAME | Brand Name | TEXT | 50 | NULL |
| LOAD | FY\_CODE | FY | TEXT | 50 | NULL |
| LOAD | MTH\_NUM | Month | MONTH | NULL | NULL |
| CUSTOM | NO\_BRKDW\_VAL | No Breakdown | NUMBER | 38 | 8 |
| CUSTOM | ATTR\_4\_VAL | ATTR\_4\_VAL | NUMBER | 38 | 8 |
| CUSTOM | ATTR\_3\_VAL | ATTR\_3\_VAL | NUMBER | 38 | 8 |
| CUSTOM | ATTR\_2\_VAL | SD NOI | NUMBER | 38 | 8 |
| CUSTOM | ATTR\_1\_VAL | SD OI | NUMBER | 38 | 8 |
| CUSTOM | CNSMR\_VAL | CNSMR\_VAL | NUMBER | 38 | 8 |
| CUSTOM | CUST\_VAL | CUST\_VAL | NUMBER | 38 | 8 |
| CUSTOM | OTHER\_VAL | OTHER\_VAL | NUMBER | 38 | 8 |
| CUSTOM | LYLTY\_CARD\_VAL | LYLTY\_CARD\_VAL | NUMBER | 38 | 8 |
| CUSTOM | SCNR\_VAL | SCNR\_VAL | NUMBER | 38 | 8 |
| CUSTOM | DIST\_DRIVE\_VAL | DIST\_DRIVE\_VAL | NUMBER | 38 | 8 |
| CUSTOM | MSP\_RSRV\_CUST\_INVC\_VAL | MSP\_RSRV\_CUST\_INVC\_VAL | NUMBER | 38 | 8 |
| CUSTOM | MSP\_RSRV\_CREDT\_NOTE\_VAL | MSP\_RSRV\_CREDT\_NOTE\_VAL | NUMBER | 38 | 8 |
| CUSTOM | QBDF\_CUST\_INVC\_VAL | QBDF\_CUST\_INVC\_VAL | NUMBER | 38 | 8 |
| CUSTOM | QBDF\_CREDT\_NOTE\_VAL | QBDF\_CREDT\_NOTE\_VAL | NUMBER | 38 | 8 |
| CUSTOM | QBDF\_ON\_INVC\_VAL | QBDF\_ON\_INVC\_VAL | NUMBER | 38 | 8 |
| CUSTOM | CORP\_PLAN\_VAL | CORP\_PLAN\_VAL | NUMBER | 38 | 8 |
| CUSTOM | INNVN\_FUND\_VAL | INNVN\_FUND\_VAL | NUMBER | 38 | 8 |
| CUSTOM | ANN\_AGRMT\_VAL | ANN\_AGRMT\_VAL | NUMBER | 38 | 8 |
| CUSTOM | CUST\_L4\_ID | CUST\_L4\_ID | TEXT | 50 | NULL |
| CUSTOM | ATTR\_6\_VAL | ATTR\_6\_VAL | NUMBER | 38 | 8 |
| CUSTOM | ATTR\_12\_VAL | ATTR\_12\_VAL | NUMBER | 38 | 8 |
| CUSTOM | ATTR\_7\_VAL | ATTR\_7\_VAL | NUMBER | 38 | 8 |
| CUSTOM | ATTR\_11\_VAL | ATTR\_11\_VAL | NUMBER | 38 | 8 |
| CUSTOM | ATTR\_9\_VAL | ATTR\_9\_VAL | NUMBER | 38 | 8 |
| CUSTOM | ATTR\_10\_VAL | ATTR\_10\_VAL | NUMBER | 38 | 8 |
| CUSTOM | ATTR\_5\_VAL | ATTR\_5\_VAL | NUMBER | 38 | 8 |
| CUSTOM | ATTR\_8\_VAL | ATTR\_8\_VAL | NUMBER | 38 | 8 |
| CUSTOM | ATTR\_13\_VAL | ATTR\_13\_VAL | NUMBER | 38 | 8 |

### TT Indirect

Market Groups:

* Europe - CCAR
* Europe - EE
* Europe - SEE
* Europe - TURKEY

Input File Definitions:

| **COLUMN TYPE** | **COLUMN NAME** | **COLUMN LABEL** | **DATA TYPE** | **LENGTH** | **PRECISION** |
| --- | --- | --- | --- | --- | --- |
| SYSTEM | sys\_row\_id | Row surrogate key | INTEGER | NULL | NULL |
| SYSTEM | sys\_invld\_ind | Is row marked as Invalid | BOOLEAN | NULL | NULL |
| SYSTEM | sys\_obslt\_ind | Is row marked as Obsolete | BOOLEAN | NULL | NULL |
| SYSTEM | sys\_init\_actn\_id | Load/Upload Action which added this row | INTEGER | NULL | NULL |
| SYSTEM | sys\_last\_uplod\_actn\_id | Identifier of most recent Upload Action of this ro | INTEGER | NULL | NULL |
| SYSTEM | sys\_mkt\_id | Identifier of Market, added if mkt\_col\_id in Input | INTEGER | NULL | NULL |
| SYSTEM | sys\_last\_mdfd\_datetm | Date of last modification including Load Process t | DATE | NULL | NULL |
| SYSTEM | sys\_last\_mdfd\_user\_name | "User who uploaded as the last or ""AUTO\_LOAD"" if th" | TEXT | 50 | NULL |
| LOAD | MKT\_GEO\_ID | Market Geo ID | TEXT | 50 | NULL |
| LOAD | MKT\_NAME | Market | TEXT | 50 | NULL |
| LOAD | RPTNG\_CUST\_NAME | Reporting Customer Name | TEXT | 50 | NULL |
| LOAD | SBSTR\_ID | Subsector ID | INTEGER | NULL | NULL |
| LOAD | SBSTR\_NAME | Subsector Name | TEXT | 50 | NULL |
| LOAD | CATEG\_ID | Category ID | INTEGER | NULL | NULL |
| LOAD | CATEG\_NAME | Category Name | TEXT | 50 | NULL |
| LOAD | FY\_CODE | FY | TEXT | 50 | NULL |
| LOAD | MTH\_NUM | Month | MONTH | NULL | NULL |
| LOAD | BRAND\_ID | Brand ID | INTEGER | NULL | NULL |
| LOAD | BRAND\_NAME | Brand Name | TEXT | 50 | NULL |
| CUSTOM | CASH\_DISC\_2\_VAL | CASH\_DISC\_2\_VAL | NUMBER | 38 | 8 |
| CUSTOM | LGSTC\_DISC\_VAL | LGSTC\_DISC\_VAL | NUMBER | 38 | 8 |
| CUSTOM | EFFCY\_DISC\_VAL | EFFCY\_DISC\_VAL | NUMBER | 38 | 8 |
| CUSTOM | DSTBR\_DISC\_VAL | DSTBR\_DISC\_VAL | NUMBER | 38 | 8 |
| CUSTOM | OTHER\_VAL | OTHER\_VAL | NUMBER | 38 | 8 |
| CUSTOM | FLEX\_FOLD\_IN\_VAL | FLEX\_FI\_VAL | NUMBER | 38 | 8 |
| CUSTOM | CASH\_DISC\_1\_VAL | CASH\_DISC\_1\_VAL | NUMBER | 38 | 8 |
| CUSTOM | NO\_BRKDW\_VAL | TT No Breakdown | PERCENT | 38 | 8 |
| CUSTOM | CUST\_L4\_ID | CUST\_L4\_ID | TEXT | 50 | NULL |
| CUSTOM | ATTR\_4\_VAL | ATTR\_4\_VAL | NUMBER | 38 | 8 |
| CUSTOM | ATTR\_3\_VAL | ATTR\_3\_VAL | NUMBER | 38 | 8 |
| CUSTOM | ATTR\_2\_VAL | ATTR\_2\_VAL | NUMBER | 38 | 8 |
| CUSTOM | ATTR\_1\_VAL | ATTR\_1\_VAL | NUMBER | 38 | 8 |
| CUSTOM | EXCDD\_CASH\_DISC\_PPR\_VAL | EXCDD\_CASH\_DISC\_PPR\_VAL | NUMBER | 38 | 8 |

### TT Direct

Market Groups:

* Europe - CE
* Europe - DACH
* Europe - EE
* Europe - IBERIA
* Europe - ITALY
* Europe - NORDICS
* Europe - SEE
* Europe - UKI

Input File Definitions:

| **COLUMN TYPE** | **COLUMN NAME** | **COLUMN LABEL** | **DATA TYPE** | **LENGTH** | **PRECISION** |
| --- | --- | --- | --- | --- | --- |
| SYSTEM | sys\_row\_id | Row surrogate key | INTEGER | NULL | NULL |
| SYSTEM | sys\_invld\_ind | Is row marked as Invalid | BOOLEAN | NULL | NULL |
| SYSTEM | sys\_obslt\_ind | Is row marked as Obsolete | BOOLEAN | NULL | NULL |
| SYSTEM | sys\_init\_actn\_id | Load/Upload Action which added this row | INTEGER | NULL | NULL |
| SYSTEM | sys\_last\_uplod\_actn\_id | Identifier of most recent Upload Action of this ro | INTEGER | NULL | NULL |
| SYSTEM | sys\_mkt\_id | Identifier of Market, added if mkt\_col\_id in Input | INTEGER | NULL | NULL |
| SYSTEM | sys\_last\_mdfd\_datetm | Date of last modification including Load Process t | DATE | NULL | NULL |
| SYSTEM | sys\_last\_mdfd\_user\_name | "User who uploaded as the last or ""AUTO\_LOAD"" if th" | TEXT | 50 | NULL |
| LOAD | MKT\_GEO\_ID | Market Geo ID | TEXT | 50 | NULL |
| LOAD | MKT\_NAME | Market | TEXT | 50 | NULL |
| LOAD | RPTNG\_CUST\_NAME | Reporting Customer Name | TEXT | 50 | NULL |
| LOAD | SBSTR\_ID | Subsector ID | INTEGER | NULL | NULL |
| LOAD | SBSTR\_NAME | Subsector Name | TEXT | 50 | NULL |
| LOAD | CATEG\_ID | Category ID | INTEGER | NULL | NULL |
| LOAD | CATEG\_NAME | Category Name | TEXT | 50 | NULL |
| LOAD | FY\_CODE | FY | TEXT | 50 | NULL |
| LOAD | MTH\_NUM | Month | MONTH | NULL | NULL |
| LOAD | BRAND\_ID | Brand ID | INTEGER | NULL | NULL |
| LOAD | BRAND\_NAME | Brand Name | TEXT | 50 | NULL |
| CUSTOM | CASH\_DISC\_2\_VAL | CASH\_DISC\_2\_VAL | NUMBER | 38 | 8 |
| CUSTOM | LGSTC\_DISC\_VAL | TT Logistic Discount | NUMBER | 38 | 8 |
| CUSTOM | EFFCY\_DISC\_VAL | TT Efficiency Discount | NUMBER | 38 | 8 |
| CUSTOM | DSTBR\_DISC\_VAL | TT Distributor Discount | NUMBER | 38 | 8 |
| CUSTOM | OTHER\_VAL | TT Other | NUMBER | 38 | 8 |
| CUSTOM | FLEX\_FOLD\_IN\_VAL | TT Flex Fold In | NUMBER | 38 | 8 |
| CUSTOM | CASH\_DISC\_1\_VAL | TT Cash Discount | PERCENT | 38 | 8 |
| CUSTOM | NO\_BRKDW\_VAL | NO\_BRKDW\_VAL | NUMBER | 38 | 8 |
| CUSTOM | CUST\_L4\_ID | CUST\_L4\_ID | TEXT | 50 | NULL |
| CUSTOM | ATTR\_4\_VAL | ATTR\_4\_VAL | NUMBER | 38 | 8 |
| CUSTOM | ATTR\_3\_VAL | ATTR\_3\_VAL | NUMBER | 38 | 8 |
| CUSTOM | ATTR\_2\_VAL | ATTR\_2\_VAL | NUMBER | 38 | 8 |
| CUSTOM | ATTR\_1\_VAL | ATTR\_1\_VAL | NUMBER | 38 | 8 |
| CUSTOM | EXCDD\_CASH\_DISC\_PPR\_VAL | TT OI ex. CD | PERCENT | 38 | 8 |

### SD National Direct

Market Groups:

* Europe - CE
* Europe - DACH
* Europe - IBERIA
* Europe - ITALY
* Europe - NORDICS
* Europe - SEE

Input File Definitions:

| **COLUMN TYPE** | **COLUMN NAME** | **COLUMN LABEL** | **DATA TYPE** | **LENGTH** | **PRECISION** |
| --- | --- | --- | --- | --- | --- |
| SYSTEM | sys\_row\_id | Row surrogate key | INTEGER | NULL | NULL |
| SYSTEM | sys\_invld\_ind | Is row marked as Invalid | BOOLEAN | NULL | NULL |
| SYSTEM | sys\_obslt\_ind | Is row marked as Obsolete | BOOLEAN | NULL | NULL |
| SYSTEM | sys\_init\_actn\_id | Load/Upload Action which added this row | INTEGER | NULL | NULL |
| SYSTEM | sys\_last\_uplod\_actn\_id | Identifier of most recent Upload Action of this ro | INTEGER | NULL | NULL |
| SYSTEM | sys\_mkt\_id | Identifier of Market, added if mkt\_col\_id in Input | INTEGER | NULL | NULL |
| SYSTEM | sys\_last\_mdfd\_datetm | Date of last modification including Load Process t | DATE | NULL | NULL |
| SYSTEM | sys\_last\_mdfd\_user\_name | "User who uploaded as the last or ""AUTO\_LOAD"" if th" | TEXT | 50 | NULL |
| LOAD | MKT\_GEO\_ID | Market Geo ID | TEXT | 50 | NULL |
| LOAD | MKT\_NAME | Market | TEXT | 50 | NULL |
| LOAD | SBSTR\_ID | Subsector ID | INTEGER | NULL | NULL |
| LOAD | SBSTR\_NAME | Subsector Name | TEXT | 50 | NULL |
| LOAD | CATEG\_ID | Category ID | INTEGER | NULL | NULL |
| LOAD | CATEG\_NAME | Category Name | TEXT | 50 | NULL |
| LOAD | FY\_CODE | FY | TEXT | 50 | NULL |
| LOAD | MTH\_NUM | Month | MONTH | NULL | NULL |
| LOAD | BRAND\_ID | Brand ID | INTEGER | NULL | NULL |
| LOAD | BRAND\_NAME | Brand Name | TEXT | 50 | NULL |
| LOAD | RPTNG\_CUST\_NAME | Reporting Customer Name | TEXT | 50 | NULL |
| LOAD | MKT\_GRP\_NAME | Market Group | TEXT | 50 | NULL |
| CUSTOM | NATL\_SD\_VAL | SD National | NUMBER | 38 | 8 |
| CUSTOM | ATTR\_4\_VAL | ATTR\_4\_VAL | NUMBER | 38 | 8 |
| CUSTOM | ATTR\_3\_VAL | ATTR\_3\_VAL | NUMBER | 38 | 8 |
| CUSTOM | ATTR\_2\_VAL | ATTR\_2\_VAL | NUMBER | 38 | 8 |
| CUSTOM | ATTR\_1\_VAL | ATTR\_1\_VAL | NUMBER | 38 | 8 |
| CUSTOM | CUST\_L4\_ID | CUST\_L4\_ID | TEXT | 50 | NULL |

### SD Customer Indirect

Market Groups:

* Europe - EE
* Europe - SEE
* Europe - TURKEY

Input File Definitions:

| **COLUMN TYPE** | **COLUMN NAME** | **COLUMN LABEL** | **DATA TYPE** | **LENGTH** | **PRECISION** |
| --- | --- | --- | --- | --- | --- |
| SYSTEM | sys\_row\_id | Row surrogate key | INTEGER | NULL | NULL |
| SYSTEM | sys\_invld\_ind | Is row marked as Invalid | BOOLEAN | NULL | NULL |
| SYSTEM | sys\_obslt\_ind | Is row marked as Obsolete | BOOLEAN | NULL | NULL |
| SYSTEM | sys\_init\_actn\_id | Load/Upload Action which added this row | INTEGER | NULL | NULL |
| SYSTEM | sys\_last\_uplod\_actn\_id | Identifier of most recent Upload Action of this ro | INTEGER | NULL | NULL |
| SYSTEM | sys\_mkt\_id | Identifier of Market, added if mkt\_col\_id in Input | INTEGER | NULL | NULL |
| SYSTEM | sys\_last\_mdfd\_datetm | Date of last modification including Load Process t | DATE | NULL | NULL |
| SYSTEM | sys\_last\_mdfd\_user\_name | "User who uploaded as the last or ""AUTO\_LOAD"" if th" | TEXT | 50 | NULL |
| LOAD | MKT\_GEO\_ID | Market Geo ID | TEXT | 50 | NULL |
| LOAD | MKT\_NAME | Market | TEXT | 50 | NULL |
| LOAD | RPTNG\_CUST\_NAME | Reporting Customer Name | TEXT | 50 | NULL |
| LOAD | SBSTR\_ID | Subsector ID | INTEGER | NULL | NULL |
| LOAD | SBSTR\_NAME | Subsector Name | TEXT | 50 | NULL |
| LOAD | CATEG\_ID | Category ID | INTEGER | NULL | NULL |
| LOAD | CATEG\_NAME | Category Name | TEXT | 50 | NULL |
| LOAD | BRAND\_ID | Brand ID | INTEGER | NULL | NULL |
| LOAD | BRAND\_NAME | Brand Name | TEXT | 50 | NULL |
| LOAD | FY\_CODE | FY | TEXT | 50 | NULL |
| LOAD | MTH\_NUM | Month | MONTH | NULL | NULL |
| CUSTOM | NO\_BRKDW\_VAL | SD No Breakdown | NUMBER | 38 | 8 |
| CUSTOM | ATTR\_4\_VAL | ATTR\_4\_VAL | NUMBER | 38 | 8 |
| CUSTOM | ATTR\_3\_VAL | ATTR\_3\_VAL | NUMBER | 38 | 8 |
| CUSTOM | ATTR\_2\_VAL | ATTR\_2\_VAL | NUMBER | 38 | 8 |
| CUSTOM | ATTR\_1\_VAL | ATTR\_1\_VAL | NUMBER | 38 | 8 |
| CUSTOM | CNSMR\_VAL | CNSMR\_VAL | NUMBER | 38 | 8 |
| CUSTOM | CUST\_VAL | CUST\_VAL | NUMBER | 38 | 8 |
| CUSTOM | OTHER\_VAL | OTHER\_VAL | NUMBER | 38 | 8 |
| CUSTOM | LYLTY\_CARD\_VAL | LYLTY\_CARD\_VAL | NUMBER | 38 | 8 |
| CUSTOM | SCNR\_VAL | SCNR\_VAL | NUMBER | 38 | 8 |
| CUSTOM | DIST\_DRIVE\_VAL | DIST\_DRIVE\_VAL | NUMBER | 38 | 8 |
| CUSTOM | MSP\_RSRV\_CUST\_INVC\_VAL | MSP\_RSRV\_CUST\_INVC\_VAL | NUMBER | 38 | 8 |
| CUSTOM | MSP\_RSRV\_CREDT\_NOTE\_VAL | MSP\_RSRV\_CREDT\_NOTE\_VAL | NUMBER | 38 | 8 |
| CUSTOM | QBDF\_CUST\_INVC\_VAL | QBDF\_CUST\_INVC\_VAL | NUMBER | 38 | 8 |
| CUSTOM | QBDF\_CREDT\_NOTE\_VAL | QBDF\_CREDT\_NOTE\_VAL | NUMBER | 38 | 8 |
| CUSTOM | QBDF\_ON\_INVC\_VAL | QBDF\_ON\_INVC\_VAL | NUMBER | 38 | 8 |
| CUSTOM | CORP\_PLAN\_VAL | CORP\_PLAN\_VAL | NUMBER | 38 | 8 |
| CUSTOM | INNVN\_FUND\_VAL | INNVN\_FUND\_VAL | NUMBER | 38 | 8 |
| CUSTOM | ANN\_AGRMT\_VAL | ANN\_AGRMT\_VAL | NUMBER | 38 | 8 |
| CUSTOM | CUST\_L4\_ID | CUST\_L4\_ID | TEXT | 50 | NULL |
| CUSTOM | ATTR\_6\_VAL | ATTR\_6\_VAL | NUMBER | 38 | 8 |
| CUSTOM | ATTR\_12\_VAL | ATTR\_12\_VAL | NUMBER | 38 | 8 |
| CUSTOM | ATTR\_7\_VAL | ATTR\_7\_VAL | NUMBER | 38 | 8 |
| CUSTOM | ATTR\_11\_VAL | ATTR\_11\_VAL | NUMBER | 38 | 8 |
| CUSTOM | ATTR\_9\_VAL | ATTR\_9\_VAL | NUMBER | 38 | 8 |
| CUSTOM | ATTR\_10\_VAL | ATTR\_10\_VAL | NUMBER | 38 | 8 |
| CUSTOM | ATTR\_5\_VAL | ATTR\_5\_VAL | NUMBER | 38 | 8 |
| CUSTOM | ATTR\_8\_VAL | ATTR\_8\_VAL | NUMBER | 38 | 8 |
| CUSTOM | ATTR\_13\_VAL | ATTR\_13\_VAL | NUMBER | 38 | 8 |

### TT Direct Forecast

Market Groups:

* Europe - FRANCE
* Europe - UKI

Input File Definitions:

| **COLUMN TYPE** | **COLUMN NAME** | **COLUMN LABEL** | **DATA TYPE** | **LENGTH** | **PRECISION** |
| --- | --- | --- | --- | --- | --- |
| SYSTEM | sys\_row\_id | Row surrogate key | INTEGER | NULL | NULL |
| SYSTEM | sys\_invld\_ind | Is row marked as Invalid | BOOLEAN | NULL | NULL |
| SYSTEM | sys\_obslt\_ind | Is row marked as Obsolete | BOOLEAN | NULL | NULL |
| SYSTEM | sys\_init\_actn\_id | Load/Upload Action which added this row | INTEGER | NULL | NULL |
| SYSTEM | sys\_last\_uplod\_actn\_id | Identifier of most recent Upload Action of this ro | INTEGER | NULL | NULL |
| SYSTEM | sys\_mkt\_id | Identifier of Market, added if mkt\_col\_id in Input | INTEGER | NULL | NULL |
| SYSTEM | sys\_last\_mdfd\_datetm | Date of last modification including Load Process t | DATE | NULL | NULL |
| SYSTEM | sys\_last\_mdfd\_user\_name | "User who uploaded as the last or ""AUTO\_LOAD"" if th" | TEXT | 50 | NULL |
| LOAD | MKT\_GEO\_ID | Market Geo ID | TEXT | 50 | NULL |
| LOAD | MKT\_NAME | Market | TEXT | 50 | NULL |
| LOAD | RPTNG\_CUST\_NAME | Reporting Customer Name | TEXT | 50 | NULL |
| LOAD | SBSTR\_ID | Subsector ID | INTEGER | NULL | NULL |
| LOAD | SBSTR\_NAME | Subsector Name | TEXT | 50 | NULL |
| LOAD | CATEG\_ID | Category ID | INTEGER | NULL | NULL |
| LOAD | CATEG\_NAME | Category Name | TEXT | 50 | NULL |
| LOAD | FY\_CODE | FY | TEXT | 50 | NULL |
| LOAD | MTH\_NUM | Month | MONTH | NULL | NULL |
| LOAD | BRAND\_ID | Brand ID | INTEGER | NULL | NULL |
| LOAD | BRAND\_NAME | Brand Name | TEXT | 50 | NULL |
| CUSTOM | CASH\_DISC\_2\_VAL | CASH\_DISC\_2\_VAL | NUMBER | 38 | 8 |
| CUSTOM | LGSTC\_DISC\_VAL | LGSTC\_DISC\_VAL | NUMBER | 38 | 8 |
| CUSTOM | EFFCY\_DISC\_VAL | EFFCY\_DISC\_VAL | NUMBER | 38 | 8 |
| CUSTOM | DSTBR\_DISC\_VAL | DSTBR\_DISC\_VAL | NUMBER | 38 | 8 |
| CUSTOM | OTHER\_VAL | OTHER\_VAL | NUMBER | 38 | 8 |
| CUSTOM | FLEX\_FOLD\_IN\_VAL | FLEX\_FI\_VAL | NUMBER | 38 | 8 |
| CUSTOM | CASH\_DISC\_1\_VAL | TT Cash Discount | PERCENT | 38 | 8 |
| CUSTOM | NO\_BRKDW\_VAL | NO\_BRKDW\_VAL | NUMBER | 38 | 8 |
| CUSTOM | CUST\_L4\_ID | CUST\_L4\_ID | TEXT | 50 | NULL |
| CUSTOM | ATTR\_4\_VAL | ATTR\_4\_VAL | NUMBER | 38 | 8 |
| CUSTOM | ATTR\_3\_VAL | ATTR\_3\_VAL | NUMBER | 38 | 8 |
| CUSTOM | ATTR\_2\_VAL | ATTR\_2\_VAL | NUMBER | 38 | 8 |
| CUSTOM | ATTR\_1\_VAL | ATTR\_1\_VAL | NUMBER | 38 | 8 |
| CUSTOM | EXCDD\_CASH\_DISC\_PPR\_VAL | TT exc. CD and PPR | PERCENT | 38 | 8 |

### SD Regional Direct Forecast

Market Groups:

* Europe - FRANCE
* Europe - UKI

Input File Definitions:

| **COLUMN TYPE** | **COLUMN NAME** | **COLUMN LABEL** | **DATA TYPE** | **LENGTH** | **PRECISION** |
| --- | --- | --- | --- | --- | --- |
| SYSTEM | sys\_row\_id | Row surrogate key | INTEGER | NULL | NULL |
| SYSTEM | sys\_invld\_ind | Is row marked as Invalid | BOOLEAN | NULL | NULL |
| SYSTEM | sys\_obslt\_ind | Is row marked as Obsolete | BOOLEAN | NULL | NULL |
| SYSTEM | sys\_init\_actn\_id | Load/Upload Action which added this row | INTEGER | NULL | NULL |
| SYSTEM | sys\_last\_uplod\_actn\_id | Identifier of most recent Upload Action of this ro | INTEGER | NULL | NULL |
| SYSTEM | sys\_mkt\_id | Identifier of Market, added if mkt\_col\_id in Input | INTEGER | NULL | NULL |
| SYSTEM | sys\_last\_mdfd\_datetm | Date of last modification including Load Process t | DATE | NULL | NULL |
| SYSTEM | sys\_last\_mdfd\_user\_name | "User who uploaded as the last or ""AUTO\_LOAD"" if th" | TEXT | 50 | NULL |
| LOAD | MKT\_GEO\_ID | Market Geo ID | TEXT | 50 | NULL |
| LOAD | MKT\_NAME | Market | TEXT | 50 | NULL |
| LOAD | RPTNG\_CUST\_NAME | Reporting Customer Name | TEXT | 50 | NULL |
| LOAD | SBSTR\_ID | Subsector ID | INTEGER | NULL | NULL |
| LOAD | SBSTR\_NAME | Subsector Name | TEXT | 50 | NULL |
| LOAD | CATEG\_ID | Category ID | INTEGER | NULL | NULL |
| LOAD | CATEG\_NAME | Category Name | TEXT | 50 | NULL |
| LOAD | FY\_CODE | FY | TEXT | 50 | NULL |
| LOAD | MTH\_NUM | Month | MONTH | NULL | NULL |
| LOAD | BRAND\_ID | Brand ID | INTEGER | NULL | NULL |
| LOAD | BRAND\_NAME | Brand Name | TEXT | 50 | NULL |
| CUSTOM | REGNL\_SD\_VAL | SD Regional | PERCENT | 38 | 8 |
| CUSTOM | ATTR\_4\_VAL | ATTR\_4\_VAL | NUMBER | 38 | 8 |
| CUSTOM | ATTR\_3\_VAL | ATTR\_3\_VAL | NUMBER | 38 | 8 |
| CUSTOM | ATTR\_2\_VAL | ATTR\_2\_VAL | NUMBER | 38 | 8 |
| CUSTOM | ATTR\_1\_VAL | ATTR\_1\_VAL | NUMBER | 38 | 8 |
| CUSTOM | CUST\_L4\_ID | CUST\_L4\_ID | TEXT | 50 | NULL |

### National Direct L4

Market Groups:

* Europe - FRANCE
* Europe - UKI

Input File Definitions:

| **COLUMN TYPE** | **COLUMN NAME** | **COLUMN LABEL** | **DATA TYPE** | **LENGTH** | **PRECISION** |
| --- | --- | --- | --- | --- | --- |
| SYSTEM | sys\_row\_id | Row surrogate key | INTEGER | NULL | NULL |
| SYSTEM | sys\_invld\_ind | Is row marked as Invalid | BOOLEAN | NULL | NULL |
| SYSTEM | sys\_obslt\_ind | Is row marked as Obsolete | BOOLEAN | NULL | NULL |
| SYSTEM | sys\_init\_actn\_id | Load/Upload Action which added this row | INTEGER | NULL | NULL |
| SYSTEM | sys\_last\_uplod\_actn\_id | Identifier of most recent Upload Action of this ro | INTEGER | NULL | NULL |
| SYSTEM | sys\_mkt\_id | Identifier of Market, added if mkt\_col\_id in Input | INTEGER | NULL | NULL |
| SYSTEM | sys\_last\_mdfd\_datetm | Date of last modification including Load Process t | DATE | NULL | NULL |
| SYSTEM | sys\_last\_mdfd\_user\_name | "User who uploaded as the last or ""AUTO\_LOAD"" if th" | TEXT | 50 | NULL |
| LOAD | MKT\_GEO\_ID | Market Geo ID | TEXT | 50 | NULL |
| LOAD | MKT\_NAME | Market | TEXT | 50 | NULL |
| LOAD | SBSTR\_ID | Subsector ID | INTEGER | NULL | NULL |
| LOAD | SBSTR\_NAME | Subsector Name | TEXT | 50 | NULL |
| LOAD | CATEG\_ID | Category ID | INTEGER | NULL | NULL |
| LOAD | CATEG\_NAME | Category Name | TEXT | 50 | NULL |
| LOAD | FY\_CODE | FY | TEXT | 50 | NULL |
| LOAD | MTH\_NUM | Month | MONTH | NULL | NULL |
| LOAD | BRAND\_ID | Brand ID | INTEGER | NULL | NULL |
| LOAD | BRAND\_NAME | Brand Name | TEXT | 50 | NULL |
| LOAD | RPTNG\_CUST\_NAME | Reporting Customer Name | TEXT | 50 | NULL |
| LOAD | MKT\_GRP\_NAME | Market Group | TEXT | 50 | NULL |
| CUSTOM | NATL\_SD\_VAL | SD National | NUMBER | 38 | 8 |
| CUSTOM | ATTR\_4\_VAL | ATTR\_4\_VAL | NUMBER | 38 | 8 |
| CUSTOM | ATTR\_3\_VAL | ATTR\_3\_VAL | NUMBER | 38 | 8 |
| CUSTOM | ATTR\_2\_VAL | ATTR\_2\_VAL | NUMBER | 38 | 8 |
| CUSTOM | ATTR\_1\_VAL | ATTR\_1\_VAL | NUMBER | 38 | 8 |
| CUSTOM | CUST\_L4\_ID | CUST\_L4\_ID | TEXT | 50 | NULL |

### National Direct L4 Forecast

Market Groups:

* Europe - FRANCE
* Europe - UKI

Input File Definitions:

| **COLUMN TYPE** | **COLUMN NAME** | **COLUMN LABEL** | **DATA TYPE** | **LENGTH** | **PRECISION** |
| --- | --- | --- | --- | --- | --- |
| SYSTEM | sys\_row\_id | Row surrogate key | INTEGER | NULL | NULL |
| SYSTEM | sys\_invld\_ind | Is row marked as Invalid | BOOLEAN | NULL | NULL |
| SYSTEM | sys\_obslt\_ind | Is row marked as Obsolete | BOOLEAN | NULL | NULL |
| SYSTEM | sys\_init\_actn\_id | Load/Upload Action which added this row | INTEGER | NULL | NULL |
| SYSTEM | sys\_last\_uplod\_actn\_id | Identifier of most recent Upload Action of this ro | INTEGER | NULL | NULL |
| SYSTEM | sys\_mkt\_id | Identifier of Market, added if mkt\_col\_id in Input | INTEGER | NULL | NULL |
| SYSTEM | sys\_last\_mdfd\_datetm | Date of last modification including Load Process t | DATE | NULL | NULL |
| SYSTEM | sys\_last\_mdfd\_user\_name | "User who uploaded as the last or ""AUTO\_LOAD"" if th" | TEXT | 50 | NULL |
| LOAD | MKT\_GEO\_ID | Market Geo ID | TEXT | 50 | NULL |
| LOAD | MKT\_NAME | Market | TEXT | 50 | NULL |
| LOAD | SBSTR\_ID | Subsector ID | INTEGER | NULL | NULL |
| LOAD | SBSTR\_NAME | Subsector Name | TEXT | 50 | NULL |
| LOAD | CATEG\_ID | Category ID | INTEGER | NULL | NULL |
| LOAD | CATEG\_NAME | Category Name | TEXT | 50 | NULL |
| LOAD | FY\_CODE | FY | TEXT | 50 | NULL |
| LOAD | MTH\_NUM | Month | MONTH | NULL | NULL |
| LOAD | BRAND\_ID | Brand ID | INTEGER | NULL | NULL |
| LOAD | BRAND\_NAME | Brand Name | TEXT | 50 | NULL |
| LOAD | RPTNG\_CUST\_NAME | Reporting Customer Name | TEXT | 50 | NULL |
| LOAD | MKT\_GRP\_NAME | Market Group | TEXT | 50 | NULL |
| CUSTOM | NATL\_SD\_VAL | SD National | PERCENT | 38 | 8 |
| CUSTOM | ATTR\_4\_VAL | ATTR\_4\_VAL | NUMBER | 38 | 8 |
| CUSTOM | ATTR\_3\_VAL | ATTR\_3\_VAL | NUMBER | 38 | 8 |
| CUSTOM | ATTR\_2\_VAL | ATTR\_2\_VAL | NUMBER | 38 | 8 |
| CUSTOM | ATTR\_1\_VAL | ATTR\_1\_VAL | NUMBER | 38 | 8 |
| CUSTOM | CUST\_L4\_ID | CUST\_L4\_ID | TEXT | 50 | NULL |

### SD National Indirect

Market Groups:

* Europe - SEE

Input File Definitions:

| **COLUMN TYPE** | **COLUMN NAME** | **COLUMN LABEL** | **DATA TYPE** | **LENGTH** | **PRECISION** |
| --- | --- | --- | --- | --- | --- |
| SYSTEM | sys\_row\_id | Row surrogate key | INTEGER | NULL | NULL |
| SYSTEM | sys\_invld\_ind | Is row marked as Invalid | BOOLEAN | NULL | NULL |
| SYSTEM | sys\_obslt\_ind | Is row marked as Obsolete | BOOLEAN | NULL | NULL |
| SYSTEM | sys\_init\_actn\_id | Load/Upload Action which added this row | INTEGER | NULL | NULL |
| SYSTEM | sys\_last\_uplod\_actn\_id | Identifier of most recent Upload Action of this ro | INTEGER | NULL | NULL |
| SYSTEM | sys\_mkt\_id | Identifier of Market, added if mkt\_col\_id in Input | INTEGER | NULL | NULL |
| SYSTEM | sys\_last\_mdfd\_datetm | Date of last modification including Load Process t | DATE | NULL | NULL |
| SYSTEM | sys\_last\_mdfd\_user\_name | "User who uploaded as the last or ""AUTO\_LOAD"" if th" | TEXT | 50 | NULL |
| LOAD | MKT\_GEO\_ID | Market Geo ID | TEXT | 50 | NULL |
| LOAD | MKT\_NAME | Market | TEXT | 50 | NULL |
| LOAD | SBSTR\_ID | Subsector ID | INTEGER | NULL | NULL |
| LOAD | SBSTR\_NAME | Subsector Name | TEXT | 50 | NULL |
| LOAD | CATEG\_ID | Category ID | INTEGER | NULL | NULL |
| LOAD | CATEG\_NAME | Category Name | TEXT | 50 | NULL |
| LOAD | FY\_CODE | FY | TEXT | 50 | NULL |
| LOAD | MTH\_NUM | Month | MONTH | NULL | NULL |
| LOAD | BRAND\_ID | Brand ID | INTEGER | NULL | NULL |
| CUSTOM | CUST\_L4\_ID | CUST\_L4\_ID | TEXT | 50 | NULL |
| LOAD | BRAND\_NAME | Brand Name | TEXT | 50 | NULL |
| LOAD | RPTNG\_CUST\_NAME | Reporting Customer Name | TEXT | 50 | NULL |
| LOAD | MKT\_GRP\_NAME | Market Group | TEXT | 50 | NULL |
| CUSTOM | NATL\_SD\_VAL | SD National | NUMBER | 38 | 8 |
| CUSTOM | ATTR\_4\_VAL | ATTR\_4\_VAL | NUMBER | 38 | 8 |
| CUSTOM | ATTR\_3\_VAL | ATTR\_3\_VAL | NUMBER | 38 | 8 |
| CUSTOM | ATTR\_2\_VAL | ATTR\_2\_VAL | NUMBER | 38 | 8 |
| CUSTOM | ATTR\_1\_VAL | ATTR\_1\_VAL | NUMBER | 38 | 8 |
| CUSTOM | CUST\_L4\_ID | CUST\_L4\_ID | TEXT | 50 | NULL |

