Technical Documentation - RTR - Profit and Loss (P&L) + Integrated Margin (IM) (WBP)

- 5.6 Roadmap

Access Management

Roles & Access

Role Code	Role Description	Explanation
ZR_RCS_CA_M432	P&L – Upload Data - Non-ERP ZPL_FILE	Access to transactions ZPL_FILE
ZR_RCS_CA_M12	PL - P&L Reporting	Role menu
ZBI_RCS_FI_A33	P & L – Profit and Loss - End User role	End user role

Authorization Object

Link to the BW Catalog of role

https://drive.google.com/open?id=10GEfKYqrT1eeTO_uHYAheL1GX7L5y_pvH0KQU64qh5I

Authorization object	Explanation
GBU	ZR_*_CA_P05
PRS Company	ZR_*_CA_P07
PRS Area(C_COMPCDE__C_MNGAREA)	ZR_*_CA_P08
GL Type(0GL_ACCOUNT__C_GL_TYPE)	ZR_*_CA_P10
Plant (C_PLANT)	ZR_*_CA_P02
Company (C_COMPCDE)	ZR_*_CA_P01
Authorization Scope (C_AUTHMA)	ZR_*_CA_P00

DataFlow

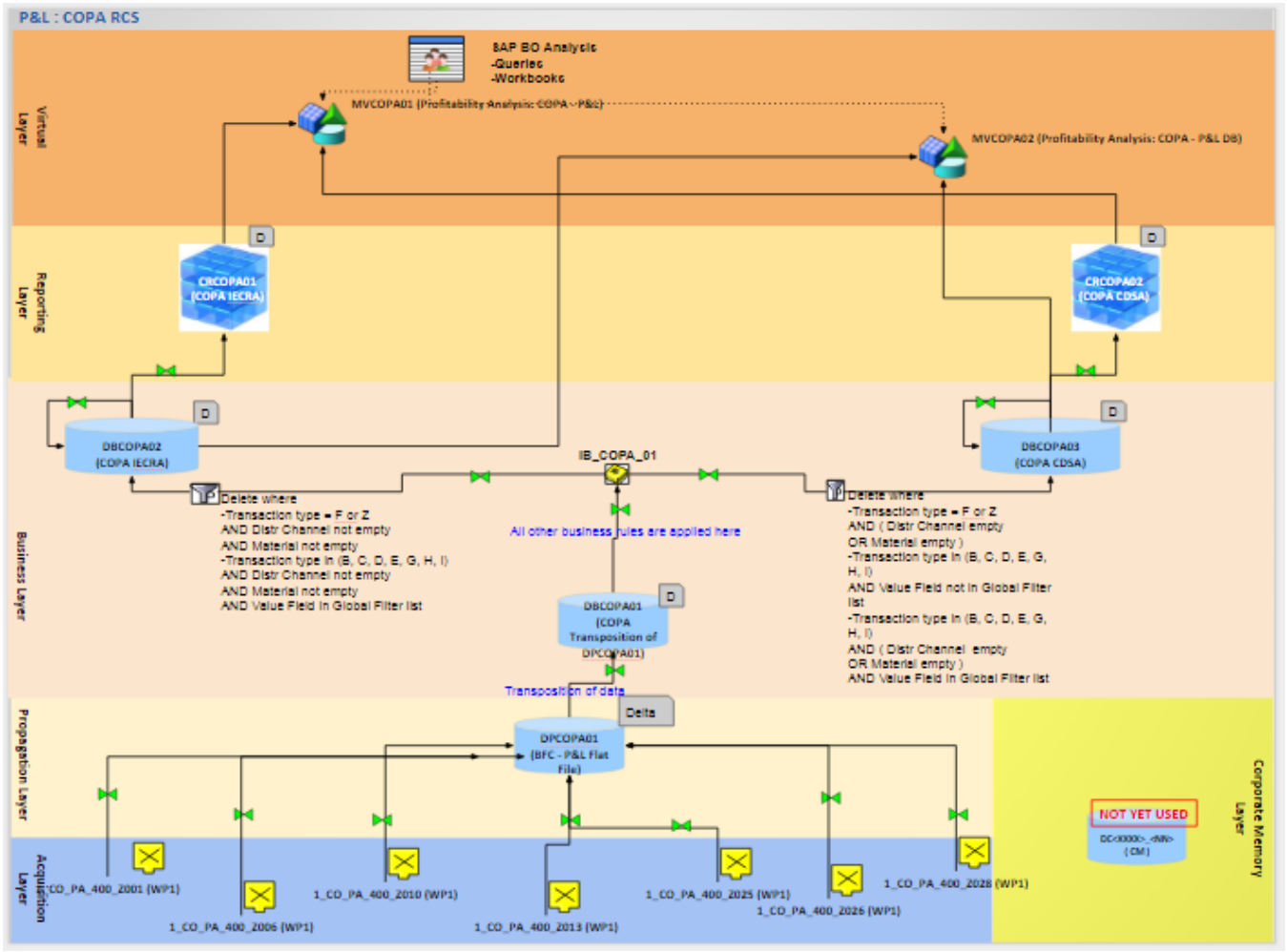
Overview

Solstice Google Drive document

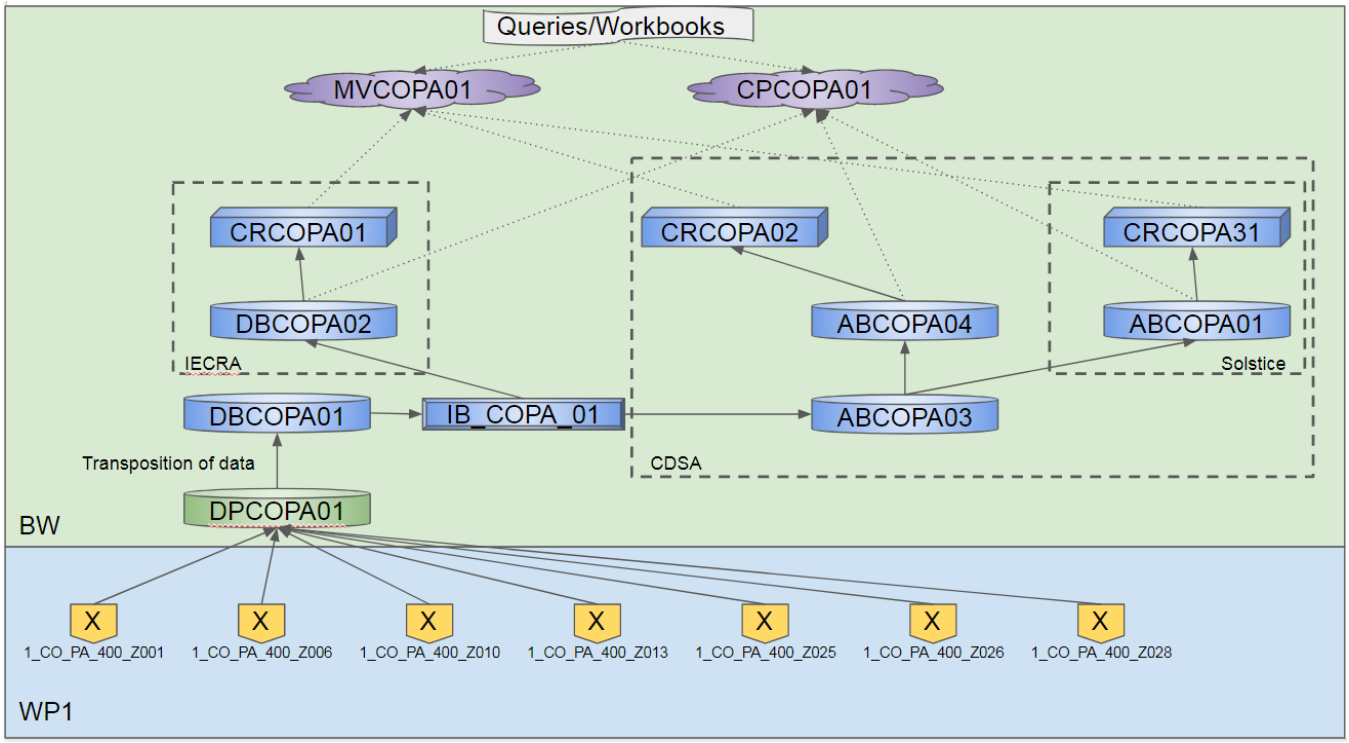
MVCOPA01

P&L

WP1

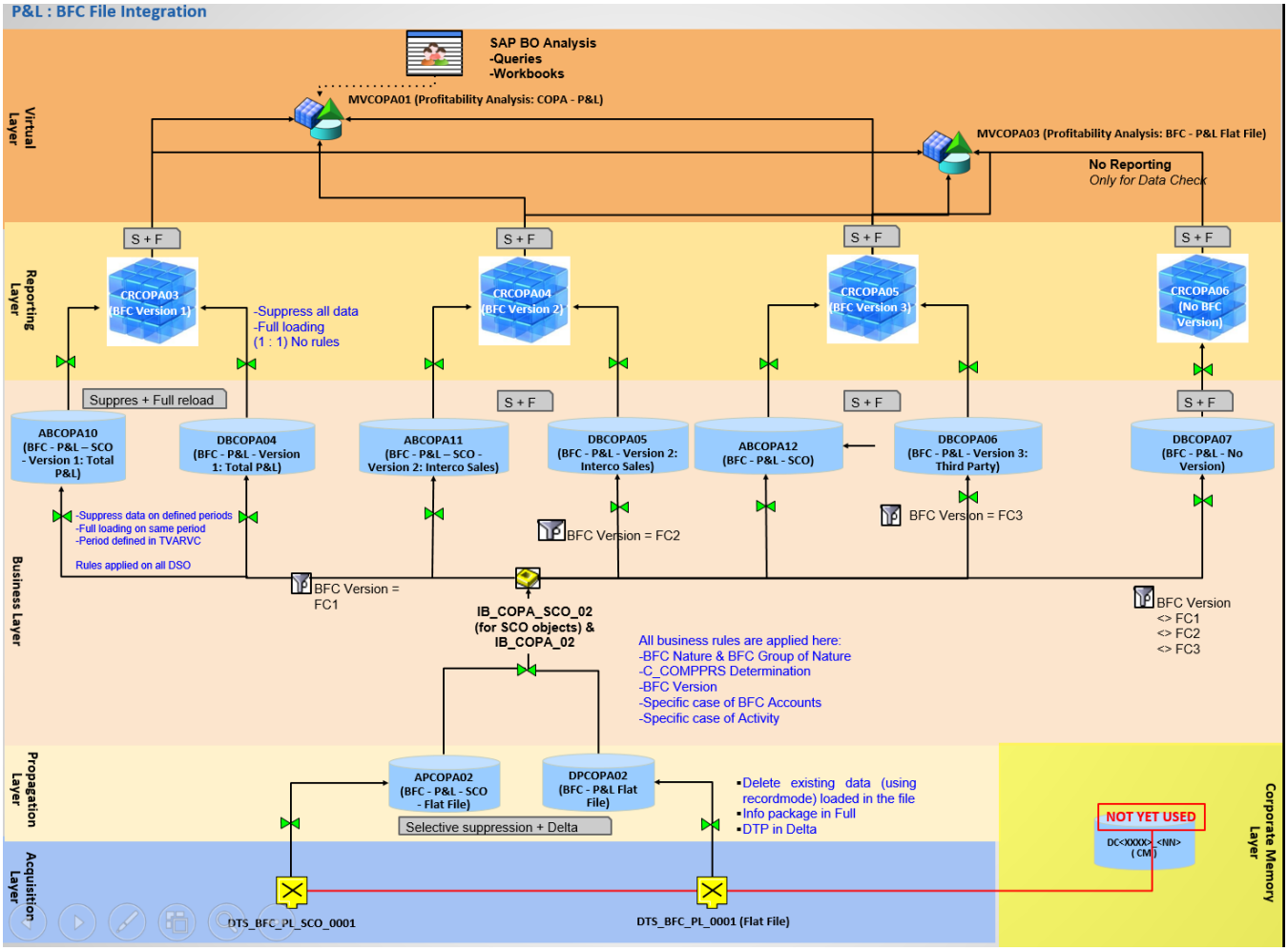


Updated dataflow after Solstice update (Rules are unchanged except for the Solstice split)