## Overview diagram

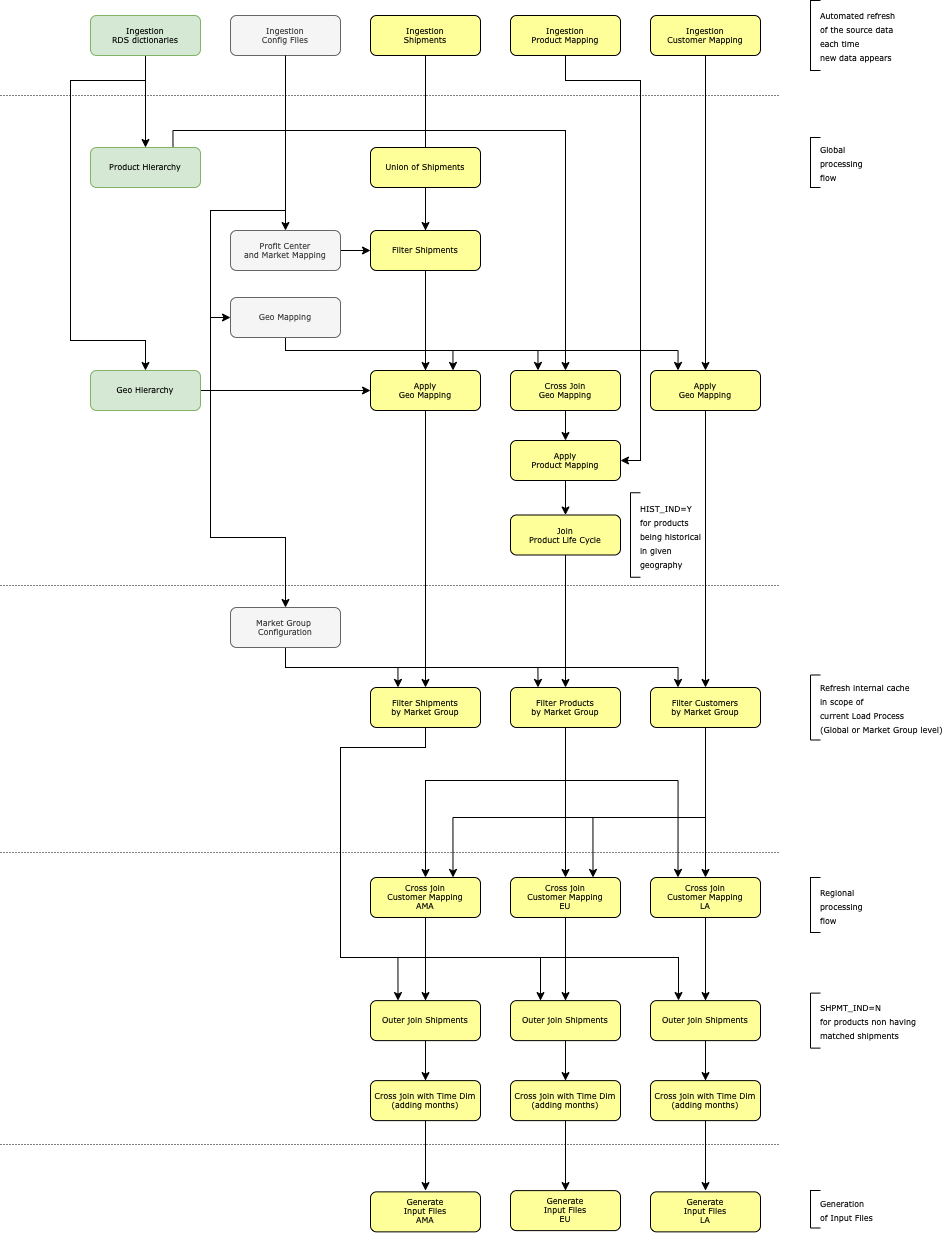


FIGURE:Overview diagram

## Geo Mapping reload.

|  | **Step** | **ADF/DB Component** | **Process** |
| --- | --- | --- | --- |
| Copy RDS data from BP2 | 0 | [ADF] in-RDSCopy | Copy RDSv2 files from BP2 to Uploader's storage account |
| Reload of Geo Mapping | 1 | [ADF] ld-GeoMappingReload | Triggers the ADB notebooks for Creating RDS Dimensions and Reloading the Geo Mapping |
|  | 2 | [ADB] 310\_CREATE\_RDS\_DIMENSIONS | Refreshes RDS Dimensions including Geo Hierarchy which is needed in Reloading Geo Mapping |
|  | 3 | [ADB] 330\_RELOAD\_GEO\_MAPPING | Generate Geo Mapping content based on Profit Center and Market Mapping file and pulled RDSv2 file (RDS GEO 705) then dump content as parquet file in storage |
|  | 4 | [ADF] ld-GeoMappingToStaging | Copy ADB output to staging Geo Map table then trigger stored procedure |
|  | 5 | [DB] main.pro\_geo\_mapng\_reld | Merge data from staging Geo Map table to final input Geo Mapping table |

### Phase 1: Ingestion of source data

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Step** | **ADF/DB Component** | **Process** |
| Ingestion of Shipments | 1 | [ADF] in-ShipmentsCopy | To copy Shipment files from BP2 to Uploader's storage account |
|  | 2 | [ADB] 100\_INGEST\_DIRECT\_SHIPMENTS | To ingest Direct Shipments data to ADB table (stage\_dirct\_shipments.ship\_step\_100\_sode\_hist\_star\_work\_tc\_lc) |
|  | 3 | [ADB] 101\_INGEST\_INDIRECT\_SHIPMENTS | To ingest Indirect Shipments data to ADB table **EU:** stage\_indir\_shipments.eu\_ind\_sh **AMA:** stage\_indir\_shipments.an\_ind\_ship\_step\_100\_rtdc\_hist\_star |
| Ingestion of Product Mapping | 1 | [ADF] in-ProductMappingCopy | To copy Product Mapping file from BP2 to Uploader's storage account |
|  | 2 | [ADB] 102\_INGEST\_MMR | To ingest Product Mapping data to ADB table (stage\_commn.mnr\_prod\_mapng\_dim\_vw) |
| Ingestion of Customer Mapping | 1 | [ADF] in-CustomerMappingCopy | To copy Customer Mapping file from BP2 to Uploader's storage account |
|  | 2 | [ADB] 102\_INGEST\_MMR | To ingest Customer Mapping data to ADB table **AMA/EU:** stage\_commn.mnr\_cust\_mapng\_dim **LA:** stage\_commn.manual\_customer\_mapping\_v |
| Ingestion of RDS data | 1 | [ADF] in-RDSCopy | To copy RDSv2 files from BP2 to Uploader's storage account |
|  | 2 | [ADB] 103\_INGEST\_RDS | To ingest RDS data to ADB table (stage\_commn.masterdata\_prod\_hier\_dim, stage\_commn.masterdata\_cust\_hier\_dim, stage\_commn.masterdata\_geo\_hier\_dim, stage\_commn.masterdata\_proft\_ctr\_hier\_dim, stage\_commn.masterdata\_prod\_life\_cycle\_dim) |

### ­ Phase 2: Global shipments processing

|  |  |  |  |
| --- | --- | --- | --- |
| Global Shipments Processing | 1 | [ADB] 111\_UNION\_DIR\_INDIR\_SHIPMENTS | To combine ingested direct and indirect shipments data into one dataset. **Output:** stage\_commn.shpmt\_dir\_indir\_union\_star\_work |
|  | 2 | [ADB] 120\_APPLY\_FACT\_FILTERING | Filter fact data using sql conditions defined in Profit Center and Market Mapping config file. **Output:** stage\_commn.shpmt\_fact\_filtr\_star\_work |
|  | 3 | [ADF] ld-DumpGeoMappingToParquet | Dump Geo Mapping data in DB as parquet file. Archiving of existing geo mapping files in storage account. |
|  | 4 | [ADB] 106\_INGEST\_GEO\_MAPPING | Re-ingest geo mapping parquet file to geo mapping table in ADB. **Output:** stage\_commn.geo\_mapng |
|  | 5 | [ADB] 200\_JOIN\_GEO\_MAPPING | Join filtered shipments data with generated Geo Mapping config file. **Output:** stage\_commn.shpmt\_join\_geo\_mapng\_star\_work |
|  |  |  |  |

### Phase 3: Regional shipments processing

|  |  |  |  |
| --- | --- | --- | --- |
| Regional Shipments Processing | 1 | [ADB] 210\_JOIN\_PRODUCT\_HIER\_GEO\_MAPPING | Join Product Hierarchy with Geo Mapping data. **Output:** stage\_commn.prod\_hier\_join\_geo\_mapng\_star\_work |
|  | 2 | [ADB] 211\_JOIN\_PRODUCT\_MAPPING\_PRODUCT\_HIER | Join Product Hierarchy with Product Mapping data. **Output:** stage\_commn.prod\_hier\_join\_prod\_mapng\_star\_work |
|  | 3 | [ADB] 212\_OUTER\_JOIN\_PRODUCT\_LIFE\_CYCLE\_PRODUCT\_HIER | Join Product Hierarchy with Product Life Cycle data. **Output:** stage\_commn.prod\_hier\_join\_prod\_life\_cycle\_star\_work |
|  | 4 | [ADB] 220\_CROSS\_JOIN\_CUSTOMER\_MAPPING\_PRODUCT\_HIER\_LA | Cross Join Product Hierarchy with Customer Mapping data for LA region. **Output:** stage\_commn.prod\_hier\_cross\_join\_cust\_mapng\_la\_star\_work |
|  | 5 | [ADB] 221\_CROSS\_JOIN\_CUSTOMER\_MAPPING\_PRODUCT\_HIER\_AMA | Cross Join Product Hierarchy with Customer Mapping data for AMA region. **Output:** stage\_commn.prod\_hier\_cross\_join\_cust\_mapng\_ama\_star\_work |
|  | 6 | [ADB] 222\_CROSS\_JOIN\_CUSTOMER\_MAPPING\_PRODUCT\_HIER\_EU | Cross Join Product Hierarchy with Customer Mapping data for EU region. **Output:** stage\_commn.prod\_hier\_cross\_join\_cust\_mapng\_eu\_star\_work |
|  | 7 | [ADB] 230\_JOIN\_CUSTOMER\_MAPPING\_LA | Outer Join Shipments with Products cross joined to Customer Mapping data for LA region. **Output:** stage\_commn.shpmt\_join\_cust\_mapng\_la\_star\_work |
|  | 8 | [ADB] 231\_JOIN\_CUSTOMER\_MAPPING\_AMA | Outer Join Shipments with Products cross joined to Customer Mapping data for AMA region. **Output:** stage\_commn.shpmt\_join\_cust\_mapng\_ama\_star\_work |
|  | 9 | [ADB] 232\_JOIN\_CUSTOMER\_MAPPING\_EU | Outer Join Shipments with Products cross joined to Customer Mapping data for EU region. **Output:** stage\_commn.shpmt\_join\_cust\_mapng\_eu\_star\_work |
|  | 10 | [ADB] 240\_CROSS\_JOIN\_TIME\_DIMENSION\_LA | Cross Join Shipments with Time Dimension data for LA region. **Output:** stage\_commn.load\_prcss\_cube\_la\_star\_work |
|  | 11 | [ADB] 241\_CROSS\_JOIN\_TIME\_DIMENSION\_AMA | Cross Join Shipments with Time Dimension data for AMA region. **Output:** stage\_commn.load\_prcss\_cube\_ama\_star\_work |
|  | 12 | [ADB] 242\_CROSS\_JOIN\_TIME\_DIMENSION\_EU | Cross Join Shipments with Time Dimension data for LA region. **Output:** stage\_commn.load\_prcss\_cube\_eu\_star\_work |
|  |  |  |  |

### Phase 4: Generating Input File Content

There are two main components responsible for starting the process:

* [ADF] ld-LoadMainGlobal - load data from incoming storage account to Uploader application for all market groups
* [ADF] ld-LoadMainSelective - load data from incoming storage account to Uploader application for selected market groups

Both contains the steps which runs the following components:

| **Step** | **ADF/DB Component** |
| --- | --- |
| 1 | [ADF] GetMarketGroupList --> [DB] *SELECT mkt\_grp\_name FROM md.mkt\_grp\_lkp\_vw WHERE mkt\_grp\_name <> 'Technical'* |
| 2 | [ADF] ExecuteLoadPrepareConfigGlobal --> [ADF] Id-PrepareConfigGlobal --> [DB] *md.mkt\_grp\_lkp\_vw* --> table to csv (ld-LoadMainGlobal only) |
| 3 | [ADF] ExecuteActualLoad --> [ADF] Id-ActualLoad |
| 3.1 | [ADF] ExecuteActualLoadConfigPrep --> [ADF] ld-DumpDBConfigFiles |
| 3.1.1 | [ADF] ExecuteLoadPrepLoadCubeColConfig --> [ADF] ld-DumpLoadCubeColumnsToCSV --> [DB] *SELECT load\_cube\_name, load\_cube\_id, col\_name, col\_num FROM md.load\_cube\_col\_prc\_vw* --> to csv |
| 3.1.2 | [ADF] ExecuteLoadPrepMktGrpLoadCubeConfig --> [ADF] ld-DumpMarketGroupsLoadCubesToCSV --> [DB] *SELECT regn\_name, mkt\_grp\_name, kt\_grp\_id, load\_cube\_id, actl\_ind, frcst\_ind, load\_cube\_tbl\_name FROM md.mkt\_grp\_load\_cube\_vw WHERE regn\_name <> 'Technical'* --> to csv |
| 3.2 | [ADF] ExecuteActualLoadConfigDump --> [ADF] ld-LoadConfigFiles |
| 3.2.1 | [ADF] LoadUploaderGeoMappingToTable --> [ADF] ld-GenericADBExecute --> [ADB] 270\_ingst\_upldr\_geo\_mapng |
| 3.2.2 | [ADF] LoadUploaderMarketGroupToTable --> [ADF] ld-GenericADBExecute --> [ADB] 290\_ingst\_upldr\_mkt\_grp\_prcsg\_cnfg |
| 3.2.3 | [ADF] LoadUploaderMarketGroupLoadCube --> [ADF] ld-GenericADBExecute --> [ADB] 291\_ingst\_upldr\_mkt\_grp\_load\_cube |
| 3.2.4 | [ADF] LoadUploaderLoadCubeColumn --> [ADF] ld-GenericADBExecute --> [ADB] 292\_ingst\_upldr\_load\_cube\_col |
| 3.3 | [ADF] ExecuteLoadPrepareStageTbls --> [ADF] ld-PrepareCubesAndStageTbls |
| 3.3.1 | [ADF] FilterShipmentsByProd --> [ADF] ld-GenericADBExecute --> [ADB] 710\_shpmt\_filtr\_by\_prod |
| 3.3.2 | [ADF] FilterShipmentsByProdCust --> [ADF] ld-GenericADBExecute --> [ADB] 720\_shpmt\_filtr\_by\_prod\_cust |
| 3.3.3 | [ADF] FilterProductsByShpmtProdCust --> [ADF] ld-GenericADBExecute --> [ADB] 730\_prod\_filtr\_by\_shpmt\_prod\_cust |
| 3.3.4 | [ADF] FilterProductsByShpmtProd --> [ADF] ld-GenericADBExecute --> [ADB] 740\_prod\_filtr\_by\_shpmt\_prod |
| 3.3.5 | [ADF] CustomerProdBrand --> [ADF] ld-GenericADBExecute --> [ADB] 810\_cust\_prod\_brand |
| 3.3.6 | [ADF] CustomerProdBrandShipment --> [ADF] ld-GenericADBExecute --> [ADB] 820\_cust\_prod\_brand\_shpmt |
| 3.3.7 | [ADF] ProdFpcShipment --> [ADF] ld-GenericADBExecute --> [ADB] 830\_prod\_fpc\_shpmt |
| 3.3.8 | [ADF] LoadCube --> [ADF] ld-GenericADBExecute --> [ADB] 840\_load\_cube |
| 3.3.9 | [ADF] ExecutePrepareStageTbls --> [ADF] Id-PrepareStageTblsFromCubes |
| 3.3.9.1 | [ADF] CreateStageTbls --> [ADF] ld-StageTblADBExecute --> [ADB] 850\_stage\_tbl |
| 3.4 | [ADF] ExecuteLoadStageTblsToDB --> [ADF] ld-UploadStageTblsToDB |
| 3.4.1 | [ADF] Truncate Staging Table --> [DB] *TRUNCATE TABLE @{pipeline().parameters.l\_db\_stage\_schma\_name}.@{item().load\_stage\_tbl\_name}; SELECT '@{item().load\_stage\_tbl\_name}' AS load\_cube\_tbl\_name;* |
| 3.4.2 | [ADF] Insert data to load Stage Table |
| 3.5 | [ADF] ExecuteRefreshInputFiles --> [ADF] ld-RefreshInputFiles |
| 3.5.1 | [ADF] ProcessRefreshInputPerMktGrp --> [DB] *[main].[pro\_load\_scope\_rfrsh]* |

## Obsolete and Invalid Rows in Load Process and Upload

### General guidelines

* each Load Input File row can be set as Obsolete
* row can be set to Obsolete or unset only by Load Process
* both Load and Non-Load Input Files optionally contain flag to mark rows as Invalid and it is editable via Upload

### Load Process

* all rows existing in Working Table not having representation (matching by Business Key) in current source data (in terms of time, product or customer dimension) are set as Obsolete
* all existing rows having such representation are set as Non-Obsolete
* existing rows are left as Valid/Invalid as prior to the upload
* new rows are by default Valid
* in Actual Input Files rows for current calendar month and future months in this FY are set as Invalid. this does not happen for Forecast Input Files where future months are left valid.

### Upload

* rows are left as Obsolete/Non-Obsolete just like they were prior to the upload
* rows marked in uploaded file as Invalid will be set in Work Table as Invalid, and Valid as Valid
* for Non-Load Input File if given Row ID is missing in file being uploaded, corresponding row is removed from the Input File Work Table

# CONFIG FILE AND SOURCE FILE REFERENCE

## Config Files

Technical/configuration files are separate from load and non-load input files.  
These are used as reference for data load processing and other technical data used for data maintenance. List of technical/configuration files:

* Profit Center and Market mapping
* Geo Mapping

### Profit Center and Market Mapping

**Description**

Profit Center and Market Mapping (PCMM) is a Config File designed to serve two purposes:

* to point parts of geography hierarchy which should be included in Geo Mapping and indirectly in whole Uploader's processing
* to provide geography based filtering of source shipments during Load Process via SQL conditions in dirct\_filtr\_cond\_sql\_txt and indir\_filtr\_cond\_sql\_txt

Below are characteristics of the mapping:

* there's no Invalid Row indicator nor Obsolete Row indicator

**Column list**

| **Technical Column Name** | **Column Label** | **Data Source** | **Description** |
| --- | --- | --- | --- |
| rds\_geo\_id | RDS Geo ID | Manual Input | Defines which RDS Geo ID to use during shipments pull |
| rds\_geo\_name | RDS Geo Name | Manual Input | Defines which RDS Geo Name to use during shipments pull |
| rds\_geo\_lvl | RDS Geo Level | Manual Input | Defines which RDS Geo Level to use during shipments pull |
| proft\_ctr\_id | Profit Center ID | Manual Input | Defines which Profit Center ID to use during shipments pull |
| proft\_ctr\_name | Profit Center Name | Manual Input | Defines which Profit Center Name to use during shipments pull |
| proft\_ctr\_lvl | Profit Center Level | Manual Input | Defines which Profit Center Level to use during shipments pull |
| rds\_prod\_hier\_lvl | RDS Product Hierarchy Level | Manual Input | Defines the Product Granularity for pulled shipments data |
| dirct\_filtr\_cond\_sql\_txt | Fact Filtering Condition | Manual Input | Used to define SQL conditions in filtering direct shipments data. |
| indir\_filtr\_cond\_sql\_txt | Fact Filtering Condition | Manual Input | Used to define SQL conditions in filtering indirect shipments data. |
| sys\_row\_ind | System Row Indicator |  | Row surrogate key |
| sys\_init\_actn\_id | System Initial Action ID |  | Load/Upload Action which added this row |
| sys\_last\_uplod\_actn\_id | System Last Upload Action ID |  | Identifier of most recent Upload Action of this row |

### Geo Mapping

**Description**

Geo Mapping is a Config File used to map Shipments to Market Group according to its geography.

Below are characteristics of Geo Mapping:

* Geo Mapping is exposed for downstream just like other Config Files
* all DQ checks for Geo Mapping are critical, automated submit happens when no validation fails
* Load Process uses submitted version of Geo Mapping
* there's no Invalid Row indicator nor Obsolete Row indicator

**List of Columns**

| **Techical Column Name** | **Column Label** | **Data Source** | **Description** |
| --- | --- | --- | --- |
| regn\_id | Region ID | RDS | Data pulled from RDS Geo Hierarchy 705 based on Profit Center & Market Mapping |
| regn\_name | Region Name | RDS | Data pulled from RDS Geo Hierarchy 705 based on Profit Center & Market Mapping |
| area\_id | Area ID | RDS | Data pulled from RDS Geo Hierarchy 705 based on Profit Center & Market Mapping |
| area\_name | Area Name | RDS | Data pulled from RDS Geo Hierarchy 705 based on Profit Center & Market Mapping |
| grp\_id | Group ID | RDS | Data pulled from RDS Geo Hierarchy 705 based on Profit Center & Market Mapping |
| grp\_name | Group Name | RDS | Data pulled from RDS Geo Hierarchy 705 based on Profit Center & Market Mapping |
| rptng\_cntry\_id | Reporting Country ID | RDS | Data pulled from RDS Geo Hierarchy 705 based on Profit Center & Market Mapping |
| rptng\_cntry\_name | Reporting Country Name | RDS | Data pulled from RDS Geo Hierarchy 705 based on Profit Center & Market Mapping |
| minor\_cntry\_id | Minor Country ID | RDS | Data pulled from RDS Geo Hierarchy 705 based on Profit Center & Market Mapping |
| minor\_cntry\_name | Minor Country Name | RDS | Data pulled from RDS Geo Hierarchy 705 based on Profit Center & Market Mapping |
| mkt\_grp\_name | Market Group | Manual Input | Custom CNOS&GC Geo Aggregation for Coefficients creation |
| custm\_regn\_name | Custom Region | Manual Input | Custom CNOS&GC Geo Aggregation for Coefficients creation |
| custm\_smo\_name | Custom SMO | Manual Input | Custom CNOS&GC Geo Aggregation for Coefficients creation |
| custm\_clstr\_name | Custom Cluster | Manual Input | Custom CNOS&GC Geo Aggregation for Coefficients creation |
| cntry\_lvl | Country Level | Manual Input | This is where user defines which GEO level should be pulled for coefficient creation |
| rds\_prod\_hier\_id | RDS Product Hierarchy ID | Manual Input | This is where user defines which RDS product hierarchy should be used for coefficients creation |
| rds\_prod\_hier\_lvl | RDS Product Hierarchy Level | PCM | Data pulled from Profit Center & Market Mapping |
| shpmt\_dirct\_ind | Shipments Direct (InfoPage) | Manual Input | This is where user defines if Direct Shipments should be used in coefficient creation |
| shpmt\_indir\_ind | Shipments Indirect (RTDC) | Manual Input | This is where user defines if Indirect Shipments should be used in coefficient creation |
| opt\_swtch\_dirct\_datetm | Optima Switch Date Dir | Manual Input | Start Date for Optima Data Pull for Direct Shipments |
| opt\_swtch\_indir\_datetm | Optima Switch Date Ind | Manual Input | Start Date for Optima Data Pull for Indirect Shipments |
| proft\_ctr\_id | Profit Center ID | PCM | Data pulled from Profit Center & Market Mapping |
| proft\_ctr\_lvl | Profit Center Level | PCM | Data pulled from Profit Center & Market Mapping |
| proft\_ctr\_name | Profit Center Name | PCM | Data pulled from Profit Center & Market Mapping |
| lc\_code | Local Currency | Manual Input | Defines the local currency of the GEO |
| srce\_sap\_crncy\_code | Source SAP Currency | Manual Input | Defines the local currency of the GEO in SAP |
| srce\_opt\_crncy\_code | Source OPTIMA Currency | Manual Input | Defines the local currency of the GEO in Optima |
| last\_mdfd\_datetm | Last Modified Date Time | Manual Input | Defines the local currency of the GEO in Optima |
| last\_mdfd\_by\_user\_name | Last Modified By User | Manual Input | Defines the local currency of the GEO in Optima |
| sys\_row\_id | System Row Indicator |  | Row surrogate key |
| sys\_init\_actn\_id | System Initial Action ID |  | Load/Upload Action which added this row |
| sys\_last\_uplod\_actn\_id | System Last Upload Action ID |  | Identifier of most recent Upload Action of this row |

## Source Files

RDS dimensions

### Profit Center Hierarchy

| **Technical Column Name** | **Column Label** | **Business Description** |
| --- | --- | --- |
| proft\_ctr\_hier\_id | Profit Center Hierarchy ID | Hierarchy ID of the Profit Center data to use |
| proft\_ctr\_hier\_eff\_date | Profit Center End Date | Start date of data's validity |
| proft\_ctr\_hier\_end\_date | Profit Center End Date | End date of data's validity |
| proft\_ctr\_orig\_lvl | Profit Center Level | Profit Center Hierarchy details |
| proft\_ctr\_1\_id | Profit Center ID 1 | Profit Center Hierarchy details (lvl 1) |
| proft\_ctr\_2\_id | Profit Center ID 2 | Profit Center Hierarchy details (lvl 2) |
| proft\_ctr\_3\_id | Profit Center ID 3 | Profit Center Hierarchy details (lvl 3) |
| proft\_ctr\_4\_id | Profit Center ID 4 | Profit Center Hierarchy details (lvl 4) |
| proft\_ctr\_5\_id | Profit Center ID 5 | Profit Center Hierarchy details (lvl 5) |
| proft\_ctr\_6\_id | Profit Center ID 6 | Profit Center Hierarchy details (lvl 6) |
| proft\_ctr\_7\_id | Profit Center ID 7 | Profit Center Hierarchy details (lvl 7) |
| proft\_ctr\_8\_id | Profit Center ID 8 | Profit Center Hierarchy details (lvl 8) |
| proft\_ctr\_9\_id | Profit Center ID 9 | Profit Center Hierarchy details (lvl 9) |
| proft\_ctr\_10\_id | Profit Center ID 10 | Profit Center Hierarchy details (lvl 10) |
| proft\_ctr\_11\_id | Profit Center ID 11 | Profit Center Hierarchy details (lvl 11) |

### Product Hierarchy

| **Technical Column Name** | **Column Label** | **Business Description** |
| --- | --- | --- |
| prod\_hier\_id | Product Hierarchy | Hierarchy ID of the Product data to use |
| prod\_hier\_eff\_date | Product Start Date | Start date of data's validity |
| prod\_hier\_end\_date | Product End Date | End date of data's validity |
| fpc\_id | FPC ID | Product Hierarchy details (FPC - lowest level) |
| brand\_id | Brand ID | Product Hierarchy details (Brand level) |
| brand\_name | Brand Name | Product Hierarchy details (Brand level) |
| categ\_id | Category ID | Product Hierarchy details (Category level) |
| categ\_name | Category Name | Product Hierarchy details (Category level) |
| sbstr\_id | Subsector ID | Product Hierarchy details (Subsector level) |
| sbstr\_name | Subsector Name | Product Hierarchy details (Subsector level) |

### Geography Hierarchy

| **Technical Column Name** | **Column Label** | **Business Description** |
| --- | --- | --- |
| geo\_hier\_id | Geo Hierarchy | Hierarchy ID of the Geo data to use |
| geo\_hier\_eff\_date | Geo Start Date | Start date of data's validity |
| geo\_hier\_end\_date | Geo End Date | End date of data's validity |
| geo\_orig\_lvl | Geo Level | Geo Hierarchy details |
| regn\_id | Region ID | Geo Hierarchy details (geo lvl 2) |
| regn\_name | Region Name | Geo Hierarchy details (geo lvl 2) |
| area\_id | Area ID | Geo Hierarchy details (geo lvl 3) |
| area\_name | Area Name | Geo Hierarchy details (geo lvl 3) |
| grp\_id | Group ID | Geo Hierarchy details (geo lvl 4) |
| grp\_name | Group Name | Geo Hierarchy details (geo lvl 4) |
| rptng\_cntry\_id | Reporting Country ID | Geo Hierarchy details (geo lvl 5) |
| rptng\_cntry\_id\_name | Reporting Country Name | Geo Hierarchy details (geo lvl 5) |
| minor\_cntry\_id | Minor Country ID | Geo Hierarchy details (geo lvl 6) |
| minor\_cntry\_name | Minor Country Name | Geo Hierarchy details (geo lvl 6) |

### Product Life Cycle

| **Technical Column Name** | **Column Label** | **Business Description** |
| --- | --- | --- |
| prod\_life\_cycle\_skid | Product Cycle SKID | Used for Technical purpose |
| curr\_ind |  |  |
| sdim\_load\_err\_ind |  |  |
| sfct\_load\_err\_ind |  |  |
| prod\_life\_cycle\_eff\_date |  |  |
| prod\_life\_cycle\_end\_date | Product Expiration date | Date on which Product becomes invalid |
| prod\_skid | PRODUCT SKID | Used for Technical purpose |
| life\_cycle\_scope\_id |  |  |
| lnch\_athrz\_code |  |  |
| stage\_id | Product Lifecycle stage | Current Active stage of Product |
| stage\_1\_date |  |  |
| stage\_2\_date |  |  |
| stage\_3\_date |  |  |
| stage\_4\_date |  |  |
| stage\_5\_date |  |  |
| stage\_6\_date |  |  |
| stage\_7\_date |  |  |
| vol\_ship\_unit\_name |  |  |
| last\_ship\_date |  |  |
| rcd\_type\_code |  |  |
| iso\_cntry\_code | Country ID | Geo the Product corresponds to. |
| mrkt\_sgmnt\_code |  |  |

### Customer Hierarchy

| **Technical Column Name** | **Column Label** | **Business Description** |
| --- | --- | --- |
| cust\_id | Customer ID | Corporate Customer ID |
| cust\_name | Customer Name | Corporate Customer Name |
| cust\_orig\_lvl |  |  |
| cust\_1\_id | Customer Level 1 ID | Corporate Level 1 Customer ID |
| cust\_1\_name | Customer Level 1 Name | Corporate Level 1 Customer Name |
| cust\_2\_id | Customer Level 2 ID | Corporate Level 2 Customer ID |
| cust\_2\_name | Customer Level 2 Name | Corporate Level 2 Customer Name |
| cust\_2\_long\_name |  |  |
| cust\_3\_id | Customer Level 3 ID | Corporate Level 3 Customer ID |
| cust\_3\_name | Customer Level 3 Name | Corporate Level 3 Customer Name |
| cust\_3\_long\_name |  |  |
| cust\_4\_id | Customer Level 4 ID | Corporate Level 4 Customer ID |
| cust\_4\_name | Customer Level 4 Name | Corporate Level 4 Customer Name |
| cust\_4\_long\_name |  |  |
| cust\_5\_id | Customer Level 5 ID | Corporate Level 5 Customer ID |
| cust\_5\_name | Customer Level 5 Name | Corporate Level 5 Customer Name |
| cust\_5\_long\_name |  |  |
| cust\_6\_id | Customer Level 6 ID | Corporate Level 6 Customer ID |
| cust\_6\_name | Customer Level 6 Name | Corporate Level 6 Customer Name |
| cust\_6\_long\_name |  |  |
| cust\_7\_id | Customer Level 7 ID | Corporate Level 7 Customer ID |
| cust\_7\_name | Customer Level 7 Name | Corporate Level 7 Customer Name |
| cust\_7\_long\_name |  |  |
| cust\_8\_id | Customer Level 8 ID | Corporate Level 8 Customer ID |
| cust\_8\_name | Customer Level 8 Name | Corporate Level 8 Customer Name |
| cust\_8\_long\_name |  |  |
| cust\_9\_id | Customer Level 9 ID | Corporate Level 9 Customer ID |
| cust\_9\_name | Customer Level 9 Name | Corporate Level 9 Customer Name |
| cust\_9\_long\_name |  |  |
| cust\_10\_id | Customer Level 10 ID | Corporate Level 10 Customer ID |
| cust\_10\_name | Customer Level 10 Name | Corporate Level 10 Customer Name |
| cust\_10\_long\_name |  |  |
| cust\_1\_long\_name |  |  |
| cust\_11\_id | Customer Level 11 ID | Corporate Level 11 Customer ID |
| cust\_11\_name | Customer Level 11 Name | Corporate Level 11 Customer Name |
| cust\_11\_long\_name |  |  |
| cust\_12\_id | Customer Level 12 ID | Corporate Level 12 Customer ID |
| cust\_12\_name | Customer Level 12 Name | Corporate Level 12 Customer Name |
| cust\_12\_long\_name |  |  |

### Customer Mapping

Note: Columns with asterisk(\*) are required for Regional Customer Mapping

| **Technical Column Name** | **Column Label** | **Region** | **Business Description** |
| --- | --- | --- | --- |
| attr\_1\_val | Attribute 1 Value | ALL |  |
| attr\_2\_val | Attribute 2 Value | ALL |  |
| attr\_3\_val | Attribute 3 Value | ALL |  |
| attr\_4\_val | Attribute 4 Value | ALL |  |
| chain\_name | Chain Name | ALL | Customer details |
| chanl\_val | Channel Name\*\* | ALL | Customer details |
| clstr\_name | Cluster Name | ALL | Cluster details for which the customer belongs |
| cmmnt | Comment | ALL |  |
| corp\_cust\_l9\_id | Corp Cust 898 Lvl 9 Id | All | Shipment Customer details |
| corp\_cust\_l9\_name | Corp Cust 898 Lvl 9 Name | All | Shipment Customer details |
| cust\_lvl | Customer Level | All | Customer details |
| custm\_smo\_name | Custom SMO | ALL | Custom CNOS&GC Geo Aggregation for Coefficients creation |
| geo\_id | Geo ID\*\* | ALL | Geography details of the customer (levels 4 to 6) |
| geo\_name | Geo Name\*\* | ALL | Geography details of the customer (levels 4 to 6) |
| minor\_cntry\_id | Minor Country ID | ALL | Geography details of the customer (level 6) |
| mnr\_cust\_mapng\_dim\_skid | MNR Customer Mapping Surrogate Key | ALL |  |
| opt\_id | Optima ID | ALL | Optima details of the customer |
| opt\_name | Optima Name | ALL | Optima details of the customer |
| org\_id | Organization ID | ALL |  |
| org\_name | Organization Name | ALL |  |
| outpt\_type | Output Type | ALL |  |
| proft\_ctr\_id | Profit Center ID | ALL |  |
| proft\_ctr\_lvl | Profit Center Level Type | ALL |  |
| proft\_ctr\_name | Profit Center Name | ALL |  |
| regn\_name | Region | ALL | Geography details of the customer |
| regnl\_cust\_name | Regional Customer Name | ALL | Geography details of the customer |
| rptng\_cntry\_id | Minor Country ID | ALL | Geography details of the customer (level 5) |
| rptng\_cust\_l1\_id | Reporting Customer Level 1 ID | ALL | Customer details |
| rptng\_cust\_l1\_name | Reporting Customer Level 1 Name | ALL | Customer details |
| rptng\_cust\_l2\_id | Reporting Customer Level 2 ID | ALL | Customer details |
| rptng\_cust\_l2\_name | Reporting Customer Level 2 Name | ALL | Customer details |
| rptng\_cust\_l3\_id | Reporting Customer Level 3 ID | ALL | Customer details |
| rptng\_cust\_l3\_name | Reporting Customer Level 3 Name | ALL | Customer details |
| rptng\_cust\_l4\_id | Reporting Customer Level 4 ID | ALL | Customer details |
| rptng\_cust\_l4\_name | Reporting Customer Level 4 Name | ALL | Customer details |
| smo\_name | SMO Name | ALL | SMO details for the product |
| src\_cust\_id | Source Customer ID | ALL | Customer Details |
| src\_cust\_name | Source Customer Name | ALL | Customer Details |
| src\_dirct\_val | Source Direct Value | ALL |  |
| src\_hybrd\_val | Source Hybdrid Value | ALL |  |
| src\_indir\_val | Source Indirect Value | ALL |  |
| srce\_cust\_id | Source Customer ID | ALL | Customer details |
| srce\_cust\_name | Source Customer Name | ALL | Customer details |
| srce\_dirct\_val | Source Direct Value | ALL |  |
| srce\_hybrd\_val | Source Hybdrid Value | ALL |  |
| srce\_indir\_val | Source Indirect Value | ALL |  |
| srce\_sys\_id | Source System ID | ALL | Customer details |
| user\_cmmnt | User Comment | ALL | Customer details |

### Product Mapping

**Product Mapping**

| **Technical Column Name** | **Column Label** | **Business Description** |
| --- | --- | --- |
| smo\_name | SMO | SMO details for the product |
| custm\_regn\_name | Custom Region Name | Custom CNOS&GC Geo Aggregation for Coefficients creation |
| custm\_smo\_name | Custom SMO | Custom CNOS&GC Geo Aggregation for Coefficients creation |
| custm\_clstr\_name | Custom Cluster | Custom CNOS&GC Geo Aggregation for Coefficients creation |
| geo\_id | Geo ID | Geography details of the product |
| geo\_name | Geo Nam | Market details of the product |
| sbstr\_id | Subsector ID | Product Hierarchy data (Subsector level) |
| sbstr\_name | Subsector Name | Product Hierarchy data (Subsector level) |
| categ\_id | Category ID | Product Hierarchy data (Category level) |
| categ\_name | Category Name | Product Hierarchy data (Category level) |
| brand\_id | Brand ID | Product Hierarchy data (Brand level) |
| brand\_name | Brand Name | Product Hierarchy data (Brand level) |
| excld\_ind | Excluded | Indicator to identify which products will be excluded in load process |
| sys\_last\_mdfd\_datetm | Last Modified date | System Generated |
| sys\_last\_mdfd\_user\_name | Last Modified by | System Generated |

### Direct Shipments for all regions

**Source Table:** stage\_common.ship\_step\_100\_sode\_hist\_star\_work\_tc\_lc

| **Column Name** | **Data Type** |
| --- | --- |
| MTH\_NUM | decimal(38,10) |
| CUST\_1\_ID | string |
| CUST\_2\_ID | string |
| CUST\_3\_ID | string |
| CUST\_4\_ID | string |
| CUST\_5\_ID | string |
| CUST\_6\_ID | string |
| CUST\_7\_ID | string |
| CUST\_8\_ID | string |
| CUST\_9\_ID | string |
| PROD\_CSU\_TYPE\_CODE | string |
| FCT\_CUST\_ID | string |
| CORP\_OFFCL\_SHIP\_FLAG | string |
| FREE\_SHIP\_FLAG | string |
| POST\_DIVST\_SHIP\_FLAG | string |
| PROD\_ID | string |
| PROD\_2\_ID | string |
| PROD\_2\_NAME | string |
| PROD\_3\_ID | string |
| PROD\_3\_NAME | string |
| PROD\_4\_ID | string |
| PROD\_4\_NAME | string |
| PROD\_5\_ID | string |
| PROD\_5\_NAME | string |
| PROD\_6\_ID | string |
| PROD\_6\_NAME | string |
| PROD\_7\_ID | string |
| PROD\_7\_NAME | string |
| PROD\_8\_ID | string |
| PROD\_8\_NAME | string |
| PROD\_12\_ID | string |
| PROD\_12\_NAME | string |
| PROD\_2\_ID\_5801 | string |
| PROD\_3\_ID\_5801 | string |
| PROD\_4\_ID\_5801 | string |
| PROD\_5\_ID\_5801 | string |
| PROD\_6\_ID\_5801 | string |
| PROD\_7\_ID\_5801 | string |
| PROD\_8\_ID\_5801 | string |
| PROD\_2\_NAME\_5801 | string |
| PROD\_3\_NAME\_5801 | string |
| PROD\_4\_NAME\_5801 | string |
| PROD\_5\_NAME\_5801 | string |
| PROD\_6\_NAME\_5801 | string |
| PROD\_7\_NAME\_5801 | string |
| PROD\_8\_NAME\_5801 | string |
| GEO\_5\_ID | string |
| GEO\_5\_NAME | string |
| GEO\_6\_ID | string |
| GEO\_6\_NAME | string |
| CRNCY\_ID | string |
| ORG\_ID | string |
| MTH\_FISC\_PERD\_NUM | decimal(38,10) |
| FISC\_YR\_SHORT\_ABBR\_NAME | string |
| MTH\_START\_DATE | date |
| HY\_CODE | string |
| QTR\_CODE | string |
| MTH\_OF\_YR\_NAME | string |
| PROFT\_CTR\_2\_ID | string |
| PROFT\_CTR\_3\_ID | string |
| PROFT\_CTR\_4\_ID | string |
| STAT\_UNIT\_QTY | double |
| GROSS\_TC\_AMT | double |
| NET\_TC\_AMT | double |
| SYS\_INSERTED\_DT | timestamp |
| REGN\_NAME | string |
| SMO\_NAME | string |
| LOCAL\_CRNCY\_CODE | string |
| GROSS\_LC\_AMT | double |
| NET\_LC\_AMT | double |

### Indirect Shipments for AMA

**Source Table:** stage\_common.an\_ind\_ship\_step\_100\_rtdc\_hist\_star

| **Column Name** | **Data Type** |
| --- | --- |
| MTH\_NUM | date |
| MTH\_START\_DATE | date |
| FISC\_YR\_SHORT\_ABBR\_NAME | string |
| CUST\_1\_ID | string |
| CUST\_2\_ID | string |
| CUST\_3\_ID | string |
| CUST\_4\_ID | string |
| CUST\_5\_ID | string |
| CUST\_6\_ID | string |
| CUST\_7\_ID | string |
| CUST\_8\_ID | string |
| CUST\_9\_ID | string |
| CHAIN\_NAME | string |
| ORG\_ID | string |
| PROD\_4\_H5005\_ID | string |
| PROD\_4\_H5005\_NAME | string |
| PROD\_5\_H5005\_ID | string |
| PROD\_5\_H5005\_NAME | string |
| PROD\_6\_H5005\_ID | string |
| PROD\_6\_H5005\_NAME | string |
| PROD\_7\_H5005\_ID | string |
| PROD\_7\_H5005\_NAME | string |
| PROD\_8\_H5005\_ID | string |
| PROD\_8\_H5005\_NAME | string |
| PROD\_12\_H5005\_ID | string |
| PROD\_12\_H5005\_NAME | string |
| PROD\_3\_H5801\_ID | string |
| PROD\_3\_H5801\_NAME | string |
| PROD\_4\_H5801\_ID | string |
| PROD\_4\_H5801\_NAME | string |
| PROD\_5\_H5801\_ID | string |
| PROD\_5\_H5801\_NAME | string |
| PROD\_6\_H5801\_ID | string |
| PROD\_6\_H5801\_NAME | string |
| PROD\_7\_H5801\_ID | string |
| PROD\_7\_H5801\_NAME | string |
| PROD\_8\_H5801\_ID | string |
| PROD\_8\_H5801\_NAME | string |
| GEO\_5\_ID | string |
| GEO\_5\_NAME | string |
| GEO\_6\_ID | string |
| GEO\_6\_NAME | string |
| CRNCY\_ID | string |
| prod\_csu\_type\_code | string |
| SU\_VOL\_QTY | decimal(38,10) |
| GROSS\_VAL\_TC\_AMT | decimal(38,10) |
| NET\_VAL\_TC\_AMT | decimal(38,10) |
| SYS\_INSERTED\_DT | timestamp |

### Indirect Shipments for Europe

**Source Table:** desc stage\_europe.eu\_ind\_sh

| **Column Name** | **Data Type** |
| --- | --- |
| MONTH\_ID | string |
| QUARTER | string |
| SEMESTER | string |
| FY\_ID | string |
| GEO\_ID | string |
| GEO\_NAME | string |
| CUST\_3\_ID | string |
| CUST\_4\_ID | string |
| CUST\_5\_ID | string |
| CUST\_6\_ID | string |
| CUST\_7\_ID | string |
| CUST\_8\_ID | string |
| CUST\_9\_ID | string |
| CUST\_10\_ID | string |
| CUST\_11\_ID | string |
| CUST\_12\_ID | string |
| SECTOR\_ID | string |
| SECTOR\_NAME | string |
| SUBSECTOR\_ID | string |
| SUBSECTOR\_NAME | string |
| CATEGORY\_ID | string |
| CATEGORY\_NAME | string |
| BRAND\_ID | string |
| BRAND\_NAME | string |
| SEGMENT\_ID | string |
| SEGMENT\_NAME | string |
| FORM\_ID | string |
| FORM\_NAME | string |
| FPC\_ID | string |
| FPC\_NAME | string |
| CRNCY\_CODE | string |
| VOL | double |
| GIV | double |
| NIV | double |

## Load Process Cube

The table below provides details about the Load Process Cube which is product of joining of all sources during Load Process.  
This data set varies across regions.

| **Technical Column Name** | **Column Name** | **Region** | **Dimension** | **Key** | **Comments** |
| --- | --- | --- | --- | --- | --- |
| actl\_ind | Set to Y if product Month is under actuals in calendar scope table | All | Technical | Yes | Custom column from Uploader processing |
| area\_id | Area ID | All | Geography | Yes | Data pulled from Geo Mapping |
| area\_name | Area Name | All | Geography | No | Data pulled from Geo Mapping |
| brand\_id | Brand ID | All | Product | Yes | Data pulled from RDS Product Hierarchy |
| brand\_name | Brand Name | All | Product | No | Data pulled from RDS Product Hierarchy |
| categ\_id | Category ID | All | Product | Yes | Data pulled from RDS Product Hierarchy |
| categ\_name | Category Name | All | Product | No | Data pulled from RDS Product Hierarchy |
| chanl\_val | Channel | AMA | Customer | No | Data pulled from Customer Mapping |
| cust\_id | Customer ID | All | Customer | No | Data pulled from Indirect/Direct Shipments depending on the customer type. Customer level 9 is pulled for LA\*\* |
| custm\_clstr\_name | Custom Cluster | All | Geography | No | Data pulled from Geo Mapping |
| custm\_regn\_name | Custom Region | All | Geography | No | Data pulled from Geo Mapping |
| custm\_smo\_name | Custom SMO | All | Geography | No | Data pulled from Geo Mapping |
| dirct\_indir\_ind | Indicates Direct/Indirect | All | Technical | Yes | Custom column from Uploader processing |
| fpc\_id | FPC | All | Product | Yes | Data pulled from Indirect/Direct Shipments depending on the customer type |
| frcst\_ind | Set to Y if product Month is under forecast in calendar scope table | All | Technical | Yes | Custom column from Uploader processing |
| fy\_code | FY | All | Time | Yes | Data pulled from Indirect/Direct Shipments depending on the customer type |
| grp\_id | Group ID | All | Geography | Yes | Data pulled from Geo Mapping |
| grp\_name | Group Name | All | Geography | No | Data pulled from Geo Mapping |
| histr\_ind | Set to Y if product is historical in PLC | All | Technical | Yes | Custom column from Uploader processing |
| minor\_cntry\_id | Minor Country ID | All | Geography | Yes | Data pulled from Indirect/Direct Shipments depending on the customer type |
| minor\_cntry\_name | Minor Country Name | All | Geography | No | Data pulled from Indirect/Direct Shipments depending on the customer type |
| mkt\_grp\_name | Market Group | All | Geography | Yes | Custom CNOS&GC Geo Aggregation for Coefficients creation |
| mkt\_geo\_id | Market Geo ID | All | Geography | No | Data pulled from Geo Mapping |
| mkt\_lvl | Market Level | All | Geography | No | Data pulled from Geo Mapping |
| mkt\_name | Market | All | Geography | Yes | Geography name related with level defined in Geo Mapping by cntry\_lvl |
| mth\_num | Month | All | Time | Yes | Data pulled from Indirect/Direct Shipments depending on the customer type |
| proft\_ctr\_id | Profit Center ID | LA, AMA | Profit center | No | Data pulled from Shipments |
| proft\_ctr\_lvl | Profit Center Level | All | Profit center | No | Data pulled from Geo Mapping |
| proft\_ctr\_name | Profit Center Name | All | Profit center | No | Data pulled from Geo Mapping |
| rds\_prod\_hier\_id | RDS Product Hierarchy ID | All | Geography | No | Data pulled from Geo Mapping |
| rds\_prod\_hier\_lvl | RDS Product Hierarchy Level | All | Geography | No | Data pulled from Geo Mapping |
| regn\_id | Region ID | All | Geography | Yes | Data pulled from Indirect/Direct Shipments depending on the customer type |
| regn\_name | Region Name | All | Geography | No | Data pulled from Geo Mapping |
| rptng\_cntry\_id | Reporting Country ID | All | Geography | Yes | Data pulled from Geo Mapping |
| rptng\_cntry\_name | Reporting Country Name | All | Geography | No | Data pulled from Geo Mapping |
| rptng\_cust\_id | Reporting Customer ID | All | Customer | No | Data pulled from Customer Mapping |
| rptng\_cust\_name | Reporting Customer Name | All | Customer | Yes | Data pulled from Customer Mapping |
| sbstr\_id | Subsector ID | All | Product | Yes | Data pulled from RDS Product Hierarchy |
| sbstr\_name | Subsector Name | All | Product | No | Data pulled from RDS Product Hierarchy |
| shpmt\_dirct\_ind | Shipments Direct (Infopage) | All | Geography | No | Data pulled from Geo Mapping |
| shpmt\_ind | Set to Y if there was any shipments | All | Technical | Yes | Custom column from Uploader processing |
| shpmt\_indir\_ind | Shipments Indirect (RTDC) | All | Geography | No | Data pulled from Geo Mapping |

## DB tables

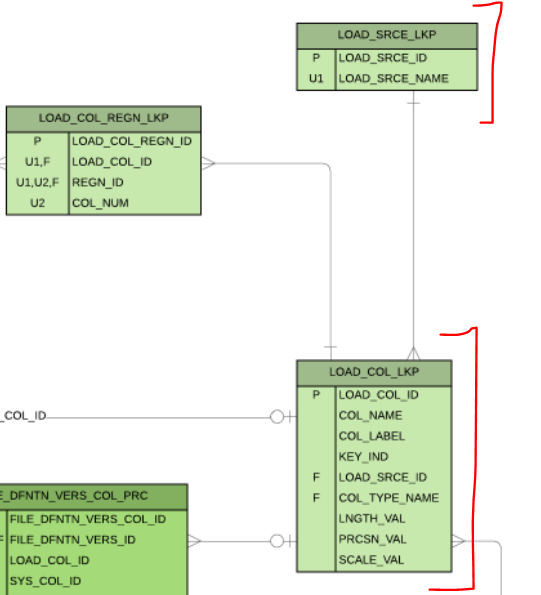


Figure: Entity diagram: DB Model

# DQ VALIDATION

## Data Validation for Input Files

For each template during creation there are created basic conditions on custom columns according to column type and by default V01 (all load columns) and V02 (all required and visible columns) conditions. Check process is triggered by pro\_file\_vldtn\_joint\_exct procedure which gathers all registers DQ checks and run them in session.

When template is created mentioned V01 and V02 are created by procedure pro\_dq\_bus\_check\_tnsfr

## Basic validations:

During each upload below data quality checks are performed on uploaded content if registered for template:

## Boolean check

Procedure checks if values provided are Y/N using pro\_file\_vldtn\_data\_type\_bool

## Date

Procedure checks if values provided are correct date format as here: Data types using pro\_file\_vldtn\_data\_type\_date

## Integer

Procedure checks if values provided are integer type using pro\_file\_vldtn\_data\_type\_int

## Month

Procedure checks if values provided are correct date format as here: Data types using pro\_file\_vldtn\_data\_type\_mth

## Number

Procedure checks if values provided are correct date format as here: Data types using pro\_file\_vldtn\_data\_type\_num

## Percent

Procedure checks if values provided are correct date format as here: Data types using pro\_file\_vldtn\_data\_type\_perc

## Text

Procedure checks if values provided are text format with length set using pro\_file\_vldtn\_data\_type\_txt

## Business validations:

### V01 - Duplicate Check

This will check if there are duplicate entries for the defined set of columns. The scope of this check applies to all data in Input File including those that are marked as invalid and obsolete.

* **Failure stops Upload:** yes
* **Failure stops Submission:** yes
* **Email report:** summary
* **Excel report:** not provided

### V02 - Nulls upon Submit

This will check if there are null values for a given column upon upload. The scope of this check only applies to the uploaded data which are not marked as invalid.

* **Failure stops Upload:** no
* **Failure stops Submission:** yes
* **Email report:** summary
* **Excel report:** failing rows

### V03 - Is the data value within defined threshold

This will check if the value that user inserts on a percent type column is within the defined threshold for the selected Input File. The scope of this check only applies to the uploaded rows.

* **Failure stops Upload:** no
* **Failure stops Submission:** yes
* **Email report:** summary
* **Excel report:** failing rows

### V04 - Check if new month appeared

This will check if at least single record for the reporting month is present in an Input File.

* **Failure stops Upload:** yes
* **Failure stops Submission:** yes
* **Email report:** detailed report
* **Excel report:** not provided

### V05 - Invalid row has entered data

This will check if there are rows marked as invalid that contains non-zero value on non-load input columns. This check **does not** consider rows that are marked as obsolete and the scope of this check only applies to the uploaded data.

* **Failure stops Upload:** no
* **Failure stops Submission:** no
* **Email report:** summary
* **Excel report:** failing rows

### V06 - All rows are Invalid

This will check if there are months that only contains invalid rows. This check **does not** consider data for future months for actuals and those that are marked as obsolete.

* **Failure stops Upload:** no
* **Failure stops Submission:** no
* **Email report:** summary
* **Excel report:** provided via email

### V07 - All rows are Invalid - grouping

This will check if there are rows marked as invalid that contains data on provided set of columns. This check **does not** consider rows that are marked as obsolete and the scope of this check applies to all rows in selected Input File

* **Failure stops Upload:** no
* **Failure stops Submission:** no
* **Email report:** summary
* **Excel report:** specific query result

### V08 - “All others” checks (Total Compare)

This will compare the sum of all "TOTAL" customer against the sum of all other customers depending on the chosen granularity and its output report will show the data that have mismatch. The scope of this check applies to valid and not obsolete data in Input File.

* **Failure stops Upload:** no
* **Failure stops Submission:** no
* **Email report:** summary
* **Excel report:** not provided

### V09 - Delta vs previous submit

This type of validation will compare all data within the Delta scope (PFY (Optional with PFY Data Freeze) + CFY till reporting month-1) versus the previously submitted data. The output report will show rows that are changed from the last publication cycle. This check will not consider rows that are marked as obsolete. It is also important to note that this check will be excluded on the first publication of CFY (September cycle).

* **Failure stops Upload:** no
* **Failure stops Submission:** no
* **Email report:** summary
* **Excel report:** failing rows

## Data Validation for Config Files

### CV01 - Proper geo\_ids with levels for Geo hierarchy

This will check if all GEO\_ID inputs with corresponding levels in Profit Center and Market Mapping belongs to RDS Geo Hierarchy.

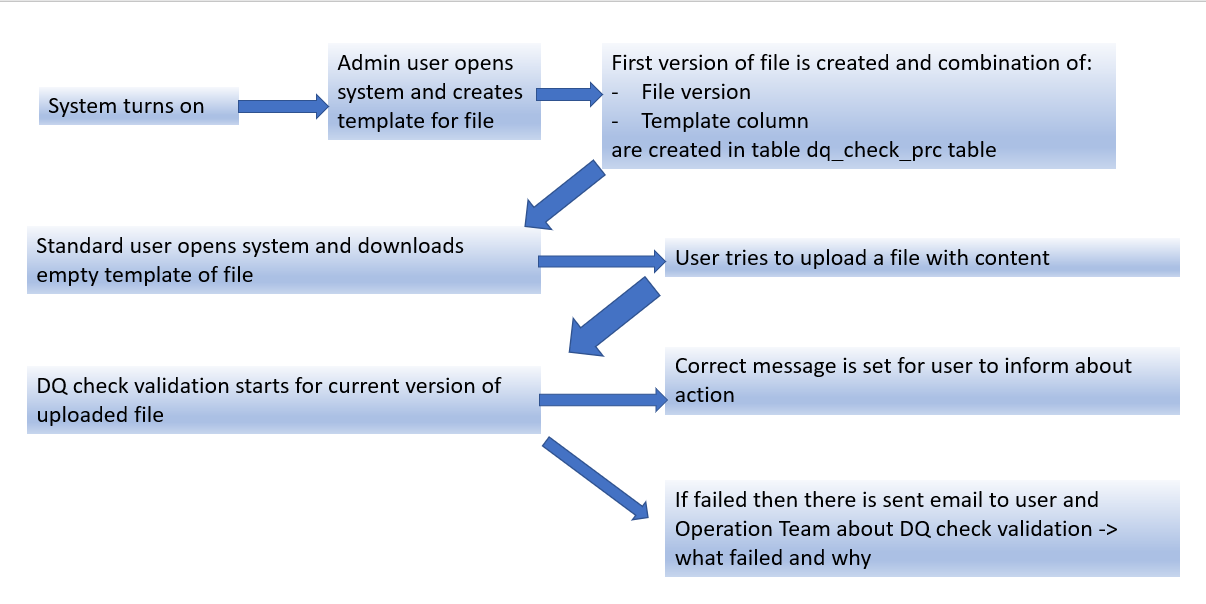
* **Failure stops Upload:** yes
* **Failure stops Submission:** yes
* **Email report:** detailed report
* **Excel report:** not provided

### CV02 - Proper proft\_ctr\_ids with levels for Profit Center hierarchy

This will check if all Profit Center ID inputs with corresponding levels in Profit Center and Market Mapping belongs to RDS Profit Center Hierarchy.

* **Failure stops Upload:** yes
* **Failure stops Submission:** yes
* **Email report:** detailed report
* **Excel report:** not provided

## DQ check steps

****

## Sample DQ report emails

### Sample email for failed Profit Center and Market Mapping upload:

**email subject**:  
CNOS & GC Uploader - upload validation report - Profit Center and Market Mapping - FAILED UPLOAD

### Sample email for upload failed on data type constraints:

**email subject**:  
CNOS & GC Uploader - upload validation report - EUROPE - France - TT Direct - FRANCE - FAILED UPLOAD

### Sample email for upload failed on business validations:

**email subject**:  
CNOS & GC Uploader - upload validation report - EUROPE - France - TT Direct - FRANCE - FAILED UPLOAD

### Sample email for failed submit:

**email subject**:  
CNOS & GC Uploader - upload validation report - EUROPE - France - TT Direct - FRANCE - FAILED SUBMIT

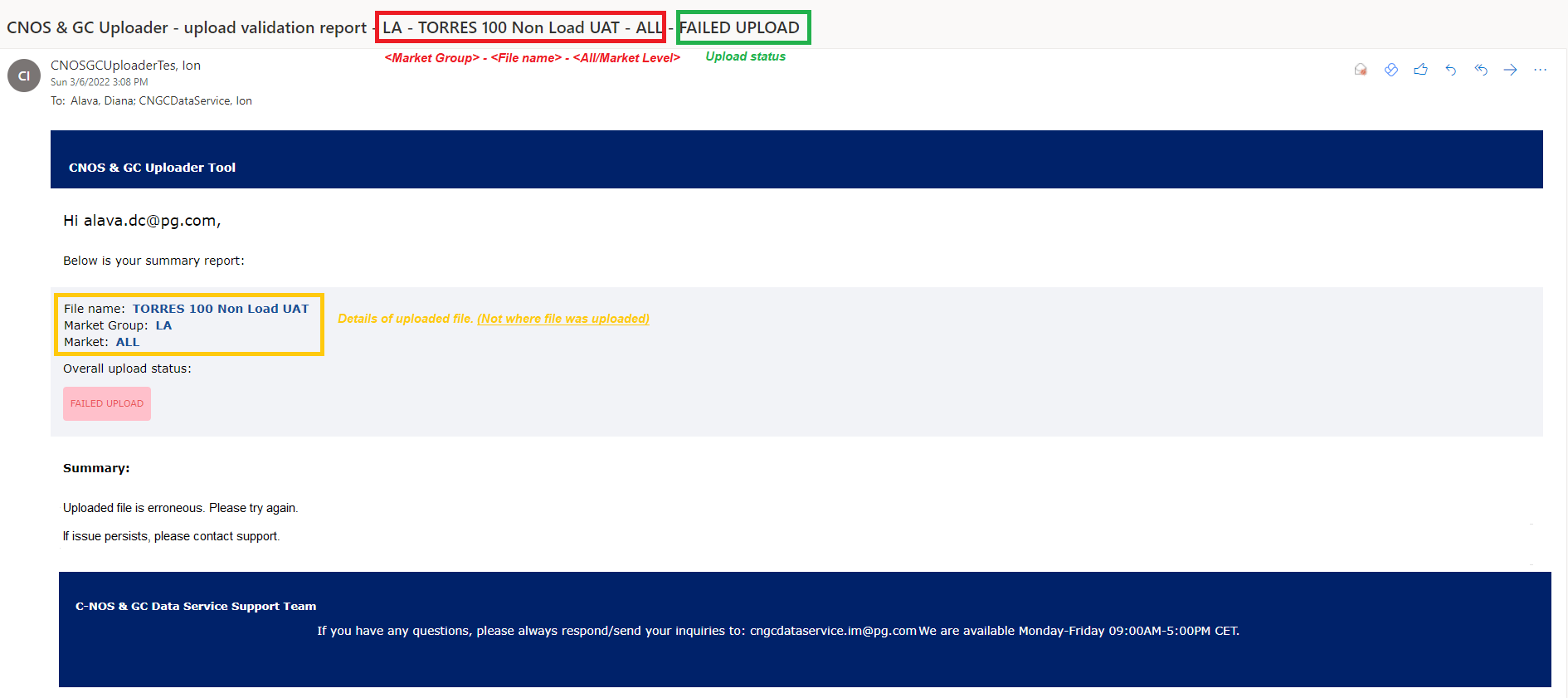
### Sample email for successful submit:

**email subject**:  
CNOS & GC Uploader - upload validation report - EUROPE - France - TT Direct - FRANCE - SUBMITTED

**General Sample Email Notification for Failed Uploaded File Issue**

The following email notification is sent to the user when there are issues with the uploaded Excel file not relating to DQ or data validations. E.g.

* Upload file mismatch; and
* Excel file formatting issues.



**Sample email - FAILED UPLOAD for Profit Centre and Market Mapping**

**File name:** **Profit Center and Market Mapping**

Overall upload status: FAILED UPLOAD

**Summary:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Code** | **Title** | **Status** | **Data** |
| CV01 | Geography ID and level check | Failed | XLSX file |
| CV02 | Profit Center ID and level check | Failed | XLSX file |

Details of failed validations:

CV01: Geography ID and level check

Number of rows with unrecognised ID: 5  
Number of rows with wrong level: 5  
Total number of affected rows: 5  
Detailed report: XLSX file

CV02: Profit Center ID and level check

Number of rows with unrecognised ID: 5  
Number of rows with wrong level: 5  
Total number of affected rows: 5  
Detailed report: XLSX file

**Sample email - FAILED UPLOAD already on datatypes**

Market Group: **EUROPE - France**

File name: **TT Direct**

Market: **FRANCE**

Overall upload status: FAILED UPLOAD

**Summary:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Code** | **Title** | **Status** | **Data** |
| INTEGER | Wrong values for column E1 | Failed | XLSX file |
| INTEGER | Wrong values for column E3 | Failed | XLSX file |
| DATE | Wrong values for column D4 | Failed | XLSX file |

**Details of failed validations:**

**INTEGER: Wrong values for column E1**

Total number of affected rows: 5  
Detailed report: XLSX file

**INTEGER: Wrong values for column E3**

Total number of affected rows: 5  
Detailed report: XLSX file

**DATE: Wrong values for column D4**

Total number of affected rows: 5  
Detailed report: XLSX file

**Sample email - FAILED UPLOAD on business validation**

**Market Group: EUROPE - France**

**File name: TT Direct**

**Market: FRANCE**

**Overall upload status: FAILED UPLOAD**

**Summary:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Code** | **Title** | **Status** | **Data** |
| V01 | Duplicates on key columns | Passed |  |
| V01 | Duplicates on other set of columns | Failed | XLSX file |
| V02 | NULLs upon submit for column C1 | Passed |  |
| V02 | NULLs upon submit for column C4 | Passed |  |
| V02 | NULLs upon submit for column C7 | Passed |  |
| V03 | Data within threshold for column D1 | Passed |  |
| V03 | Data within threshold for column D3 | Passed |  |
| V04 | New month appeared for column D1 on G1 | Failed | XLSX file |
| V05 | Data entered in Invalid Rows | Passed |  |
| V06 | All rows are Invalid | Passed |  |
| V07 | All rows are Invalid for Customer related columns | Passed |  |
| V07 | All rows are Invalid for Some other set of columns | Passed |  |
| V08 | All others | Passed |  |
| V09 | Delta vs previous submit | Passed |  |

Details of failed validations:

V01: Duplicates on other set of columns

Check columns: C1, C2, C3  
Reporting columns: D1, D2, D3  
Number of duplicates: 2  
Total number of affected rows: 5  
Detailed report: XLSX file

V04: New month appeared for column D1 on G1

Grouping column: G1  
Month column: D1  
Number of groupings without new month: 2  
Detailed report: XLSX file

**Sample email - FAILED SUBMIT**

**email content**:

Market Group: **EUROPE - France**

File name: **TT Direct**

Market: **FRANCE**

Overall upload status: FAILED SUBMIT

Summary:

|  |  |  |  |
| --- | --- | --- | --- |
| **Code** | **Title** | **Status** | **Data** |
| V01 | Duplicates on key columns | Passed |  |
| V01 | Duplicates on other set of columns | Passed |  |
| V02 | NULLs upon submit for column C1 | Failed | XLSX file |
| V02 | NULLs upon submit for column C4 | Passed |  |
| V02 | NULLs upon submit for column C7 | Passed |  |
| V03 | Data within threshold for column D1 | Passed |  |
| V03 | Data within threshold for column D3 | Failed | XLSX file |
| V04 | New month appeared for column D1 on G1 | Passed |  |
| V05 | Data entered in Invalid Rows | Passed |  |
| V06 | All rows are Invalid | Passed |  |
| V07 | All rows are Invalid for Customer related columns | Passed |  |
| V07 | All rows are Invalid for Some other set of columns | Passed |  |
| V08 | All others | Passed |  |
| V09 | Delta vs previous submit | Passed |  |

**Details of failed validations:**

**V02: NULLs upon submit for column C1**

Total number of affected rows: 5  
Detailed report: XLSX file

**V03: Data within threshold for column D3**

Total number of affected rows: 5  
Detailed report: XLSX file

**Sample email – SUBMITTED.**

**Market Group: EUROPE - France**

**File name: TT Direct**

**Market: FRANCE**

**Overall upload status: SUBMITTED**

**Summary:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Code** | **Title** | **Status** | **Data** |
| V01 | Duplicates on key columns | Passed |  |
| V01 | Duplicates on other set of columns | Passed |  |
| V02 | NULLs upon submit for column C1 | Passed |  |
| V02 | NULLs upon submit for column C4 | Passed |  |
| V02 | NULLs upon submit for column C7 | Passed |  |
| V03 | Data within threshold for column D1 | Passed |  |
| V03 | Data within threshold for column D3 | Passed |  |
| V04 | New month appeared for column D1 on G1 | Passed |  |
| V05 | Data entered in Invalid Rows | Non-critical failure | XLSX file |
| V06 | All rows are Invalid | Non-critical failure |  |
| V07 | All rows are Invalid for Customer related columns | Non-critical failure | XLSX file |
| V07 | All rows are Invalid for Some other set of columns | Passed |  |
| V08 | All others | Non-critical failure | XLSX file |
| V09 | Delta vs previous submit | Rows reported | XLSX file |

## Details of failed validations:

### V05: Data entered in Invalid Rows

Total number of affected rows: 5  
Detailed report: XLSX file

### V07: All rows are Invalid for Customer related columns

Grouping columns: C1, C2, C3  
Total number of affected rows: 5  
Detailed report: XLSX file

### V08: All others

Total number of affected rows: 5  
Detailed report: XLSX file

### V09: Delta vs previous submit

Total number of affected rows: 43  
Detailed report: XLSX file

**DQ Check Front End JSON format**

DQ check conditions set for template in the Front End will be send to Back End via JSON format file. As we have static DQ checks, there will be a JSON file created by FE and send as one parameter.

JSON structure will be as below (to be discussed with FE Team):

{

"DUPLICATES":

{

"check\_columns":

[

"mkt\_id",

"mkt\_name",

"col3"

],

"report\_columns":

[

"mkt\_name",

"categ\_id",

"mth\_num"

]

},

"NULLS":

{

"check\_columns":

[

"NULL1",

"NULL2",

"NULL3"

],

"report\_columns":

[

"NULL10",

"NULL20",

"NULL30"

]

},

"NewMthChk":

{

"check\_columns":

[

"smo\_name",

"creat\_date"

]

}

}

CODE FOR TESTING (if you need use it):

begin

declare

@json\_example varchar(max);

SET @json\_example = '{

"DUPLICATES":

{

"check\_columns":

[

"mkt\_id",

"mkt\_name",

"col3"

],

"report\_columns":

[

"mkt\_name",

"categ\_id",

"mth\_num"

]

},

"NULLS":

{

"check\_columns":

[

"NULL1",

"NULL2",

"NULL3"

],

"report\_columns":

[

"NULL10",

"NULL20",

"NULL30"

]

},

"NewMthChk":

{

"check\_columns":

[

"smo\_name",

"creat\_date"

]

}

}';

exec [main].[pro\_dq\_check\_cond\_upsrt]

@in\_parnt\_comp\_exctn\_id = NULL,

@in\_user\_name = 'TEST',

@in\_json\_dq\_strct = @json\_example,

@in\_file\_dfntn\_vers\_id = 1;

end

* FOR DEVELOPER:

**Please update this structure when you work on the task**

HOW TO PARSE JSON in SQL Server: https://www.sqlshack.com/how-to-parse-json-in-sql-server/

NULLS part:

As per Kamil's clarification - For now we can have the following way to accepting parameters from FE for eg:  
Threshold\_Condition - between  
threshold\_Lower - -2  
treshold\_Upper - 10

Both limits should adjust & accept for negative values and this check valids for only percentage custom coulmns in input files  
Edited

**HTML final content for email preparation**

HTML final report sent when validation fails/passes will compose of table with structure below. Each procedure will have separate pre-template with placeholders which will be replaced during final HTML content creation by ADF. Below proposed row with placeholder:

<td>%CHECK\_TYPE\_CODE%</td>

<td>%PROC\_NAME%</td>

<td %CLASS\_RESULT\_COLOR%>%STATUS%</td>

<td>%RESULT\_VALUE%</td>

<td>%LINK\_TO\_AZURE\_STORAGE%</td>

HTML row code:

<td> %CHECK\_TYPE\_CODE% </td>

<td> %PROC\_NAME% </td>

<td class = %CLASS\_RESULT\_COLOR% > %RESULT\_VALUE% </td>

<td> %RESULT\_VALUE% </td>

<td> %LINK\_TO\_AZURE\_STORAGE% </td>

<head>

<style>

td.td\_wrong {color: red;}

td.td\_correct {color: green;}

</style>

</head>

* Below are two examples of final content:

1. First with Failed/Passed in one table
2. The rows which passed will be in separate table when there will be no Azure Storage Link to CSV created
3. If we want to have separate table for failing/success we could create additional field in dq\_check\_type\_lkp for HTML\_SUCCESS and there put success template format

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Summary of Data Quality checks for "template" | | | | |
| **Check code** | **Procedure** | **Status** | **# of failed rows** | **Link to CSV file** |
| V01 - Duplications | pro\_dq\_V01\_check | Failed | 100 | "//AZURE/SomeDirectory/CSVDuplicationRows.csv |
| V02 - Nulls | pro\_dq\_V02\_check | Failed | 500 | "//AZURE/SomeDirectory/CSVNullsRows.csv |
| SV01 - Boolean | pro\_dq\_S01\_check | Passed | 0 | "//AZURE/SomeDirectory/CSVBooleanRows.csv |

|  |  |  |
| --- | --- | --- |
| Summary of Data Quality checks for "template" | | |
| **Check code** | **Procedure** | **Status** |
| V07 - All rows are Invalid - grouping | pro\_dq\_V07\_check | Passed |
| V02 - Nulls check | pro\_dq\_V02\_check | Passed |
| SV01 - Boolean | pro\_dq\_S01\_check | Passed |

# TECHNICAL DETAILS

## Back-end environment setup

**Environment configuration**

Note: This page is to show step by step how and what to configure environment on database side.

1. Connection to the P&G VPN
2. Change configuration in the HOSTS file on windows machine with below data. It requires to be admin and type of the file is **"FILE"**

* Path to the file is: **C:\Windows\System32\drivers\etc**
* Go inside the file and paste following data and save the file. Note: after saving file should be FILE type!  
  127.0.0.1 sqlserver-cnosgc-uploader-n01.database.windows.net   
  127.0.0.1 sqlserver-cnosgc-uploader-qa-n01.database.windows.net   
  127.0.0.1 sqlserver-cnosgc-uploader-uat-n01.database.windows.net   
  127.0.0.1 sqlserver-cnosgc-uploader-prod-n01.database.windows.net

1. Configure connection for the bastion server using for example PUTTY:

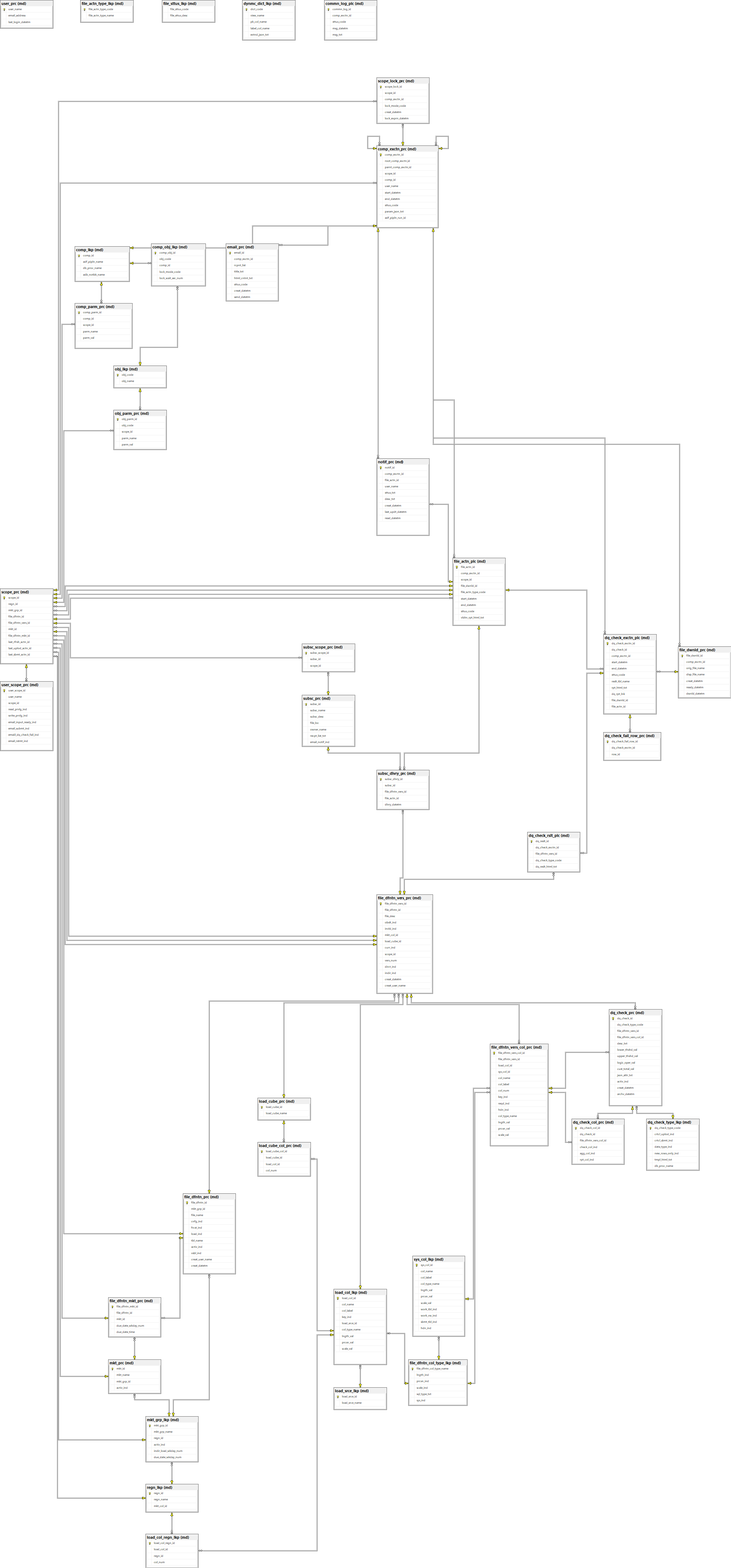
* Next go to the following path on the left to the **Tunnels** and create entry. It should be in format of:  
  Destination: **Address\_of\_database\_server:Port**  
  Source port: **1433**

1. Connect to newly created entry and log into as PG email and follow instructions for the password

If all was good and you have been granted access to that machine you will be connected. After this you can start using Management Studio for Azure database management.