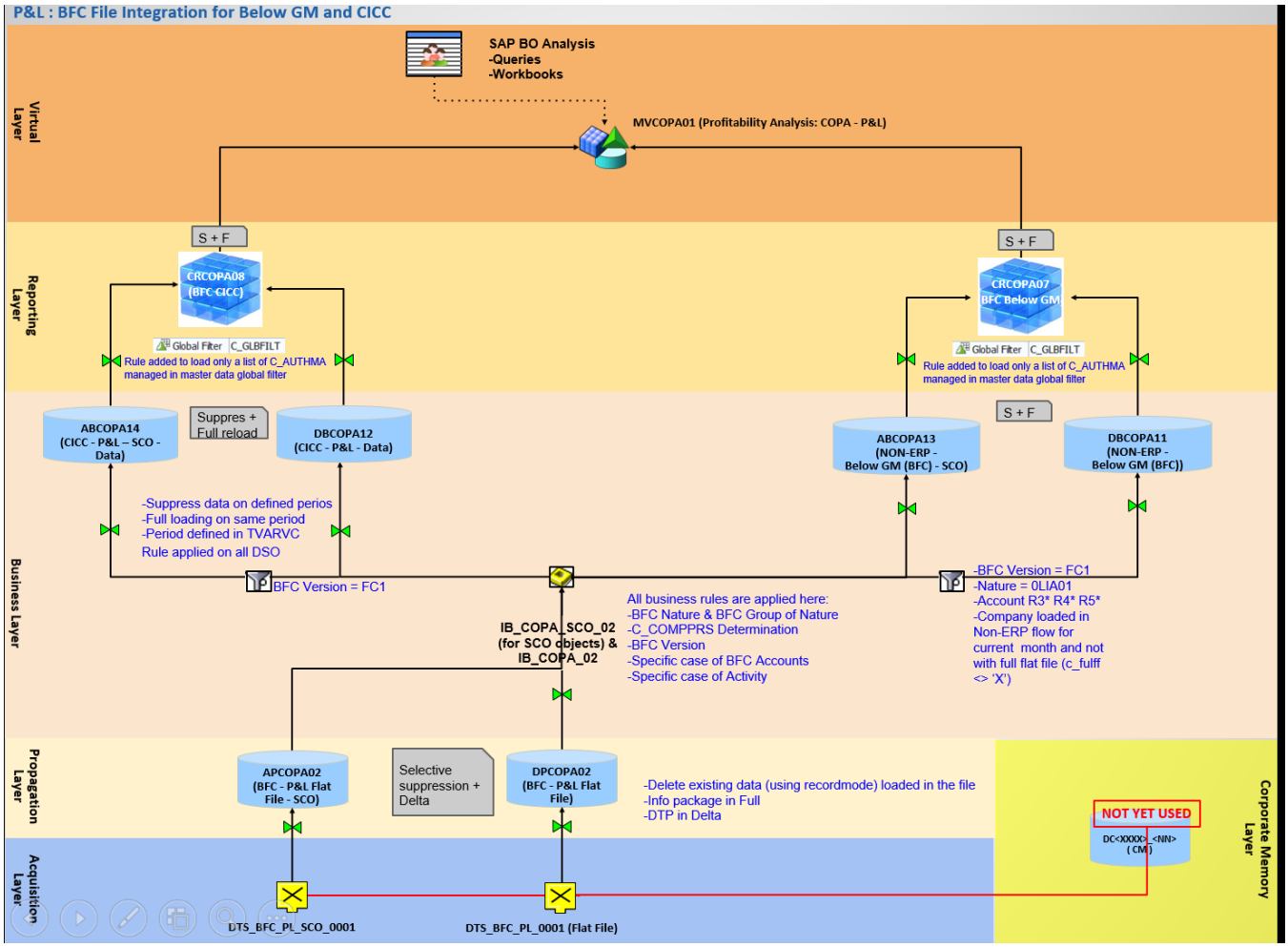


BFC

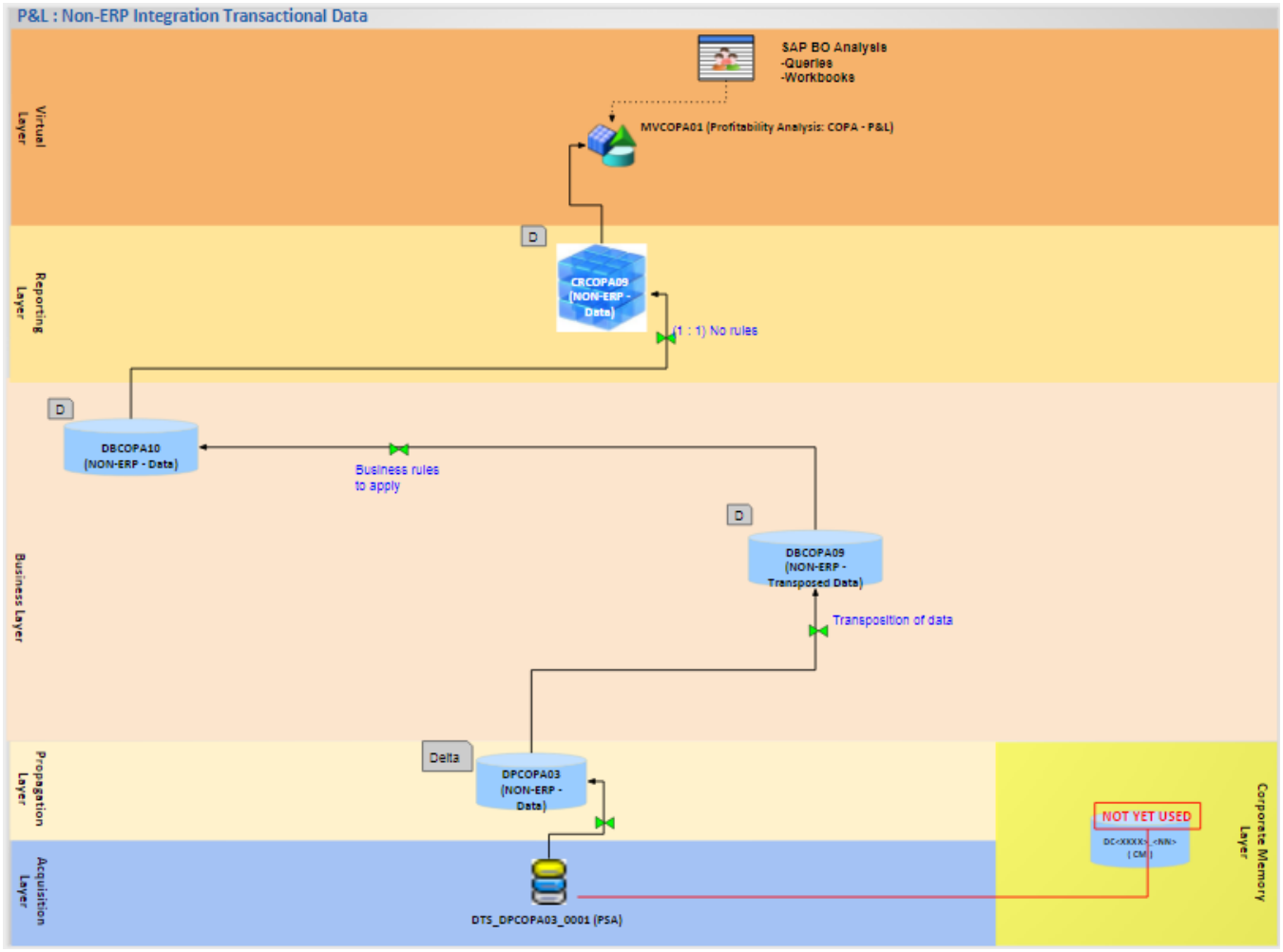
Last change : PO2 Project 08/2023



Last change : PO2 Project 08/2023

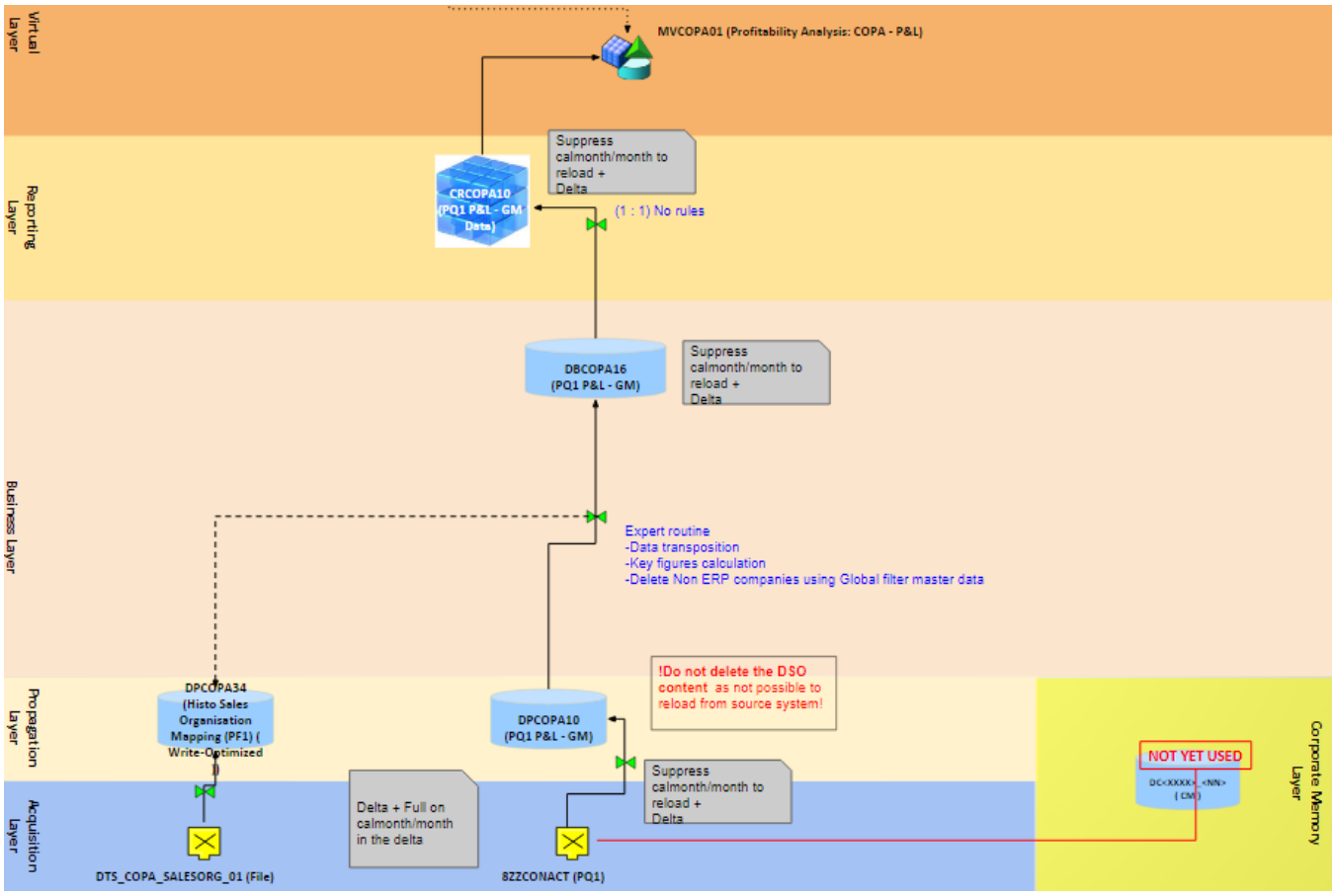

P&L : BFC File Integration for Below GM and CICC
Virtual Layer
Reporting Layer
Business Layer
Propagation Layer
Acquisition Layer
Corporate Memory Layer
SAP BO Analysis
-Queries
-Workbooks
MVCOPA01 (Profitability Analysis: COPA - P&L)
S + F
CRCOPA08 (BFC-CICC)
Global Filter C_GLBFILT
Rule added to load only a list of C_AUTHMA managed in master data global filter
S + F
CRCOPA07 BFC Below GM
Global Filter C_GLBFILT
Rule added to load only a list of C_AUTHMA managed in master data global filter
ABCOPA14 (CICC - P&L – SCO - Data)
Suppres + Full reload
DBCOPA12 (CICC - P&L - Data)
S + F
ABCOPA13 (NON-ERP - Below GM (BFC) - SCO)
DBCOPA11 (NON-ERP - Below GM (BFC))
-Suppress data on defined perios
-Full loading on same period
-Period defined in TVARVC
Rule applied on all DSO
BFC Version = FC1
IB_COPA_SCO_02 (for SCO objects) & IB_COPA_02
All business rules are applied here:
-BFC Nature & BFC Group of Nature
-C_COMPPRS Determination
-BFC Version
-Specific case of BFC Accounts
-Specific case of Activity
-BFC Version = FC1
-Nature = 0LIA01
-Account R3* R4* R5*
-Company loaded in Non-ERP flow for current month and not with full flat file (c_fulff <> 'X')
APCOPA02 (BFC - P&L Flat File - SCO)
Selective suppression + Delta
DPCOPA02 (BFC - P&L Flat File)
-Delete existing data (using recordmode) loaded in the file
-Info package in Full
-DTP in Delta
NOT YET USED
DC<XXXX>_<NN> (CM)
DTS_BFC_PL_SCO_0001
DTS_BFC_PL_0001 (Flat File)

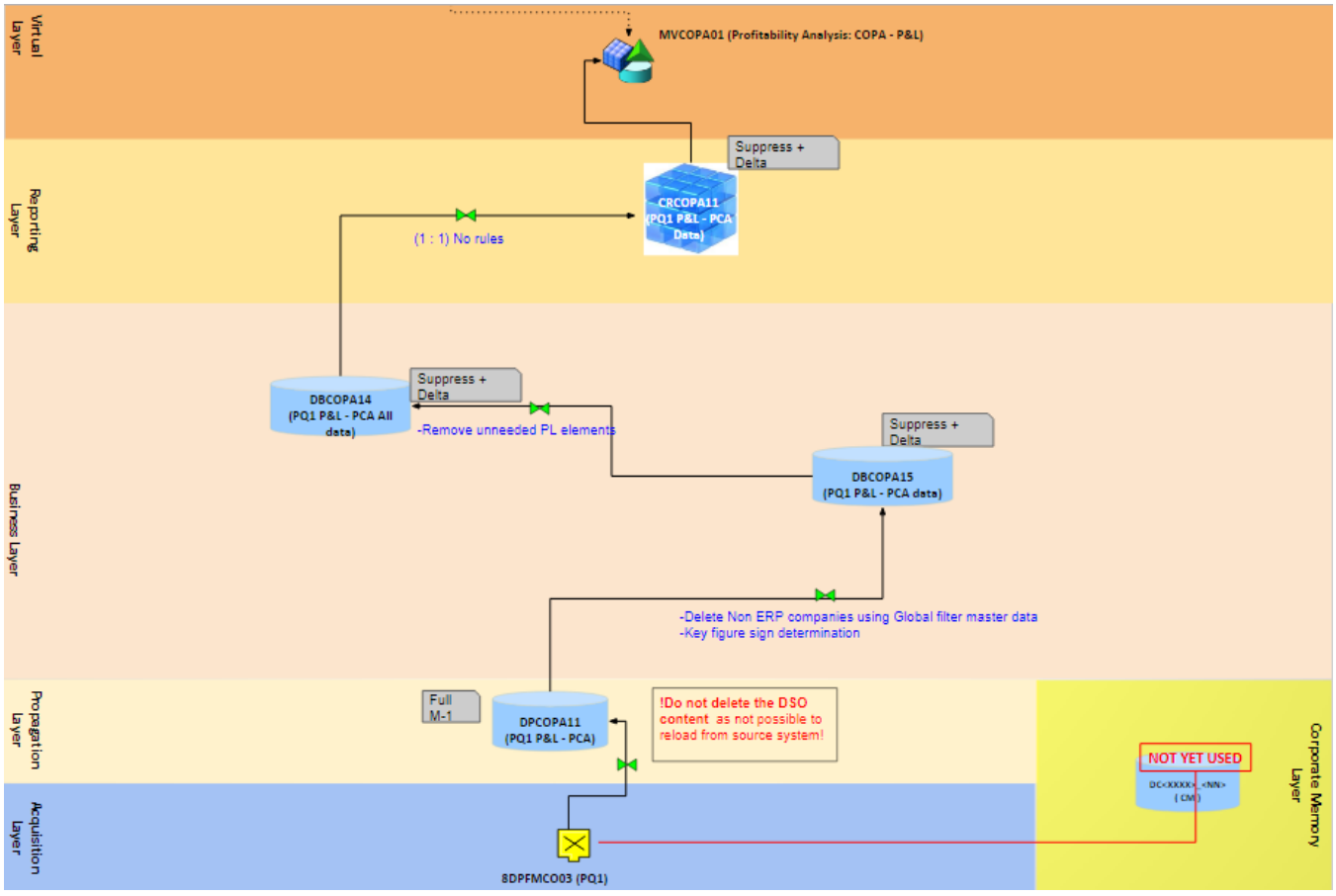
Non ERP

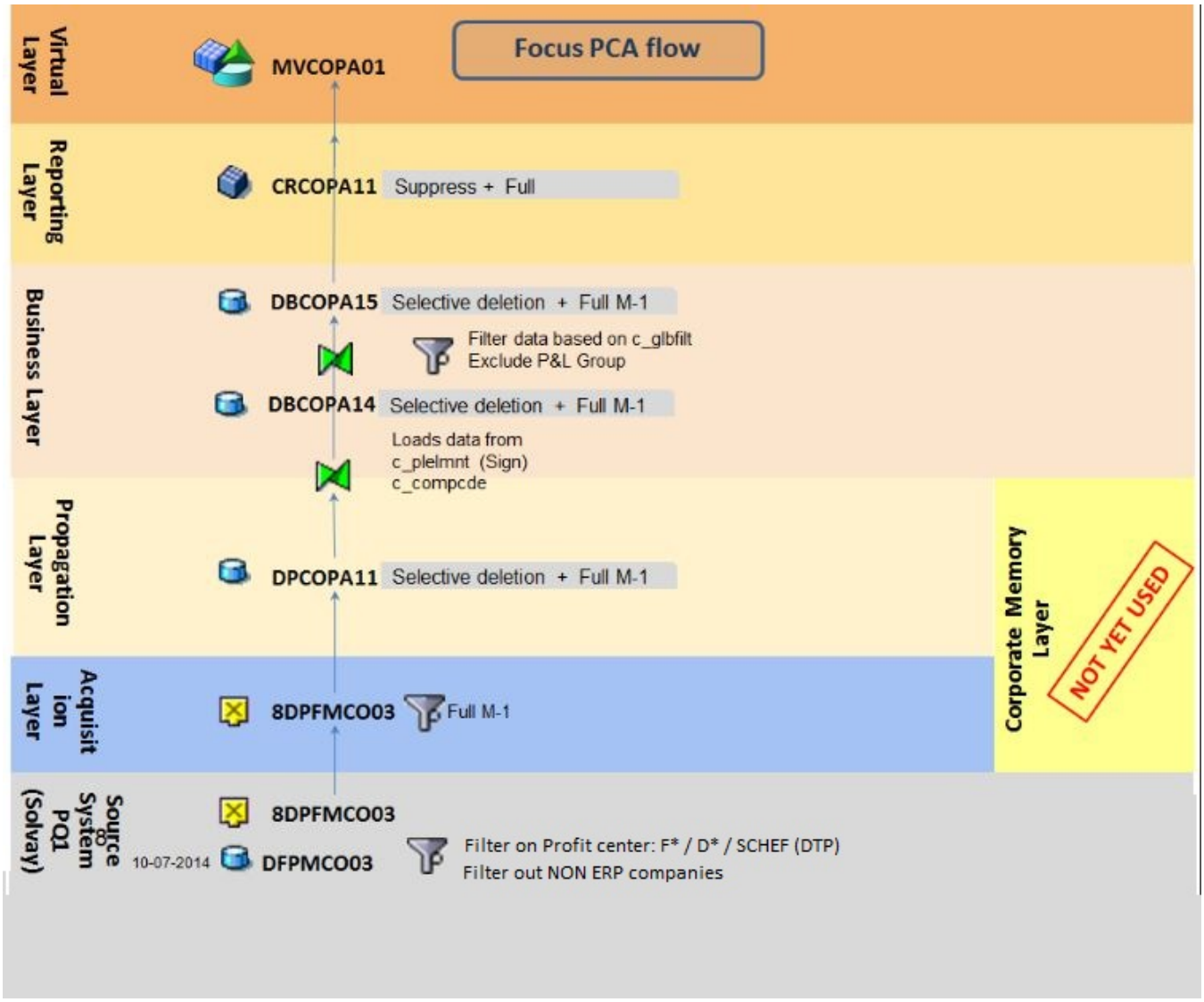


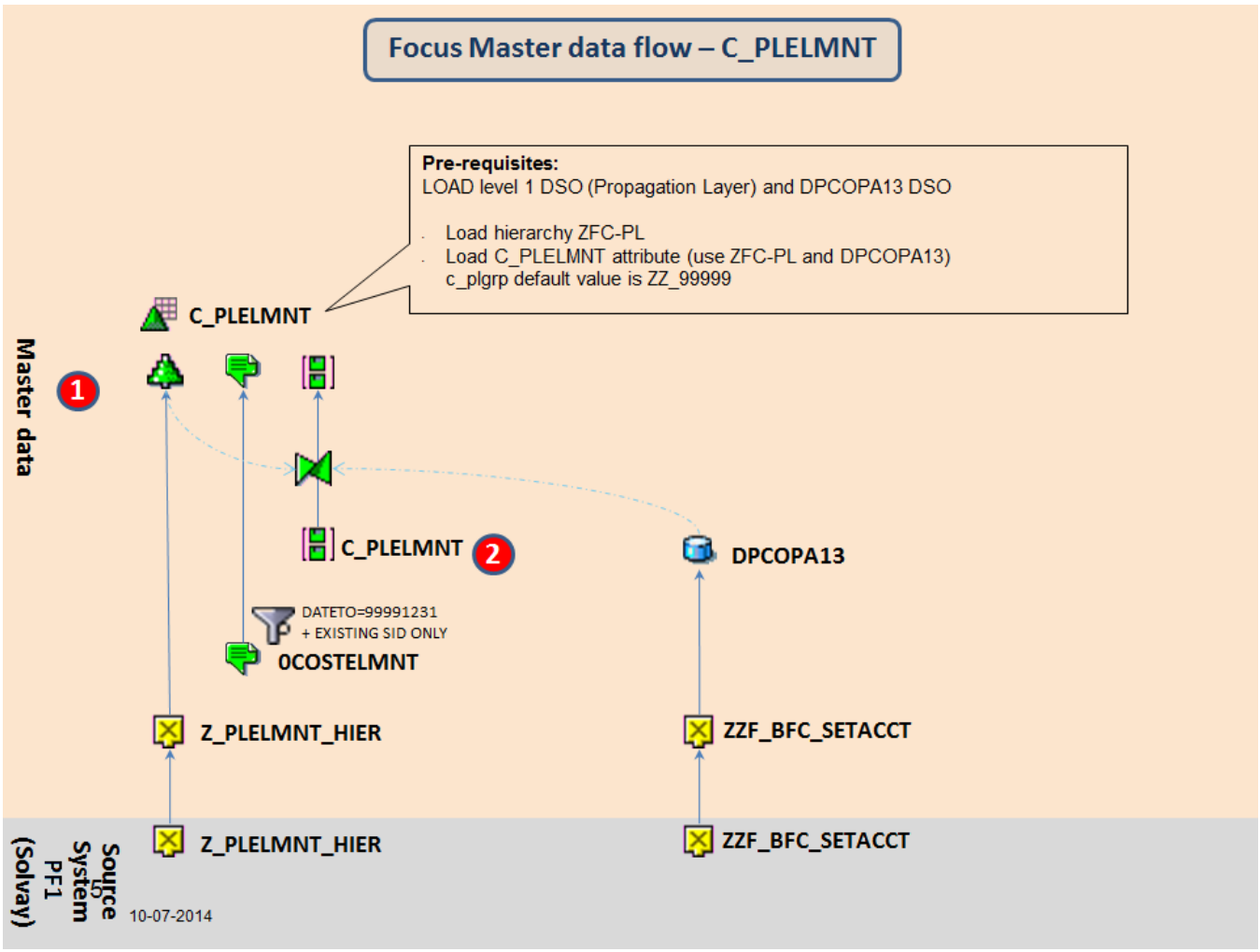
PQ1

PQ1 flows are kept only for historical data, they are replaced by PF1 flows since 01.2018.