1. Clone respective database repository to your local machine and connect it in the Visual Studio. There should be structure as:  
   5.1) Solution  
   5.2) Uploader-DB  
   5.3) Uploader-DB.Tests  
   5.4) Uploader-DB.Framework.tSQLt

If all was connected correctly in Visual Studio, repository was cloned then you can start creating branches and push changes to the external repository.



## Downloading Excel with 1M Records

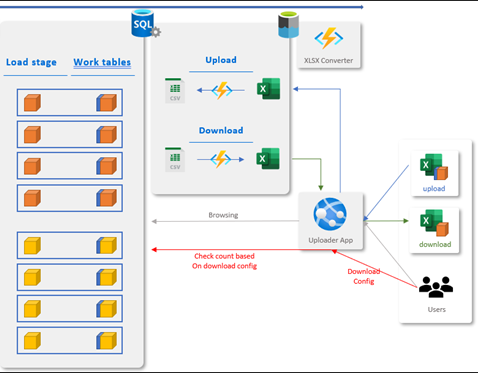
When a user tries to trigger download processing for a load input file with 1 million Records, the following warning message is to be displayed in the UI:

| **Download Error                                                                                                                                                                         Unable to start action because your selection exceeds 1M rows. Please apply filters to narrow your selection** |
| --- |

**Email reference for change request**

RE Decision for more than 1M Records.msg

### Technical Diagram for Downloading Excel with 1M Records

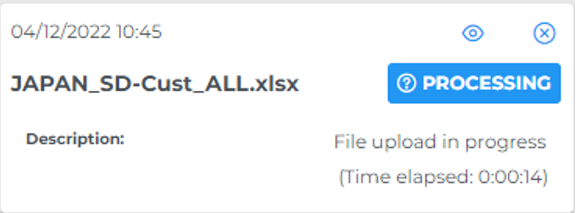


## Upload Notifications

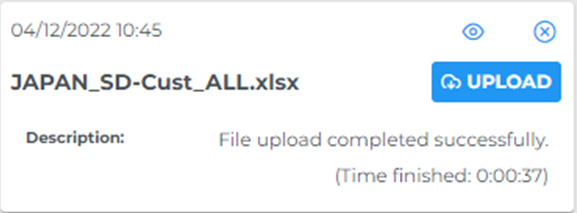
Upload Notifications

Upload notifications will use existing Uploader notifications mechanism. Upload process will push notification messages to DB and UI will present them to the user with notifications widget.

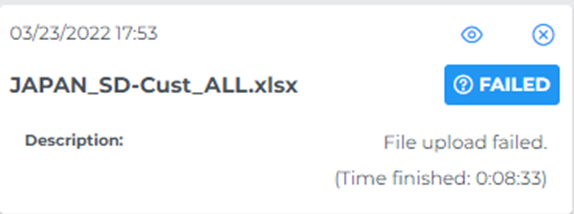
* Upload started: "File upload in progress"



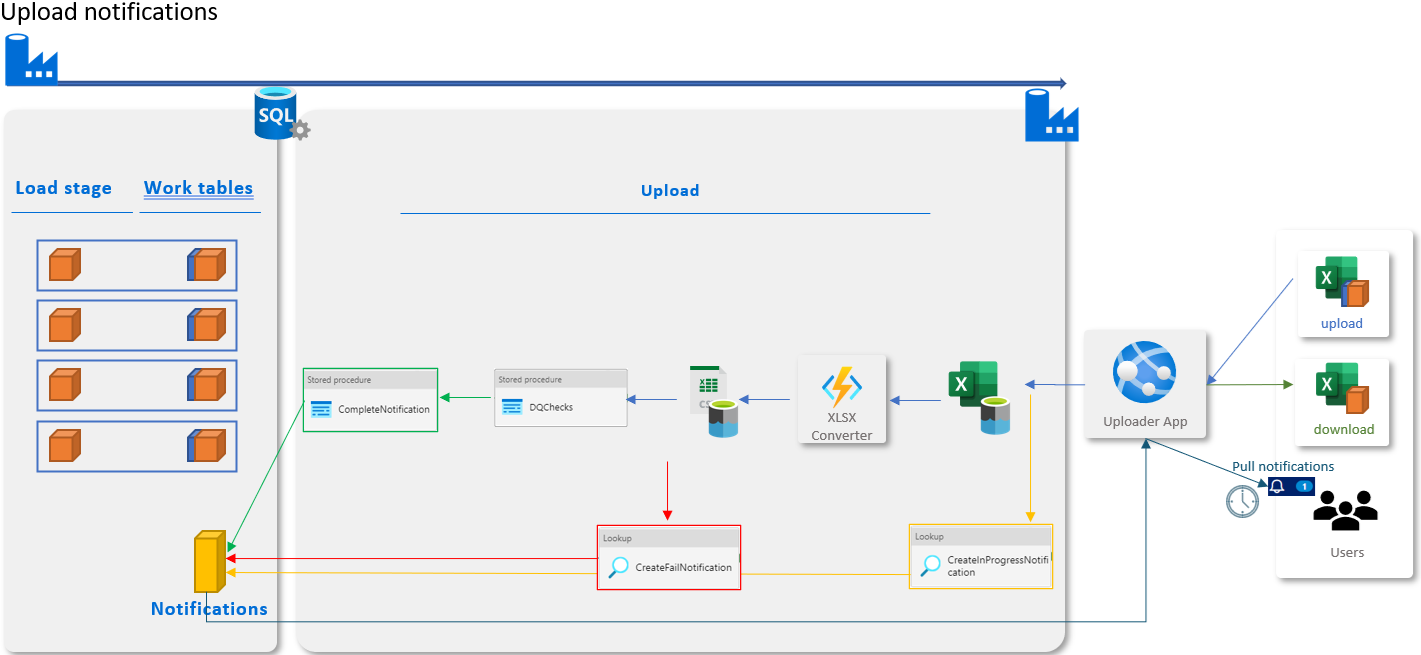
* Upload completed: "File upload completed successfully"



* Upload failed "File upload failed"



### Technical Diagram of Upload Notification



## Concurrency protection

Document describes the options for protecting the data consistency in various scenarios of concurrent Operations and End User activity.

**Supported scenarios.**

**Table 1**

| **Concurrent Process 2** | **Concurrent Process 1** | **Result** |
| --- | --- | --- |
| Upload/download file X | Upload/download file X | Single Download/Upload of specific file should be enabled only |
| Upload/download attempted | Load process | Download/upload of files for market group impacted by Load Process should be disabled |

### Upload/download vs Upload/download

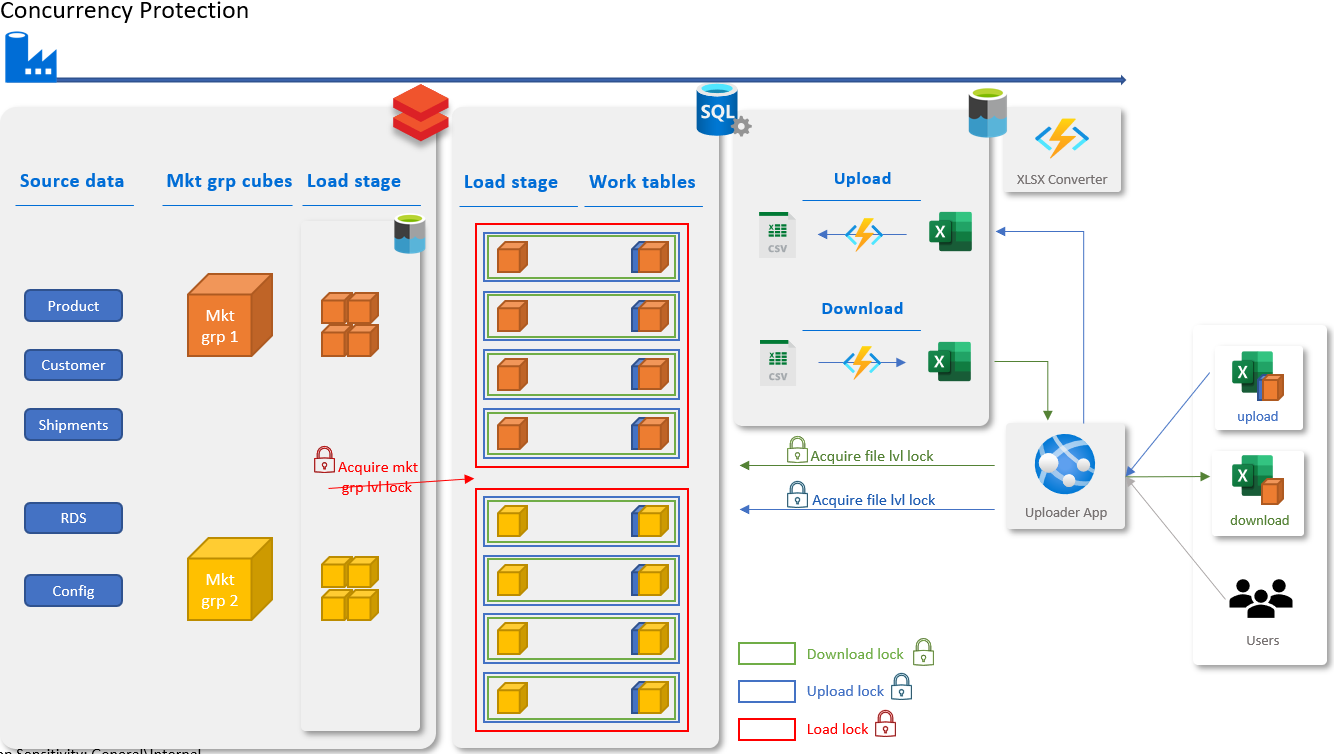
* Download/Upload of specific file should be processed only for one action at a time
* Other concurrent Download/Upload attempts for the same file version will be failed
* Single download at a time
* Single Upload at a time
* Owner of Download/Upload process that was failed will get a notification

### Upload/download vs Load process

* Load process locks all files for market groups in scope of refresh
* Load process will wait for acquiring the lock before processing for defined period of time. After this time it will fail.
* Download/Upload attempts during Load process files for market groups in scope of refresh will be failed
* Owner of Download/Upload process that was failed will get notification (See **Table 2** for scenarios)
* Upload/download of no-load input files is allowed during a load

**Table 2**

| **Scenario** | **Modal Message of Uploader Tool** |
| --- | --- |
| [Concurrent download] 2 or more users will download the same Input Files at the same time | Unable to start action because the file is currently being processed by other user or load process. If issue persists, please contact support. |
| [Concurrent Load process & Download] User will download a Load Input File while there is an ongoing Load Process | Unable to start action because the file is currently being processed by other user or load process. If issue persists, please contact support. |
| [Concurrent Load process & Upload] User will upload a Load Input File while there is an ongoing Load Process | Unable to start action because the file is currently being processed by other user or load process. If issue persists, please contact support. |
| [Concurrent upload] 2 or more users will upload the same Input Files at the same time | Unable to start action because the file is currently being processed by other user or load process. If issue persists, please contact support. |



### Scope locking interface

Acquiring scope lock will be done by updating scope\_lock\_prc table. Process that needs to lock a given scope will add an entry to this table. If this scope is locked already (entry for given scope or an overlapping scope exists in table) entry will not be added and acquiring lock will not succeed.

**Table 4**

| **Column name** | **Data type** | **Description** |
| --- | --- | --- |
| scope\_id | INT | Scope identifier |
| comp\_exctn\_id | INT | Component execution identifier that created the lock |
| lock\_exprn\_datetm | DATETIME | Lock expiration date |
| creat\_datetm | DATETIME | Lock creation date |

Acquiring the locks in scope\_lock\_prc will be done with pro\_scope\_lock\_acq procedure:

* Parameters:
  + @in\_scope\_id - id of scope. Does not accept nulls
  + @in\_lock\_mode\_code - 'S' for shared lock, 'E' for exclusive
  + @in\_exprn\_secnd - maximum number of seconds for lock to be held
* Returns: nothing
* Throws: error when can't acquire lock

Releasing the locks in scope\_lock\_prc table will be done with pro\_comp\_exctn\_lock\_relse procedure:

* Parameters: none

For load process, it is required to create multiple locks, one per market group. This will be handled by pro\_scope\_mkt\_grp\_lock\_acq. pro\_scope\_mkt\_grp\_lock\_acq procedure should create lock for scopes on Market Group level for each requested market groups. The same pro\_comp\_exctn\_lock\_relse will release lock for all requested market groups.

pro\_scope\_mkt\_grp\_lock\_acq

* Parameters:
  + @in\_mkt\_grp\_name\_list\_json\_txt - json with a list of market group names to lock taking for similar to: {"Market Groups": ["Europe - FRANCE", "Europe - UKI"]}
* Returns: nothing
* Throws: error when can't acquire lock for all market groups

### Scope locking for upload and download

Upload and download are a type of file action. As such, every upload and download process starts with calling pro\_file\_actn\_open and ends with calling pro\_file\_actn\_close. Those procedures update file\_actn\_plc which is a complete record of all actions related to a given file. Each file action is unique and has file\_actn\_id PK.

In order to prevent multiple uploads or downloads for a given scope, pro\_scope\_lock\_acq will be called for uploads or downloads in pro\_file\_actn\_open (action type can be checked by file\_actn\_type\_code). pro\_file\_actn\_close will call pro\_comp\_exctn\_lock\_relse for uploads or downloads.

### Scope locking for Load Process

Load process will try to acquire the lock at the very beginning and release locks once done. While from technical point of view, the scope of the lock could be smaller (reduced to file refresh in DB) locking uploads and downloads from the very beginning supports business needs better and reduces the risk of data inconsistencies and possible rework for end-users.

Locking will be added to both of the root Load process ADF pipelines

* ld-LoadMainGlobal
* ld-LoadMainSelective

Additional action will be added at the beginning of the pipeline calling pro\_scope\_mkt\_grp\_lock\_acq. If action will succeed, execution will continue, if not, pipeline will retry a number of times with defined interval (6 retries, with 10 minute interval). If lock will not be acquired, pipeline will fail. At the end of pipeline action will be added calling pro\_comp\_exctn\_lock\_relse.

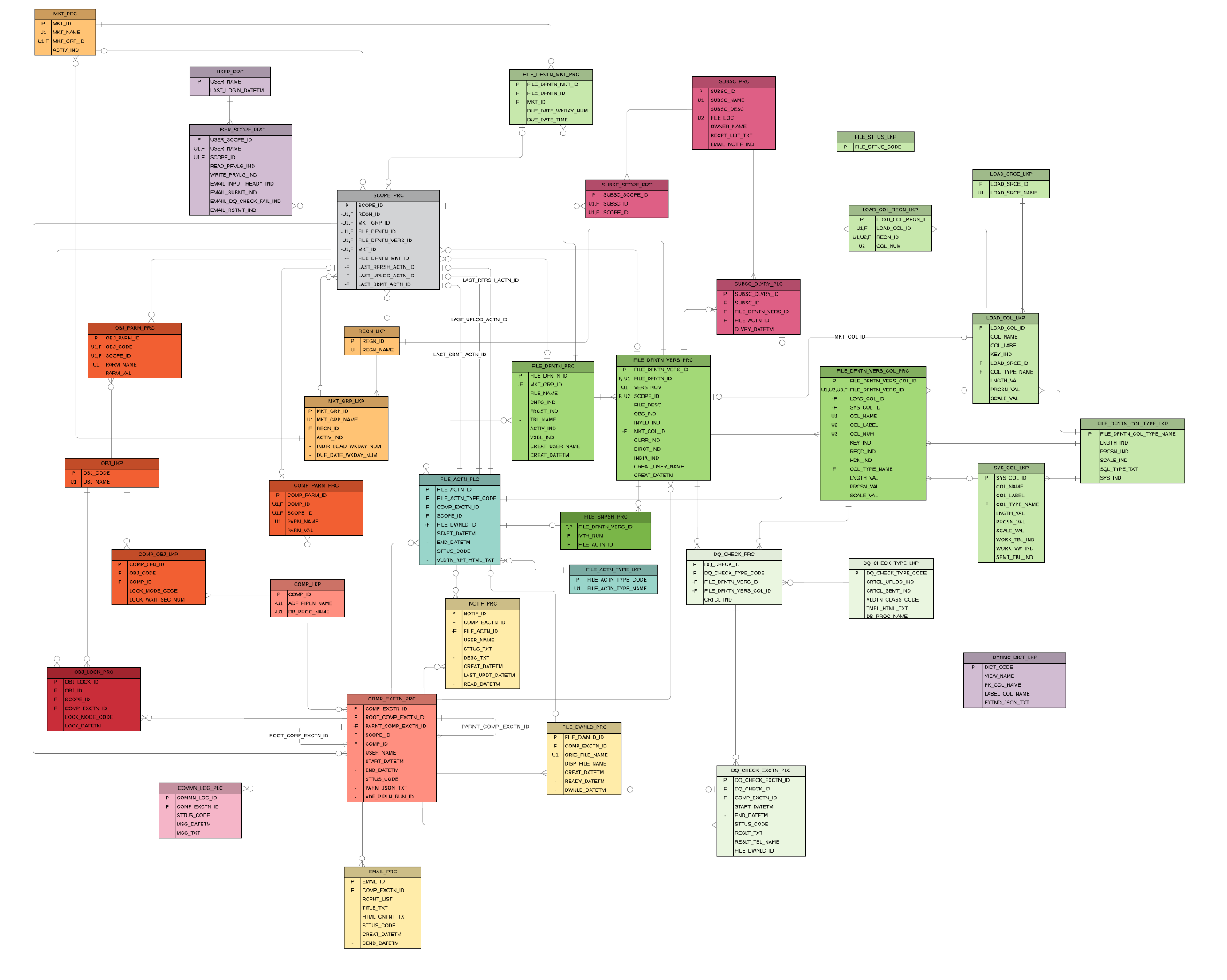
## Local Database Development Setup

**Entity diagram**

**Description:**

* P: primary key
* F: foreign key, if the column name is not equal to referenced column it's marked on relation arrow
* U{n}: unique key, for example all columns with U1 form together unique key
* "-": column can be nullable

Diagram



## Metadata tables

### commn\_log\_plc

Common log table.

| **Column name** | **Data type** | **Description** |
| --- | --- | --- |
| commn\_log\_id | INT | Row identifier for log table |
| comp\_exctn\_id | INT | ID of component execution |
| sttus\_code | CHAR(3) | Message type: OK/ERR/WRN |
| msg\_datetm | DATETIME2(7) | Date when message was created |
| msg\_txt | VARCHAR(MAX) | Content of the logged message |

### comp\_exctn\_prc

List of Component Executions.

| **Column name** | **Data type** | **Description** |
| --- | --- | --- |
| comp\_exctn\_id | INT | ID of Component Execution |
| main\_comp\_exctn\_id | INT | ID of root Component Execution |
| parnt\_comp\_exctn\_id | INT | ID of parent Component Execution |
| scope\_id | INT | ID of the Scope |
| comp\_id | INT | ID of Component |
| user\_name | VARCHAR(20) | P&G short name |
| start\_datetm | DATETIME2(7) | Start of execution |
| end\_datetm | DATETIME2(7) | End of the execcution |
| sttus\_code | VARCHAR(5) | Execution status: A-active, C-completed, F-failed |
| param\_json\_txt | VARCHAR(MAX) | JSON containing parameters of the execution |
| adf\_pipln\_run\_id | VARCHAR(50) | ID of the ADF pipeline execution |

### comp\_lkp

**Summary**

List of backend Components.  
Component can be:

* an ADF pipeline
* a DB procedure  
  Dictionaty used for tracking Component Execution.

**Columns**

| **Column name** | **Data type** | **Description** |
| --- | --- | --- |
| comp\_id | INT | Row identifier |
| adf\_pipln\_name | VARCHAR(50) | Name of ADF pipeline |
| db\_proc\_name | VARCHAR(50) | Name of DB procedure |

### comp\_obj\_lkp

List Objects related to Component.

| **Column name** | **Data type** | **Description** |
| --- | --- | --- |
| comp\_obj\_id | INT | Row identifier |
| obj\_code | VARCHAR(50) | Objects identifier |
| comp\_id | INT | Component identifier |
| lock\_mode\_code | VARCHAR(20) | SHARED/EXCLUSIVE |
| lock\_wait\_sec\_num | INT | How long can execution wait for the resource |

### comp\_parm\_prc

List Objects related to Component.

| **Column name** | **Data type** | **Description** |
| --- | --- | --- |
| comp\_parm\_id | INT | Row identifier |
| comp\_id | INT | Component identifier |
| scope\_id | INT | Scope identifier |
| parm\_name | VARCHAR(30) | Name of parameter |
| parm\_val | VARCHAR(30) | Parameter value |

### dynmc\_dict\_lkp

Dictionary containing list of dynamic lookups for Front End.

| **Column name** | **Data type** | **Description** | **Sample data** |
| --- | --- | --- | --- |
| dict\_code | VARCHAR(50) | Identifier of dictionary | mkt\_grp |
| view\_name | VARCHAR(50) | Name of the view | mkt\_grp\_prc\_vw |
| pk\_col\_name | VARCHAR(50) | Which column holds dictionary PK | mkt\_grp\_id |
| label\_col\_name | VARCHAR(50) | Which column holds label of the row | mkt\_grp\_name |
| extnd\_json\_txt | VARCHAR(MAX) | Additional dictionary attributes |  |

**List of expected dynamic dictionaries.**

| **Proposed name** | **Proposed base object** | **Proposed PK column** | **Proposed label column** | **Purpose** |
| --- | --- | --- | --- | --- |
| mkt\_grp | mkt\_grp\_lkp\_vw | mkt\_grp\_id | mkt\_grp\_name | list all market groups |
| mkt | mkt\_lkp\_vw | mkt\_id | mkt\_name | list all markets |
| file\_dfntn\_vers | file\_dfntn\_vers\_prc\_vw | file\_dfntn\_vers\_id | file\_name | list all current file versions (both visible/invisible, and valid/invalid) |
| file\_sttus | file\_sttus\_lkp\_vw | file\_sttus\_code | file\_sttus\_code | list all file statuses |
| user | user\_vw | user\_name | user\_name | distinct list of all users |
| scope | scope\_prc\_vw | scope\_id | scope\_name | list of scopes |
| file\_dfntn\_col\_type | file\_dfntn\_col\_type\_lkp\_vw | file\_dfntn\_col\_type\_name | file\_dfntn\_col\_type\_name | list of column data types |

**extnd\_json\_txt structure:**

{

"additionalColumns" ["mkt\_grp\_name", "mkt\_name", "file\_name"],

"additionalFilters" ["mkt\_grp\_name", "mkt\_name", "file\_name"]

}

* additionalColumns - list of columns which will be added to SELECT
* additionalFilters - list of columns which can be added to WHERE if present in request parameters
* incidentally both collections can list same columns, but can be mutually exclusive too

### dq\_check\_prc

List Objects related to Data Quality Check.

| **Column name** | **Data type** | **Description** |
| --- | --- | --- |
| dq\_check\_id | INT | Row identifier |
| dq\_check\_type\_code | VARCHAR(50) | Foreign object type identifier |
| file\_dfntn\_vers\_id | INT | File version definition identifier |
| file\_dfntn\_vers\_col\_id | VARCHAR(20) | Column ID of file's version |
| crtcl\_ind | CHAR(1) | Control identifier |
| check\_col\_ind | CHAR(1) | DQ check column identifier |
| rpt\_col\_ind | CHAR(1) | DQ check report column identifier |
| actv\_ind | CHAR(1) | Column current status identifier |
| creat\_datetm | DATETIME2(7) | Time of creation |
| archv\_datetm | DATETIME2(7) | Time of archiving |

### dq\_check\_exctn\_plc

List Objects related with Data Quality Check.

| **Column name** | **Data type** | **Description** |
| --- | --- | --- |
| dq\_check\_exctn\_id | INT | Row identifier |
| dq\_check\_id | INT | DQ Check ID |
| comp\_exctn\_id | INT | Component execution ID |
| file\_actn\_id | INT | ID of Upload action |
| start\_datetm | DATETIME2(7) | Start date time |
| end\_datetm | DATETIME2(7) | End date time |
| sttus\_code | VARCHAR(10) | Status code |
| reslt\_txt | VARCHAR(500) | HTML description of validation result to be included in email |
| reslt\_tbl\_name | VARCHAR(50) | Result table name |
| file\_dwnld\_id | INT | File download identifier |

### dq\_check\_type\_lkp

List Objects related with Data Quality Check.

| **Column name** | **Data type** | **Description** |
| --- | --- | --- |
| dq\_check\_type\_code | INT | Row identifier |
| crtcl\_upload\_ind | INT | Critical upload indicator |
| crtcl\_sbmt\_ind | INT | Critical submit indicator |
| vldtn\_class\_code | VARCHAR2(10) | Validation class code: data\_type/upload/submit |
| tmpl\_html\_txt | DATETIME2(7) | Template structure |
| db\_proc\_name | VARCHAR(10) | Name of procedure |

### email\_prc

Dictionary containing Market Groups.

| **Column name** | **Data type** | **Description** |
| --- | --- | --- |
| email\_id | INT | Row identifier for email |
| comp\_exctn\_id | INT | ID of Component Execution |
| rcpnt\_list | VARCHAR(MAX) | List of people to send email to |
| title\_txt | VARCHAR(200) | Tile of the email |
| html\_cntnt\_txt | VARCHAR(MAX) | Content of the email in HTML form |
| sttus\_code | VARCHAR(10) | Status of the email (NEW/SENDING/SENT) |
| creat\_datetm | DATETIME2(7) | Date when email was created |
| send\_datetm | DATETIME2(7) | Date when email was sent |

### file\_actn\_plc

Dictionary containing Market Groups.

| **Column name** | **Data type** | **Description** | **Sampl Value** |
| --- | --- | --- | --- |
| file\_actn\_id | INT | Row identifier |  |
| file\_actn\_type\_code | CHAR(1) | Action type from file\_actn\_type\_lkp | S |
| comp\_exctn\_id | INT | ID of Component Execution in comp\_exctn\_prc |  |
| scope\_id | INT | Scope identifier |  |
| file\_dwnld\_id | INT | Identifier of the file prepared for download |  |
| start\_datetm | DATETIME2(7) | Start of the action |  |
| end\_datetm | DATETIME2(7) | End of the action |  |
| sttus\_code | VARCHAR(10) | Completion status: A-active, C-completed, F-failed |  |
| vldtn\_rpt\_html\_txt | VARCHAR(MAX) | HTML content of validation report to be displayed in notification and sent in email |  |

### file\_actn\_type\_lkp

List of actions which might be performed on Input File.

| **Column name** | **Data type** | **Description** |
| --- | --- | --- |
| file\_actn\_type\_code | VARCHAR(15) | Type of code for file action, can be one of U/D/S/R (Upload, Download, Submit, Refresh) |
| file\_actn\_type\_name | VARCHAR(50) | Name of the action |

Details:

* Unique key (file\_actn\_type\_name)

### file\_arch\_prc

List of monthly snapshots of Input File.

| **Column name** | **Data type** | **Description** |
| --- | --- | --- |
| file\_dfntn\_vers\_id | INT | ID of Input File version |
| mth\_num | INT | Month in form of number: 202104 |
| file\_actn\_id | INT | ID of action which created a snapshot |

### file\_dfntn\_col\_type\_lkp

List of available Column Types.

| **Column name** | **Data type** | **Description** |
| --- | --- | --- |
| file\_dfntn\_col\_type\_name | VARCHAR(20) | Row indentifier |
| lngth\_ind | CHAR(1) | Should data type be characterized by length? Y for string fields, otherwise N |
| prcsn\_ind | CHAR(1) | Should data type be characterized by precision? Y for numeric fields, otherwise N |
| scale\_ind | CHAR(1) | Should data type be characterized by scale? Y for numeric fields, otherwise N |
| sql\_type\_txt | VARCHAR(MAX) | SQL data type to be used for this data type |
| sys\_ind | CHAR(1) | Is data type reserved for System Columns only? |

### file\_dfntn\_mkt\_prc

**Summary**

Attributes of Input File for given Market

**Columns**

| **Column name** | **Data type** | **Description** |
| --- | --- | --- |
| file\_dfntn\_mkt\_id | INT | Row identifier |
| file\_dfntn\_id | INT | Input File identifier |
| mkt\_id | INT | Market identifier |
| due\_date\_wkday\_num | INT | Working day due date for Input File in Market |
| due\_date\_time | DATETIME2(7) | Time of day (CET) representing deadline for Submission |

### file\_dfntn\_prc

**Summary**

List of Input Files.

**Columns**

| **Column name** | **Data type** | **Description** |
| --- | --- | --- |
| file\_dfntn\_id | INT | Row indetifier |
| mkt\_grp\_id | INT | Market Group identifier |
| file\_name | VARCHAR(100) | Name of file |
| cnfg\_ind | CHAR(1) | Y for Config Files |
| frcst\_ind | CHAR(1) | Y for Forecast identifier |
| tbl\_name | VARCHAR(100) | Name of table |
| activ\_ind | CHAR(1) | Y for active files |
| vsbl\_ind | CHAR(1) | Y for active visibility |
| creat\_user\_name | VARCHAR(50) | P&G short name who created the file |
| creat\_datetm | DATETIME2(7) | The date of file creation |

### file\_dfntn\_vers\_col\_prc

**Summary**

Dictionary containing columns for given Inut File version.

**Columns**

| **Column name** | **Data type** | **Description** | **Sample value** |
| --- | --- | --- | --- |
| file\_dfntn\_vers\_col\_id | INT | Row identifier of table |  |
| file\_dfntn\_vers\_id | INT | Version of file |  |
| load\_col\_id | INT | Identifier of Load Column (if row represents Load Column) |  |
| sys\_col\_id | INT | Identifier of System Column |  |
| col\_name | VARCHAR(50) | Column name from specific table |  |
| col\_label | VARCHAR(50) | Label for column from specific table |  |
| col\_num | INT | Position of the column in the specific table |  |
| key\_ind | CHAR(1) | Grid row identifier, NULL for new rows (new columns in Input File definition). Only single column should have it set to Y | Y/N |
| reqd\_ind | CHAR(1) | Is field content mandatory upon load and upload? |  |
| hdn\_ind | CHAR(1) | Should column be presented in UI and visible in Excel? |  |
| col\_type\_name | VARCHAR(20) | Column type as defined in file\_dfntn\_col\_type\_lkp |  |
| lngth\_val | INT | Length of the string field |  |
| prcsn\_val | INT | Precision of the column for numeric field |  |
| scale\_val | INT | Length of decimal digits that are stored to the right of the decimal point for numeric field |  |

**Unique keys**

* file\_dfntn\_col\_prc\_uk1 (file\_dfntn\_vers\_id, col\_name)
* file\_dfntn\_col\_prc\_uk2 (file\_dfntn\_vers\_id, col\_label)
* file\_dfntn\_col\_prc\_uk3 (file\_dfntn\_vers\_id, col\_num)

### file\_dfntn\_vers\_prc

**Summary**

List of versions of Input File.

**Columns**

| **Column name** | **Data type** | **Description** |
| --- | --- | --- |
| file\_dfntn\_vers\_id | INT | Row identifier |
| file\_dfntn\_id | INT | Identifier of Input File |
| vers\_num | INT | Identifier of file's version |
| scope\_id | INT | Identifier of Scope ID |
| file\_desc | VARCHAR(50) | Description of Input File version |
| obs\_ind | CHAR(1) | Should there be option to mark rows as Obsolete |
| invld\_ind | CHAR(1) | Should there be option to mark rows as Invalid |
| mkt\_col\_id | INT | Which column serves as source of Market |
| load\_cube\_id | INT | Load Cube ID from load\_cube\_prc |
| curr\_ind | CHAR(1) | Is this a current version of the Input File? |
| dirct\_ind | CHAR(1) | Indicator Y/N |
| indir\_ind | CHAR(1) | Indicator Y/N |
| creat\_user\_name | VARCHAR(50) | P&G short name who created the file version |
| creat\_datetm | DATETIME2 | The date of file version creation |

### file\_dwnld\_prc

Dictionary with list of files prepared for download.  
Can be used both for Input Files and for Data Quality Reports.

| **Column name** | **Data type** | **Description** |
| --- | --- | --- |
| file\_dwnld\_id | INT | Row identifier |
| comp\_exctn\_id | INT | ID of Component Execution |
| orig\_file\_name | VARCHAR(50) | Name of the file on Storage Account including container and path |
| disp\_file\_name | VARCHAR(50) | Default name used for saving file locally by End User |
| creat\_datetm | DATETIME2(7) | When was the file requested |
| ready\_datetm | DATETIME2(7) | When was the file actually put on the Storage Account |
| dwnld\_datetm | DATETIME2(7) | When was the file most recently downloaded |

Constraints:  
**file\_dwnld\_prc\_uk1** - Unique constraint on **[orig\_file\_name]**

### file\_sttus\_lkp

Dictionary with list of available file status

| **Column name** | **Data type** | **Description** |
| --- | --- | --- |
| file\_sttus\_code | VARCHAR(20) | Row identifier |

### load\_cube\_prc

List of Load Cubes.

| **Column name** | **Data type** | **Description** |
| --- | --- | --- |
| load\_cube\_id | INT | Row identifier |
| load\_cube\_desc\_txt | VARCHAR(50) | Load Cube description |

### load\_cube\_col\_prc

| **Column name** | **Data type** | **Description** |
| --- | --- | --- |
| load\_cube\_col\_id | INT | Row identifier |
| load\_cube\_id | INT | Load Cube ID |
| load\_col\_id | INT | ID of Load Column |
| col\_num | INT | Column order |

### load\_col\_lkp

List of Load Columns.

| **Column name** | **Data type** | **Description** |
| --- | --- | --- |
| load\_col\_id | INT | Row identifier |
| col\_name | VARCHAR(50) | Name of the column |
| col\_label | VARCHAR(50) | Label for the column |
| key\_ind | CHAR(1) | Is this a PK column? |
| load\_srce\_id | INT | Which is the source of this column? |
| col\_type\_name | VARCHAR(20) | Column type as defined in file\_dfntn\_col\_type\_lkp |
| lngth\_val | INT | Length of the string field |
| prcsn\_val | INT | Precision of the column for numeric field |
| scale\_val | INT | Length of decimal digits that are stored to the right of the decimal point for numeric field |

### load\_col\_regn\_lkp

List of columns being product of ADB Load Process divided per Region.

| **Column name** | **Data type** | **Description** |
| --- | --- | --- |
| load\_col\_regn\_id | INT | Row identifier |
| load\_col\_id | INT | Foreign key to load\_col\_lkp table |
| regn\_id | INT | Foreign key to regn\_lkp table |
| col\_num | INT |  |

References:

* load\_col\_lkp -> reference table
* regn\_lkp -> reference table

### load\_srce\_lkp

List of the sources of the Load Process:

* Shipments
* Customer Mapping
* Product Mapping
* Geo Mapping
* RDS Product Hierarchy

| **Column name** | **Data type** | **Description** |
| --- | --- | --- |
| load\_srce\_id | INT | Row indentifier |
| load\_srce\_name | VARCHAR(200) | Name of the source |

Unique keys:

* load\_srce\_lkp\_uk1 (load\_srce\_name)

### mkt\_grp\_lkp

**Summary**

Dictionary containing Market Groups.

**Columns**

| **Column name** | **Data type** | **Description** |
| --- | --- | --- |
| mkt\_grp\_id | INT | Row identifier |
| mkt\_grp\_name | VARCHAR(50) | Name of the Market Group |
| regn\_id | INT | Region identifier |
| activ\_ind | CHAR(1) | Indicates if Market Group is being active (Y/N) |
| indir\_load\_wkday\_num | INT | Working day to trigger Load for Indirects |
| due\_date\_wkday\_num | INT | Due date for Input Files in Market Group |

**Unique keys**

* mkt\_grp\_lkp\_uk1 (mkt\_grp\_name)

### mkt\_prc

**Summary**

Dictionary containing Markets.

**Columns**

| **Column name** | **Data type** | **Description** |
| --- | --- | --- |
| mkt\_id | INT | Row indentifier |
| mkt\_name | VARCHAR(50) | Market name |
| mkt\_grp\_id | INT | Market Group identifier |
| activ\_ind | CHAR(1) | Should data for Market be visible for End Users |

**Unique keys**

* mkt\_prc\_uk1 (mkt\_name, mkt\_grp\_id)

### notif\_prc

List of notifications.

| **Column name** | **Data type** | **Description** |
| --- | --- | --- |
| notif\_id | INT | Row identifier |
| comp\_exctn\_id | INT | ID of Component Execution |
| file\_actn\_id | INT | ID of related File Action |
| user\_name | VARCHAR(20) | P&G short name |
| sttus\_txt | VARCHAR(20) | Short description of action status: new, starting, processing, completed, invalid, failed, etc. |
| desc\_txt | VARCHAR(1000) | Optional notification text |
| create\_datetm | DATETIME2(7) | When was notification created |
| last\_updt\_datetm | DATETIME2(7) | When was notification most recently updated |
| read\_datetm | DATETIME2(7) | When was notification most recently read |

### obj\_lkp

| **Column name** | **Data type** | **Description** |
| --- | --- | --- |
| obj\_code | VARCHAR(20) | Row indentifier |
| obj\_name | VARCHAR(50) | Name of object |

Constraints:

**obj\_lkp\_uk1** - unique **obj\_name**

### scope\_lock\_prc

| **Column name** | **Data type** | **Description** |
| --- | --- | --- |
| scope\_lock\_id | INT | Row indentifier |
| scope\_id | INT | Scope ID |
| comp\_exctn\_id | INT | Component execution ID |
| lock\_mode\_code | CHAR(1 | Type of the lock: S - Shared, E - Exclusive |
| creat\_datetm | DATETIME2(7) | Time of lock acquisition |
| lock\_exprn\_datetm | DATETIME2(7) | Time from which lock will be considered as expired (no longer valid) |

### obj\_parm\_prc

List Parameters which can be set for given Objects.

| **Column name** | **Data type** | **Description** |
| --- | --- | --- |
| obj\_parm\_id | INT | Row identifier |
| obj\_code | VARCHAR(50) | Objects identifier |
| scope\_id | INT | Scope identifier |
| parm\_name | VARCHAR(50) | Name of the parameter |
| parm\_val | VARCHAR(50) | Parameter value |

### regn\_lkp

**Summary**

Dictionary containing Regions.