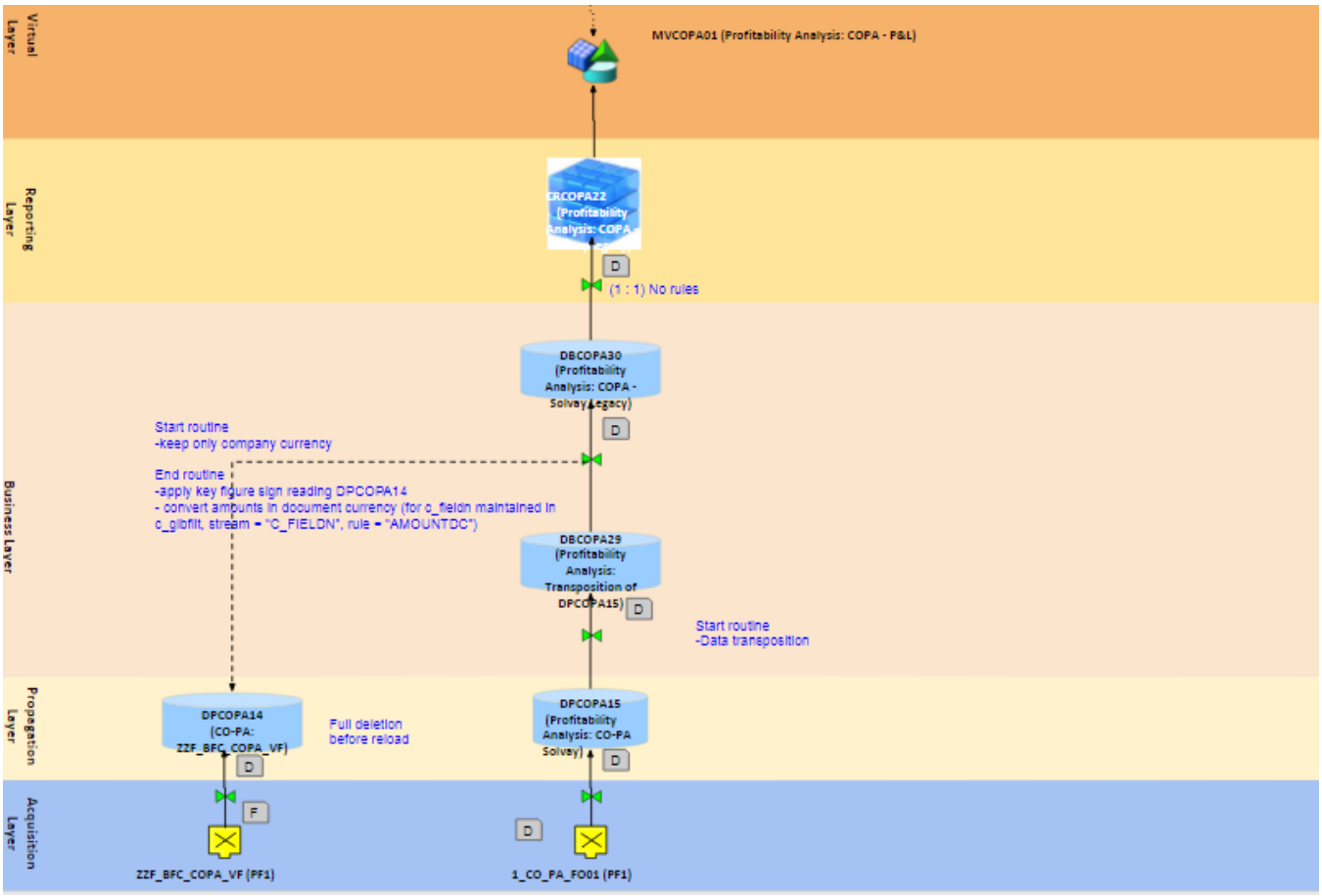

Virtual Layer
Reporting Layer
Business Layer
Propagation Layer
Acquisition Layer
Corporate Memory Layer
MVCOPA01 (Profitability Analysis: COPA - P&L)
CRCOPA10 (PQ1 P&L - GM Data)
Suppress calmonth/month to reload + Delta
(1 : 1) No rules
DBCOPA16 (PQ1 P&L - GM)
Suppress calmonth/month to reload + Delta
Expert routine
-Data transposition
-Key figures calculation
-Delete Non ERP companies using Global filter master data
DPCOPA34 (Histo Sales Organisation Mapping (PF1) (Write-Optimized))
DPCOPA10 (PQ1 P&L - GM)
!Do not delete the DSO content as not possible to reload from source system!
Suppress calmonth/month to reload + Delta
Delta + Full on calmonth/month in the delta
NOT YET USED
DTS_COPA_SALESORG_01 (File)
8ZZCONACT (PQ1)

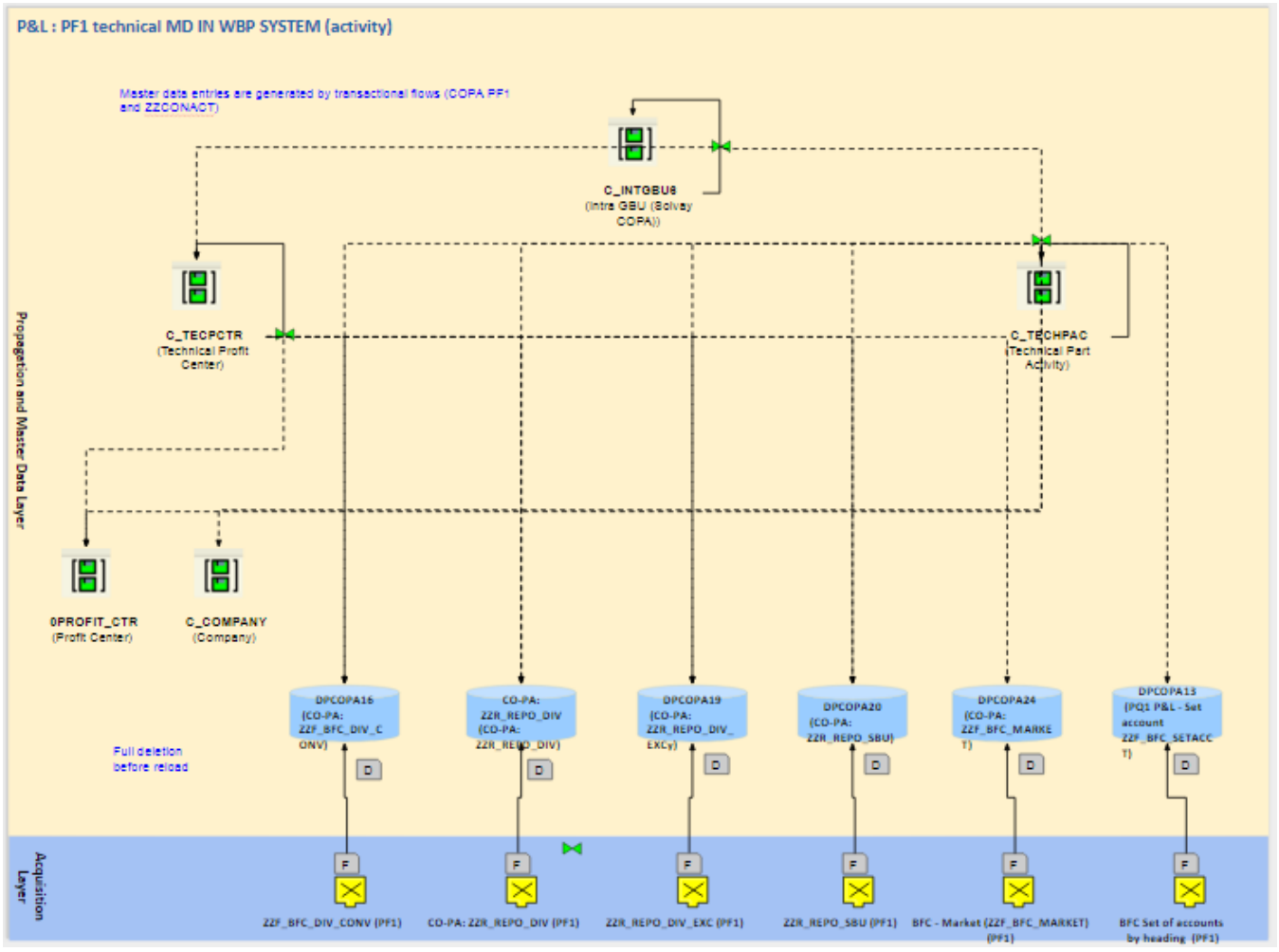

Virtual Layer
Reporting Layer
Business Layer
Propagation Layer
Acquisition Layer
Corporate Memory Layer
MVCOPA01 (Profitability Analysis: COPA - P&L)
Suppress + Delta
CRCOPA11 (PQ1 P&L - PCA Data)
(1 : 1) No rules
DBCOPA14 (PQ1 P&L - PCA All data)
Suppress + Delta
-Remove unneeded PL elements
Suppress + Delta
DBCOPA15 (PQ1 P&L - PCA data)
-Delete Non ERP companies using Global filter master data
-Key figure sign determination
Full M-1
DPCOPA11 (PQ1 P&L - PCA)
!Do not delete the DSO content as not possible to reload from source system!
NOT YET USED
8DPFMCO03 (PQ1)

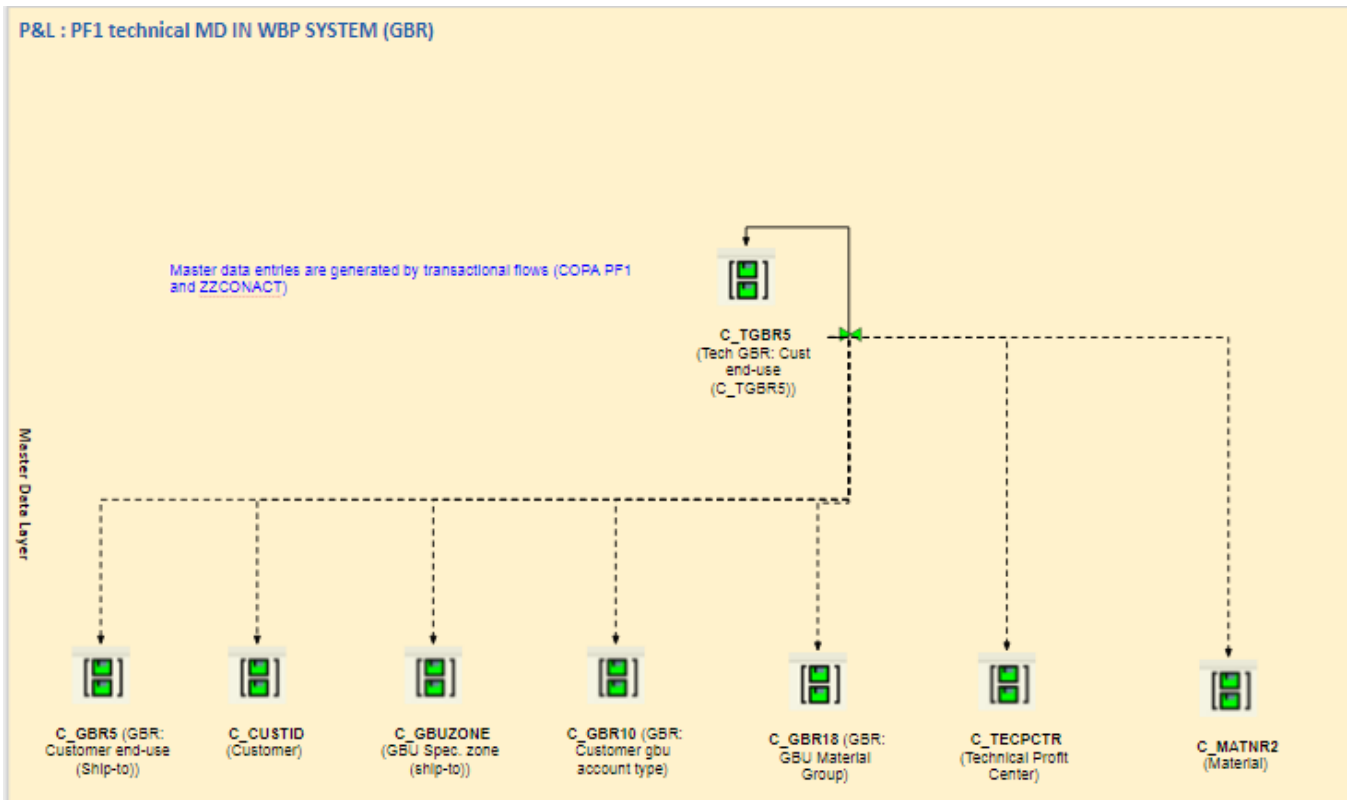

Virtual Layer
MVCOPA01
Focus PCA flow
Reporting Layer
CRCOPA11
Suppress + Full
Business Layer
DBCOPA15
Selective deletion + Full M-1
Filter data based on c_glbfilt
Exclude P&L Group
DBCOPA14
Selective deletion + Full M-1
Loads data from
c_plelmnt (Sign)
c_compcde
Propagation Layer
DPCOPA11
Selective deletion + Full M-1
Acquisition Layer
8DPFMCO03
Full M-1
Source System PQ1 (Solvay)
8DPFMCO03
10-07-2014
DFPMCO03
Filter on Profit center: F* / D* / SCHEF (DTP)
Filter out NON ERP companies
Corporate Memory Layer
NOT YET USED


Focus Master data flow – C_PLELMNT
Pre-requisites:
LOAD level 1 DSO (Propagation Layer) and DPCOPA13 DSO
Load hierarchy ZFC-PL
Load C_PLELMNT attribute (use ZFC-PL and DPCOPA13)
c_plgrp default value is ZZ_99999
C_PLELMNT
Master data
1
C_PLELMNT
2
DPCOPA13
DATETO=99991231
+ EXISTING SID ONLY
0COSTELMNT
Z_PLELMNT_HIER
ZZF_BFC_SETACCT
Z_PLELMNT_HIER
ZZF_BFC_SETACCT
Source System PF1 (Solvay)
10-07-2014

PF1:

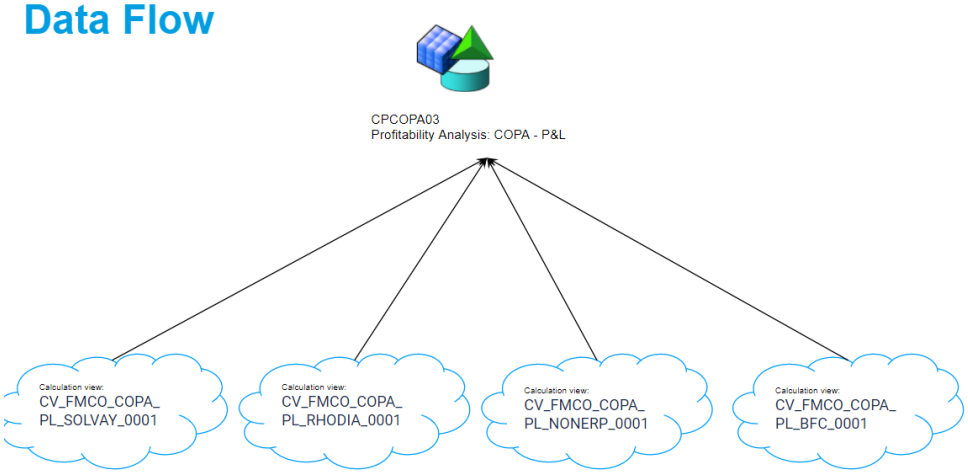

Virtual Layer
Reporting Layer
Business Layer
Propagation Layer
Acquisition Layer
MVCOPA01 (Profitability Analysis: COPA - P&L)
CRCOPA22 (Profitability Analysis: COPA -
D
(1 : 1) No rules
DBCOPA30 (Profitability Analysis: COPA - Solvay Legacy)
D
Start routine
-keep only company currency
End routine
-apply key figure sign reading DPCOPA14
- convert amounts in document currency (for c_fieldn maintained in c_glbflt, stream = "C_FIELDN", rule = "AMOUNTDC")
DBCOPA29 (Profitability Analysis: Transposition of DPCOPA15)
D
Start routine
-Data transposition
DPCOPA14 (CO-PA: ZZF_BFC_COPA_VF)
Full deletion before reload
DPCOPA15 (Profitability Analysis: CO-PA Solvay)
D
D
F
D
ZZF_BFC_COPA_VF (PF1)
1_CO_PA_FO01 (PF1)