**Columns**

| **Column name** | **Data type** | **Description** |
| --- | --- | --- |
| regn\_id | INT | Row identifier |
| regn\_name | VARCHAR(50) | Name of the region |

**Unique keys**

* regn\_lkp\_uk1 (regn\_name)

### scope\_prc

**Summary**

Dictionary representing Market/File scope for any processing/action in Uploader.  
Scopes represent hierarchy and can be of these levels:

* global
* regional
* Market Group level
* Input File in Market Group (and all its versions)
* particular version of Input File in Market Group
* particular version of Input File limited to single Market

**Columns**

| **Column name** | **Data type** | **Description** |
| --- | --- | --- |
| scope\_id | INT | Row identifier |
| regn\_id | INT | Region identifier |
| mkt\_grp\_id | INT | Market Group identifier |
| file\_dfntn\_id | INT | Identifier of Input File |
| file\_dfntn\_vers\_id | INT | Identifier of Input File version |
| mkt\_id | INT | Market identifier |
| file\_dfntn\_mkt\_id | INT | FK for Market/Input File attributes |
| last\_rfrsh\_actn\_id | INT | FK for File Action table |
| last\_uplod\_actn\_id | INT | FK for File Action table |
| last\_sbmt\_actn\_id | INT | FK for File Action table |

### subsc\_dlvry\_prc

Records of Input Files delivered in scope of given Subscription.

| **Column name** | **Data type** | **Description** |
| --- | --- | --- |
| subsc\_dlvry\_id | INT | Row identifier |
| subsc\_id | INT | Identifier of Subscription |
| file\_dfntn\_vers\_id | INT | Identifier of Input File version |
| file\_actn\_id | INT | Identifier of File Action |
| dlvry\_datetm | DATETIME2(7) | Delivery time |

### subsc\_prc

List of Subscriptions.

| **Column name** | **Data type** | **Description** |
| --- | --- | --- |
| subsc\_id | INT | Subscription identifier |
| subsc\_name | VARCHAR(50) | Subscription name |
| subsc\_desc | VARCHAR(50) | Subscription description |
| file\_loc | VARCHAR(50) | Path to which files should be exposed |
| owner\_name | VARCHAR(20) | P&G short name of user responsible for subscription |
| recpt\_list\_txt | VARCHAR(MAX) | List of recipients to be notified upon delivery |
| email\_notif\_ind | CHAR(1) | Should recipients be notified via email about delivery |

### subsc\_scope\_prc

List of Scopes for given Subscription

| **Column name** | **Data type** | **Description** |
| --- | --- | --- |
| subsc\_scope\_id | INT | Row identifier |
| subsc\_id | INT | ID of the Subscription |
| scope\_id | INT | ID of the Scope |

### sys\_col\_lkp

List of System Columns.

| **Column name** | **Data type** | **Description** |
| --- | --- | --- |
| sys\_col\_id | INT | Row identifier |
| col\_name | VARCHAR(50) | Name of column kept in system |
| col\_label | VARCHAR(200) | Label for column |
| col\_type\_name | VARCHAR(20) | Column type as defined in file\_dfntn\_col\_type\_lkp |
| lngth\_val | INT | Length of the string field |
| prcsn\_val | INT | Precision of the column for numeric field |
| scale\_val | INT | Length of decimal digits that are stored to the right of the decimal point for numeric field |
| work\_tbl\_ind | CHAR(1) | Table work indicator |
| work\_vw\_ind | CHAR(1) | View work indicator |
| sbmt\_tbl\_ind | CHAR(1) | Submit table indicator |

### user\_prc

| **Column name** | **Data type** | **Description** |
| --- | --- | --- |
| user\_name | VARCHAR(50) | P&G short name |
| email\_address | VARCHAR(100) | P&G email address |
| last\_login\_datetm | DATETIME2(7) | Last login date |

### user\_scope\_prc

**Summary**

Attribute of users in given Scope.

**Columns**

| **Column name** | **Data type** | **Description** |
| --- | --- | --- |
| user\_scope\_id | INT | Row identifier |
| user\_name | VARCHAR(20) | P&G short name |
| scope\_id | INT | Scope identifier |
| read\_prvlg\_ind | CHAR(1) | Y if user has read access to the Scope |
| write\_prvlg\_ind | CHAR(1) | Y if user has write access to the Scope |
| email\_input\_ready\_ind | CHAR(1) | Should user receive email notification on data refresh? |
| email\_submt\_ind | CHAR(1) | Should user receive email notification after submission in the Scope? |
| email\_dq\_check\_fail\_ind | CHAR(1) | Should user receive notification upon validation errors? |
| email\_rstmt\_ind | CHAR(1) | Should user be notified in case of restatement (submission after due date) for given Scope? |

## Generic objects

**Staging table for Input File**

Schema: **stage**  
Name: **load\_stage\_<NNNN>\_<MMMM>\_sfct** (N stands for Market Group ID, N for File Version ID)  
Purpose: holds grouping of load columns relevant for particular Input File

Description:  
As a part of reloading Input File, Load Cube for given Market Group is reduced (with distinct grouping) to set of data relevant for given Input File.

**DQ validation results**

Schema: **vldtn\_reslt**  
Name: **valdtn\_<yyyyMMdd><FFFFFFFF><DQDQDQDQ>\_sfct**

* "yyyyMMdd" stands for time
* "FFFFFFFF" stands for file action ID in 8 digit format
* "DQDQDQDQ" stands for DQ check execution ID in 8 digit format

**Description:**

Holds results of Input File validation. Table stored in validation result schema "**vldtn\_reslt**", used by ADF pipeline process for preparing Email content of validations for user when detected wrong state of uploaded data.

Table created by each dq check condition check's logic holding rows which have not met applied dq condition for template.

**Dropping mechanism:** There is no implemented mechanism of dropping tables. Initial idea was to delete everything where date is grater than set value. This step can be part of uploaded procedure as a final step for cleaning up or completely separate logic.

**Referenced by:** This wiki page is connected with following page: DQ check summary

**Load Cube staging table for Market Group**

Schema: **vldtn\_reslt**  
Name: **valdtn\_<yyyyMMdd><FFFFFFFF><DQDQDQDQ>\_sfct**

* "yyyyMMdd" stands for time
* "FFFFFFFF" stands for file action ID in 8 digit format
* "DQDQDQDQ" stands for DQ check execution ID in 8 digit format

**Description:**

Holds results of Input File validation. Table stored in validation result schema "**vldtn\_reslt**", used by ADF pipeline process for preparing Email content of validations for user when detected wrong state of uploaded data.

Table created by each dq check condition check's logic holding rows which have not met applied dq condition for template.

**Dropping mechanism:** There is no implemented mechanism of dropping tables. Initial idea was to delete everything where date is grater than set value. This step can be part of uploaded procedure as a final step for cleaning up or completely separate logic.

**Referenced by:** This wiki page is connected with following page: DQ check summary

**Load Cube staging table for Market Group**

Schema: **stage**  
Name: **load\_cube\_<NNNN>\_<MMMM>\_sfct** (N stands for mkt\_grp\_id, M for load\_cube\_id)  
Purpose: holds product of Databricks Load Process for given Load Cube in Market Group

Description:  
Columns list taken from load\_cube\_col\_prc\_vw.  
Name of the table is held in mkt\_grp\_load\_cube\_vw.load\_cube\_tbl\_name.  
Created by pro\_mkt\_grp\_load\_cube\_tbl\_creat upon creating new File Definition.

Related objects:

* pipeline ld-UploadLoadCubeToDB
* procedure pro\_file\_reld\_grpng\_creat

**Submit Table for Input File version**

Schema: **sbmt**  
Name: **input\_<NNNN>\_<MMMM>\_sbmt\_<YYYYMM>\_fct**

* N stands for mkt\_grp\_id
* M for file\_dfntn\_vers\_id
* YYYYMM month of the publication cycle

Description:  
Snapshot table which holds data being exposed to downstream (all rows except Invalid and Obsolete Rows).

This table is updated/created only when all DQ checks assigned to template have passed data quality checks!

Created by: System Submit process  
Dropped by: Never dropped. When there is no new template version created such table is truncated and latest data is exposed to the downstream

**Working View for Input File version**

Schema: **input**  
Name: **input\_<NNNN>\_<MMMM>\_work\_fct\_vw**

* N stands for mkt\_grp\_id
* M for file\_dfntn\_vers\_id

Description:  
Working data set to which relate download and upload actions.  
Contains columns of Working Table extended by special column as described here:  
sys\_last\_mdfd\_datetm and sys\_last\_mdfd\_user\_name are taken from file\_actn\_plc\_vw (start\_datetm and user\_name) via sys\_last\_uplod\_actn\_id.

Created by: pro\_file\_work\_tbl\_creat

**Working Table for Input File version**

Schema: **input**  
Name: **input\_<NNNN>\_<MMMM>\_work\_fct**

* N stands for mkt\_grp\_id
* M for file\_dfntn\_vers\_id

**Description:**

Working table holds template's data available on the FE for upload and download. It is sourced by main load process and later updated by user's actions (download/upload).

Created by: **New template's version action creation on the FE**  
Dropped by: **Currently no dropping**

Database API

**Errors**

DB schema: **main**  
Used by: **FE API**  
Context: **\***  
Purpose: to communicate errors that happen during DB processing in response to user actions.

**Description**

Errors which occur in DB in response to user actions should provide some feedback about the situation.

Instead of providing full explanation as free text, the error should provide relevant information in structured manner which can then be used to properly format the message presented to the user.

A DB error should contain at least following structured information:

| **information** |  | **meaning** |
| --- | --- | --- |
| internal error reference | error-ref | - presented on UI unconditionally when available - for DB consists of Component Execution ID to find relevant DB log record - for FE conversation ID - will be presented as is, data type or pattern does not matter |
| message text | message-text | - message to be presented as is |

To provide structure FE API will inspect DB error messages for following pattern:

... @@error-ref@@message-text@@ ...

## Procedures and functions

### fn\_get\_parm\_val

DB schema: **main**  
Used by: **DB internal**  
Context: **Input File maintenance**  
Returning value: parameter value from [md].[obj\_parm\_prc]

| **Parameter name** | **Data type** | **Description** | **Sample value** |
| --- | --- | --- | --- |
| obj\_code | VARCHAR(50) | Object Code from [md].[obj\_lkp] | upldr.db |
| parm\_name | VARCHAR(50) | Parameter name | vers |

Description:  
Function looks up parameter value from [md].[obj\_parm\_prc] given object code and parameter name.

### fn\_dup\_cond\_list

DB schema: **main**  
Used by: **FE application**  
Context: **FE list of columns for duplication check**  
Returning value: Table with list of already created data quality checks for particular template

| **Parameter name** | **Data type** | **Description** | **Sample value** |
| --- | --- | --- | --- |
| file\_name | VARCHAR(200) | Name of file to show DQ check conditions |  |

Description:

Returns table of following structure with list of already assigned columns for DQ check:

* col\_name
* check\_ind
* rpt\_ind

### fn\_get\_load\_col\_list

DB schema: **main**  
Used by: **DB internal**  
Context: **Input File maintenance**  
Returning value: string with list of columns as requested by expression

| **Parameter name** | **Data type** | **Description** | **Sample value** |
| --- | --- | --- | --- |
| in\_file\_dfntn\_vers\_id | INT | Version of Input File definition |  |
| in\_col\_patrn\_txt | VARCHAR(MAX) | Describes how should single column be presented. May contain any of 3 special patterns to be replaced: <col\_name>, <load\_col\_name> and <phys\_data\_type>. See example below |  |
| in\_sprtr\_txt | VARCHAR(100) | Describes what should be delimiter on the column list. See example below |  |
| in\_load\_col\_ind | CHAR(1) | Should list of columns be restricted to those having certain value of load\_ind. Not taken into account if NULL. |  |
| in\_work\_tbl\_ind | CHAR(1) | Should list of columns be restricted to those having certain value of work\_tbl\_ind. Not taken into account if NULL. |  |
| in\_work\_vw\_ind | CHAR(1) | Should list of columns be restricted to those having certain value of work\_vw\_ind. Not taken into account if NULL. |  |
| in\_sbmt\_tbl\_ind | CHAR(1) | Should list of columns be restricted to those having certain value of sbmt\_tbl\_ind. Not taken into account if NULL. |  |

Description:  
Function returns string listing columns in Input File as described in expression and flag list. Columns are ordered by col\_num.

For example for Input File having two column "a" and "b", a call with in\_col\_patrn\_txt="<col\_name>" and in\_sprtr\_txt=" - " will return: "a - b".

### fn\_get\_scope

DB schema: **main**  
Used by: **DB internal and ADF**  
Context: **Maket and File hierarchy**  
Returning value: scope identifier for given Region/Market Group/Market or File (INT)

| **Parameter name** | **Data type** | **Description** | **Sample value** |
| --- | --- | --- | --- |
| in\_regn\_id | INT | Identifier of Region |  |
| in\_regn\_name | VARCHAR | Region name |  |
| in\_mkt\_grp\_id | INT | Identifier of Market Group |  |
| in\_mkt\_grp\_name | VARCHAR | Market Group name |  |
| in\_file\_dfntn\_id | INT | Input File definition |  |
| in\_file\_dfntn\_vers\_id | INT | Version of Input File definition |  |
| in\_mkt\_id | INT | Identifier of Market |  |
| in\_mkt\_name | VARCHAR | Market name |  |

Description:  
Returns the scope\_id for given entry data. Throws error if scope not found in scope\_prc. This function should source from scope\_prc\_vw.  
Sample data and evaluations can be found here 

### pro\_all\_rows\_invld\_check

DB schema: **main**  
Used by: **FE API**  
Context: **Database**  
Purpose: Validate data as per DQ Check V06

| **Parameter name** | **Data type** | **Description** | **Sample value** |
| --- | --- | --- | --- |
| in\_parnt\_comp\_exctn\_id | INT | Identifier of callers Component Execution |  |
| in\_user\_name | VARCHAR(50) | Name of the user performing the action |  |
| in\_tbl\_name | VARCHAR(200) | Table name as a source for procedure |  |
| in\_file\_dfntn\_id | INT | File ID |  |
| in\_file\_dfntn\_vers\_id | INT | File Version ID |  |
| out\_sttus\_val | INT | Status of procedure (1 -> OK, -1 -> Failed) |  |

Description:  
-Procedure responsible for checking if there are months having all rows as invalid. Requirements as per WIKI: V06 DQ Validation

### pro\_comp\_exctn\_set\_scope

DB schema: **main**  
Used by: **DB internal and ADF**  
Context: **logging**  
Purpose: to register execution of code unit

| **Parameter name** | **Data type** | **Description** | **Sample value** |
| --- | --- | --- | --- |
| in\_comp\_exctn\_id | INT | Identifier of Component Execution |  |
| in\_scope\_id | INT | Scope identifier |  |

Description:  
Sets scope\_id for given Component Execution which we assume is opened.  
Used if scope needs to be calculated at the beginning of the execution.

### pro\_dup\_check

DB schema: **main**  
Used by: **FE API**  
Context: **Database**  
Purpose: Validate data as per DQ Check V01

| **Parameter name** | **Data type** | **Description** | **Sample value** |
| --- | --- | --- | --- |
| in\_parnt\_comp\_exctn\_id | INT | Identifier of callers Component Execution |  |
| in\_user\_name | VARCHAR(50) | Name of the user performing the action |  |
| in\_tbl\_name | VARCHAR(200) | Table name as a source for procedure |  |
| in\_file\_dfntn\_id | INT | File ID |  |
| in\_file\_dfntn\_vers\_id | INT | File Version ID |  |
| out\_sttus\_val | INT | Status of procedure (1 -> OK, -1 -> Failed) |  |

Description:  
-Procedure responsible for checking if there dyplicate values using combination of provided columns in check. Requirements as per WIKI: V01 DQ Validation

### pro\_email\_creat

DB schema: **main**  
Used by: **Azure Data Factory/Logic App**  
Context: **Database actions**  
Purpose: Creating email entry with details

| **Parameter name** | **Data type** | **Description** | **Sample value** |
| --- | --- | --- | --- |
| in\_parnt\_comp\_exctn\_id | INT | Identifier of callers Component Execution |  |
| in\_user\_name | VARCHAR(50) | Name of the user performing the action |  |
| in\_scope\_id | INT | Scope identifier |  |
| in\_rcpnt\_list | VARCHAR(MAX) | List of recipients for sending emails |  |
| in\_title\_txt | VARCHAR(200) | Title of email |  |
| in\_html\_cntnt\_txt | VARCHAR(MAX) | Detailed content of the email |  |

Description:  
Procedure performs following steps:

* creates new entry in email\_prc table with email details

### pro\_email\_delet

DB schema: **main**  
Used by: **Azure Data Factory/Logic App**  
Context: **Database actions**  
Purpose: Deleting emails older than provided retention value

| **Parameter name** | **Data type** | **Description** | **Sample value** |
| --- | --- | --- | --- |
| in\_parnt\_comp\_exctn\_id | INT | Identifier of callers Component Execution |  |
| in\_user\_name | VARCHAR(50) | Name of the user performing the action |  |

Description:  
Procedure performs following steps:

* deletes entries in email\_prc table with email details older than retention value

IMPORTANT:  
For now, retention value is set to be 30 days

### pro\_email\_sttus\_updt

DB schema: **main**  
Used by: **Azure Data Factory/Logic App**  
Context: **Database actions**  
Purpose: Updates email status code.

| **Parameter name** | **Data type** | **Description** | **Sample value** |
| --- | --- | --- | --- |
| in\_parnt\_comp\_exctn\_id | INT | Identifier of callers Component Execution |  |
| in\_user\_name | VARCHAR(50) | Name of the user performing the action |  |
| in\_email\_id | INT | Email ID for update |  |
| in\_email\_sttus\_code | VARCHAR(10) | Status for email (NEW,SENDING,SENT) |  |

Description:  
Procedure performs following steps:

* checks what status was passed into procedure. If not correct for current email status it throws error  
  a) If status is "NEW" then **incoming email\_sttus\_code** should be "SENDING"  
  b) If status is "SENDING" then **incoming email\_sttus\_code** should be "SENT"  
  c) If not above case then throws error
* changes status of email with provided incoming parameter value in table email\_prc

### pro\_file\_actn\_open

DB schema: **main**  
Used by: **FE API**  
Context: **Database**  
Purpose: Insert new Action related with file.

| **Parameter name** | **Data type** | **Description** | **Sample value** |
| --- | --- | --- | --- |
| in\_parnt\_comp\_exctn\_id | INT | Identifier of callers Component Execution |  |
| in\_user\_name | VARCHAR(50) | Name of the user performing the action |  |
| in\_scope\_id | INT | Scope Identifier |  |
| in\_file\_actn\_type\_code | CHAR(1) | Action type code |  |
| out\_file\_actn\_id | INT | Returned action type ID |  |

Description:  
Procedure creates new entry in table md.file\_actn\_plc with status = "A"

### pro\_file\_actn\_close

DB schema: **main**  
Used by: **FE API**  
Context: **Database**  
Purpose: Insert new Action related with file.

| **Parameter name** | **Data type** | **Description** | **Sample value** |
| --- | --- | --- | --- |
| in\_parnt\_comp\_exctn\_id | INT | Identifier of callers Component Execution |  |
| in\_user\_name | VARCHAR(50) | Name of the user performing the action |  |
| in\_file\_actn\_id | INT | File Action Identifier |  |
| in\_sttus\_code | CHAR(1) | Status code to be assigned for closing the file action | C, F |

Description:  
Procedure closes the file action in table md.file\_actn\_plc

### pro\_file\_dfntn\_save

**Summary**

DB schema: **main**  
Used by: **FE API**  
Context: **Admin Panel**  
Purpose: to validate and save into DB Input File definition.

**Parameters**

| **Parameter name** | **Data type** | **Description** | **Sample value** |
| --- | --- | --- | --- |
| in\_parnt\_comp\_exctn\_id | INT | Identifier of callers Component Execution |  |
| in\_user\_name | VARCHAR(50) | Name of the user performing the action |  |
| in\_file\_dfntn\_vers\_id | INT | Identifier of Input File version |  |
| in\_file\_dfntn\_vers\_json\_txt | VARCHAR | complete (not only changes) JSON with attributes of Input File itself and the list of columns with its attributes. See more info below. |  |
| out\_file\_dfntn\_vers\_id | INT | Input File identifier for new version entry or in\_file\_dfntn\_vers\_id if no changes performed |  |
|  |  |  |  |

**Description**

**New file version creation**

Increment the vers\_num value with value of 1 to create a new file version.

**Procedure performs following steps:**

* in\_file\_dfntn\_vers\_json\_txt is checked if valid JSON, completeness, and correctness.
* if in\_file\_dfntn\_vers\_id does not exists throw error.
* if Input File version for in\_file\_dfntn\_vers\_id's definition was changed:
  1. Upsert (update or insert) version in file\_dfntn\_vers\_prc table via pro\_file\_dfntn\_vers\_upsrt and pro\_file\_dfntn\_vers\_col\_upsrt
  2. The old one (in\_file\_dfntn\_vers\_id) is set to sunset (curr\_ind = N).
* links File Definition version with Load Cube via pro\_file\_load\_cube\_rgstr
* If validation fails error is thrown with explanation.

**Checks being performed on Input File Definition:**

* structure of in\_file\_dfntn\_vers\_json\_txt:
  + form\_data: input data from file definition file\_dfntn\_vers\_prc\_vw. The content will be a JSON object with "field\_name": value pairs.
  + grid\_data: grid data changes (ie. changes done to the actual data presented on the grid) from file\_dfntn\_vers\_col\_prc\_vw. The content will be a JSON array with objects representing each row from the grid for the specific version.

### pro\_file\_dwnld\_cmplt

DB schema: **main**  
Used by: **FE API**  
Context: **Notifications**  
Purpose: marks file as downloaded and deletes it from Storage Account

| **Parameter name** | **Data type** | **Description** | **Sample value** |
| --- | --- | --- | --- |
| in\_parnt\_comp\_exctn\_id | INT | Identifier of callers Component Execution |  |
| in\_user\_name | VARCHAR(50) | Name of the user performing the action |  |
| in\_file\_dwnld\_id | INT | Notification identifier |  |

Description:  
Updates dwnld\_datetm in file\_dwnld\_prc.

### pro\_file\_dwnld\_creat

DB schema: **main**  
Used by: **ADF**  
Context: **file download**  
Purpose: to track creation of temporary file in Storage Account

| **Parameter name** | **Data type** | **Description** | **Sample value** |
| --- | --- | --- | --- |
| in\_parnt\_comp\_exctn\_id | INT | Identifier of callers Component Execution |  |
| in\_user\_name | VARCHAR(200) | Name of the user performing the action |  |
| in\_orig\_file\_name | VARCHAR(200) | Storage Account location including container and path |  |
| in\_disp\_file\_name | VARCHAR(200) | Default name for local saving of the file by user |  |
| out\_file\_dwnld\_id | INT | ID of newly created entry |  |

Description:  
Inserts record into file\_dwnld\_prc.

### pro\_file\_dwnld\_ready

DB schema: **main**  
Used by: **ADF**  
Context: **file download**  
Purpose: to notify user when file is ready for download.

| **Parameter name** | **Data type** | **Description** | **Sample value** |
| --- | --- | --- | --- |
| in\_parnt\_comp\_exctn\_id | INT | Identifier of callers Component Execution |  |
| in\_user\_name | VARCHAR(50) | Name of the user performing the action |  |
| in\_file\_dwnld\_id | INT | Notification identifier |  |

Description:  
Updates ready\_datetm field in [file\_dwnld\_prc](https://dev.azure.com/dh-platforms-devops/app-cngc-dataservice-Notification entry with devops/\_wiki/wikis/app-cngc-dataservice-devops.wiki/4429/file\_dwnld\_prc) entry.  
Creates new Notification entry with pro\_notif\_upsrt

### pro\_file\_filtr\_to\_sql

| **Parameter name** | **Data type** | **Description** | **Sample value** |
| --- | --- | --- | --- |
| in\_parnt\_comp\_exctn\_id | INT | Identifier of callers Component Execution |  |
| in\_user\_name | VARCHAR(50) | Name of the user performing the action |  |
| in\_scope\_id | INT | Scope identifier |  |
| in\_filtr\_json\_txt | VARCHAR | JSON describing what filters were used when requesting download as defined in fl-FileDownload |  |
| out\_filtr\_sql\_txt | VARCHAR | in\_filtr\_json\_txt converted into SQL condition, contains also Market condition |  |
| out\_ordr\_sql\_txt | VARCHAR | ORDER BY clause extracted from in\_filtr\_json\_txt |  |

Creates SQL clauses to be used in WHERE and ORDER BY upon downloading Input File content.  
If scope represents Market level (mkt\_id IS NOT NULL), out\_filtr\_sql\_txt should also have added condition "sys\_mkt\_id = <mkt\_id>".

### pro\_file\_load\_cube\_rgstr

DB schema: **main**  
Used by: **DB internal**  
Context: **File Definition**  
Purpose: Links the File Definition version with Load Cube

| **Parameter name** | **Data type** | **Description** | **Sample value** |
| --- | --- | --- | --- |
| in\_parnt\_comp\_exctn\_id | INT | Identifier of callers Component Execution |  |
| in\_user\_name | VARCHAR(50) | Name of the user performing the action |  |
| file\_dfntn\_vers\_id | INT | File Definition version ID |  |

Description:

* looks for the Load Cube which contains all Load Columns being part of the File Definition
* creates Load Cube table in Market Group with pro\_mkt\_grp\_load\_cube\_tbl\_creat if the table does not exist yet
* updates file\_dfntn\_vers\_prc.load\_cube\_id

### pro\_file\_sbmt\_tbl\_creat

Params:  
in\_file\_dfntn\_vers\_id

Steps:

* if table exists:
  + truncate the table file\_dfntn\_vers\_prc\_vw. sbmt\_curr\_snpsh\_tbl\_name
* else
  + create Submit Table with pro\_file\_tbl\_creat (in\_sbmt\_tbl\_ind=Y) under name file\_dfntn\_vers\_prc\_vw.sbmt\_curr\_snpsh\_tbl\_name
  + switch synonym file\_dfntn\_vers\_prc\_vw.smbt\_synm\_name to the current snapshot
* insert all Valid and Non-Obsolete rows into this table (list of columns for dynamic query get with fn\_get\_col\_list again with in\_sbmt\_tbl\_ind=Y)

Existing field file\_dfntn\_vers\_prc\_vw.sbmt\_tbl\_name to be replaced with:

* sbmt\_curr\_snpsh\_tbl\_name (ex. input\_0021\_0009\_sbmt\_202110\_fct)
* sbmt\_curr\_snpsh\_tbl\_mask\_txt (ex. input\_0021\_0009\_sbmt\_<MONTH>\_fct) (for future use to remove old snapshots)
* smbt\_synm\_name (ex. input\_0021\_0009\_sbmt\_fct)  
  Fields should also work for Config Files in some way.

Snapshot tables and synonym should be maintained in sbmt schema.

### pro\_file\_stage\_creat

DB schema: **main**  
Used by: **DB internal and ADF**  
Context: **logging**  
Purpose: to create staging table for uploading data into DB

| **Parameter name** | **Data type** | **Description** | **Sample value** |
| --- | --- | --- | --- |
| in\_parnt\_comp\_exctn\_id | INT | Identifier of callers Component Execution |  |
| in\_user\_name | VARCHAR(50) | Name of the user performing the action |  |
| in\_file\_dfntn\_vers\_id | INT | Version of Input File definition |  |
| in\_data\_type\_rflcd\_ind | CHAR(1) | Should mirroring stage table have column types just like working table (Y) or have them all as VARCHARs to allow inserting any value (N) |  |
| out\_tbl\_name | VARCHAR | Name of the table created in stage schema |  |

Description:  
Procedure creates in schema stage table <work\_tbl\_name>\_ <main\_comp\_exctn\_id\_lpadded\_to\_10\_digits>\_<YYYYMMHH24MISS>\_sfct having column list just like work table name for in\_file\_dfntn\_vers\_id (file\_dfntn\_vers\_prc\_vw.work\_tbl\_name).  
Depending on the in\_data\_type\_rflctd\_ind columns will be just like in original table or unified to VARCHAR(MAX).  
ADF user is granted INSERT, UPDATE and SELECT privs to the table.

### pro\_file\_tbl\_creat

DB schema: **main**  
Used by: **DB Internally**  
Context: **Input File maintenance**  
Purpose: to create any table related with Input File

| **Parameter name** | **Data type** | **Description** | **Sample value** |
| --- | --- | --- | --- |
| in\_parnt\_comp\_exctn\_id | INT | Identifier of callers Component Execution |  |
| in\_user\_name | VARCHAR(50) | Name of the user performing the action |  |
| in\_file\_dfntn\_vers\_id | INT | Version of Input File definition |  |
| in\_schma\_name | VARCHAR(50) | Name of the schema to create the table |  |
| in\_tbl\_name | VARCHAR(MAX) | Name of the table to be created |  |
| in\_data\_type\_rflcd\_ind | CHAR(1) | Should mirroring stage table have column types just like working table (Y) or have them all as VARCHARs to allow inserting any value (N) |  |
| in\_load\_col\_ind | CHAR(1) | Should list of columns be restricted to those having certain value of load\_ind. Not taken into account if NULL. |  |
| in\_work\_tbl\_ind | CHAR(1) | Should list of columns be restricted to those having certain value of work\_tbl\_ind. Not taken into account if NULL. |  |
| in\_sbmt\_tbl\_ind | CHAR(1) | Should list of columns be restricted to those having certain value of sbmt\_tbl\_ind. Not taken into account if NULL. |  |

Description:  
Procedure creates a table representing given version of the Input File.  
Additional parameters determine:

* schema in which table should be created
* table name
* which column types are to be included in definition  
  If the table exists, nothing happens.

### pro\_file\_tbl\_pk\_creat

DB schema: **main**  
Used by: **DB Internally**  
Context: **Input File maintenance**  
Purpose: to create primary key based on **sys\_row\_id** column for any table

| **Parameter name** | **Data type** | **Description** | **Sample value** |
| --- | --- | --- | --- |
| in\_parnt\_comp\_exctn\_id | INT | Identifier of callers Component Execution | 1 |
| in\_user\_name | VARCHAR(50) | Name of the user performing the action | doe.j |
| in\_schma\_name | VARCHAR(50) | Name of the schema | input |
| in\_tbl\_name | VARCHAR(MAX) | Name of the table to have PK generated | input\_0001\_0004\_work\_fct |

Description:  
Procedure creates primary key on **sys\_row\_id** column (hardcoded) for given table on selected db schema.

If primary key exists, nothing happens.

### pro\_file\_uplod

DB schema: **main**  
Used by: **DB internal and ADF**  
Context: **File Upload**  
Purpose: to merge stage table content with working table

| **Parameter name** | **Data type** | **Description** | **Sample value** |
| --- | --- | --- | --- |
| in\_parnt\_comp\_exctn\_id | INT | Identifier of callers Component Execution |  |
| in\_user\_name | VARCHAR(50) | Name of the user performing the action |  |
| in\_uplod\_actn\_id | INT | ID of File Action |  |
| in\_stage\_tbl\_name | VARCHAR(50) | Name of the staging table |  |

Description:  
Procedures is part of Upload processing. It merges content of stage table into working table performing those DQ validations which are necessary prior to changing content of work table.

Steps:

* **Step 1:** performs DQ validation of data type correctness with pro\_file\_vldtn\_joint\_exct with in\_data\_type\_ind=Y
* **Step 2:** creates temp stage table (temp\_tbl1) with proper data types with pro\_file\_stage\_creat
* **Step 3:** inserts into temp\_tbl1 all rows from in\_stage\_tbl\_name
* **Step 4:** performs DQ validations having new\_rows\_only\_ind=Y with pro\_file\_vldtn\_joint\_exct on temp\_tbl1
* **Step 5:** creates temp stage table (temp\_tbl2) with proper data types with pro\_file\_stage\_creat
* **Step 6:** inserts into temp\_tbl2 merged data from working table and temp\_tbl1 with rows joined on sys\_row\_id:
  + rows only in Work Table remain untouched
  + rows only in temp\_tbl1 are inserted for Non-Load Input File with:
    - new sys\_row\_ids
    - updated sys\_init\_actn\_id and sys\_last\_uplod\_actn\_id
  + rows matching both:
    - load columns values from Work Table
    - custom columns from temp\_tbl1
    - updated sys\_last\_uplod\_actn\_id
* **Step 7:** performs DQ validations having new\_rows\_only\_ind=N with pro\_file\_vldtn\_joint\_exct on temp\_tbl2
* **Step 8:** if no critical failure in DQ\_CHECK\_EXCTN\_PLC with crtcl\_uplod\_ind=Y:
  + work table is swapped with temp\_tbl2
* else action is closed with F
* **Step 9:** drops stage tables: temp\_tbl1, temp\_tbl2 and in\_stage\_tbl\_name

### pro\_file\_vldtn\_data\_type\_bool

DB schema: **main**  
Used by: **DB internal and ADF**  
Context: **File Upload**  
Purpose: **Check data correctness**

| **Parameter name** | **Data type** | **Description** | **Sample value** |
| --- | --- | --- | --- |
| in\_parnt\_comp\_exctn\_id | INT | Identifier of callers Component Execution |  |
| in\_user\_name | VARCHAR(50) | Name of the user performing the action |  |
| in\_tbl\_name | VARCHAR | Uploaded table name |  |
| in\_dq\_check\_id | INT | DQ Check ID |  |
| in\_file\_dfntn\_vers\_id | INT | File version ID |  |
| in\_file\_dwnld\_id | INT | File download ID |  |
| in\_file\_actn\_id | INT | File action ID |  |

Description:  
Procedure checks if data in boolean column has values **Y/N**.

### pro\_file\_vldtn\_data\_type\_date

DB schema: **main**  
Used by: **DB internal and ADF**  
Context: **File Upload**  
Purpose: **Check data correctness**

| **Parameter name** | **Data type** | **Description** | **Sample value** |
| --- | --- | --- | --- |
| in\_parnt\_comp\_exctn\_id | INT | Identifier of callers Component Execution |  |
| in\_user\_name | VARCHAR(50) | Name of the user performing the action |  |
| in\_tbl\_name | VARCHAR | Uploaded table name |  |
| in\_dq\_check\_id | INT | DQ Check ID |  |
| in\_file\_dfntn\_vers\_id | INT | File version ID |  |
| in\_file\_dwnld\_id | INT | File download ID |  |
| in\_file\_actn\_id | INT | File action ID |  |

Description:  
Procedure checks if data in date column is in line with Data types

### pro\_file\_vldtn\_data\_type\_int

DB schema: **main**  
Used by: **DB internal and ADF**  
Context: **File Upload**  
Purpose: **Check data correctness**

| **Parameter name** | **Data type** | **Description** | **Sample value** |
| --- | --- | --- | --- |
| in\_parnt\_comp\_exctn\_id | INT | Identifier of callers Component Execution |  |
| in\_user\_name | VARCHAR(50) | Name of the user performing the action |  |
| in\_tbl\_name | VARCHAR | Uploaded table name |  |
| in\_dq\_check\_id | INT | DQ Check ID |  |
| in\_file\_dfntn\_vers\_id | INT | File version ID |  |
| in\_file\_dwnld\_id | INT | File download ID |  |
| in\_file\_actn\_id | INT | File action ID |  |

Description:  
Procedure checks if data in column is integer.

### pro\_file\_vldtn\_data\_type\_mth

DB schema: **main**  
Used by: **DB internal and ADF**  
Context: **File Upload**  
Purpose: **Check data correctness**

| **Parameter name** | **Data type** | **Description** | **Sample value** |
| --- | --- | --- | --- |
| in\_parnt\_comp\_exctn\_id | INT | Identifier of callers Component Execution |  |
| in\_user\_name | VARCHAR(50) | Name of the user performing the action |  |
| in\_tbl\_name | VARCHAR | Uploaded table name |  |
| in\_dq\_check\_id | INT | DQ Check ID |  |
| in\_file\_dfntn\_vers\_id | INT | File version ID |  |
| in\_file\_dwnld\_id | INT | File download ID |  |
| in\_file\_actn\_id | INT | File action ID |  |

Description:  
Procedure checks if data in date column is in line with Data types

### pro\_file\_vldtn\_data\_type\_num

DB schema: **main**  
Used by: **DB internal and ADF**  
Context: **File Upload**  
Purpose: **Check data correctness**

| **Parameter name** | **Data type** | **Description** | **Sample value** |
| --- | --- | --- | --- |
| in\_parnt\_comp\_exctn\_id | INT | Identifier of callers Component Execution |  |
| in\_user\_name | VARCHAR(50) | Name of the user performing the action |  |
| in\_tbl\_name | VARCHAR | Uploaded table name |  |
| in\_dq\_check\_id | INT | DQ Check ID |  |
| in\_file\_dfntn\_vers\_id | INT | File version ID |  |
| in\_file\_dwnld\_id | INT | File download ID |  |
| in\_file\_actn\_id | INT | File action ID |  |

Description:  
Procedure checks if data in number column is in line with Data types

### pro\_file\_vldtn\_data\_type\_perc

DB schema: **main**  
Used by: **DB internal and ADF**  
Context: **File Upload**  
Purpose: **Check data correctness**

| **Parameter name** | **Data type** | **Description** | **Sample value** |
| --- | --- | --- | --- |
| in\_parnt\_comp\_exctn\_id | INT | Identifier of callers Component Execution |  |
| in\_user\_name | VARCHAR(50) | Name of the user performing the action |  |
| in\_tbl\_name | VARCHAR | Uploaded table name |  |
| in\_dq\_check\_id | INT | DQ Check ID |  |
| in\_file\_dfntn\_vers\_id | INT | File version ID |  |
| in\_file\_dwnld\_id | INT | File download ID |  |
| in\_file\_actn\_id | INT | File action ID |  |

Description:  
Procedure checks if data in percentage column is in line with Data types

### pro\_file\_vldtn\_data\_type\_txt

DB schema: **main**  
Used by: **DB internal and ADF**  
Context: **File Upload**  
Purpose: **Check data correctness**

| **Parameter name** | **Data type** | **Description** | **Sample value** |
| --- | --- | --- | --- |
| in\_parnt\_comp\_exctn\_id | INT | Identifier of callers Component Execution |  |
| in\_user\_name | VARCHAR(50) | Name of the user performing the action |  |
| in\_tbl\_name | VARCHAR | Uploaded table name |  |
| in\_dq\_check\_id | INT | DQ Check ID |  |
| in\_file\_dfntn\_vers\_id | INT | File version ID |  |
| in\_file\_dwnld\_id | INT | File download ID |  |
| in\_file\_actn\_id | INT | File action ID |  |

Description:  
Procedure checks if data in text column is proper length.