P&L : PF1 technical MD IN WBP SYSTEM (activity)
Master data entries are generated by transactional flows (COPA PF1 and ZZCONACT)
C_INTGBU8 (Intra GBU (Solvay COPA))
C_TECPCTR (Technical Profit Center)
C_TECHPAC (Technical Part Activity)
Propagation and Master Data Layer
0PROFIT_CTR (Profit Center)
C_COMPANY (Company)
DPCOPA16 (CO-PA: ZZF_BFC_DIV_CONV)
CO-PA: ZZR_REPO_DIV (CO-PA: ZZR_REPO_DIV)
DPCOPA19 (CO-PA: ZZR_REPO_DIV_EXCy)
DPCOPA20 (CO-PA: ZZR_REPO_SBU)
DPCOPA24 (CO-PA: ZZF_BFC_MARKET)
DPCOPA13 (PQ1 P&L - Set account ZZF_BFC_SETACCT)
Full deletion before reload
D
F
Acquisition Layer
ZZF_BFC_DIV_CONV (PF1)
CO-PA: ZZR_REPO_DIV (PF1)
ZZR_REPO_DIV_EXC (PF1)
ZZR_REPO_SBU (PF1)
BFC - Market (ZZF_BFC_MARKET) (PF1)
BFC Set of accounts by heading (PF1)


P&L : PF1 technical MD IN WBP SYSTEM (GBR)
Master data entries are generated by transactional flows (COPA PF1 and ZZCONACT)
C_TGBR5 (Tech GBR: Cust end-use (C_TGBR5))
Master Data Layer
C_GBR5 (GBR: Customer end-use (Ship-to))
C_CUSTID (Customer)
C_GBUZONE (GBU Spec. zone (ship-to))
C_GBR10 (GBR: Customer gbu account type)
C_GBR18 (GBR: GBU Material Group)
C_TECPCTR (Technical Profit Center)
C_MATNR2 (Material)

CPCOPA03

Data Flow



More details here.

General presentation

Objective of the application

P&L

P&L application provides repor ts fully aligned with BFC structure (BFC Headings) down to REBIT/REBITDA able to analyze P&L starting from the BFC view (Company/Activity) and ability to drill down to Customer Material level.

A profit and loss statement (P&L) is a financial statement that summarizes the revenues, costs and expenses incurred during a specific period of time, usually a fiscal quarter or year.

These records provide information about a company's ability – or lack thereof – to generate profit by increasing revenue, reducing costs, or both.

It was done in 2 phases :

Phase 1 : COPA model of Rhodia legacy (RCS)

Phase 2 :

- Solvay legacy (PQ1)
- BFC Data
- CICC
- Below Gross Margin
- Non ERP companies

Then, with SPS project, Solvay legacy PQ1 solution was replaced by PF1 flows.

IM

Integrated Margin is linked with P&L data and providers BW and BO reports with details of the calculation of the integrated costs

Query Technical Name: BW_QRY_MVCOPA01_0004

Note : Don't use the info objects C_GBR34, C_PROD, C_MAGNIT2 from the multi provider : MVCOPA01 in any of the queries built on the Multi provider MVCOPA01 as these objects are linked to only NON-ERP.

Documentation for IM is here:

Technical Documentation - BW Integrated Margin (WBP)

Solstice

An additional split (both in the main P&L dataflow and the IM dataflow from WP1) was deemed necessary to collect the Main Specification, Program and associated Market only for the concerned Solstice companies.

Owner : SL Finance Data & Reporting

SL Finance Data & Reporting Manager is ROLLIER, Charlotte

Usage information

More than 500 users have access to P&L applications. Those users are worldwide and most of them are controllers.

History

P&L was done in 2015 and IM in 2016. Solstice split was done in 2019.

Technical Rules on Workbench

Functional rules

COPA flows

- COPA module is the ancestor of BW
- COPA is divided in “tight” perimeters (in customizing, in VV list, for certain characteristics, for BW data sources) : Operating concern
- The lines of the ERP data (COPA value fields from WP1 system) must be translated in BFC account. This translation should be dynamic.

Value Fields

- Value Fields are the lowest key figures in COPA module (For BCS sourcing)
- Value Fields are the central elements of the BFC account determination. Since CROCO project, the determination of the BFC account depends of the value field and also the OECD activity for PF1 COPA flow.
- P&L analysis is based on BFC Account with a detail by Value Field. For now, it is not useful to have an analysis directly by value field

Rhodia legacy

- Historic data have been recalcuted with the new allocation Value Field / BFC Account
- Historic views are avaible in BFC (not restated)

The organization determination must be as flexible and as dynamic as possible. W e had to consider 2 cases:

- CASE A : COPA line items with Material and Distribution Channel valid
- CASE B : COPA line items with Material empty and/or Distribution Channel empty

Solvay legacy data (PQ1 based solution)

- Existing application in PQ1 with a lot of intelligence
- Easier to load from PQ1 than redoing what has already been done - it was a temporary solution and it is replaced with SPS project (P&L with COPA in PF1, go live was on January 2018)

BFC Data

- Loaded by flat files
- 3 levels:
    - Version 1: Total P&L
    - Version 2: Interco Sales
    - Version 3 : Third Party

Another DSO for data without version

08/2023 : PO2 Project

BFC are data are now feeded from 2 BFC application (BFC ECO - the existing one and BFC SCO)

BFC SCO will send data from 09/2023

We still have 3 levels at cube level

- Version 1: Total P&L (CRCOPA03)
- Version 2: Interco Sales (CRCOPA04)
- Version 3 : Third Party (CRCOPA05)

But at DSO level

- ECO:
    - Version 1: Total P&L (DBCOPA04)
    - Version 2: Interco Sales (DBCOPA05)
    - Version 3 : Third Party (DBCOPA06)
- SCO:
    - Version 1: Total P&L (ABCOPA11)
    - Version 2: Interco Sales (ABCOPA10)
    - Version 3 : Third Party (ABCOPA12)
- Inside the cube we segregate the information of the three cubes from BFC ECO and BFC SCO identifiable thanks to C_INFOPRO

ECO	SCO
[C_INFOPRO] InfoProvider [DBCOPA06] BFC - P&L - Version 3: Third Party [DBCOPA05] BFC - P&L - Version 2: Interco Sales [DBCOPA04] BFC - P&L - Version 1: Total P&L	[C_INFOPRO] InfoProvi [ABCOPA11] BFC - P [ABCOPA10] BFC - P [ABCOPA12] BFC - P

BFC data are used in 1 Core query : BW_BW_QRY_MVCOPA01_0003

For PO2 project the content of BFC amounts (BFC ECO or BFC SCO) are depending

The conso view (0, 1, 2 or 3)

The Date of analysis (before/After BFC SCO deployment ; before/After Spin Off)

We will receive 2 BFC files (some companies will be in Both) with YTD Legal and YTD Conso

TO BE							
	BFC SOLVAY (BFC ECO)				BFC SCO		
	YTD LEGAL	YTD CONSO	M LEGAL		YTD LEGAL	YTD CONSO	M LEGAL

If the month of analysis is before BFC SCO deployment (09/2023)

	BFC comparaIson	**BEFORE 09/2023 (<)**		
P&L	Legal	Conso SOLVAY	Conso ECO	Conso SCO
Conso view	0	1	2	3
Scope	ECO + SCO or ECO or SCO	ECO + SCO or ECO or SCO	ECO + SCO or ECO or SCO	ECO + SCO or ECO or SCO
	YTD LEGAL for ALL Companies	YTD CONSO for ALL Companies	YTD CONSO for ECO Companies	YTD CONSO for SCO Companies

If the month of analysis is after BFC SCO deployment (09/2023) BUT before SPIN OFF

	BFC comparaison	**AFTER 09/2023 (incl)**		
P&L	Legal	Conso SOLVAY	Conso ECO	Conso SCO
Conso view	0	1	2	3
Scope	ECO + SCO or ECO or SCO	ECO + SCO or ECO or SCO	ECO + SCO or ECO or SCO	ECO + SCO or ECO or SCO
	YTD LEGAL	YTD CONSO for ALL Companies	YTD CONSO for ECO Companies	
				YTD CONSO for SCO Companies

If the month of analysis is after SPIN OFF

	BFC comparaison	**AFTER spin off (>=2024)**		
P&L	Legal	Conso SOLVAY	Conso ECO	Conso SCO
Conso view	0	1	2	3
Scope	ECO + SCO or ECO or SCO	ECO + SCO or ECO or SCO	ECO + SCO or ECO or SCO	ECO + SCO or ECO or SCO
	YTD LEGAL	0	YTD CONSO for ECO Companies	
				YTD CONSO for SCO Companies

<--- 08/2023 : PO2 Project

CICC

- Loaded by BFC flat files
- Only for CICC companies (companies in c_compprs MD with landscape = "CICC") or Adjustment companies maintained in c_glbfilt (PL_BFC_ADJ /COMPANY)

Below Gross Margin

- Loaded by BFC flat files, only for companies loaded in Non ERP for the concerned month (companies can move from Non ERP to ERP) when Below Gross Margin is not uploaded by flat file (= only Non ERP system for which a non full flat file is loaded).

Non ERP companies

- Some companies are in Solvay group but don't use RCS or Solvay ERP : companies belonging to NOVECARE or SODA GBU

Systems involved

- Rhodia ERP WP1
- Solvay BW PQ1 (PQ1 is the view of the P&L for PF1 data)

- BFC (flat files)
- Flat files for Non ERP

Organisation

- BFC GBU
- BFC Group of Activities
- BFC Activity

Technical rules

P&L WP1

We have 2 steps in the business layer.

From the propagation layer, we transpose data from DPCOPA01 to DBCOPA01.

In DPCOPA01, each value field is in a dedicated key figures. Objective of the transposition, we have 1 key figure and 1 field in key (called Value Field) to identify them.

After the transposition, we split data in 2 target DSO: DBCOPA02 and ABCOPA03.
As we have common rules for both DSO, we use an Info source to have those rules inside an unique transformation.

Transformations on DBCOPA02 DBCOPA02 and ABCOPA03 ABCOPA03 are used to determine the flag intra gbu (C_INTGBUF). It compares the GBU (0g_cwwe01__cpfctr1_2 for DBCOPA02 and c_tecmat for ABCOPA03) and the partner GBU (c_tecpara__cpfctr1_2). It was not possible to use a technical Master Data to fill this requirement as the needed key (the keys that would permit to determine the combination GBU and partner GBU) would have been too long. These loadings are done only once a week.

BFC heading determination: the BFC heading is determined in master data C_FIELDN using DSO dpcopaa10 (data from WP1 table ZWFAT110).

Solstice (WP1)

An additional split is done from ABCOPA03 to 2 ADSO: ABCOPA01 and ABCOPA04. ABCOPA01 contains Solstice data and ABCOPA04 contains non-Solstice data. Split is done on a list of Company Code in the DTPs.

Delta load to ABCOPA01 determines the Specification List, Main Specification, Program, Market and Program Percentage for each new entry, using lookups in Master Data C_ITM_SD (Main Specification and Specification List) and ADSO ABDPDY01 (Program, Market, Program Percentage).

Delta load to ABCOPA01 cannot be determined solely from new data in ABCOPA03, as we must take into account the updates of the added fields Main Specification, Program, Market, Program Percentage. This requires to know if these values have been changed in the lookup sources. This issue is solved by the field 0RSPL_TSTMP (Time Stamp of Last Change) in ABCOPA03, that is updated by 3 different Delta loads:

- Delta from ABDPDY01: If Program, Market or Program Percentage have been modified/created.
- Delta from ABDPDY02 & from ADPSDAE01: Both ADSOs are used to feed Master Data C_ITM_SD, if Main Specification or Specification List are created/changed.

P&L PF1

We have 2 steps in the business layer.

From the propagation layer, we transpose data from DPCOPA15 to DBCOPA29.

In DPCOPA15, each value field is in a dedicated key figures. Objective of the transposition, we have 1 key figure and 1 field in key (called Value Field) to identify them.

After the transposition, we use a second level of Business layer for other business rules: DBCOPA29 to DBCOPA30

- we apply the sign of the key figures according to DSO DPCOPA14
- we calculate document currency amounts for the key figures maintained under c_glbfilt (stream = “C_FIELDN”, rule = “AMOUNTDC”): to calculate the document currency amounts, we use the rate stored in the document (rate type "M"), however, it doesn't take into account the conversion factors (TCURF). In order to retrieve these factors to translate correctly the amounts, we use the **remote** Function Module " ZFM_BW_EXCHANGE_RATE_RFC" that calls in PF1, for all concerned currencies, the function module 'READ_EXCHANGE_RATE'. Note that an alternative solution was to load in WBP 'READ_EXCHANGE_RATE' results from PF1 for all possible combinations, however, it took a lot of time (the datasource created for this is DTS_EXCHANGE_RATE).
  We retrieve the conversion factors for the document creation date => theorically, the TCURF factors could change during a day, so the factors at the beginning and at the end of the day could be different. If this occurs, a workaround in order to find the correct factors for a document could be to get the rate using 'READ_EXCHANGE_RATE' for the creation date and also for the creation date -1 and to use the factors for the day when the rate is the nearer to the document rate.

- Maintained below Global filter code to rectify the issue with the currency conversions (INR - JPY,SGD-JPY,CNY-JPY , THB-JPY & KRW - JPY) for which the amount in company code currency value is wrong in SAP .Due to this issue , Amount in document currency keyfigures K_AMNTDC & K_AMNTNDC are showing wrong values in BW.

```
623  *----Get Global filter data for DBCOPA29 - PF1 TCURF Exchange rate type
624   *M data is maintained.
625        clear ls_tcurf_m.
626        clear LS_C_RULE.
627        concatenate <result_fields>-doc_currcy'-'
628                    <result_fields>-currency'-'
629                    'M' into LS_C_RULE.
630        select single * from /BIC/PC_GLBFILT
631               into ls_tcurf_m
632               where /BIC/C_STREAM = 'DBCOPA29' and
633                     /BIC/C_RULE = LS_C_RULE and
634                     /BIC/C_GLBFILT = '001' and
635                     OBJVERS = 'A'.
636        if sy-subrc NE 0.
637           ls_tcurf_m-/bic/c_low = 1.
638           ls_tcurf_m-/bic/c_high = 1.
639        endif.
640    "applying PF1 TCURF factor from global filter
641     w_amount = w_amount * ( ls_tcurf_m-/bic/c_low /
642               ls_tcurf_m-/bic/c_high ).
643
```

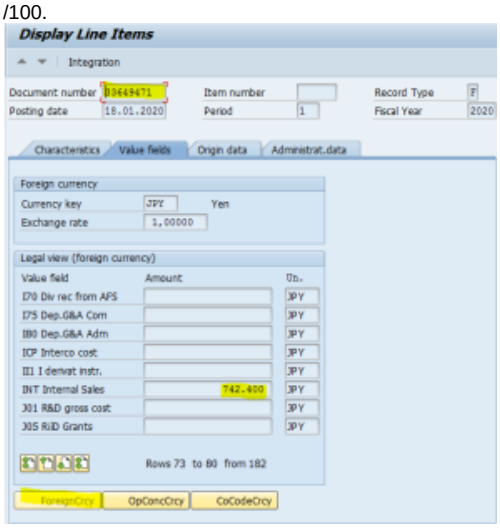
Global filter data for Stream DBCOPA29 :

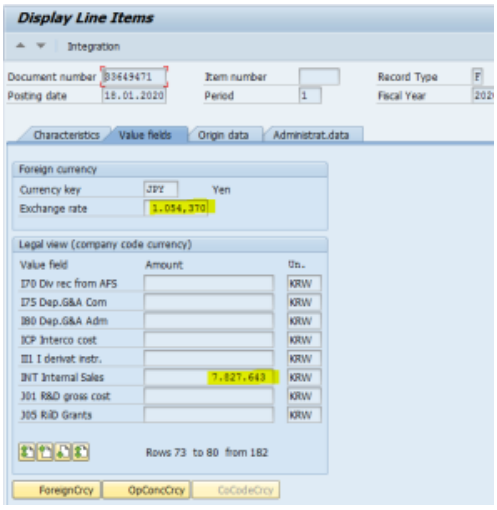
Data Browser: Table /BIC/PC_GLBFILT Select Entries 5

/BIC/C_STREAM	/BIC/C_RULE	/BIC/C_GLBFILT	OBJVERS	CHANGED	/BIC/C_DESC	/BIC/C_SIGN	/BIC/C_OPTION	/BIC/C_LOW	/BIC/C_HIGH	/BIC/C_ACTIVE
DBCOPA29	JPY-CNY-M	001	A			I	EQ	100	1	Y
DBCOPA29	JPY-INR-M	001	A			I	EQ	100	1	Y
DBCOPA29	JPY-KRW-M	001	A			I	EQ	100	1	Y
DBCOPA29	JPY-SGD-M	001	A			I	EQ	100	1	Y
DBCOPA29	JPY-THB-M	001	A			I	EQ	100	1	Y

Example :

Company Code Currency : KRW
Document Currency : JPY
COPA Document : 33649471
Amount in Document Currency = 742.400 ,Exchange Rate : 1.054,370
Amount in Comp currency should be 742.400 * 1.054,370 = 782.764288
But since we have conversion factor 100 maintained in TCURF table for JPY to KRW , in KE24 it's showing as 7.827.643 which is 782.764288 /100.





Requirement is to convert amount in Currency to Doc currency – KRW to JPY using the exchange rate and store the converted amount in K_AMNTDC (Doc Currency) in the target DSO DBCOPA30

But since the Exchange rate function module ZFM_BW_EXCHANGE_RATE_RFC is not considering the factors properly for such kind of cases ((INR - JPY,SGD-JPY,CNY-JPY , THB-JPY & KRW - JPY) ,we are maintaining these cases in Global filter and multiplying the amount with 100 so that the Amount in document currency value matches with KE24 value .

Data Browser: Table /BIC/ADBCOPA3000 Select Entries 484

/BIC/C_PABEL	/BIC/C_PAPOSNR	CO_AREA	/BIC/C_VERSN2	CURTYPE	VTY	/BIC/C_FIELDN	RECORDMODE	FISCYEAR	FISCPER	AMOUNT	/BIC/K_AMNTDC	/BIC/K_AMNTNDC
0033649471		CHEF	000	10	010	FERTF		2020	2020001	2,015,940-	0	0.000
0033649471		CHEF	000	10	010	FERTP		2020	2020001	2,910,785-	0	0.000
0033649471		CHEF	000	10	010	QUANTITY		2020	2020001	0	0	0.000
0033649471		CHEF	000	10	010	VVF00		2020	2020001	1,006,614-	0	0.000
0033649471		CHEF	000	10	010	VVINT		2020	2020001	7,827,643	742,400	742,400.000

Handled all the currency scenarios as below in End Routine .

Step 1 : Get data from TCURX for CompanyCode Currency ,If currency found update as '0' else 1

```
565          select single * from tcurx
566                 into ls_tcurx
567                 where currkey = <result_fields>-currency.
568          if sy-subrc eq 0 .
569            "If currency found, set 0 or 1 as per TCURX
570             ls_currbool-compcurr = 0.
571
572          else.               "If currency NOT found, set as 1
573            ls_currbool-compcurr = 1.
574          endif.
575          clear ls_tcurx.
```

Step 2 :Get data from TCURX for Document Currency ,If currency found update as '0' else 1

```
578          select single * from tcurx
579                 into ls_tcurx
580                 where currkey = <result_fields>-doc_currcy.
581          if sy-subrc eq 0 .
582            "If currency found, set 0 or 1 as per TCURX
583             ls_currbool-doccurr = 0.
584          else.                "If currency NOT found, set as 1
585            ls_currbool-doccurr = 1.
586          endif.
587          clear ls_tcurx.
```

Step 3 :I dentify HIGH and LOW using TCURX Info

*---1:HIGH indicates there are decimals in the currency like:EUR,USD
*---0:LOW indicates there are No decimals in the currency like:JPY,ITL,KRW

```
591        IF NOT <result_fields>-currency = <result_fields>-doc_currcy.
592     case ls_currbool.
593            when '00'.  "LOW CCcurr ====>LOW DocCurr
594   "LOW CCcurr ====>HIGH DocCurr | Only external conversion of CCcurr
595              call function 'BAPI_CURRENCY_CONV_TO_EXTERNAL'
596                exporting
597                  currency        = <result_fields>-currency
598                  amount_internal = w_amount
599                importing
600                  amount_external = w_amount.
601            when '11'.  "HIGH CCcurr ====>HIGH DocCurr
602   *          Do Nothing
603            when '01'.
604   "LOW CCcurr ====>HIGH DocCurr | Only external conversion of CCcurr
605              call function 'BAPI_CURRENCY_CONV_TO_EXTERNAL'
606                exporting
607                  currency        = <result_fields>-currency
608                  amount_internal = w_amount
609                importing
610                  amount_external = w_amount.
611
612            when '10'.
613   *"HIGH CCcurr ====>LOW DocCurr | Only internal conversion input DocCurr
614              call function 'BAPI_CURRENCY_CONV_TO_INTERNAL'
615                exporting
616                  currency             = <result_fields>-doc_currcy
617                  amount_external      = w_amount
618                  max_number_of_digits = 22
619                importing
620                  amount_internal      = w_amount.
621
622          endcase.
```

Above cases details are given below

Cases	Currency in TCURX	Document currency in TCURX	Action in the result conversion	Comment
00	Yes	Yes	External conversion of CCcurr	Currency <> Document Currency
11	No	No	No Action	Currency <> Document Currency

01	Yes	No	External conversion of CCcurr	Currency <> Document Currency
10	No	Yes	Internal conversion of DocCcurr	Currency <> Document Currency

Step 4 : Maintained below Global filter code to rectify the issue with the currency conversions (INR - JPY,SGD-JPY,CNY-JPY , THB-JPY & KRW - JPY) for which the amount in company code currency value is wrong in SAP .Due to this issue ,Keyfigures Amount in document currencies K_AMNTDC & K_AMNTNDC are showing wrong values in BW.

```
*----Get Global filter data for DBCOPA29 - PF1 TCURF Exchange rate type
*M data is maintained.
     clear ls_tcurf_m.
     clear LS_C_RULE.
     concatenate <result_fields>-doc_currcy'-'
                 <result_fields>-currency'-'
                 'M' into LS_C_RULE.
     select single * from /BIC/PC_GLBFILT
            into ls_tcurf_m
            where /BIC/C_STREAM = 'DBCOPA29' and
                  /BIC/C_RULE = LS_C_RULE and
                  /BIC/C_GLBFILT = '001' and
                  OBJVERS = 'A'.
     if sy-subrc NE 0.
        ls_tcurf_m-/bic/c_low = 1.
        ls_tcurf_m-/bic/c_high = 1.
     endif.
 "applying PF1 TCURF factor from global filter
  w_amount = w_amount * ( ls_tcurf_m-/bic/c_low /
            ls_tcurf_m-/bic/c_high ).
```

Step 5 : APPLY CONVERSION RATE TO CALCULATE AMOUNT IN DC

```
 * APPLY CONVERSION RATE TO CALCULATE AMOUNT IN DC
           IF <result_fields>-gt_tkrate <= 0.
             <result_fields>-/bic/k_amntdc = w_amount * abs(
             <result_fields>-gt_tkrate ).
           ELSEIF <result_fields>-gt_tkrate > 0.
             <result_fields>-/bic/k_amntdc = w_amount / abs(
             <result_fields>-gt_tkrate ).
           ENDIF.
```

Step 6 : Apply the correct factors using exchange rate function module .

```
          READ TABLE itb_h_exchange_rate ASSIGNING <fs_exchge_rate>
            WITH TABLE KEY
                 zdate         = <result_fields>-createdon
                 zforeign_curr = <result_fields>-currency
                 zlocal_curr   = <result_fields>-doc_currcy
                 zrate_type    = c_rate_type.

*If no result => ratio 1:1
          IF sy-subrc = 0.

            w_amount = <result_fields>-/bic/k_amntdc.
            <result_fields>-/bic/k_ffact =
              <fs_exchge_rate>-zforeign_factor.
            <result_fields>-/bic/k_tfact =
              <fs_exchge_rate>-zlocal_factor.

            w_amount = w_amount *
                       <result_fields>-/bic/k_tfact /
                       <result_fields>-/bic/k_ffact.

            <result_fields>-/bic/k_amntdc = w_amount.

          ENDIF.
     ELSE.
```

Step 7 : Apply external format to the amount without dimension K_AMNTNDC

```
          CALL FUNCTION 'BAPI_CURRENCY_CONV_TO_EXTERNAL'
            EXPORTING
              currency        = <result_fields>-doc_currcy
              amount_internal = w_amount
            IMPORTING
              amount_external = w_amount.

          <result_fields>-/bic/k_amntndc = w_amount.

        ENDIF.

      ENDLOOP.
```

BFC heading determination:

- in IM flow (ZZCONACT), the BFC heading is determined in master data C_FIELDN (=value field) using DSO dpcopa14 (data from PF1 table ZZF_BFC_COPA_VF) for the OECD activity = #.
- in COPA flow, the BFC heading is determined in master data C_FIELDN2 (=value field and OECD activity) using DSO dpcopa14 (data from PF1 table ZZF_BFC_COPA_VF) depending of the OECD activity (if no correspondance on the activity, take by default the BFC heading for activity = #). The BFC heading attribute of the MD c_plelmn2 does not take into account the OECD activity, it is available in the report as the "Initial Heading".