### pro\_file\_vldtn\_email\_creat

DB schema: **main**  
Used by: **Azure Data Factory**  
Context: **Database actions**  
Purpose: Prepare DQ check report in HTML format

| **Parameter name** | **Data type** | **Description** | **Sample value** |
| --- | --- | --- | --- |
| in\_parnt\_comp\_exctn\_id | INT | Identifier of callers Component Execution |  |
| in\_user\_name | VARCHAR(50) | Name of the user performing the action |  |
| in\_file\_actn\_id | INT | File action identifier |  |

Description:  
Procedure performs following steps:

* check if there are failed data type validation for the provided file\_actn\_id in dq\_check\_exctn\_plc.
* if found, creates the report for data type validation and creates an entry in email\_prc.
* if there are no failed data type validation for the provided file\_actn\_id, proceed in checking status of business validation and creates an entry in email\_prc with generated html report.
* overall status in the report is identified by file\_dfntn\_vers\_prc\_vw.file\_sttus\_code

### pro\_file\_vldtn\_exct

Params:

* in\_comp\_exctn\_id
* in\_user\_name
* in\_tbl\_name
* in\_dq\_check\_id
* in\_upload\_actn\_id

Steps:

* creates entry in dq\_check\_exctn\_plc
* executes db\_proc\_name for given dq\_check\_type(\_lkp) passing all necessary params to it.

**Below Logic to following ( as reference, can be modified)**

select \* from md.dq\_check\_type\_lkp a  
join md.dq\_check\_prc b on a.dq\_check\_type\_code = b.dq\_check\_type\_code;

select distinct a.db\_proc\_name, a.vldtn\_class\_code from md.dq\_check\_type\_lkp a  
join md.dq\_check\_prc b on a.dq\_check\_type\_code = b.dq\_check\_type\_code;

exec pro\_dup\_check @parameters

LOOP

execute procedure to check DQ check

Save NAME OF DQ procedure which failed and value (true false) into some result variable

END

FInal step:

Sum of all TRUE FALSE and see if we have the same ammount as exected procedures invoked.

**END**

### pro\_file\_vldtn\_joint\_exct

| **Parameter name** | **Data type** | **Description** | **Sample value** |
| --- | --- | --- | --- |
| in\_parnt\_comp\_exctn\_id | INT | Identifier of callers Component Execution |  |
| in\_user\_name | VARCHAR(50) | Name of the user performing the action |  |
| in\_file\_dfntn\_vers\_id | INT | File definition version identifier |  |
| in\_uplod\_actn\_id | INT | ID of File Action |  |
| in\_crtcl\_upload\_ind | VARCHAR2(1) | Critical upload indicator, might be NULL |  |
| in\_crtcl\_sbmt\_ind | VARCHAR2(1) | Critical submit indicator, might be NULL |  |
| in\_data\_typ\_ind | VARCHAR2(1) | Validation class code: data\_type/upload/submit, might be NULL |  |
| in\_stage\_tbl\_name | VARCHAR(50) | Name of the staging table |  |
| out\_sttus\_val | VARCHAR(50) | Overall Status |  |

Description:  
Executes all data quality check procedures for given in\_file\_dfntn\_vers\_id having set given:

* in\_crtcl\_upload\_ind
* in\_crtcl\_sbmt\_ind
* in\_data\_typ\_ind

According to above parameters' combination, executes pro\_file\_vldtn\_data\_type% procedures (based on entries in md.dq\_check\_type\_lkp).

New view:  
**md.dq\_check\_type\_lkp\_vw**  
to be created, to merge data from md.dq\_check\_prc and md.dq\_check\_type\_lkp (required for cursor to iterate on file's dq checks).

### pro\_file\_work\_tbl\_creat

DB schema: **main**  
Used by: **DB Internally**  
Context: **Input File maintenance**  
Purpose: to create working table for Input File

| **Parameter name** | **Data type** | **Description** | **Sample value** |
| --- | --- | --- | --- |
| in\_parnt\_comp\_exctn\_id | INT | Identifier of callers Component Execution |  |
| in\_user\_name | VARCHAR(50) | Name of the user performing the action |  |
| in\_file\_dfntn\_vers\_id | INT | Version of Input File definition |  |

Description:  
Procedure creates working table for given version of the Input File.

Table has a primary key generated based on sys\_row\_id column.

### pro\_file\_xlsx\_cnvrt\_dq\_parm\_prep

**Summary**

Prepare JSON param for XLSX Converter reference.

**Parameters**

| **Parameter name** | **Data type** | **Description** | **Sample value** |
| --- | --- | --- | --- |
| in\_parnt\_comp\_exctn\_id | INT | Identifier of callers Component Execution |  |
| in\_user\_name | VARCHAR(50) | Name of the user performing the action |  |
| in\_dq\_check\_exctn\_id | INT | File definition version id of the file to perform the conversion upon | 7 |
| out\_parm\_json\_txt | VARCHAR | JSON output of the procedure. It contains file details and its columns | See XLSX converter sample |

**Description**

1. Based on in\_dq\_check\_exctn\_id identify table name for the file (table DQ\_CHECK\_EXCTN\_PLC.RESLT\_TBL\_NAME).
2. Based on table name collect column list from DB system table.
3. For each column, read column label from FILE\_DFNTN\_VER\_COL\_PRC. If label is unavailable, get column name.
4. Format the JSON using the File details and all of its column information.

### pro\_file\_xlsx\_cnvrt\_parm\_prep

**Summary**

Prepare JSON param for XLSX Converter reference.

**Parameters**

| **Parameter name** | **Data type** | **Description** | **Sample value** |
| --- | --- | --- | --- |
| in\_parnt\_comp\_exctn\_id | INT | Identifier of callers Component Execution |  |
| in\_user\_name | VARCHAR(50) | Name of the user performing the action |  |
| in\_file\_dfntn\_vers\_id | INT | File definition version id of the file to perform the conversion upon | 7 |
| in\_file\_actn\_id | INT | Id to identify the specified the action from the list of actions per file version | 38 |
| in\_cnvrt\_type\_code | VARCHAR(50) | Convert type code (CSV\_TO\_CSV, CSV\_TO\_XLSX, XLSX\_TO\_CSV) | CSV\_TO\_CSV |
| out\_parm\_json\_txt | VARCHAR | JSON output of the procedure. It contains file details and its columns | See XLSX converter sample |
|  |  |  |  |

**Description**

1. Identify the source and target type by splitting the in\_cnvrt\_type\_code with the character \_ and setting the value respectively.
2. Format the JSON using the File details and all its column information.

### pro\_geo\_mapng\_reld

DB schema: **main**  
Used by: **ADF**  
Context: **Load Process**  
Purpose: merging new Geo Mapping content produced by Databricks into working Geo Mapping table

| **Parameter name** | **Data type** | **Description** | **Sample value** |
| --- | --- | --- | --- |
| in\_parnt\_comp\_exctn\_id | INT | Identifier of callers Component Execution |  |
| in\_user\_name | VARCHAR(50) | Name of the user performing the action |  |
| in\_stage\_tbl\_name | VARCHAR | Name of staging table in stage schema |  |

Description:  
Procedure performs following steps:

* creates table tmp.<work\_tbl\_name>\_ <main\_comp\_exctn\_id\_lpadded\_to\_10\_digits>\_<YYYYMMHH24MISS>\_sfct mirroring structure of working table (file\_dfntn\_vers\_prc\_vw.work\_tbl\_name)
* left outer joins staging table with working table and inserts result into tmp table
  + manual input column from working table
  + others from stage
* truncates destination working table
* performs **switch operation** between tmp and working tables
* drops temporary table
* drops staging table

### pro\_load\_cube\_col\_upsrt

DB schema: **main**  
Used by: **Init scripts**  
Context: **Load Process**  
Purpose: Inserts row into load\_cube\_col\_prc table

| **Parameter name** | **Data type** | **Description** | **Sample value** |
| --- | --- | --- | --- |
| in\_parnt\_comp\_exctn\_id | INT | Identifier of callers Component Execution |  |
| in\_user\_name | VARCHAR(50) | Name of the user performing the action |  |
| in\_load\_cube\_name | VARCHAR(50) | Name of the Load Cube |  |
| in\_col\_name | VARCHAR(50) | File action ID associated with notification |  |
| in\_col\_num | INT | Column position on the list for the Load Cube |  |

Description:

* Creates or updates row in load\_cube\_col\_prc

### pro\_load\_file\_dfntn\_vers\_rfrsh

DB schema: **main**  
Used by: **Ops Team**  
Context: **reloading Input File**  
Purpose: Update Working Table for single Input File with Load Cube content

| **Parameter name** | **Data type** | **Description** | **Sample value** |
| --- | --- | --- | --- |
| in\_parnt\_comp\_exctn\_id | INT | Identifier of callers Component Execution |  |
| in\_user\_name | VARCHAR | Name of the user performing the action |  |
| in\_file\_dfntn\_vers\_id | INT | File version ID |  |

Steps:

* register file action with pro\_file\_actn\_creat
* create tmp table with distinct on used Load Columns with pro\_load\_grpng\_creat
* create tmp table with pro\_file\_stage\_creat
  + mark all rows not in tmp as obsolete
  + mark all rows in tmp as non-obsolete
  + insert new rows
* switch tmp table with Work Table
* drop tmp table

### pro\_load\_grpng\_creat

DB schema: **main**  
Used by: **Ops Team**  
Context: **reloading Input File**  
Purpose: creates Input File stage table from Load Cube

| **Parameter name** | **Data type** | **Description** | **Sample value** |
| --- | --- | --- | --- |
| in\_parnt\_comp\_exctn\_id | INT | Identifier of callers Component Execution |  |
| in\_user\_name | VARCHAR | Name of the user performing the action |  |
| in\_file\_dfntn\_vers\_id | INT | File version ID |  |

Description:  
Creates stage table with distinct groupings on Load Columns used in particular Input File version selecting data from Load Cube table for Market Group.

Takes into account different fact data needed for various file types:

* for forecast takes only rows having histr\_ind=N
* for actuals takes only rows having shpmt\_ind=Y
* for Input File based on Direct Shipments includes rows having dirct\_indir\_ind=D
* for Input File based on Indirect Shipments includes rows having dirct\_indir\_ind=I

### pro\_load\_scope\_rfrsh

DB schema: **main**  
Used by: **Ops Team**  
Context: **reloading Input File**  
Purpose: Refresh all active Input Files in given Market Group with Load Cube content

| **Parameter name** | **Data type** | **Description** | **Sample value** |
| --- | --- | --- | --- |
| in\_parnt\_comp\_exctn\_id | INT | Identifier of callers Component Execution |  |
| in\_user\_name | VARCHAR | Name of the user performing the action |  |
| in\_regn\_id | INT | Identifier of Region |  |
| in\_regn\_name | VARCHAR | Region name |  |
| in\_mkt\_grp\_id | INT | Identifier of Market Group |  |
| in\_mkt\_grp\_name | VARCHAR | Market Group name |  |

Steps:

* for each current IF version of Load Input File matching given Market Group/Region parameters executes pro\_load\_file\_dfntn\_vers\_rfrsh  
  It's enough if any of those four parameters is set.

### pro\_log\_dynmc\_sql

DB schema: **main**  
Used by: **ADF, DB**  
Context: **Load Process**  
Purpose: Running **dynamic** SQL generated by BE code.

| **Parameter name** | **Data type** | **Description** | **Sample value** |
| --- | --- | --- | --- |
| comp\_exctn\_id | INT | Identifier of callers Component Execution |  |
| sql\_txt | VARCHAR(MAX) | Dynamic SQL generated for execution |  |

Description:

* Provided sql\_txt is logged into pro\_log\_msg table
* Provided statement is executed
* If statement returns rows then pro\_log\_msg is called to log result

BE Developers:

* Procedure does not call **Component Execution procs (open, close)**.

### pro\_log\_msg

DB schema: **main**  
Used by: **Database**  
Context: **Database actions**  
Purpose: Logging actions performed on database

| **Parameter name** | **Data type** | **Description** | **Sample value** |
| --- | --- | --- | --- |
| in\_comp\_exctn\_id | INT | Unique process identity which performed action on database |  |
| in\_sttus\_code | CHAR(3) | Status of action performed (OK,ERR) |  |
| in\_msg\_txt | VARCHAR(MAX) | Detailed information about action that was performed on database |  |

Description:  
Procedure performs following steps:

* Takes incoming parameters and inserts record into commn\_log\_plc table

### pro\_mkt\_grp\_load\_cube\_tbl\_creat

DB schema: **main**  
Used by: **DB internal**  
Context: **File Definition**  
Purpose: Creates staging table for given Load Cube in Market Group

| **Parameter name** | **Data type** | **Description** | **Sample value** |
| --- | --- | --- | --- |
| in\_parnt\_comp\_exctn\_id | INT | Identifier of callers Component Execution |  |
| in\_user\_name | VARCHAR(50) | Name of the user performing the action |  |
| mkt\_grp\_id | INT | Market Group ID |  |
| load\_cube\_id | INT | Load Cube ID |  |

Description:  
Creates staging table if it does not exist.

### pro\_notif\_mark\_read

DB schema: **main**  
Used by: **FE API**  
Context: **Notifications**  
Purpose: marks Notification as read

| **Parameter name** | **Data type** | **Description** | **Sample value** |
| --- | --- | --- | --- |
| in\_parnt\_comp\_exctn\_id | INT | Identifier of callers Component Execution |  |
| in\_user\_name | VARCHAR(50) | Name of the user performing the action |  |
| in\_notif\_id | INT | Notification identifier |  |

**Description**:  
Updates record in notif\_prc setting read\_datetm to current date, throws error if not found.

If in\_notif\_id parameter is null, then all unread notifications should be marked as read. This is needed to cover the case, where user wants to reset the unread notifications counter (= read all unread notifications in one batch).

### pro\_notif\_upsrt

DB schema: **main**  
Used by: **FE API**  
Context: **Database**  
Purpose: Insert or updates notification instance

| **Parameter name** | **Data type** | **Description** | **Sample value** |
| --- | --- | --- | --- |
| in\_parnt\_comp\_exctn\_id | INT | Identifier of callers Component Execution |  |
| in\_user\_name | VARCHAR(50) | Name of the user performing the action |  |
| in\_notif\_id | INT | Notification Identifier |  |
| in\_file\_actn\_id | INT | File action ID associated with notification |  |
| in\_notif\_user\_name | VARCHAR(50) | Name of the user related with notification |  |
| in\_sttus\_txt | TEXT(50) | Status of notification |  |
| in\_desc\_txt | TEXT(1000) | Description of notification |  |

Description:

* Creates or updates row in notif\_prc

## Procedures for metadata maintenance

### pro\_dq\_bus\_check\_tnsfr

DB schema: **md**  
Used by: **BE**  
Context: **metadata maintenance**  
Purpose: **creates business dq conditions V01 and V02**

| **Parameter name** | **Data type** | **Description** | **Sample value** |
| --- | --- | --- | --- |
| in\_parnt\_comp\_exctn\_id | INT | Identifier of callers Component Execution |  |
| in\_user\_name | VARCHAR | Name of the user performing the action |  |
| in\_file\_dfntn\_vers\_id | INT | File version ID |  |
| in\_load\_ind | VARCHAR | BOOLEAN | Y/N indicating file type |

Description:

This procedure is responsible for creating business conditions V01 and V02 described here DQ conditions when new version of template is created.

Creation:

* file type is LOAD then V01 is created.
* V02 is created for all required and visible columns in template

### pro\_comp\_exctn\_open

DB schema: **main**  
Used by: **DB internal and ADF**  
Context: **logging**  
Purpose: to register execution of code unit

| **Parameter name** | **Data type** | **Description** | **Sample value** |
| --- | --- | --- | --- |
| in\_parnt\_comp\_exctn\_id | INT | Identifier of callers Component Execution |  |
| in\_user\_name | VARCHAR(50) | Name of the user performing the action |  |
| in\_scope\_id | INT | Scope identifier |  |
| in\_db\_proc\_name | VARCHAR(50) | Name of procedure which created log |  |
| in\_adf\_pipln\_name | VARCHAR(50) | Name of the pipeline from Azure Data Factory |  |
| in\_param\_json\_txt | VARCHAR(MAX) | JSON describing input parameters for the Component Execution |  |
| in\_adf\_pipln\_run\_id | VARCHAR(50) | ID of ADF pipeline execution |  |
| out\_param\_json\_txt | VARCHAR(MAX) | JSON describing Objects and Component Params |  |
| out\_comp\_exctn\_id | INT | ID of this Component Execution |  |

Steps:

* checks if in\_parnt\_comp\_exctn\_id or in\_user\_name is not NULL (otherwise throws error)
* if in\_scope\_id is NULL, than global scope (record with all FKs NULL in scope\_prc) should be looked for with fn\_get\_scope
* creates entry in comp\_exctn\_prc
* [in future] acquires locks for all objects linked with comp\_obj\_lkp
* [in future] reads parameters stored in comp\_param and obj\_param\_prc linking with comp\_obj\_prm and stores them in out\_param\_json\_txt

### pro\_comp\_exctn\_close

DB schema: **main**  
Used by: **DB internal and ADF**  
Context: **logging**  
Purpose: to close execution of code unit

| **Parameter name** | **Data type** | **Description** | **Sample value** |
| --- | --- | --- | --- |
| in\_comp\_exctn\_id | INT | ID of this Component Execution |  |
| in\_sttus\_code | CHAR(1) | Completion status: C - completed, F - failed |  |
| in\_err\_msg\_txt | VARCHAR(MAX) | Additional failure message |  |

Steps:

* closes any unfinished file action in file\_actn\_plc for comp\_exctn\_id by setting end\_datetm and sttus\_code
* sets end\_datetm and sttus\_code in comp\_exctn\_prc

### pro\_dq\_check\_cond\_upsrt

DB schema: **md**  
Used by: **Ops Team**  
Context: **metadata maintenance**  
Purpose: adds conditions to dq\_check\_prc table from JSON FE file

| **Parameter name** | **Data type** | **Description** | **Sample value** |
| --- | --- | --- | --- |
| in\_parnt\_comp\_exctn\_id | INT | Identifier of callers Component Execution |  |
| in\_user\_name | VARCHAR | Name of the user performing the action |  |
| in\_json\_dq\_strct | VARCHAR | JSON structure with all selected, active conditions from FE |  |
| in\_file\_dfntn\_vers\_id | INT | File version ID |  |
| in\_cnfg\_ind | CHAR | Configuration indicator (Y/N) |  |

Description:  
This procedure inserts data to dq\_check\_prc from provided JSON file format from FE application

### pro\_dq\_check\_upsrt

DB schema: **md**  
Used by: **Ops Team**  
Context: **metadata maintenance**  
Purpose: creates or updates record in load\_col\_regn\_lkp

| **Parameter name** | **Data type** | **Description** | **Sample value** |
| --- | --- | --- | --- |
| in\_parnt\_comp\_exctn\_id | INT | Identifier of callers Component Execution |  |
| in\_user\_name | VARCHAR | Name of the user performing the action |  |
| in\_regn\_name | VARCHAR | Region name |  |
| in\_col\_name | VARCHAR | Column name |  |
| in\_col\_num | CHAR | Number of column (Load Process Cube) |  |

Description:  
This procedure inserts data to load\_col\_regn\_lkp table provided by init script: init\_load\_col\_regn\_lkp.sql (Visual Studio script) with definition of Load Process Cube

### pro\_load\_col\_regn\_upsrt

DB schema: **md**  
Used by: **Ops Team**  
Context: **metadata maintenance**  
Purpose: creates or updates record in load\_col\_regn\_lkp

| **Parameter name** | **Data type** | **Description** | **Sample value** |
| --- | --- | --- | --- |
| in\_parnt\_comp\_exctn\_id | INT | Identifier of callers Component Execution |  |
| in\_user\_name | VARCHAR | Name of the user performing the action |  |
| in\_regn\_name | VARCHAR | Region name |  |
| in\_col\_name | VARCHAR | Column name |  |
| in\_col\_num | CHAR | Number of column (Load Process Cube) |  |

Description:  
This procedure inserts data to load\_col\_regn\_lkp table provided by init script: init\_load\_col\_regn\_lkp.sql (Visual Studio script) with definition of Load Process Cube

### pro\_comp\_upsrt

Params:  
in\_adf\_pipln\_name  
in\_db\_proc\_name

Steps:

* creates row in comp\_lkp

### pro\_file\_dfntn\_upsrt

DB schema: **main**  
Used by: **Ops Team**  
Context: **metadata maintenance**  
Purpose: creates or updates record in file\_dfntn\_prc and scope\_prc

| **Parameter name** | **Data type** | **Description** | **Sample value** |
| --- | --- | --- | --- |
| in\_parnt\_comp\_exctn\_id | INT | Identifier of callers Component Execution |  |
| in\_user\_name | VARCHAR | Name of the user performing the action |  |
| in\_regn\_name | VARCHAR | Region name |  |
| in\_mkt\_grp\_name | VARCHAR | Market Group name |  |
| in\_file\_name | VARCHAR | File Name |  |
| in\_new\_file\_name | VARCHAR | New File Name |  |
| in\_mkt\_name | VARCHAR | Market Name |  |
| in\_cnfg\_ind | CHAR | Configuration Indicator (Y/N) |  |
| in\_frcst\_ind | CHAR | Forecast Indicator (Y/N) |  |
| in\_tbl\_name | VARCHAR | Table Name |  |
| in\_activ\_ind | CHAR | Indicates if File Definition is being active (Y/N) |  |
| in\_vsbl\_ind | CHAR | Indicates if File Definition is being visible (Y/N) |  |

Description:

* check if row exist in file\_dfntn\_prc\_vw
* if does not exist in transaction:
  + if Market Group does not exist, throws an error
  + inserts the row into file\_dfntn\_prc
  + calls pro\_scope\_upsrt
* else checks what columns need to be updated (if any) and performs the update  
  (cnfg\_ind, frcst\_ind and tbl\_name should not be updatable)

### pro\_file\_dfntn\_vers\_col\_upsrt

DB schema: **main**  
Used by: **Ops Team**  
Context: **metadata maintenance**  
Purpose: creates or updates record in file\_dfntn\_vers\_col\_prc

| **Parameter name** | **Data type** | **Description** | **Sample value** |
| --- | --- | --- | --- |
| in\_parnt\_comp\_exctn\_id | INT | Identifier of callers Component Execution |  |
| in\_user\_name | VARCHAR | Name of the user performing the action |  |
| in\_regn\_name | VARCHAR | Region name |  |
| in\_mkt\_grp\_name | VARCHAR | Market Group name |  |
| in\_file\_name | VARCHAR | File Name |  |
| in\_vers\_num | INT | Indicates the version number of a file |  |
| in\_col\_name | VARCHAR | Column Name |  |
| in\_load\_col\_name | VARCHAR | Load Column Name |  |
| in\_sys\_col\_name | VARCHAR | System Column Name |  |
| in\_col\_label | VARCHAR | Column Label |  |
| in\_col\_num | INT | Column Number |  |
| in\_key\_ind | CHAR | Key Indicator (Y/N) |  |
| in\_reqd\_ind | CHAR | Required Indicator (Y/N) |  |
| in\_hdn\_ind | CHAR | Hidden Indicator (Y/N) |  |
| in\_col\_type\_name | VARCHAR | Column Type Name |  |
| in\_lngth\_val | INT | Indicates the length value |  |
| in\_prcsn\_val | INT | Indicates the precision value |  |
| in\_scale\_val | INT | Indicates the scale value |  |

Description:

* check if rows exist via pro\_file\_dfntn\_vers\_col\_vw
* if does not exist in transaction:
  + if Input File Version does not exist, throws an error
  + inserts the row into pro\_file\_dfntn\_vers\_col\_prc
  + inserts a row into dq\_check\_prc for the column with pro\_dq\_check\_upsrt if column is one of below:
    - sys\_invld\_ind
    - Non-Load and Non-System column
* else checks what columns need to be updated (if any) and performs the update.

### pro\_file\_dfntn\_vers\_upsrt

DB schema: **main**  
Used by: **Ops Team**  
Context: **metadata maintenance**  
Purpose: creates or updates record in file\_dfntn\_vers\_prc and scope\_prc

| **Parameter name** | **Data type** | **Description** | **Sample value** |
| --- | --- | --- | --- |
| in\_parnt\_comp\_exctn\_id | INT | Identifier of callers Component Execution |  |
| in\_user\_name | VARCHAR | Name of the user performing the action |  |
| in\_regn\_name | VARCHAR | Region name |  |
| in\_mkt\_grp\_name | VARCHAR | Market Group name |  |
| in\_file\_name | VARCHAR | File Name |  |
| in\_vers\_num | INT | Indicates the version number of a file |  |
| in\_mkt\_name | VARCHAR | Market Name |  |
| in\_file\_desc | VARCHAR | Description of the file |  |
| in\_obs\_ind | CHAR |  |  |
| in\_invld\_ind | CHAR | Indicates if the File version is invalid (Y/N) |  |
| in\_mkt\_col\_name | VARCHAR | Market Column Name |  |
| in\_curr\_ind | CHAR | Current Indicator (Y/N) |  |
| in\_dirct\_ind | CHAR | Direct Indicator (Y/N) |  |
| in\_indir\_ind | CHAR | Indirect Indicator (Y/N) |  |

Description:

* check if rows exist in file\_dfntn\_vers\_prc\_vw
* if does not exist in transaction:
  + throws an error if Input File does not exist
  + inserts the row into file\_dfntn\_vers\_prc
  + calls pro\_scope\_upsrt
* else checks what columns need to be updated (if any) and performs the update

Proper mkt\_col\_id needs to be taken from load\_col\_lkp\_vw for provided in\_mkt\_col\_name.

### pro\_mkt\_grp\_upsrt

DB schema: **md**  
Used by: **Ops Team**  
Context: **metadata maintenance**  
Purpose: creates or updates record in mkt\_grp\_lkp

| **Parameter name** | **Data type** | **Description** | **Sample value** |
| --- | --- | --- | --- |
| in\_parnt\_comp\_exctn\_id | INT | Identifier of callers Component Execution |  |
| in\_user\_name | VARCHAR | Name of the user performing the action |  |
| in\_regn\_name | VARCHAR | Region name |  |
| in\_mkt\_grp\_name | VARCHAR | Market Group name |  |
| in\_active\_ind | CHAR | Indicates if Market Group is being active (Y/N) |  |
| in\_indir\_load\_wkday\_num | INT | Working day to trigger Load for Indirects |  |
| in\_due\_date\_wkday\_num | INT | Due date for Input Files in Market Group |  |

Description:

* inserts the record into mkt\_grp\_lkp.
* ensures there's a proper entry in scope\_prc.
* ensures there's a stage table for Market Group Load Cube

### pro\_mkt\_upsrt

DB schema: **md**  
Used by: **Ops Team**  
Context: **metadata maintenance**  
Purpose: creates or updates record in **mkt\_prc**

| **Parameter name** | **Data type** | **Description** | **Sample value** |
| --- | --- | --- | --- |
| in\_parnt\_comp\_exctn\_id | INT | Identifier of callers Component Execution |  |
| in\_user\_name | VARCHAR | Name of the user performing the action |  |
| in\_regn\_name | VARCHAR | Region name |  |
| in\_mkt\_grp\_name | VARCHAR | Market Group name |  |
| @in\_mkt\_name | VARCHAR | Market name |  |
| in\_active\_ind | CHAR | Indicates if Market Group is being active (Y/N) |  |

Steps:

* Checks if row exists via mkt\_vw

1. Inserts the row into pro\_mkt\_prc if not exists
2. If exists then update active\_ind if needed
3. For existing market inserts new scope\_id in scope\_prc for latest file version

### pro\_regn\_upsrt

DB schema: **md**  
Used by: **Ops Team**  
Context: **metadata maintenance**  
Purpose: creates or updates record in regn\_lkp

| **Parameter name** | **Data type** | **Description** | **Sample value** |
| --- | --- | --- | --- |
| in\_parnt\_comp\_exctn\_id | INT | Identifier of callers Component Execution |  |
| in\_user\_name | VARCHAR | Name of the user performing the action |  |
| in\_regn\_name | VARCHAR | Region name |  |

Description:  
Inserts the record into regn\_lkp.  
Ensures there's a proper entry in scope\_prc.

### pro\_scope\_upsrt

DB schema: **main**  
Used by: **Ops Team**  
Context: **metadata maintenance**  
Purpose: creates or updates record in scope\_prc

| **Parameter name** | **Data type** | **Description** | **Sample value** |
| --- | --- | --- | --- |
| in\_parnt\_comp\_exctn\_id | INT | Identifier of callers Component Execution |  |
| in\_user\_name | VARCHAR | Name of the user performing the action |  |
| in\_regn\_name | VARCHAR | Region name |  |
| in\_mkt\_grp\_name | VARCHAR | Market Group name |  |
| in\_file\_name | VARCHAR | File Name |  |
| in\_vers\_num | INT | Indicates the version number of a file |  |
| in\_mkt\_name | VARCHAR | Market Name |  |
| in\_last\_rfrsh\_actn\_id | INT | Indicates the last refresh action |  |
| in\_last\_uplod\_actn\_id | INT | Indicates the last upload action |  |
| in\_last\_sbmt\_actn\_id | INT | Indicates the last submit action |  |

Description:

* check if rows exist via scope\_prc\_vw
* if row does not exist, inserts the row into scope\_prc
* else checks what columns need to be updated (if any) and performs the update

### pro\_user\_scope\_delet

**Summary**

DB schema: **md**  
Used by: **FE API**  
Context: **Privileges**  
Purpose: to remove privileges

**Parameters**

| **Parameter name** | **Data type** | **Description** | **Sample value** |
| --- | --- | --- | --- |
| in\_parnt\_comp\_exctn\_id | INT | Identifier of callers Component Execution |  |
| in\_user\_name | VARCHAR(50) | Name of the user performing the action |  |
| in\_user\_scope\_json\_list | VARCHAR(MAX) | List of pairs (scope ID, user name) | [ ["doe.jd", 3], ["doe.jd", 5], ["doe.jd", 7], ["smith.js", 3], ["smith.js", 5], ["smith.js", 7] ] |

**Description**

Delete records on user\_scope\_prc.

**Database Permission**

For permission, procedure name should be added here - main.sql

### pro\_user\_scope\_set

**Summary**

DB schema: **main**  
Used by: **FE API**  
Context: **Privileges**  
Purpose: to assign/unassign specific privileges

**Parameters**

| **Parameter name** | **Data type** | **Description** | **Sample value** |
| --- | --- | --- | --- |
| in\_parnt\_comp\_exctn\_id | INT | Identifier of callers Component Execution |  |
| in\_user\_name | VARCHAR(50) | Name of the user performing the action |  |
| in\_user\_scope\_json\_list | VARCHAR(MAX) | List of pairs (scope ID, user name) | [ ["doe.jd", 3], ["doe.jd", 5], ["doe.jd", 7], ["smith.js", 3], ["smith.js", 5], ["smith.js", 7] ] |
| in\_sttng\_json\_txt | VARCHAR(MAX) | List all settings which need to be remembered | { "read\_prvlg\_ind": true, "write\_prvlg\_ind": true, "email\_input\_ready\_ind": false, "email\_submt\_ind": false, "emaiil\_dq\_check\_fail\_ind": false, "email\_rstmt\_ind": false } |

**Description**

For each pair user/scope in user\_scope\_json\_list the procedure sets privileges as described in sttng\_json\_txt using pro\_user\_scope\_upsrt whose combinations are contained on md.user\_scope\_prc..  
Scopes need to represent Input File in Market Group or Market level.  
In future can be used to set user mailing preferences and other users attributes.

**Database Permission**

For permission, procedure name should be added here - main.sql

### pro\_user\_scope\_upsrt

**Summary**

DB schema: **md**  
Used by: **FE API**  
Context: **Privileges**  
Purpose: to assign/unassign specific privileges

**Parameters**

| **Parameter name** | **Data type** | **Description** | **Sample value** |
| --- | --- | --- | --- |
| in\_parnt\_comp\_exctn\_id | INT | Identifier of callers Component Execution |  |
| in\_user\_name | VARCHAR(50) | Name of the user performing the action |  |
| in\_asign\_user\_name | VARCHAR(20) | P&G short name |  |
| in\_scope\_id | INT | Scope identifier |  |
| in\_read\_prvlg\_ind | CHAR(1) | Y if user has read access to the Scope |  |
| in\_write\_prvlg\_ind | CHAR(1) | Y if user has write access to the Scope |  |
| in\_email\_input\_ready\_ind | CHAR(1) | Should user receive email notification on data refresh? |  |
| in\_email\_submt\_ind | CHAR(1) | Should user receive email notification after submission in the Scope? |  |
| in\_emaiil\_dq\_check\_fail\_ind | CHAR(1) | Should user receive notification upon validation errors? |  |
| in\_email\_rstmt\_ind | CHAR(1) | Should user be notified in case of restatement (submission after due date) for given Scope? |  |

**Description**

Set privileges of a single record based on the **user\_name** and **scope\_id** combination contained on md.user\_scope\_prc.

**Database Permission**

For permission, procedure name should be added here - main.sql

### pro\_user\_upsrt

**Summary**

DB schema: **main**  
Used by: **FE API**  
Context: **Authentication**  
Purpose: to add user if does not exist and update last login date

**Parameters**

| **Parameter name** | **Data type** | **Description** | **Sample value** |
| --- | --- | --- | --- |
| in\_email\_address | VARCHAR(100) | P&G email address |  |

**Description**

**Steps**

1. Check if the user exists on user\_prc\_vw using the email address provided.
2. If exists, update the last\_login\_datetm using current date.
3. If not yet on the database, insert the new user.

**Database Permission**

For permission, procedure name should be added here - main.sql

## Database Views

### dynmc\_dict\_lkp\_vw

DB schema: **md**  
Used by: **ADF**  
Context: **Home Panel**  
Content: details of dictionaries used to provide data for dropdown

Columns like in dynmc\_dict\_lkp

Description:  
Used by FE to dynamically find details of dictionaries used to fill the dropdowns.

### email\_prc\_vw

DB schema: **md**  
Used by: **ADF**  
Context: **emails**  
Content: List of emails

Columns like in email\_prc

Description:  
Lists of emails to be sent to users with the statuses.

### file\_actn\_extnd\_vw

**Summary**

DB schema: **md**  
Used by: **FE API**  
Context: **file action**  
Content: Extension of file action linking upload action with corresponding submit action (if submit proceeded and no DQ validation errors encountered for the file being uploaded)  
Purpose: used by [main].[pro\_file\_vldtn\_email\_creat] to obtain email title for the upload

**Columns**

| **Column name** | **Description** | **Sample value** |
| --- | --- | --- |
| uplod\_file\_actn\_id | file\_actn\_id of upload activity from file\_actn\_plc\_vw |  |
| file\_sttus\_code | overall upload status used to create email subject |  |
| email\_title\_txt | email subject |  |

### file\_actn\_plc\_vw

**Summary**

DB schema: **md**  
Used by: **FE API**  
Context: **file action**  
Content: File actions of given version of Input File

**Columns**

Columns beside columns already in file\_actn\_plc:

| **Column name** | **Data type** | **Description** | **Sample value** |
| --- | --- | --- | --- |
| scope\_name | VARCHAR2(1000) | From scope\_prc\_vw |  |
| file\_actn\_type\_name | VARCHAR2(20) | From file\_actn\_type\_lkp |  |
| file\_hash\_txt | VARCHAR2(100) | From file\_dwnld\_prc\_vw |  |
| disp\_file\_name | VARCHAR2(100) | From file\_dwnld\_prc\_vw |  |
| user\_name | VARCHAR(20) | From comp\_exctn\_prc | doe.j |
| phys\_file\_name | VARCHAR2(100) | File name with value of: <aVVVVVVVV>\_<AAAAAAAA>\_<YYYYMMDD>\_<hhmmss> where a stands for file\_actn\_type\_code, V stands for file\_dfntn\_vers\_id, A stands for file\_actn\_id, YYYYMMDD for start\_datetm(date) and hhmmss for start\_datetm(time) (because there might be parallel actions for the same file version) | d00000005\_00000119\_20211015\_053813 |

**Description**

Lists attributes of File Action

### file\_dfntn\_col\_type\_lkp\_vw

DB schema: **md**  
Used by: **ADF**  
Context: **emails**  
Content: List of emails

Columns like in file\_dfntn\_col\_type\_lkp

Description:  
Lists of emails to be sent to users with the statuses.