WBS element determination:

The standard WBS element from COPA, for CROCO billing types, is always empty. In this case, the WBSE is retrieved in PF1 table Z1F_CRC_INV (Z1F_CRC_INV-PSPNR), during COPA extraction, searching on the Sales order item (VGBEL/VGPOS). It is not possible to found several WBSE for the same sales order item. This enhancement is limited to record type "F" and to CROCO billing types maintained in TVARVC parameter ZBW_FKART_CRC.

Z1F_CRC_INV-PSPNR (8 characters) must be convert into external format (POSID -> 24 characters) for BW (key of c_wbs_el2), we use the function "SBW_PRPS8_TO_PRPS24" for mass translation.

The following technical MD are used in these flows:

-c_tecpct2 (reference c_tecpctr) - Technical Profit Center:
This object is used to determine dynamically the organisational structure (activity1, GBU...).
It follows the business rules in COPA, these rules have changed after SPS project and the old model technical MD, c_techba, is now used for historical data only (before 2018).
cf. part "8.7 Organisation structure" of the specification document: https://drive.google.com/file/d/19tbW1LIskpgOQHslJAH1Aby85wZYjmRsKc6BBtE5SVM/view

The entries in this MD are generated only by the SID generation in transactional flows then transformation from the MD to itself determines the attributes.

-c_techpac - Technical Part Activity:
This object is used to determine dynamically the partner organisational structure.
It follows the business rules in COPA, these rules have changed after SPS project and the old model technical MD, c_techpba, is now used for historical data only (before 2018).
cf. part "8.8 Partner Organisation structure" of the specification document: https://drive.google.com/file/d/19tbW1LIskpgOQHslJAH1Aby85wZYjmRsKc6BBtE5SVM/view

The entries in this MD are generated only by the SID generation in transactional flows then transformation from the MD to itself determines the attributes.

The MD is compounded with the Partner activity. This field is used for the partner organisation structure determination in the COPA flow, not in IM flow, and it allows to keep the Partner activity determinated in the document.

-c_intgbu7 (reference c_intgbu6) - Intra GBU (Solvay COPA):
This object is used to determine dynamically the intra GBU flag, reading the GBU and partner GBU in c_tecpctr and c_techpac.

The entries in this MD are generated only by the SID generation in transactional flows then transformation from the MD to itself determines the attributes.

The MD is compounded with the Partner activity. This field is used for the partner organisation structure determination in the COPA flow, not in IM flow, and it allows to keep the Partner activity determinated in the document.

-c_tgbr5:
This object is used to determine dynamically the GBR axis.
cf. specifications: https://drive.google.com/file/d/1AL5UzU3oIFqyzMAwU9Dv6Ko6SNWQcF6Z992nB5WxhAU/view

The entries in this MD are generated only by the SID generation in transactional flows then transformation from the MD to itself determines the attributes.

BFC

08/2023 : PO2 Project changes

BFC Data comes from extraction sent by Marie-Yolande KUCZYNSKI

- Every day between 3rd and 20th day + 25th day of the month at 3:30AM (FR)
- Format: txt
- Directory: /BW/exploit/PandL
- Loaded files:
  - BFC_vs_BW_PandL.txt
  - **BFC_vs_BW_PandL_SCO.txt**

The file is loaded by Process chain BFC: Global Process Chain (*PC_COPA_PL_12*). This one is a subchain called in metachains:

- COPA: PL Global Process Chain (*PC_COPA_PL_GLOBAL*), called in Daily Process chain (RSP_DAILY). So it is executed everyday (except saturday and sunday), even during closing period, between 1H45 and 2H30.

- COPA: PL - 2 extra loadings during the closing period - P&L (*PC_COPA_PL_22*). Chain *PC_COPA_PL_12* is at the end of the process chain. It is executed only during closing months, between 2nd and 5th closing day. In fact it depends on value of global filters:
  - Rule RUN_CAL_EX, Stream PL_COPA, Active = 'Y'. Calendar to use (=50 for Solvay working days)
  - Rule RUN_PER_EX, Stream PL_COPA, Active = 'Y'. Days on which process chain is executed. Now between 2 and 5.
  - Rule CHECK_LOAD, Stream PL_ELEMENT, Low = PL, Active = 'Y'. Hours on which proces chain should be executed (in the period indicated by the 2 variables above). Now: 08H ,13H ,15H, 20H.

  Notice that process chain is scheduled for every hours, (except saturday and sunday). But execution of loadings is done with enhanced decision tree, accoriding to those variable values.

The file loaded by cancel and replace (using recordmode to delete active records) to propagation layer. For this part, we don't respect LSA methodology.

All business rules calculated between propa DSO DPCOPA02 and Info source

- Easier for maintenance
- Very simple business rules
- Only exclusion of data between info source and business layer

For business and reporting layers, we load by Delete + Full.

From the BFC File **(BFC ECO or BFC SCO)** , we also load 2 other targets :

- CICC system and adjustment companies
- Companies not working on ERP : only a part of P&L

Those targets are in business layer.

(BFC ECO or BFC SCO) we mange in C_GLBFILT (Stream PL_BFC_ADJ) the authorization scope allowed by Origin DSO

For instance Cube CRCOPA08

Table: /BIC/PC_GLBFILT

/BIC/C_STREA	/BIC/C_RULE	/BIC/..	OBJVERS	Cl	/BIC/C_DESC	/BIC/C_SIGN	/BIC/C_OPTION	/BIC/C_LOW	/BIC/C_HIGH	/BIC/C_ACTIVE
PL_BFC_A..	ABCOPA14	001	A		Authorize Login Scope in the Cube CICC CRCOPA08	I	EQ	ECO		N
PL_BFC_ADJ	ABCOPA14	002	A		Authorize Login Scope in the Cube CICC CRCOPA08	I	EQ	SCO		Y
PL_BFC_ADJ	DBCOPA12	001	A		Authorize Login Scope in the Cube CICC CRCOPA08	I	EQ	ECO		Y
PL_BFC_ADJ	DBCOPA12	002	A		Authorize Login Scope in the Cube CICC CRCOPA08	I	EQ	SCO		N

Only Company with authorization scope ECO are allowed from DSO DBCOPA12

Only Company with authorization scope SCO are allowed from DSO ABCOP14

For instance Cube CRCOPA07

Table: /BIC/PC_GLBFILT

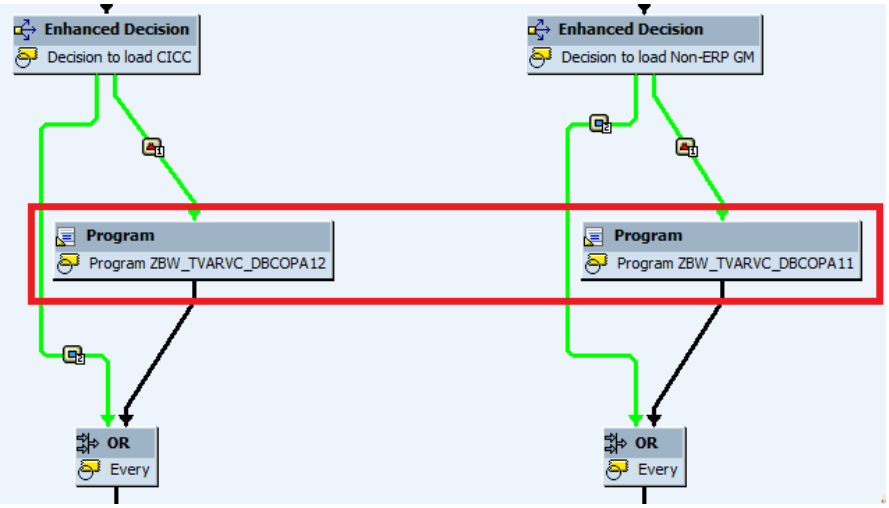
/BIC/C_STRE..	/BIC/C_RULE	/BIC..	O..	Cl	/BIC/C_DESC	/BI..	/BIC/C_OPTIO..	/BIC/C_LOW	/BIC/C_HIGH	/BIC/C_ACTIVE
PL_BFC_ADJ	ABCOPA13	001	A		Authorize Login Scope in the Cube NON ERP CRCOPA07	I	EQ	ECO		N
PL_BFC_ADJ	ABCOPA13	002	A		Authorize Login Scope in the Cube NON ERP CRCOPA07	I	EQ	SCO		N
PL_BFC_ADJ	DBCOPA11	001	A		Authorize Login Scope in the Cube NON ERP CRCOPA07	I	EQ	ECO		Y
PL_BFC_ADJ	DBCOPA11	002	A		Authorize Login Scope in the Cube NON ERP CRCOPA07	I	EQ	SCO		Y

Company with authorization scope ECO and SCO are coming from DBCOPA11

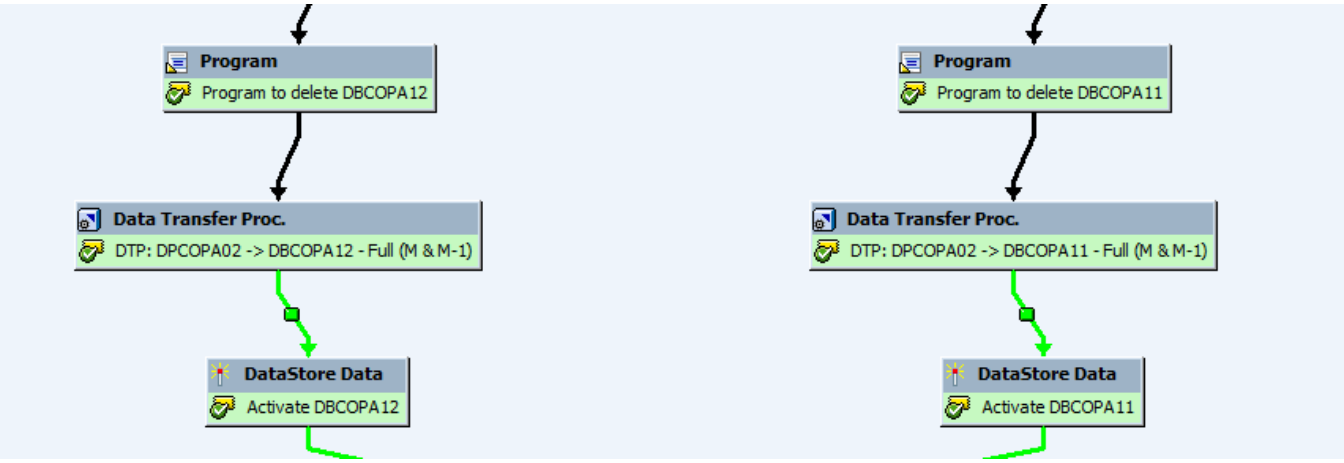
Those parameters will allow to manage the period BEFORE BFC SCO ; Between BFC SCO -> SPIN OFF and after SPIN OFF

We load them by Delete + Full for current and previous months :

determination in process chains to update TVARVC table



We do selective deletion using programs and loading with selections in DTP



The time reference could be changed manually. There is a written procedure for that

The reporting layer is loaded by deletion and full loading.

Adjustment companies and CICC companies shared the same flow. The source system has to be determined:

- CICC companies : companies in c_compprs with landscape = "CICC" and conso method = "10" and "30" => source system is PI1_020
- Adjustment companies: PRS company code maintained in c_gblfilt (stream = "PL_BFC_ADJ", rule = "COMPANY") => the source system is obtained from c_compprs MD, the read of the MD is restricted to the landscape listed in c_glbfilt (stream = "PL_BFC_ADJ", rule = "LANDSCAPE").

This flow is restricted to the magnitude nature 0LIA01 for CICC and 1RET90 (maintained in c_glbfilt PL_BFC_ADJ/NATMAG) for Adjustment companies.

Non-ERP

We needed a solution for companies not working on ERP. We have created a specific transaction : **ZPL_FILE**

The transaction is based on program : **ZBW_PL_PC**

Documentation about this transaction :

https://docs.google.com/document/d/1gF426ok7VvnF1-idij_FCsMt7HMYQuY4LiN34CseSZs/edit

With this transaction, user can load excel file for

- Customer master data
- Material master data
- Non-ERP transactional data

This program will save the file from user upload to format filename + upload time + user ID in folder /exploit/BW/PandL

and copy to the constant files basing on user selection what the data is updated to

- customer_master_data.csv
- material_master_data.csv
- non-erp_data.csv

Example of user manual to upload non-ERP file

Dataflow :

https://docs.google.com/presentation/d/1fmDajNiJ7SF12m8K_ytJgScvxoaMmcEnUCVP7-S1d8w/edit

Process chain:

- PC_COPA_PL_06
- PC_COPA_PL_07
- PC_COPA_PL_08

The transaction available in SOLIA portal. For information, it's better to use IE and not Chrome

We have created a new authorization object (C_SRSYST) to be sure that users can only load data for companies they have access to.

We test user's authorizations with **DPAUTH01** DSO

For master data, we can display content of master data in the screen and also save it in an excel file

For master data and transactional data, user can load excel file.