### file\_dfntn\_vers\_col\_prc\_vw

**Columns**

Additional columns to those in the file\_dfntn\_vers\_col\_prc

| **Column name** | **Data type** | **Description** | **Sample value** |
| --- | --- | --- | --- |
| regn\_name | VARCHAR(50) | See regn\_lkp |  |
| mkt\_grp\_name | VARCHAR(50) | See mkt\_grp\_lkp |  |
| file\_name | VARCHAR(100) | See file\_dfntn\_prc |  |
| file\_dfntn\_id | INT |  |  |
| vers\_num | INT |  |  |
| load\_col\_name | VARCHAR(50) |  |  |
| phys\_data\_type\_txt | VARCHAR(34) |  |  |
| col\_srce | VARCHAR(6) |  |  |
| work\_tbl\_ind | CHAR(1) |  |  |
| work\_vw\_ind | CHAR(1) |  |  |
| sbmt\_tbl\_ind | CHAR(1) |  |  |
| load\_col\_ind | CHAR(1) |  |  |

### file\_dfntn\_vers\_mkt\_vw

**Description**

Shows Input File details for each scope on level of IF version or IF version in Market (all scopes having non-NULL file\_dfntn\_vers\_id).  
Has the same set of columns as file\_dfntn\_vers\_prc\_vw plus additionally:

* mkt\_id (from the scope)
* mkt\_name
* sql\_filtr\_cond\_txt: condition to pull file data "sys\_mkt\_id = <mkt\_id>" for scopes on mkt level, otherwise NULL
* mkt\_grp\_scope\_id: scope for Market Group level (equal to scope\_id for Market Group level, or different when on Market Level)

**Columns**

Columns from scope\_prc INNER JOIN file\_dfntn\_vers\_prc\_vw AND LEFT JOIN mkt\_prc with additional below:

| **Column name** | **Data type** | **Description** | **Sample value** |
| --- | --- | --- | --- |
| sql\_filtr\_cond\_txt | VARCHAR | condition to pull file data | sys\_mkt\_id = 1 |

### file\_dfntn\_vers\_prc\_vw

**Summary**

DB schema: **md**  
Used by: **FE API**  
Context: **file definition**  
Content: attributes of given version of Input File

**Columns**

Columns beside columns already in file\_dfntn\_vers\_prc:

| **Column name** | **Data type** | **Description** | **Sample value** |
| --- | --- | --- | --- |
| sbmt\_tbl\_name | VARCHAR | sbmt\_tbl\_name from file\_dfntn\_prc (if not NULL) otherwise input\_NNNN\_MMMM\_sbmt (NNNN and MMMM for mkt\_grp\_id and file\_dfntn\_vers\_id) |  |
| work\_tbl\_name | VARCHAR | work\_tbl\_name from file\_dfntn\_prc (if not NULL) otherwise input\_NNNN\_MMMM\_fct (NNNN and MMMM for mkt\_grp\_id and file\_dfntn\_vers\_id) |  |
| last\_sttus\_chng\_datetm | DATETIME2(7) | Last status change |  |
| last\_user\_actn\_datetm | DATETIME2(7) | Last user action date |  |
| last\_mdfd\_user\_name | VARCHAR | Last modified by |  |
| load\_ind | CHAR(1) | Load indicator ('Y' or 'N') using this logic | Y |
| file\_sttus\_code | VARCHAR | Current status of Input File version explained using this logic | SUBMITTED |

**Description**

Lists of files prepared for download used for Input File download and for downloading of DQ Validation HTML reports.

**Load indicator logic**

**'Y'** if file\_dfntn\_vers\_id has any columns WHERE load\_col\_id IS NOT NULL otherwise, **'N'**

**File Status Code Logic**

Input File status

* **NEW**

load\_ind = 'N' AND file\_actn\_id IS NULL

* **PENDING LOAD**

load\_ind = 'Y' AND file\_actn\_id IS NULL

* **PENDING INPUT**

load\_ind = 'N' AND MONTH(start\_datetm AS latst\_datetm) < MONTH(CURRENT\_TIMESTAMP)

* **SUBMITTED**

file\_actn\_type\_code = 'S' AND sttus\_code = 'C'

* **AVAILABLE**

file\_actn\_type\_code = 'L' AND load\_ind = 'N'

* **UPLOAD IN PROGRESS**

file\_actn\_type\_code = 'U' AND sttus\_code = 'A'

* **FAILED UPLOAD**

file\_actn\_type\_code = 'U' AND sttus\_code = 'F'

* **FAILED VALIDATION**

file\_actn\_type\_code = 'U' AND sttus\_code = 'C' -- AND ?

### file\_dwnld\_prc\_vw

DB schema: **md**  
Used by: **FE API**  
Context: **generic download**  
Content: list of files for generic download

Columns beside columns already in file\_dwnld\_prc:

| **Column name** | **Data type** | **Description** | **Sample value** |
| --- | --- | --- | --- |
| file\_hash\_txt | VARCHAR | Hash key for download links taken as SHA\_256  for concatenation of file\_dwnld\_id and comp\_exctn\_id |  |

Description:  
Lists of files prepared for download used for Input File download and for downloading of DQ Validation HTML reports.

### file\_sttus\_lkp\_vw

DB schema: **md**  
Used by: **FE API**  
Context: **Home Panel**  
Content: list of valid states of the Input File

| **Column name** | **Data type** | **Description** | **Sample value** |
| --- | --- | --- | --- |
| file\_sttus\_code | VARCHAR | Code of file status |  |

Description:  
List of allowed statuses of Input File.

### load\_col\_lkp\_vw

**Summary**

DB schema: **md**  
Used by: **FE API**  
Context: **Load**  
Content: List of load columns

**Columns**

Columns beside columns already in load\_col\_lkp:

| **Column name** | **Data type** | **Description** | **Sample value** |
| --- | --- | --- | --- |
| phys\_data\_type | VARCHAR | Physical data type | INT |
|  |  |  |  |

**Description**

Gets the list of Load Columns

**Column phys\_data\_type logic**

Source: load\_col\_lkp\_vw.sql

| **col\_type\_name value** | **phys\_data\_type value** |
| --- | --- |
| INTEGER | INT |
| TEXT | VARCHAR(lngth\_val). See lngth\_val |
| NUMBER | NUMERIC(prcsn\_val, scale\_val). See prcsn\_val and scale\_val |
| BOOLEAN | CHAR(1) |
|  |  |

### load\_col\_regn\_lkp\_vw

DB schema: **md**  
Used by: **ADF**  
Context: **Load**  
Content: List of columns in load cube for given region

Columns beside columns already in load\_col\_regn\_lkp:

| **Column name** | **Data type** | **Description** | **Sample value** |
| --- | --- | --- | --- |
| col\_name | VARCHAR | name of the column |  |
| regn\_name | VARCHAR | name of the Region |  |

Description:  
There will be entry for each column/region combination as in description.  
Sample data:

| **Column name** | **Region name** |
| --- | --- |
| fpc\_id | AMA |
| fpc\_id | EU |
| regn\_id | AMA |
| regn\_id | EU |
| regn\_id | LA |

### mkt\_grp\_lkp\_vw

DB schema: **md**  
Used by: **FE API**  
Context: **Home Panel**  
Content: list of Market Groups

Additional fields:

| **Column name** | **Data type** | **Description** | **Sample value** |
| --- | --- | --- | --- |
| regn\_name | VARCHAR | Region name |  |

Description:  
Gets the list of Market Groups

### mkt\_prc\_vw

DB schema: **md**  
Used by: **FE API**  
Context: **Home Panel**  
Content: list of Markets

| **Column name** | **Data type** | **Description** | **Sample value** |
| --- | --- | --- | --- |
| mkt\_grp\_id | INT | Identifier of Market Group |  |
| mkt\_grp\_name | VARCHAR | Market Group name |  |
| mkt\_id | INT | Identifier of Market |  |
| mkt\_name | VARCHAR | Market name |  |
| activ\_ind | CHAR(1) | Shows if Market is currently in use | Y/N |
| extnd\_attr\_json\_txt | VARCHAR | For future use |  |

Description:  
Gets the list of Markets, do not list "ALL" like in MMR

### notif\_prc\_vw

DB schema: **md**  
Used by: **FE API**  
Context: **Notifications**  
Content: list of Notifications

Columns beside columns already in notif\_prc:

| **Column name** | **Data type** | **Description** | **Sample value** |
| --- | --- | --- | --- |
| actn\_type\_name | VARCHAR2(20) | Name of the action from file\_actn\_plc\_vw | Upload, Download, etc. If no action related then Generic |
| svrty\_code | VARCHAR2(20) | Meaning of current status can map to specific icon or color (PrimeNG color coding of severity ) | info - to show is process is pending or progressing, success, warning, danger - to show that process has ended and what was the outcome |
| scope\_id | INT | From file\_actn\_plc\_vw |  |
| scope\_name | VARCHAR2(1000) | From file\_actn\_plc\_vw |  |
| file\_hash\_txt | VARCHAR2(100) | From file\_actn\_plc\_vw |  |
| disp\_file\_name | VARCHAR2(100) | From file\_actn\_plc\_vw |  |
| Description: |  |  |  |
| Gets the list of Notifications |  |  |  |

### scope\_prc\_vw

DB schema: **md**  
Used by: **FE API**  
Context: **Home Panel**  
Content: list of scopes

Columns beside columns already in scope\_prc:

| **Column name** | **Data type** | **Description** | **Sample value** |
| --- | --- | --- | --- |
| scope\_name | VARCHAR | Concatenation of Region / Market Group / File / Market or "GLOBAL" for global scope. | EU / Europe UKI / TT Direct Non-Standard / IRELAND |

Description:  
Lists of files prepared for download used for Input File download and for downloading of DQ Validation HTML reports.

### user\_file\_asign\_vw

**Summary**

DB schema: **md**  
Used by: **FE API**  
Context: **Home Panel/Admin Panel**  
Content: lists cartesian product of all users and all scopes presenting Input File details and actual assignments

**Columns**

| **Column name** | **Data type** | **Description** | **Sample value** |
| --- | --- | --- | --- |
| user\_name | VARCHAR | User's P&G short name | test.user |
| email\_address | VARCHAR | User's email address | test.user@pg.com |
| asign\_ind | VARCHAR(1) | Y if row represents actual row in user\_scope\_prc, N otherwise |  |
| scope\_id | INT | Scope identifier |  |
| mkt\_grp\_id | INT | Identifier of Market Group |  |
| mkt\_grp\_name | VARCHAR | Market Group name |  |
| mkt\_id | INT | Identifier of Market |  |
| mkt\_name | VARCHAR | Market name |  |
| file\_dfntn\_id | INT | Identifier of File Defintion |  |
| file\_dfntn\_vers\_id | INT | Identifier of particular version of File Defintion |  |
| file\_name | VARCHAR | Name of the Input File |  |
| file\_desc | VARCHAR | Description of Input File |  |
| frcst\_ind | CHAR(1) | Forecast column indicator |  |
| cnfg\_ind | CHAR(1) | Configuration column indicator |  |
| load\_ind | CHAR(1) | Load column indicator |  |
| curr\_ind | CHAR(1) | Indicator if file version is the current/latest |  |
| all\_mkt\_ind | CHAR(1) | Indicator if Market is 'ALL' |  |
| file\_sttus\_code | VARCHAR | Current status of Input File version |  |
| actl\_read\_prvlg\_ind | CHAR(1) | Indicator of actual read access for this particular scope |  |
| actl\_write\_prvlg\_ind | CHAR(1) | Indicator of actual write access for this particular scope |  |
| eff\_read\_prvlg\_ind | CHAR(1) | Indicator of effective (perhaps inherited) read access for this particular scope |  |
| eff\_write\_prvlg\_ind | CHAR(1) | Indicator of effective (perhaps inherited) write access for this particular scope |  |
| load\_datetm | DATETM | Last time when data was refreshed by Load Process |  |
| uplod\_datetm | DATETM | Last time content was uploaded |  |
| uplod\_user\_name | VARCHAR | User who uploaded most recently |  |
| sbmt\_datetm | DATETM | Last time Input File was submitted |  |
| sbmt\_user\_name | VARCHAR | User who submitted most recently |  |
| due\_datetm | DATETM | Due date for Input File to be uploaded with final content |  |
| extnd\_attr\_json\_txt | VARCHAR | Additional attributes for UI |  |

**Description**

Returns complete list of pairs user\_name/scope for current versions of Input Files being active. Rows witch represent actual assignment have asign\_ind=Y.  
Added rows for Market Group level if not explicitly assigned.

Specification of extnd\_attr\_json\_txt:

* not used for now, but can be used later on, when some extra attributes should control the appearance of an assignment.

### user\_scope\_prc\_vw

**Summary**

Privileges of users with the associated scope.

**Columns**

All columns on the table md.user\_scope\_prc

### user\_prc\_vw

DB schema: **md**  
Used by: **FE API**  
Context: **Dropdowns**  
Content: list of users having any assignments in Uploader

| **Column name** | **Data type** | **Description** | **Sample value** |
| --- | --- | --- | --- |
| user\_name | VARCHAR | P&G short name |  |
| last\_login\_datetm | DATETIME2(7) | Date and time of user's last login | 2021-07-27 14:22:51.2066667 |

Description:  
List of all users having assignments in Uploader.

## Data Factory reference

### Pipelines

ADF Pipelines are grouped into 5 main areas:

* Common
* Email
* File
* Ingestion
* Load

**Maintenance**

**Contains Details about the pipelines used for cleanup of redundant tables in the database.**

### Mt-montly\_db\_cleanup

**Purpose:** To list out all the Temporary tables created in the Database and filter them out based on Date (only reporting month - 1 tables should be dropped.)

**Caller Pipeline:** ld-ActualLoad

| **Parameter name** | **Data type** | **Description** | **Sample value** |
| --- | --- | --- | --- |
| path | string | Partial Notebook path in ADB | /Uploader/ |
| in\_parnt\_comp\_exctn\_id | string | Component Execution ID for Logging | 1473 |
| in\_user\_name | string | username of the user who triggered the pipeline | colaco.cc |

**Description:** This pipeline is called every time whenever there is a load initiated by the user. It consists of a copy activity which will query the DB for all tables under the schema's 'tmp', 'sbmt'. once this config is dumped into internal storage, an ADB notebook is called, which filter out all the tables which are needed to be dropped.

such a list is then passed to the **Mt-GenericParallelDeleter** which will actually drop those table.

Additionally, the Out pipeline activity is configured on **Completed**, hence even if this pipeline fails for some reason, further processing would not stop.

### Mt-GenericParallelDeleter

**Purpose:** Drops all the tables from the database based on inputs received from caller pipeline.

**Caller Pipeline:** Mt-montly\_db\_cleanup

| **Parameter name** | **Data type** | **Description** | **Sample value** |
| --- | --- | --- | --- |
| input | array | Expects a list of tables to be dropped | ["table\_1","table\_2"] |
| schema | string | Expects the DB Schema of above mentioned table. | tmp |

**Description:**  
This pipeline contains a for loop which will loop over the array received in the input parameter and append it with the string received in schema parameter.  
example. "tmp.table\_1"  
moreover a there is lookup which will drop all such tables from the database.

**Common**

### cm-DumpDBInputFileToCSV

**Summary**

**Purpose**: to dump CSV files on a **selected location** from an input file defined from an **in\_file\_dfntn\_vers\_id**  
**Used by**: **Other Pipeline**

**Parameters**

| **Parameter name** | **Data type** | **Description** | **Sample value** |
| --- | --- | --- | --- |
| in\_cond\_txt | String | Condition clause for the query | "**reg\_id = 25**" |
| in\_cntnr\_name\_txt | String | Dump location - blob container name | test-adf-common |
| in\_path\_txt | String | Dump location - folder name within the specified container in **in\_cntnr\_name\_txt**, adheres to guidelines Naming and referencing shares, directories, files, and metadata | dbinputtoparquet |
| in\_file\_name\_txt | String | Dump location - file name within the specified folder path in **in\_path\_txt** | test1 |
| in\_parnt\_comp\_exctn\_id | String | Parent component execution Id |  |
| in\_ordr\_txt | String |  |  |
| in\_db\_schema\_txt | String |  |  |
| in\_db\_tbl\_txt | String |  |  |
| in\_user\_name | String | Username of who executed the pipeline. Can be used for audit trail | test\_user |
| in\_select\_col\_list | String |  |  |

**Description:**

Pipeline performs following steps:

1. Get the submit table name within the input schema from file\_dfntn\_vers\_prc\_vw where file\_dfntn\_vers\_id = **in\_file\_dfntn\_vers\_id**
2. Get the submit table columns within the input schema from file\_dfntn\_vers\_col\_prc\_vw where file\_dfntn\_vers\_id = **in\_file\_dfntn\_vers\_id**
3. Formulate the query including the condition clause, if specified, in parameter **in\_cond\_txt**
4. Call the pipeline cm-DumpDBSelectToCSV passing the parameters:  
   **l\_qry\_txt**  
   **in\_path\_txt**  
   **in\_file\_name\_txt**  
   **in\_parnt\_comp\_exctn\_id**

### cm-DumpDBInputFileToParquet

**Summary**

**Purpose**: to dump parquet files on a **selected location** from an input file defined from an **in\_file\_dfntn\_vers\_id**  
**Used by**: **Other Pipeline**

**Parameters**

| **Parameter name** | **Data type** | **Description** | **Sample value** |
| --- | --- | --- | --- |
| in\_file\_dfntn\_vers\_id | Int | File definition Id |  |
| in\_cond\_txt | String | Condition clause for the query | "**reg\_id = 25**" |
| in\_cntnr\_name\_txt | String | Dump location - blob container name | test-adf-common |
| in\_path\_txt | String | Dump location - folder name within the specified container in **in\_cntnr\_name\_txt**, adheres to guidelines Naming and referencing shares, directories, files, and metadata | dbinputtoparquet |
| in\_archv\_path\_txt | String |  | NOT NEEDED |
| in\_file\_name\_txt | String | Dump location - file name within the specified folder path in **in\_path\_txt** | test1 |
| in\_parnt\_comp\_exctn\_id | String | Parent component execution Id |  |
| in\_user\_name | String | Username of who executed the pipeline. Can be used for audit trail | test\_user |
| in\_schema\_name | String | in\_schema\_name |  |

**Description:**

Pipeline performs following steps:

1. Get the submit table name within the input schema from file\_dfntn\_vers\_prc\_vw where file\_dfntn\_vers\_id = **in\_file\_dfntn\_vers\_id**
2. Get the submit table columns within the input schema from file\_dfntn\_vers\_col\_prc\_vw where file\_dfntn\_vers\_id = **in\_file\_dfntn\_vers\_id**
3. Formulate the query including the condition clause, if specified, in parameter **in\_cond\_txt**
4. Call the pipeline cm-DumpDBSelectToParquet passing the parameters:  
   **l\_qry\_txt**  
   **in\_path\_txt**  
   **in\_file\_name\_txt**  
   **in\_parnt\_comp\_exctn\_id**

### cm-DumpDBSelectToCSV

Purpose: to dump parquet files on a **selected location** using a **custom query**  
Used by: **Other Pipeline**, **cm-DumpDBInputFileToCSV**

| **Parameter name** | **Data type** | **Description** | **Sample value** |
| --- | --- | --- | --- |
| in\_qry\_txt | String | Custom select query | SELECT TOP 25 \* FROM [input].[cnfg\_geo\_mapng\_fct] WHERE regn\_id = 924 |
| in\_cntnr\_name\_txt | String | Dump location - blob container name | test-adf-common |
| in\_path\_txt | String | Dump location - folder name within the specified container in **in\_cntnr\_name\_txt** | dbinputtoparquet |
| in\_file\_name\_txt | String | Dump location - file name within the specified folder path in **in\_path\_txt** | test1 |
| in\_parnt\_comp\_exctn\_id | String | Parent component execution Id |  |
| in\_user\_name | String | Username of who executed the pipeline. Can be used for audit trail | test\_user |

Description:

Pipeline performs following steps:

1. Run the query specified in parameter **in\_qry\_txt**
2. Dump the query results from **Step 1** to specified location using parameters **in\_cntnr\_name\_txt** / **in\_path\_txt** / **in\_file\_name\_txt**

* e.g. **test-adf-common**/**dbinputtocsv**/**test1**

### cm-DumpDBSelectToParquet

Purpose: to dump parquet files on a **selected location** using a **custom query**  
Used by: **Other Pipeline**, **cm-DumpDBInputFileToParquet**

| **Parameter name** | **Data type** | **Description** | **Sample value** |
| --- | --- | --- | --- |
| in\_qry\_txt | String | Custom select query | SELECT TOP 25 \* FROM [input].[cnfg\_geo\_mapng\_fct] WHERE regn\_id = 924 |
| in\_cntnr\_name\_txt | String | Dump location - blob container name | test-adf-common |
| in\_path\_txt | String | Dump location - folder name within the specified container in **in\_cntnr\_name\_txt** | dbinputtoparquet |
| in\_file\_name\_txt | String | Dump location - file name within the specified folder path in **in\_path\_txt** | test1 |
| in\_parnt\_comp\_exctn\_id | String | Parent component execution Id |  |
| in\_user\_name | String | Username of who executed the pipeline. Can be used for audit trail | test\_user |

Description:

Pipeline performs following steps:

1. Run the query specified in parameter **in\_qry\_txt**
2. Dump the query results from **Step 1** to specified location using parameters **in\_cntnr\_name\_txt** / **in\_path\_txt** / **in\_file\_name\_txt**

* e.g. **test-adf-common**/**dbinputtoparquet**/**test1**

### cm-GenerateDQCheckReport

Purpose: to prepare Input File data for file download by End User  
Used by: **FE API**

| **Parameter name** | **Data type** | **Description** | **Sample value** |
| --- | --- | --- | --- |
| in\_parnt\_comp\_exctn\_id | Int | Identifier of callers Component Execution |  |
| in\_dq\_check\_exctn\_id | Int | DQ check Execution ID for a particular file template |  |
| in\_file\_actn\_id | Int | DQ check Execution corresponding to file action type |  |
| in\_file\_dfntn\_vers\_id | Int | File definition version of input file template |  |
| in\_user\_name | String | Username who deals with this run of pipeline |  |
| in\_cntnr\_name\_txt | String | Container name in SA |  |
| in\_path\_txt | String | Path to be mentioned for csv file |  |

**Description:**

Pipeline performs following steps:

* executes [main].[pro\_comp\_exctn\_open]
* should query out rows from [md].[dq\_check\_exctn\_plc] with provided information of dq\_check\_exctn\_id & file\_actn\_id
* Invoke pipeline to iterate over each row obtained from [md].[dq\_check\_exctn\_plc] result
* Invoke pipeline cm-DumpDBSelectToCSV for each iteration per row, it should generate csv file in SA folder with name from reslt\_table\_name

### cm-PumpCSVInputFileToDB

**Summary**

**Purpose**

To pump csv files from a **selected location** to a **created stage table**

**Used by**

Other Pipeline

**Parameters**

| **Parameter name** | **Data type** | **Description** | **Sample value** |
| --- | --- | --- | --- |
| in\_cntnr\_name\_txt | String | Source location - blob container name | incoming |
| in\_path\_txt | String | Source location - folder name within the specified container in **in\_cntnr\_name\_txt** | delimited |
| in\_file\_name\_txt | String | Source location - file name within the specified folder path in **in\_path\_txt** without file extension | stage cnfg\_geo\_mapng 1 |
| in\_file\_dfntn\_vers\_id | Int | See notes [1] |  |
| in\_tbl\_name | String |  | NONE |
| in\_parnt\_comp\_exctn\_id | Int | See notes [1] |  |
| in\_user\_name | String | See notes [1] |  |

**Notes**

1. See the parameter on pro\_file\_stage\_creat

**Process**

Pipeline performs following steps:

1. Get the file metadata (column names, data type, etc.) from the source location using parameters **in\_cntnr\_name\_txt** / **in\_path\_txt** / **in\_file\_name\_txt**.

test-adf-common/dbinputtoparquet/test1

1. Create stage table via pro\_file\_stage\_creat.
2. Get the target metadata of the created stage table from Step 2.
3. Create a dynamic json mapping using the meta data of the source on Step 1 and map it into sink meta data on Step 3.
4. Insert the data into the created stage table from Step 2 using the mapping configuration on Step 4.  
   ***Note:*** Assuming CSV has the same structure of the table (same column name, column number, etc.), inserting will encounter no problems.

### cm-PumpFileIntoDB

**Parameters**

| **Parameter name** | **Data type** | **Description** | **Sample value** |
| --- | --- | --- | --- |
| in\_parnt\_comp\_exctn\_id | String | Parent component execution Id | 0 |
| in\_user\_name | String | Username of who executed the pipeline. Can be used for audit trail | ADF, doe.j@pg.com |
| in\_cntnr\_name\_txt | String | Dump location - blob container name | test-adf-common |
| in\_path\_txt | String | Dump location - folder name within the specified container in **in\_cntnr\_name\_txt**, adheres to guidelines Naming and referencing shares, directories, files, and metadata | dbinputtoparquet |
| in\_file\_name\_txt | String | Dump location - file name within the specified folder path in **in\_path\_txt** | original |
| in\_trgt\_tbl\_name | String | Target table on the database which is usually under the stage schema | cnfg\_geo\_mapng\_work\_fct\_0000001837\_202108163749\_sfct |

**Description**

Loads file content into given target table.  
List of queries derived from metadata.  
Dataset parametrized in line with file extension.

**Column list query for dynamic mapping**

SELECT col\_name

FROM {mtdta\_tbl\_name}

WHERE {mtdta\_key\_col\_name} = {mtdta\_key\_col\_val}

ORDER BY col\_num;

**Sample queries**

* Query 1
* SELECT col\_name, phys\_data\_type
* FROM file\_dfntn\_vers\_col\_prc\_vw
* WHERE file\_dfntn\_vers\_id = 1
* ORDER BY col\_num;
* Query 2
* SELECT col\_name
* FROM load\_col\_regn\_lkp\_vw
* WHERE regn\_id = 2

ORDER BY col\_num;

### [Obsolete] cm-GenericDBLog

Purpose: to log pipeline activity to db  
Used by: **Other ADF Pipelines**

| **Parameter name** | **Data type** | **Description** | **Sample value** |
| --- | --- | --- | --- |
| in\_comp\_exctn\_id | INT | Identifier of the Component Execution |  |
| in\_msg\_txt | String | Message to log on DB |  |
| in\_sttus\_code | String | Status code if 'OK' or 'ERR' |  |

Description:

Pipeline performs following steps:

* Call the main.pro\_log\_msg procedure in DB using the pipeline parameters.
* There are two possible results of this execution.
  1. Success - pipeline will proceed to next execution.
  2. Failed - execution will halt including the calling pipeline.

**Email**

### em-EmailCleanup

Purpose: to remove old email entries from the queue table  
Used by: **executed by ADF trigger**

| **Parameter name** | **Data type** | **Description** | **Sample value** |
| --- | --- | --- | --- |
| in\_parnt\_comp\_exctn\_id | INT | Identifier of callers Component Execution |  |
| in\_user\_name | VARCHAR | User requesting download |  |
| Description: |  |  |  |

Pipeline performs following steps:

* calls the main.pro\_email\_delet procedure in DB to drop the email if it is marked as SENT.

### em-EmailDispatch

**Summary**

Purpose: to send email by **email\_id**  
Used by: **Other Pipeline**, **em-EmailService**

**Parameters**

| **Parameter name** | **Data type** | **Description** | **Sample value** |
| --- | --- | --- | --- |
| in\_parnt\_comp\_exctn\_id | INT | Identifier of callers Component Execution |  |
| in\_user\_name | VARCHAR | User requesting download |  |
| in\_email\_id | int | Email id from email\_prc table |  |

**Description**

Pipeline performs following steps:

1. Get full email details using email\_prc\_vw by email\_id.
   * list of recipients
   * title/subject
   * email content (HTML)
2. Call the pro\_email\_sttus\_updt procedure in DB to mark it as SENDING.
3. Call logicapp-cnosgc-uploader-n01 through HTTP Request to send the email with the provided email details.
4. Call the pro\_email\_sttus\_updt procedure in DB to mark it as SENT.
5. Delete old emails greater than retention period using pro\_email\_delet.

### em-EmailService

Purpose: to send emails ready in the queue  
Used by: **ADF trigger**

| **Parameter name** | **Data type** | **Description** | **Sample value** |
| --- | --- | --- | --- |
| in\_parnt\_comp\_exctn\_id | INT | Identifier of callers Component Execution |  |
| in\_user\_name | VARCHAR | User requesting download |  |

Description:

Pipeline performs following steps:

* Get email list from email\_prc\_vw with status NEW.
* Call em-EmailDispatch

**File**

### fl-Process-V06

Purpose:  
To Execute and generate a Report for V06 DQ check.

Used by:  
fl-Execute-non-critical-dq-checks

| **Parameter name** | **Data type** | **Description** | **Sample value** |
| --- | --- | --- | --- |
| in\_user\_name | string | username of the user who is performing action on file | colaco,cc@pg.com |
| in\_file\_dfntn\_vers\_id | int | file definition version of the input template which is being currently modified | 100 |
| input\_tbl\_name | string | name of the staging table which is created during upload | stage.input\_v06\_tbl |
| scope\_id | int | scope-id of the user who is performing the file action | 310 |
| in\_parnt\_comp\_exctn\_id | int | Component Execution ID for Logging | 654312 |
| file\_action\_id | int | file-action ID which is generated in the DB for current file action | 896 |

Description:

This pipeline will execute the stored procedure, which will generate a report for V06 DQ check,  
Post that if there are any rows captured in the DQ check, an Excel report will be generated. Same information will be updated in the non-critical-DQ-check framework, so that e-mail is generated accordingly.

### fl-Process-V09

**Purpose:**  
To Execute and generate a Report for V09 DQ check.

**Used by:**  
fl-Execute-non-critical-dq-checks

| **Parameter name** | **Data type** | **Description** | **Sample value** |
| --- | --- | --- | --- |
| in\_user\_name | string | username of the user who is performing action on file | colaco.cc@pg.com |
| in\_file\_dfntn\_vers\_id | int | file definition version of the input template which is being currently modified | 156 |
| input\_tbl\_name | string | name of the staging table which is created during upload | stage.input\_1234\_5678 |
|  |  |  |  |
| scope\_id | int | scope-id of the user who is performing the file action | 489 |
| in\_parnt\_comp\_exctn\_id | int | Component Execution ID for Logging | 7896 |
| file\_action\_id | int | file-action ID which is generated in the DB for current file action | 741 |
| load\_ind | boolean | Signifies if the template is Load or Non-load | N |
| out\_email\_id | int | primary key which is used to track the email which is sent out once upload is completed | 497 |

**Description:**

This pipeline will execute the stored procedure, which will generate a report for V09 DQ check,  
Post that if there are any rows captured in the DQ check, an Excel report will be generated. Same information will be updated in the non-critical-DQ-check framework, so that e-mail is generated accordingly.

### fl-Process-V05

**Purpose:**  
To Execute and generate a Report for V05 DQ check.  
**Used by:**  
fl-Execute-non-critical-dq-checks

| **Parameter name** | **Data type** | **Description** | **Sample value** |
| --- | --- | --- | --- |
| in\_user\_name | string | username of the user who is performing action on file | colaco.cc@pg.com |
| in\_file\_dfntn\_vers\_id | int | file definition version of the input template which is being currently modified | 145 |
| input\_tbl\_name | string | name of the staging table which is created during upload | stage.input\_1234\_5678 |
| scope\_id | int | scope-id of the user who is performing the file action | 741 |
| in\_parnt\_comp\_exctn\_id | int | Component Execution ID for Logging | 78549 |
| file\_action\_id | int | file-action ID which is generated in the DB for current file action | 369 |

**Description:**

This pipeline will execute the stored procedure, which will generate a report for V05 DQ check,  
Post that if there are any rows captured in the DQ check, an Excel report will be generated. Same information will be updated in the non-critical-DQ-check framework, so that e-mail is generated accordingly.

### fl-Execute-non-critical-dq-checks

**Purpose**: To Execute Non-Critical DQ checks after file Submit

**Used by**: fl-FileUploadAndFollowUp

| **Parameter name** | **Data type** | **Description** | **Sample value** |
| --- | --- | --- | --- |
| in\_user\_name | string | username of the user who is performing action on file | colaco.cc@pg.com |
| in\_file\_dfntn\_vers\_id | int | file definition version of the input template which is being currently modified | 23 |
| input\_tbl\_name | string | name of the staging table which is created during upload | stage.input\_1234\_5678 |
| scope\_id | int | scope-id of the user who is performing the file action | 230 |
| in\_parnt\_comp\_exctn\_id | int | Component Execution ID for Logging | 1587 |
| out\_email\_id | int | primary key which is used to track the email which is sent out once upload is completed | 897 |
| file\_action\_id | int | file-action ID which is generated in the DB for current file action | 7899 |

**Description:**  
This pipeline is executed once the all critical DQ checks are passed. The purpose of this pipeline is to trigger the execution of non critical DQ checks.

### fl-FileConverter

**Parameters**

| **Parameter name** | **Data type** | **Description** | **Sample value** |
| --- | --- | --- | --- |
| in\_converter\_param\_json | String |  |  |
| in\_converter\_base\_url | String |  | https://func-cnosgc-uploader-excel-converter-n01.ase4xnp.pgcloud.com |
| in\_function\_name | String |  | convert |

### fl-FileDownload

Purpose: to download data from DB as csv and prepare it for the converter app  
Used by: **FE API**

| **Parameter name** | **Data type** | **Description** | **Sample value** |
| --- | --- | --- | --- |
| in\_parnt\_comp\_exctn\_id | INT | Identifier of callers Component Execution |  |
| in\_user\_name | VARCHAR(50) | Name of the user performing the action |  |
| in\_scope\_id | INT | Scope identifier |  |
| in\_filtr\_json\_txt | VARCHAR | JSON describing what filters were used when requesting download |  |

**Description:**

Pipeline performs following steps:

* opens component execution with pro\_comp\_exctn\_open
* creates File Action with pro\_file\_actn\_open
* triggers PROCESSING notification in UI via pro\_notif\_upsrt
* registers downloaded file in DB via pro\_file\_dwnld\_creat
* converts filtering and ordering JSON params (in\_filtr\_json\_txt) to SQL statement via pro\_file\_filtr\_to\_sql
* retrieves request body for converter (for csv to xlsx conversion) via pro\_file\_xlsx\_cnvrt\_parm\_prep
* runs Copy activity to dump DB data as csv in converter-input storage container, path: converter-input/input
* runs Copy activity to dump all necessary converter parameters as JSON file in converter-input storage container, path: converter-input/json/input

**Specification for in\_filtr\_json\_txt:**

We need to pass filter criteria (an array of criteria) but also result ordering.

* order - contains a map of column name and sorting direction (either asc or desc). Nulls are always last. Assume that only one column can sort at a time, but if this can be implemented to be generic from the start, please do.
* criteria - contains an array of objects which represent a criteria defined per column. Possible operators are
  + LIKE (for TEXT data type) - the value will come with % already in place.
  + = (for any numerical data type)
* value - can contain | which can be used to specify multiple alternative values.

### fl-FileDownloadAndFollowUp

Purpose: to download data from DB table as Excel file  
Used by: **FE API**

| **Parameter name** | **Data type** | **Description** | **Sample value** |
| --- | --- | --- | --- |
| in\_parnt\_comp\_exctn\_id | INT | Identifier of callers Component Execution |  |
| in\_user\_name | VARCHAR(50) | Name of the user performing the action |  |
| in\_scope\_id | INT | Scope identifier |  |
| in\_filtr\_json\_txt | VARCHAR | JSON describing what filters were used when requesting download |  |
| in\_qry\_txt | VARCHAR | SQL query to be used to query DB table |  |
| in\_phys\_file\_name | VARCHAR | File name to be downloaded |  |
| in\_converter\_param\_json | VARCHAR | JSON with parameters |  |
| in\_file\_actn\_id | INT | File action ID |  |
| in\_file\_dwnld\_id | INT | File download ID |  |

**Description:**

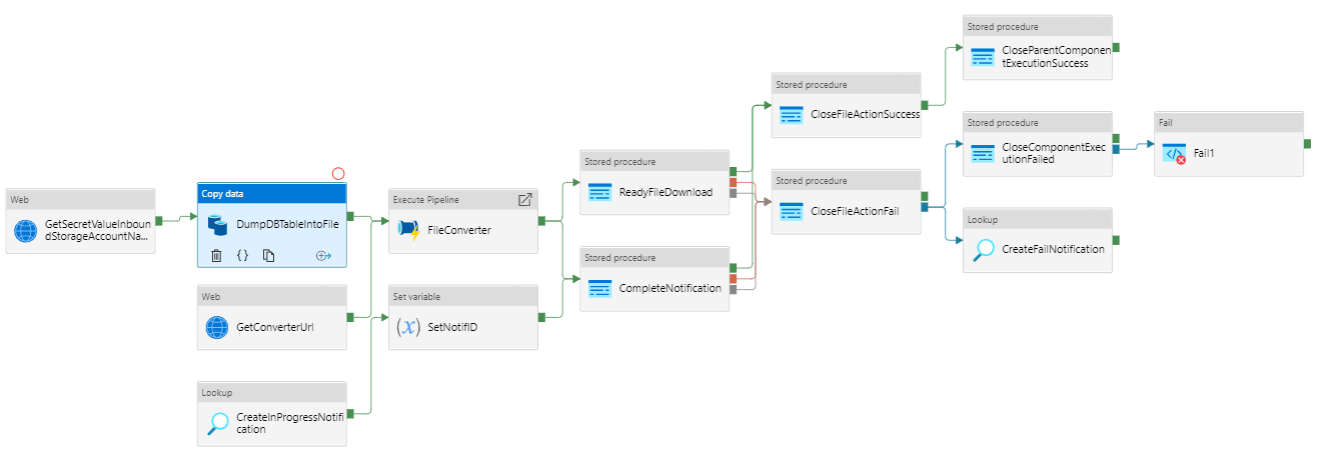
Pipeline performs following steps:

* establishes connection to Converter and Storage account
* Creates notification status
* dumps data to file and performs Converter conditions on columns
* marks as ready prepared file for user's download
* closes all opened components and actions

**Specification for in\_filtr\_json\_txt:**

We need to pass filter criteria (an array of criteria) but also result ordering.

* order - contains a map of column name and sorting direction (either asc or desc). Nulls are always last. Assume that only one column can sort at a time, but if this can be implemented to be generic from the start, please do.
* criteria - contains an array of objects which represent a criteria defined per column. Possible operators are
  + LIKE (for TEXT data type) - the value will come with % already in place.
  + = (for any numerical data type)
* value - can contain | which can be used to specify multiple alternative values.



### fl-FileDownloadFollowUp

Purpose: to complete the download by adding the link to the newly converted xlsx file and marking file as ready

Triggered By: ConverterFileDownloadOutputJsonCreated (Blob trigger)

| **Parameter name** | **Data type** | **Description** | **Sample value** |
| --- | --- | --- | --- |
| in\_file\_name | STRING | file name dumped by converter app in storage path: converter-input/json/output/filedownload | 20211202105713-d00000170\_00001267\_20211202\_105243.json |

**Description:**

Pipeline performs following steps:

* looks up the config JSON file sent by converter in converter-input/json/output/filedownload path and assigns the variables of the pipeline parameters (from fl-FileDownload pipeline)
* opens new component execution with pro\_comp\_exctn\_open
* marks download file as ready with pro\_file\_dwnld\_ready
* triggers completion of notification in UI via pro\_notif\_upsrt
* closes the file action via pro\_file\_actn\_close
* closes the new component execution with pro\_comp\_exctn\_close
* closes the parent component execution with pro\_comp\_exctn\_close

### fl-FileDQValDownload

Purpose: to download data from DB as csv and prepare it for the converter app  
Used by: **FE API**

| **Parameter name** | **Data type** | **Description** | **Sample value** |
| --- | --- | --- | --- |
| in\_parnt\_comp\_exctn\_id | INT | Identifier of callers Component Execution |  |
| in\_user\_name | STRING | Name of the user performing the action |  |
| in\_dq\_check\_exctn\_id | INT | DQ check execution identifier |  |
| in\_reslt\_tbl\_name | STRING | JSON describing what filters were used when requesting download |  |
| in\_scope\_id | INT | Scope identifier |  |
| in\_file\_actn\_id | INT | File action identifier |  |

**Description:**

Pipeline performs following steps:

* opens component execution with pro\_comp\_exctn\_open
* registers downloaded file in DB via pro\_file\_dwnld\_creat
* retrieves request body for converter (for csv to xlsx conversion) via pro\_file\_xlsx\_cnvrt\_dq\_parm\_prep
* runs Copy activity to dump DB data as csv in converter-input storage container, path: converter-input/input
* runs Copy activity to dump all necessary converter parameters as JSON file in converter-input storage container, path: converter-input/json/input

### fl-FileDQValDownloadFollowUp

Purpose: to complete the DQ val download by adding the link to the newly converted xlsx file and marking file as ready

Triggered By: ConverterFileDQValOutputJsonCreated (Blob Trigger)

| **Parameter name** | **Data type** | **Description** | **Sample value** |
| --- | --- | --- | --- |
| in\_file\_name | STRING | file name dumped by converter app in storage path: converter-input/json/output/fileDQval | 20211216142454-vldtn\_20211216\_00000086\_00000210\_sfct.json |

**Description:**

Pipeline performs following steps:

* looks up the config JSON file sent by converter in converter-input/json/output/fileDQval path and assigns the variables of the pipeline parameters (from fl-FileDQValDownload pipeline)
* opens new component execution with pro\_comp\_exctn\_open
* marks download file as ready with pro\_file\_dwnld\_ready
* closes the new component execution with pro\_comp\_exctn\_close
* closes the parent component execution with pro\_comp\_exctn\_close

### fl-FileSubmit

Purpose: to copy single Input File from DB to CNGC exposed storage

Triggered By: fl-FileUploadAndFollowUp

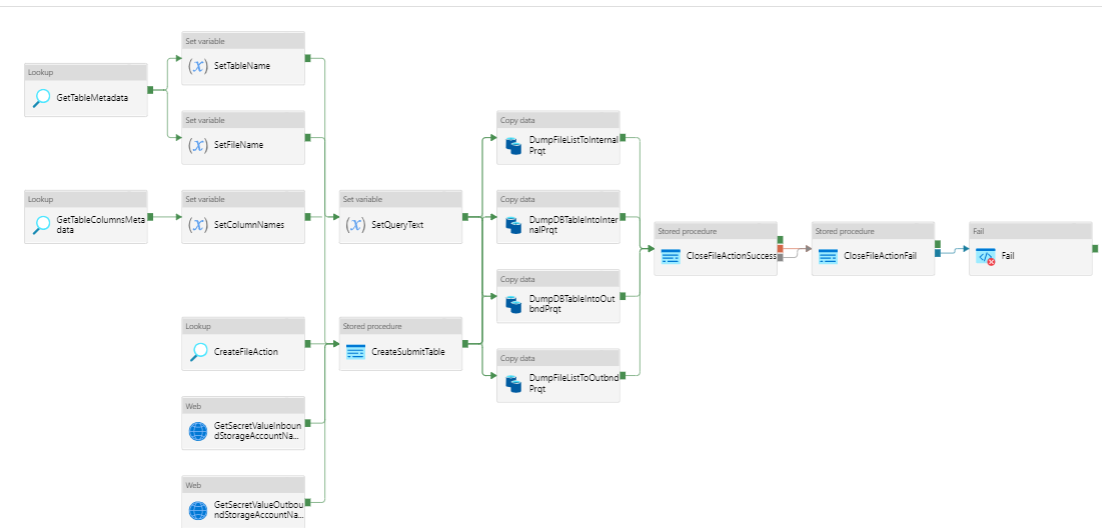
| **Parameter name** | **Data type** | **Description** | **Sample value** |
| --- | --- | --- | --- |
| in\_parnt\_comp\_exctn\_id | INT | Identifier of callers Component Execution |  |
| in\_user\_name | VARCHAR | User requesting download |  |
| in\_file\_dfntn\_vers\_id | INT | Input File version ID |  |
| in\_scope\_id | INT | Scope ID |  |

**Description:**

**User** privileges are assumed to be checked before either by Front-End or by the fact that only authorised user can trigger this ADF pipeline.

Pipeline performs following steps:

* Gathers submit table name and all related columns
* Creates file action ID connected with Submit process
* Gathers all details for IN and OUT storage accounts
* Runs submit procedure pro\_file\_sbmt\_tbl\_creat dump data from working table to submit table
* Dumps data to parquet files for both In and out storage accounts using prepared query and target storage account details
* Closes file action



### fl-FileUpload

Purpose: to prepare the upload file for the converter app (to be converted from xlsx to csv)  
Used by: **FE API**

| **Parameter name** | **Data type** | **Description** | **Sample value** |
| --- | --- | --- | --- |
| in\_parnt\_comp\_exctn\_id | INT | Identifier of callers Component Execution |  |
| in\_scope\_id | INT | Scope identifier |  |
| in\_user\_name | VARCHAR | User requesting download |  |
| in\_file\_loc | VARCHAR | Location of the file on Uploader's internal storage |  |

**Description:**  
User privileges are assumed to be checked before either by Front-End or by the fact that only authorised user can trigger this ADF pipeline.

Pipeline performs following steps:

* opens component execution with pro\_comp\_exctn\_open
* creates File Action with pro\_file\_actn\_open
* creates Stage table in DB with pro\_file\_stage\_creat
* retrieves request body for converter (for xlsx to csv conversion) via pro\_file\_xlsx\_cnvrt\_parm\_prep
* renames the file uploaded (based on the output of previous step) by creating new file with new name and deleting the original one
* runs Copy activity to dump all necessary converter parameters as JSON file in converter-input storage container, path: converter-input/json/input

### fl-FileUploadAndFollowUp

Purpose: to upload data from Excel file to DB template  
Used by: **FE API**

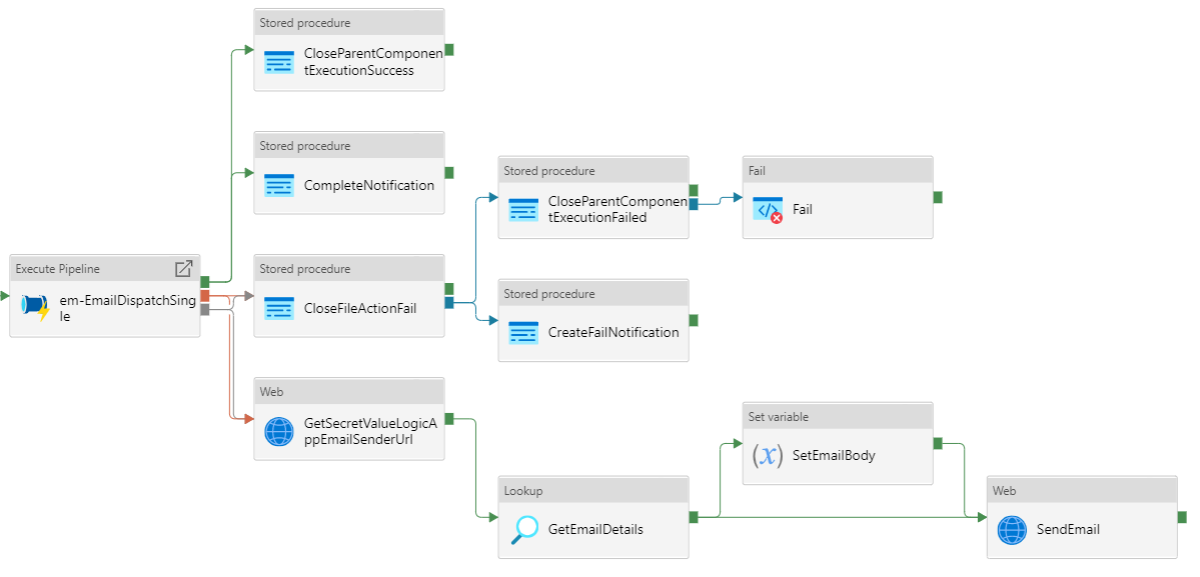
| **Parameter name** | **Data type** | **Description** | **Sample value** |
| --- | --- | --- | --- |
| in\_parnt\_comp\_exctn\_id | INT | Identifier of callers Component Execution |  |
| in\_user\_name | VARCHAR(50) | Name of the user performing the action |  |
| in\_scope\_id | INT | Scope identifier |  |
| in\_file\_dfntn\_vers\_id | INT | File definition identifier |  |
| in\_file\_actn\_id | INT | File action identifier |  |
| in\_converter\_param\_json | VARCHAR | Parameters passed as String to Converter module |  |
| in\_phys\_file\_name | VARCHAR | Name of the uploaded file |  |

**Description:**

Pipeline performs following steps:

* sets up connection with Converter app, creates notification message, creates stage table for uploaded content, establishes storage account details
* inserts data provided via Excel to created stage table
* runs main upload procedure
* performs DQ checks file creation using pipeline: fileDQValDownloadAndFollowUp if any exists during process
* verifies if there is possibility to run Submit pipeline if no critical indicators found
* closes file with respective status to DQ checks found if any
* prepares DQ check result email with all DQ check conditions with respective statuses
* closes all opened actions, components and sets notification status
* sends email to user

Pipeline body:



### fl-FileUploadFollowUp

Purpose: to complete upload procedure by pumping converted csv file to stage table in DB, merge stage table with working table in DB, prepare DQ validations and submit file to outbound CNGC storage

Triggered By: ConverterFileUploadOutputJsonCreated (Blob Trigger)

| **Parameter name** | **Data type** | **Description** | **Sample value** |
| --- | --- | --- | --- |
| in\_file\_name | STRING | file name dumped by converter app in storage path: converter-input/json/output/fileupload | 20211202105713-u00000170\_00001267\_20211202\_105243.json |

**Description:**  
User privileges are assumed to be checked before either by Front-End or by the fact that only authorised user can trigger this ADF pipeline.

Pipeline performs following steps:

* looks up the config JSON file sent by converter in converter-input/json/output/fileupload path and assigns the variables of the pipeline parameters (from fl-FileDownload pipeline)
* opens new component execution with pro\_comp\_exctn\_open
* runs Copy activity to pump recently converted csv file into the stage table in the DB
* merges stage table into working table via procedure pro\_file\_uplod (which also performs DQ validation)
* prepares failed validation reports (if any) via repetitive calls (for each failed DQ validation) of pipeline fl-FileDQValDownload
* validates that there are no critical submit DQ validations failed, and then executes the pipeline fl-FileSubmit to store input file as parquet in external CNGC storage
* prepares DQ validation results email or submit status email via pro\_file\_vldtn\_email\_creat
* sends email via em-EmailDispatch pipeline
* closes the file action ("C" if no DQ validations for critical upload failed and "F" otherwise) via pro\_file\_actn\_close
* closes the new component execution with pro\_comp\_exctn\_close
* closes the parent component execution with pro\_comp\_exctn\_close

**Ingestion**

### in-CopyCustomerMapping

Purpose: To copy latest Customer Mapping files from BP2 to Uploader's incoming storage account  
Used by: Backend

| **Parameter name** | **Data type** | **Description** | **Sample value** |
| --- | --- | --- | --- |
| in\_parnt\_comp\_exctn\_id | INT | Identifier of callers Component Execution | 1 |
| in\_user\_name | STRING | Name of the user performing the action | surname.ns@pg.com |

**Description:**

Pipeline performs following steps:

* establishes connection to Storage account
* opens component
* gets list of customer mapping files
* copy latest customers files from BP2 to Uploader incoming storage account
* closes all opened components and actions

**Specification for CustomerMappingCopy.json:**

[

{

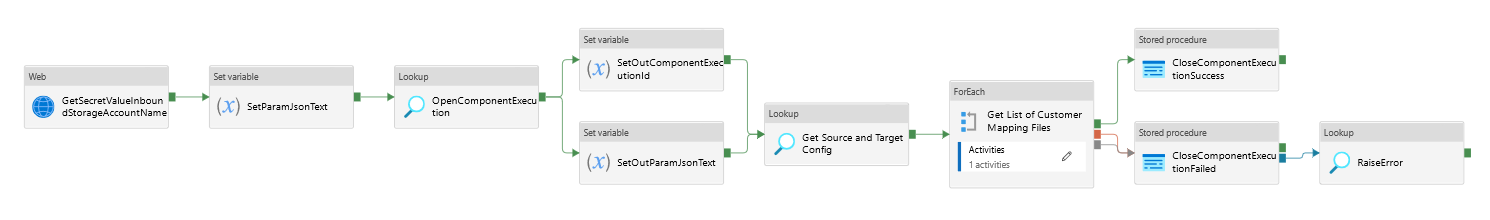
"source":"MDM\_CustomerMappings/",

"target":"MDM\_CustomerMappings"

}

]

"source" - source folder in BP2  
"target" - destination folder in Uploader 'incoming' container



### in-CopyProductMapping

Purpose: To copy latest Product Mapping files from BP2 to Uploader's incoming storage account  
Used by: Backend

| **Parameter name** | **Data type** | **Description** | **Sample value** |
| --- | --- | --- | --- |
| in\_parnt\_comp\_exctn\_id | INT | Identifier of callers Component Execution | 1 |
| in\_user\_name | STRING | Name of the user performing the action | surname.ns@pg.com |

**Description:**

Pipeline performs following steps:

* establishes connection to Storage account
* opens component
* gets list of product mapping files
* copy latest products files from BP2 to Uploader incoming storage account
* closes all opened components and actions

**Specification for ProductMappingCopy.json:**

**[**

{

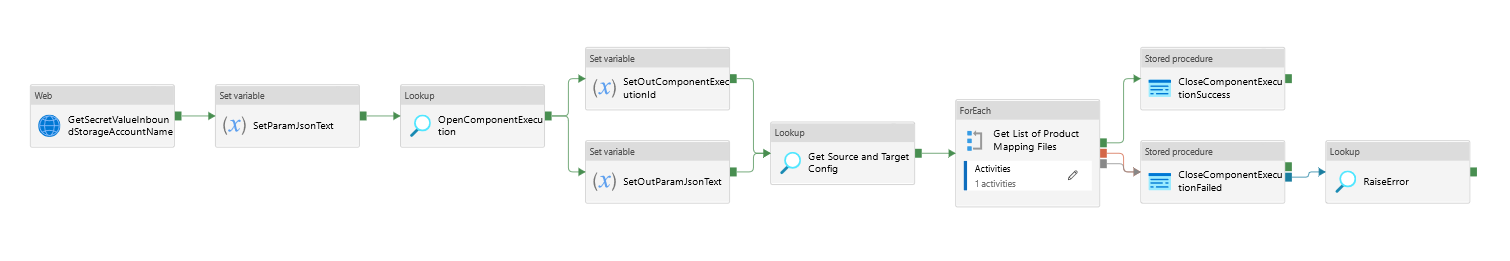
"source":"MDM\_ProductMappings/",

"target":"MDM\_ProductMappings"

}

]

"source" - source folder in BP2  
"target" - destination folder in Uploader 'incoming' container



### in-CopyRDSDictionaries

Purpose: To copy latest RDSv2 files from BP2 to Uploader's incoming storage account  
Used by: Backend

| **Parameter name** | **Data type** | **Description** | **Sample value** |
| --- | --- | --- | --- |
| in\_parnt\_comp\_exctn\_id | INT | Identifier of callers Component Execution | 1 |
| in\_user\_name | STRING | Name of the user performing the action | surname.ns@pg.com |

**Description:**

Pipeline performs following steps:

* establishes connection to Storage account
* opens component
* gets list of RDS files
* copy latest RDS files from BP2 to Uploader incoming storage account
* closes all opened components and actions

**Specification for RDSCopy.json:**

[

{

"source":"cdl\_unrefined\_masterdata/prod\_hier\_dim/",

"target":"cdl\_unrefined\_masterdata/prod\_hier\_dim"

},

{

"source":"cdl\_unrefined\_masterdata/cust\_hier\_dim/",

"target":"cdl\_unrefined\_masterdata/cust\_hier\_dim"

},

{

"source":"cdl\_unrefined\_masterdata/geo\_hier\_dim/",

"target":"cdl\_unrefined\_masterdata/geo\_hier\_dim"

},

{

"source":"cdl\_unrefined\_masterdata/proft\_ctr\_hier\_dim/",

"target":"cdl\_unrefined\_masterdata/proft\_ctr\_hier\_dim"

},

{

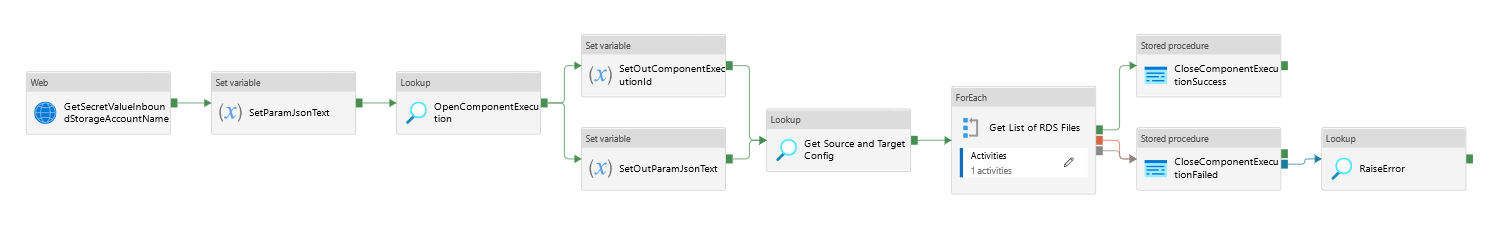
"source":"cdl\_unrefined\_masterdata/prod\_life\_cycle\_dim/",

"target":"cdl\_unrefined\_masterdata/prod\_life\_cycle\_dim"

}

]

"source" - source folder in BP2  
"target" - destination folder in Uploader 'incoming' container



### in-CopyShipments

Purpose: To copy latest Shipment files from BP2 to Uploader's incoming storage account  
Used by: Backend

| **Parameter name** | **Data type** | **Description** | **Sample value** |
| --- | --- | --- | --- |
| in\_parnt\_comp\_exctn\_id | INT | Identifier of callers Component Execution | 1 |
| in\_user\_name | STRING | Name of the user performing the action | surname.ns@pg.com |

**Description:**

Pipeline performs following steps:

* establishes connection to Storage account
* opens component
* gets list of shipments files
* copy latest shipments files from BP2 to Uploader incoming storage account
* closes all opened components and actions

**Specification for ShipmentsCopy.json:**

[

{

"source":"ship\_step\_100\_sode\_hist\_star\_work\_tc\_lc/",

"target":"shipments/ship\_step\_100\_sode\_hist\_star\_work\_tc\_lc"

},

{

"source":"an\_ind\_ship\_step\_100\_rtdc\_hist\_star/",

"target":"shipments/an\_ind\_ship\_step\_100\_rtdc\_hist\_star"

},

{

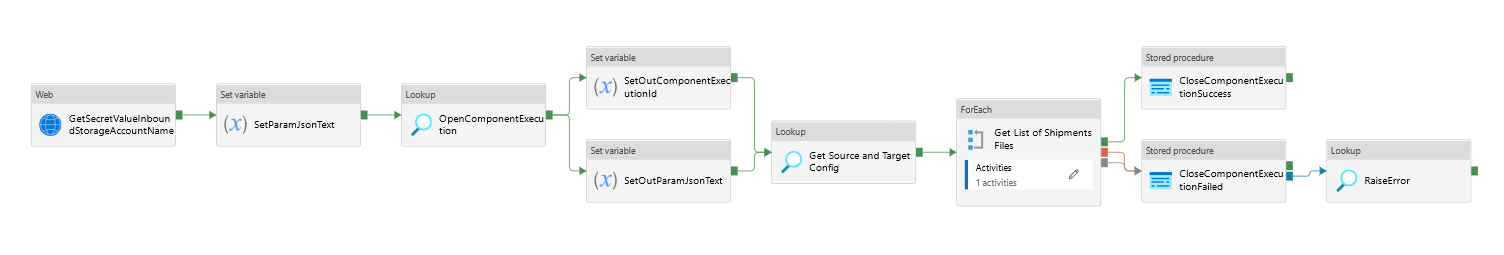
"source":"eu\_ind\_sh/",

"target":"shipments/eu\_ind\_sh"

}

]

"source" - source folder in BP2  
"target" - destination folder in Uploader 'incoming' container



**Load**

### ld-AutomatedCommunication

**Purpose:** To Send an Automated E-mail to the Ops team, after Load completion.

**Used by:** ld-LoadMainGlobal, ld-LoadMainSelective

| **Parameter name** | **Data type** | **Description** | **Sample value** |
| --- | --- | --- | --- |
| load\_type | string | Type of the Load being performed | Global / Selective |
| input\_json | string/json | In case load tyoe is selective, an Input Json will signify all the markets for which selective load is being performed | [{"in\_mkt\_grp\_name":"Europe - EE","in\_cust\_prcss\_ind":"Y","in\_prod\_prcss\_ind":"Y","in\_shpmt\_prcss\_ind":"Y","in\_direct\_prcss\_ind":"N","in\_indirect\_prcss\_ind":"Y"},{"in\_mkt\_grp\_name":"Europe - TURKEY","in\_cust\_prcss\_ind":"Y","in\_prod\_prcss\_ind":"Y","in\_shpmt\_prcss\_ind":"Y","in\_direct\_prcss\_ind":"N","in\_indirect\_prcss\_ind":"Y"},{"in\_mkt\_grp\_name":"Europe - CCAR","in\_cust\_prcss\_ind":"Y","in\_prod\_prcss\_ind":"Y","in\_shpmt\_prcss\_ind":"Y","in\_direct\_prcss\_ind":"N","in\_indirect\_prcss\_ind":"Y"}] |
| laod\_status | string | Indication whether the Load was success or failure | Completed/Fail |
| Pipelinestarttime | timestamp | The Start time of the pipeline to calculate the runtime of the Laod | 2023-02-28 03:50:12.570 |

**Descrition:**  
This Pipeline is used to send an E-mail to the Ops team, once the Load (Selective / Global) is either completed or failed.  
In case the Load type is selective the, the E-mail will also list out the markets for which were part of the laod, in case its a global laod it will written as 'ALL'.  
Along with the market information, the runtime of the Load is also shared via E-mail.

### ld-LoadRefreshVldInd

**Purpose:** Used to refresh the sys\_invld\_ind for all input templates which are been refreshed during the load. All the rows for actuals templates in uploader must be marked as valid for the reporting rows.

**Used by:** ld-ActualLoad

| **Parameter name** | **Data type** | **Description** | **Sample value** |
| --- | --- | --- | --- |
| Path | string | Partial Notebook path in ADB | /Uploader/ |
| in\_parnt\_comp\_exctn\_id | string | Component Execution ID for Logging | 1473 |
| in\_user\_name | string | username of the user who triggered the pipeline | colaco.cc |
| in\_param\_json\_txt | json | Parameters for Loading | [{"in\_cust\_prcss\_ind":"Y","in\_prod\_prcss\_ind":"Y","in\_shpmt\_prcss\_ind":"Y"}] |

**Description:** This pipeline is executed as part of the ld-ActualLoad pipeline. This pipeline uses the mkt\_grp\_prcsg\_cnfg.csv file to gather all the market groups which are been processed. using this information, query is generated using ADB notebook (sys\_inv\_chk\_pick\_mkt\_grp). This query is then used to list out all the input templates which are in scope for the current load. Additionally a lookup is used to update the sys\_invld\_ind = 'N' for all the rows which belong to the reporting month.

### ld-AutomatedGlobalLoad

**Purpose:** To Trigger the following pipelines

in-CopyShipments  
in-CopyRDSDictionaries  
in-CopyCustomerMapping  
in-CopyProductMapping  
ld-LoadMainGlobal

**Parameters:** None

**Triggering Mechanism**:This Pipeline will be triggered from CNGC-ADF via REST API as part of the 3WD pipeline(common\_3wd-pipeline). Goal of this pipeline is to automate the global load, so that there is no waste of time due to manual intervention.

### ld-RefreshTDCLoad

**Purpose:** To Refresh the Europe - ALL TDC input file.

**Used by**: ld-ActualLoad

| **Parameter name** | **Data type** | **Description** | **Sample value** |
| --- | --- | --- | --- |
| Path | string | Partial Notebook path in ADB | /Uploader/ |
| in\_parnt\_comp\_exctn\_id | string | Component Execution ID for Logging | 1473 |
| l\_param\_json\_txt | json | Parameters for Loading | [{"in\_cust\_prcss\_ind":"Y","in\_prod\_prcss\_ind":"Y","in\_shpmt\_prcss\_ind":"Y"}] |

**Description:**

As Europe - ALL is a derived input file (dervied from all European Marktes) a separate pipeline is created to Complile all the TDC inputs from all Europe Marktes and Load them into Europe - ALL TDC input file. This pipeline is triggered from ld-ActualLoad where there is a request to perform a Global or a selective Load.

### ld-LoadMainGlobal

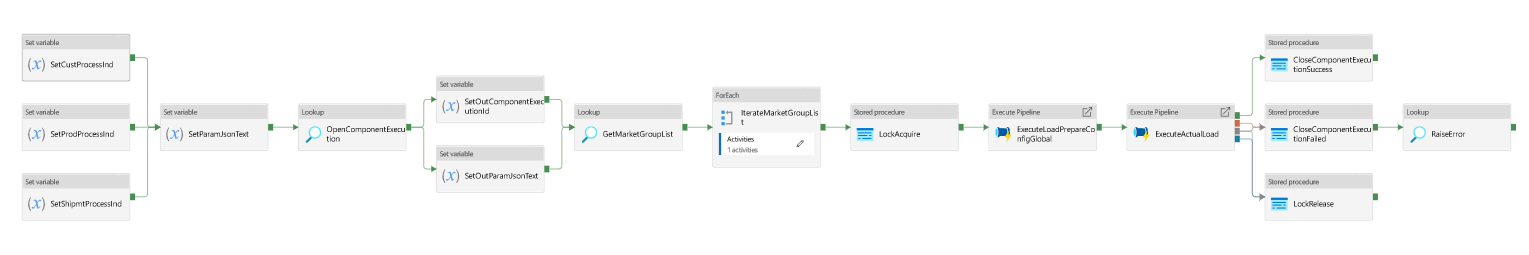
Purpose: to load data from incoming storage account to Uploader application for all market groups  
Used by: Backend

| **Parameter name** | **Data type** | **Description** | **Sample value** |
| --- | --- | --- | --- |
| in\_parnt\_comp\_exctn\_id | STRING | Identifier of callers Component Execution | 1 |
| in\_user\_name | STRING | Name of the user performing the action | surname.ns@pg.com |
| in\_cust\_prcss\_ind | STRING | Indicator if customer data should be loaded | N |
| in\_prod\_prcss\_ind | STRING | Indicator if product data should be loaded | N |
| in\_shpmt\_prcss\_ind | STRING | Indicator if shipment data should be loaded | N |

**Description:**

Pipeline performs the following steps:

* establishes connection to storage account
* opens new component
* gets marked group list
* loads data for all market groups for selected areas (customers, products, shipments)
* closes all opened components



### ld-LoadMainSelective

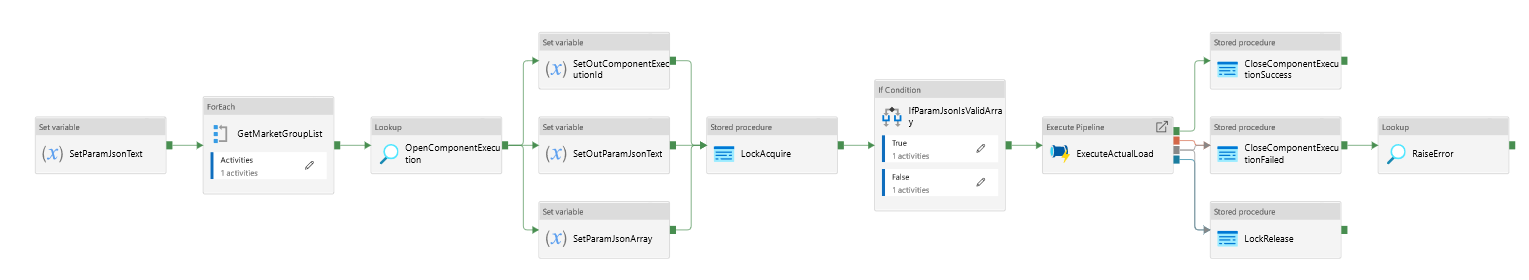
Purpose: to load data from incoming storage account to Uploader application for selected groups  
Used by: Backend

| **Parameter name** | **Data type** | **Description** | **Sample value** |
| --- | --- | --- | --- |
| in\_parnt\_comp\_exctn\_id | STRING | Identifier of callers Component Execution | 1 |
| in\_user\_name | STRING | Name of the user performing the action | surname.ns@pg.com |
| in\_json\_txt | STRING | Definition which data should be loaded | [{"in\_mkt\_grp\_name":"AMA","in\_cust\_prcss\_ind":"Y","in\_prod\_prcss\_ind":"N","in\_shpmt\_prcss\_ind":"N"},{"in\_mkt\_grp\_name":"LA","in\_cust\_prcss\_ind":"Y","in\_prod\_prcss\_ind":"N","in\_shpmt\_prcss\_ind":"N"}] |

**Description:**

Pipeline performs the following steps:

* establishes connection to storage account
* opens new component
* gets marked group list
* loads data for selected market groups for selected areas (customers, products, shipments)
* closes all opened components



### ld-DumpProfitCenterAndMarketMappingToParquet

Purpose: to dump profit center and market mapping from database to parquet file  
Used by: **Other Pipeline**

| **Parameter name** | **Data type** | **Description** | **Sample value** |
| --- | --- | --- | --- |
| l\_proft\_ctr\_mkt\_mapng\_file\_name | string | File name to get the file definition version ID | Profit Center and Market Mapping |

Description:

Pipeline performs following steps:

1. Get file\_dfntn\_vers\_id from md.file\_dfntn\_vers\_prc\_vw where file\_name = 'Profit Center and Market Mapping'.
2. Calls Common/cm-DumpDBTableIntoFile pipeline to dump the file with supplied paramaters:

* in\_file\_dfntn\_vers\_id
* in\_cond\_txt: empty
* in\_cntnr\_name\_txt: internal
* in\_path\_txt: cnfg\_files
* in\_file\_name\_txt: profit\_center\_and\_market\_mapping

### ld-ReloadGeoMapping

**Summary**

**Purpose**

To pump csv files from a **selected location** to a **created stage table**

**Used by**

Other Pipeline

**Parameters**

| **Parameter name** | **Data type** | **Description** | **Sample value** |
| --- | --- | --- | --- |
| in\_cntnr\_name\_txt | String | Source location - blob container name | incoming |
| in\_path\_txt | String | Source location - folder name within the specified container in **in\_cntnr\_name\_txt** | delimited |
| in\_file\_name\_txt | String | Source location - file name within the specified folder path in **in\_path\_txt** without file extension | stage cnfg\_geo\_mapng 1 |
| in\_file\_dfntn\_vers\_id |  | See notes [1] |  |
| in\_data\_type\_rflcd\_ind |  | See notes [1] |  |
| in\_parnt\_comp\_exctn\_id |  | See notes [1] |  |
| in\_user\_name |  | See notes [1] |  |
|  |  |  |  |

**Notes**

1. See the parameter on pro\_file\_stage\_creat

**Process**

Pipeline performs following steps:

1. Get the file metadata (column names, data type, etc.) from the source location using parameters **in\_cntnr\_name\_txt** / **in\_path\_txt** / **in\_file\_name\_txt**.

test-adf-common/dbinputtoparquet/test1

1. Create stage table via pro\_file\_stage\_creat.
2. Get the target metadata of the created stage table from Step 2.
3. Create a dynamic json mapping using the meta data of the source on Step 1 and map it into sink meta data on Step 3.
4. Insert the data into the created stage table from Step 2 using the mapping configuration on Step 4.  
   ***Note:*** Assuming CSV has the same structure of the table (same column name, column number, etc.), inserting will encounter no problems.

### System columns

This page describes what are System Columns in Input Files and which objects contains them.

| **Column name** | **Data type** | **Working table** | **Working view** | **Submit table** | **Description** |
| --- | --- | --- | --- | --- | --- |
| sys\_invld\_ind | CHAR(1) | Yes(\*) | Yes(\*) | No | Is row marked as Invalid, (\*): added if file\_dfntn\_vers\_prc.invld\_ind=Y |
| sys\_obslt\_ind | CHAR(1) | Yes(\*\*) | Yes(\*\*) | No | Is row marked as Obsolete, (\*\*) added if file\_dfntn\_vers\_prc.obslt\_ind=Y |
| sys\_last\_mdfd\_datetm | DATETIME2(7) | No | Yes | No | Date of last modification including Load Process time for new rows |
| sys\_last\_mdfd\_user\_name | VARCHAR(50) | No | Yes | No | User who uploaded as the last or "AUTO\_LOAD" if the only action so far was Load Process |
| sys\_row\_id | INT | Yes | Yes | Yes | Row surrogate key |
| sys\_init\_actn\_id | INT | Yes | Yes | Yes | Load/Upload Action which added this row |
| sys\_last\_uplod\_actn\_id | INT | Yes | Yes | Yes | Identifier of most recent Upload Action of this row |
| sys\_mkt\_id | INT | Yes(\*\*\*) | Yes(\*\*\*) | Yes(\*\*\*) | Identifier of Market, (\*\*\*): added if file\_dfntn\_vers\_prc.mkt\_col\_id is not NULL, so it's available only in Load Input Files |

### Load columns

This page describes full list of load columns which can be added into the file definition.

| **Description** | **Column name** | **Data type** |
| --- | --- | --- |
| Is month in range of Actual Input File | actl\_ind | BOOLEAN |
| Area ID | area\_id | TEXT(20) |
| Area Name | area\_name | TEXT(50) |
| Brand ID | brand\_id | INTEGER |
| Brand Name | brand\_name | TEXT(50) |
| Brand Form ID | brand\_form\_id | INTEGER |
| Brand Form Name | brand\_form\_name | TEXT(500) |
| Category ID | categ\_id | INTEGER |
| Category Name | categ\_name | TEXT(50) |
| Channel | chanl\_val | TEXT(50) |
| Customer ID | cust\_id | INTEGER |
| Custom Cluster | custm\_clstr\_name | TEXT(50) |
| Custom Region | custm\_regn\_name | TEXT(50) |
| Custom SMO | custm\_smo\_name | TEXT(50) |
| Indicates Direct/Indirect | dirct\_indir\_ind | TEXT(50) |
| FPC ID | fpc\_id | INTEGER |
| FPC Name | fpc\_name | TEXT(500) |
| Is month in range of Forecast Input File | frcst\_ind | BOOLEAN |
| FY | fy\_code | TEXT(50) |
| Group ID | grp\_id | TEXT(20) |
| Group Name | grp\_name | TEXT(50) |
| Historical indicator | histr\_ind | BOOLEAN |
| Local Currency | local\_crncy\_code | TEXT(3) |
| Minor Country ID | minor\_cntry\_id | TEXT(10) |
| Minor Country Name | minor\_cntry\_name | TEXT(50) |
| Market Group | mkt\_grp\_name | TEXT(50) |
| Market Geo ID | mkt\_geo\_id | TEXT(50) |
| Market Level | mkt\_lvl | TEXT(50) |
| Market | mkt\_name | TEXT(50) |
| Month | mth\_num | MONTH |
| Profit Center ID | proft\_ctr\_id | INTEGER |
| Profit Center Level | proft\_ctr\_lvl | INTEGER |
| Profit Center Name | proft\_ctr\_name | TEXT(50) |
| RDS Product Hierarchy ID | rds\_prod\_hier\_id | INTEGER |
| RDS Product Hierarchy Level | rds\_prod\_hier\_lvl | TEXT(50) |
| Region ID | regn\_id | TEXT(10) |
| Region Name | regn\_name | TEXT(50) |
| Reporting Country ID | rptng\_cntry\_id | INTEGER |
| Reporting Country Name | rptng\_cntry\_name | TEXT(50) |
| Reporting Customer ID | rptng\_cust\_id | INTEGER |
| Reporting Customer Name | rptng\_cust\_name | TEXT(50) |
| Subsector ID | sbstr\_id | INTEGER |
| Subsector Name | sbstr\_name | TEXT(50) |
| Shipments Direct (Infopage) | shpmt\_dirct\_ind | BOOLEAN |
| Shipments indicator | shpmt\_ind | BOOLEAN |
| Shipments Indirect (RTDC) | shpmt\_indir\_ind | BOOLEAN |

## Database users, roles, schemas

The following users are created in SQL databases:

* dbrcnosgcuploadern01 (Azure Databricks user, with Service Principal)
* adf-cnosgc-uploader-n01, adf-cnosgc-uploader-uat-n01, adf-cnosgc-uploader-prod-n01 (Azure Data Factory users, with Managed Identity)
* front\_end\_api\_user (used for FE API)
* az\_deploy\_uzer (used for deployment process)

The following DB users are associated with Genie Groups (FROM EXTERNAL PROVIDER clause):

* GDSG-CNOSGCUploader-DB-read
* GDSG-CNOSGCUploader-DB-write
* GDSG-CNOSGCUploader-DB-execute

SQL Database roles with specific access:

* uploader\_read\_role (SELECT)
* uploader\_write\_role (INSERT, UPDATE, DELETE)
* uploader\_execute\_role (EXECUTE)
* adf\_role - specific privileges needed for ADF user
* adb\_role - specific privileges needed for Databricks user
* front\_end\_api\_role - role for FE connection

Schemas in SQL database:

* archive - archived submissions
* main - main schema to store SQL logic
* md - applications Metadata
* input - schema for Input File maintenance
* stage - persistent processing objects
* tmp - temporary objects

Databricks Schemas

The following schemas are used in Azure Databricks:

* SET spark.uploader.read.schema=upldr\_main;
* SET spark.uploader.write.schema=upldr\_main;