We control content of the file to be sure that there is no error (missing field or incorrect value)

Once data is loaded:

- Save file in csv in /BW/exploit/BW/PandL
- Execute process chain
  - By event
  - Load file to PSA
  - Manage "delta queue" is several loadings at the same time
- Another process chains to load from PSA to BW Info provider
  - 1 for Master data
  - 1 for DSO and cubes
  - Process chain runs 3 times a day (managed in Global filter master data)

Link to the functional document:

Link to the files description:

Profit Center check:

As per the business requirement, program do lot of validations before loading the data. As part of this, For few source systems, there should not be profit center check. Where as for few other source systems, there must be a check for profit center (it should check against the profit center master data). The new source systems can be added into that list.

To have this functionality, program is modified and introduced global filter.

Stream	Rule	Counter	Global Filter Descri	Sign	Option	Low	Hight	Active
ZBW_PL_PC	NERP	1	Non ERP Syststems	I	EQ	NERP001		Y
ZBW_PL_PC	NERP	2	Non ERP Syststems	I	EQ	NERP002		N
ZBW_PL_PC	NERP	3	Non ERP Syststems	I	EQ	NERP003		N
ZBW_PL_PC	NERP	4	Non ERP Syststems	I	EQ	NERP004		N
ZBW_PL_PC	NERP	5	Non ERP Syststems	I	EQ	NERP005		N
ZBW_PL_PC	NERP	6	Non ERP Syststems	I	EQ	NERPCYT		Y

NERP system with Y No profit center

NERP system with N Profit center is mandatory & it checks the profit center against master data

In future, based on the business decision, by controlling the active parameter to Y/N for each system, we can handle whether profit center check is required or not.

This change is handled as part of UC#4000005270

Currency check:

The currency is retrieved from c_compprs where c_compprs__company = c_company in the file during the integration by the transaction ZPL_FILE. An error is raised if the currency can't be determined.

Full flat file check:

Using c_glbfilt parameter (steam = "FULLFF", rule = "NERP"), we control the list of source systems for which a full flat file is required:

Stream	Rule	Counter	Global Filter Descri	Sign	Option	Low	Hight	Active
FULLFF	NERP	1	NON ERP SOURCE SYSTEM FULL FLAT FILE	I	EQ	NERPCYT		Y

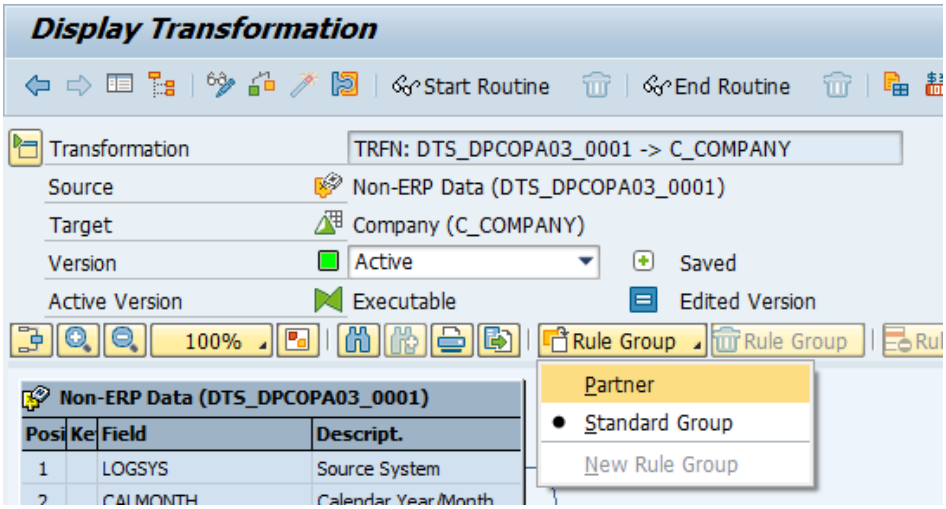
When a full flat file is expected, that means below gross margin part should not comes from BFC but from the file uploaded using ZPL_FILE. The key figures only in full flat file are maintained in the structure ZNERP_FF_KF and a warning is raised if none of those key figures is filled. It is forbidden to fill one of those key figures if the source system is not in the c_glbfilt list.

Full flat file information is kept under the propagation DSO (DPCOPA03) using infoObject c_fullff = "X".

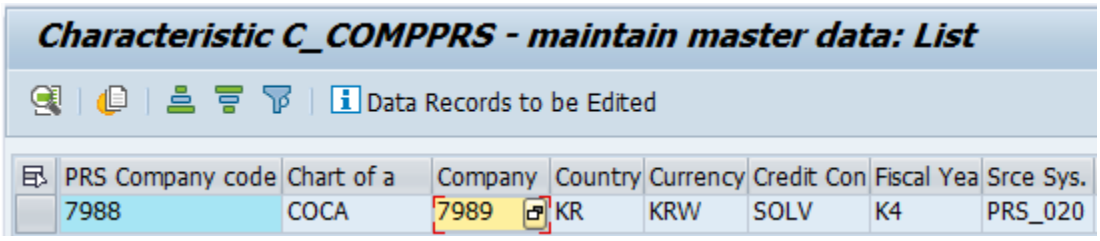
MD Company update:

The MD company is updated for Non ERP source system from the company and the trading partner in the flat file.

All attributes are retrieved from MD PRS company (c_compprs). The rules are the same for the company and the trading partner, a rule group "Partner" was created to process them in the same transformation:



Problem with old BFC company code: If the BFC company in the file has no correspondance in c_compprs, it is certainly because the file has the old BFC company code => we must use the current BFC company code even for historical data upload. For instance, BFC company code 7989 must be used instead of 7988:

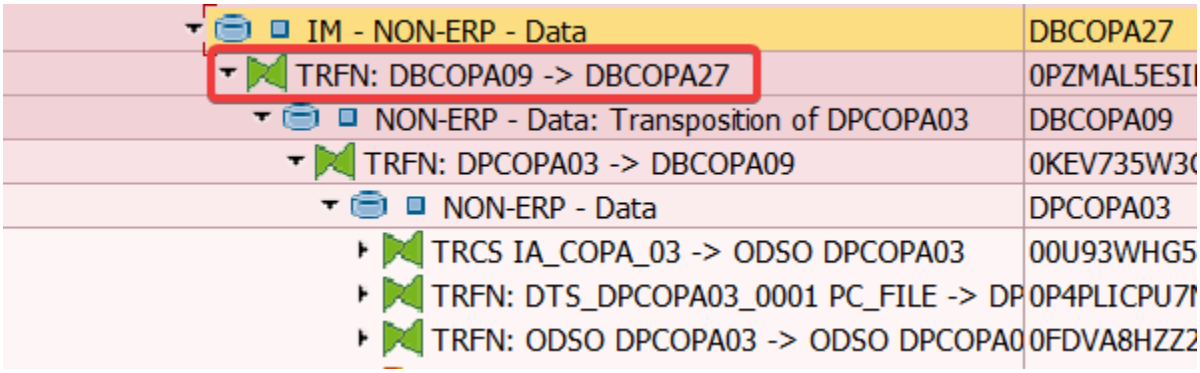


IM valuation for Non ERP entries in P&L flows :

The goal is to valuate IM Cost for NON ERP flow using WP1 Material IM Cost unit (for a restricted list of companies); if no correspondence can be found, current business rules is applied (IM Costs = Standard Costs)

Link to the functional document

BW implementation :



In "start routine" of the transformation TRFN: DBCOPA09 -> DBCOPA27, for a company list defined in the master data C_GLBFILT (See after bellow).

We read the DSO of the integration margin ODS_PCP4 with the specific key.

In case we find a matching entry with the requested KPI different from 0, we create a new line by adjusting the amount and the Magnitu account as follows:

- C_MGN_ACC = R1540C / 0AMOUNT = Qty VV04 * Integrated variable Cost Unit (C_AMT_CP) ***-1/ Lot Size**
- C_MGN_ACC= R1290C / 0AMOUNT = Qty VV04* Integrated Duty costs (K_INTDUT) ***-1/ Lot Size**
- C_MGN_ACC = R1291C / 0AMOUNT = Qty VV04* KeyFigure Integrated Transport Cost K_INTFRG ***-1/ Lot Size**
- C_MGN_ACC = R2549C / 0AMOUNT = Qty VV04* Integrated CNP part (C_MVR_CNP) *-1 **/ Lot Size**

```
* APPEND FOR EACH RULES

          IF <fs_pcp4>-/bic/c_amt_cp IS NOT INITIAL AND
          <fs_pcp4>-/bic/c_amt_cp <> 0              AND
          <fs_pcp4>-/bic/c_amt_cp <> '0.00'         AND
          <fs_pcp4>-/bic/k_lotsize <> 0.
            wa_source_package-/bic/C_MGN_ACC  = 'R1540C'.
            wa_source_package-amount          = -1 *
             ( wa_source_package-g_qvva01 * w_c_amt_cp ) / w_lotsize.
            W_FLAG = 1.
            COLLECT wa_source_package INTO it_source_package.
          ENDIF.

          IF <fs_pcp4>-/bic/k_intdut IS NOT INITIAL AND
          <fs_pcp4>-/bic/k_intdut <> 0              AND
          <fs_pcp4>-/bic/k_intdut <> '0.00'         AND
          <fs_pcp4>-/bic/k_lotsize <> 0.
            wa_source_package-/bic/C_MGN_ACC  = 'R1290C'.
            wa_source_package-amount          = -1 *
             ( wa_source_package-g_qvva01 * w_k_intdut ) / w_lotsize.
            W_FLAG = 1.
            COLLECT wa_source_package INTO it_source_package.
          ENDIF.
```

Unit and currency Conversion :

In case the above calculation is applied be careful to standardize the units and currency between the two sources via the updated function module :

- Z_MD_CONVERT_MATERIAL_UNIT3
- Z_CONVERT_CURRENCY

Master Data C_GLBFILT - Global Filter :

The list of companies can be changed with the Master Data global filter with the following parameters :

Global Filter Stream (Application)	"Global Filter Rule"	"Global Filter"	Global Filter Active	Global Filter Description	Global Filter Hight Value	Global Filter Low Value	Global Filter Option	Global Filter Sign
CO_NONERP	COMPCDE	1	Y	Company Code		*07735*	EQ	E

PQ1

PQ1 flows are kept only for historical data, they are replaced by PF1 flows since 01.2018!

Glossary

PCA: Profit Center Accounting
GM: Gross-Margin

Presentation

We needed a solution to get the P&L data for Solvay legacy.

- part of data are in GM Flow at detailed level loaded from a ZZ program
- rest of data are in PCA flow at aggregated level
- data are merged at multiprovider level, excluding duplicate data at business layer level

Specific rules

We synchronize process chain between PQ1 and WBP systems:

- WBP PCA **PC_COPA_PL_17** is executed from PQ1 **ZZKPCA_PL_TR** and **ZZKPCA_PL_TR_2**
- WBP GM **PC_COPA_PL_23** is executed from PQ1 **ZZCONEXIA_04**
- WBP C_TECHSEM **PC_COPA_PL_14** and TECHBA / TECHPBA are executed from PQ1 **ZZF_MD_BFC_PL**

Key figure aggregation : Overwrite in WBP and Summation in PQ1

DSO keys are identical in PQ1 and WBP

Aggregation is done during transformation between DSO and Cube

P&L Element attributes

- Determined from ZFC-PL hierarchy for C_PLGRP (P&L Group)
  - P&L Group is the father node of P&L Element
  - When P&L Element is out of hierarchy (no P&L Group), the default P&L Group is ZZ_99999 (these values need to be updated by business in hierarchy)
- From dpcopa13 DSO for c_mgn_acc (BFC Account) and sign

C_TECHBA and C_TECHPBA attributes are determined by reading 0G_CWWE01 attributes

Sales view:

The distribution chanel and sales organisation in PQ1 flow is determined, depending of the customer and business area, reading the DSO DPCOPA34. It was updated by a flat file based on PQ1 combinations (the link customer / business area sales organisation is not unique, so this Business rule was estimated to be correct at 98%).

The file loaded in WBP was stored the following directory: https://drive.google.com/drive/folders/1sPk5Jh8nJBNVeYIV2dX2yOB23XDQGjyU

GBU: the GBU in the reporting is the attribute of c_techba. The GBU stored in the DSO from the Business transformation layer is also the attribute of c_techba (replaced between propagation and business layer) and not the GBU from PQ1 which is linked to the business area in ZZCONACT.

Some objects in the DSO, as the GBU, the division and material group, are stored instead of being determined dynamically. It was required in order to determine some attributes (GBR axis or sales employee) in the reporting and it was considered not necessary to develop a more complex solution for this historical data flow.

GM Flow

Classic delta loading is currently not possible in Solvay side due to cumulated key figure:

after delta infopackage ran, the list of companies in the PSA is updated in TVARVC parameter PL_TVARVC_COMPANY_GM by the program Z_UPDATE_COMP_TVARVC_GM in Process chains. Moreover, the calmonth in the PSA is read in a decision form in the Process Chain and updated in TVARVC parameter PL_TVARVC_CALMONTH_GM => the Process chain allow only to load one month, if there is several month in the PSA, the loads are aborted.

In order to be sure to have all data concerning the companies/month in the delta, a full infopackage is then executed using the companies in parameter PL_TVARVC_COMPANY_GM and the calmonth in PL_TVARVC_CALMONTH_GM.

Before, to load each target in this flow, a selective deletion is done on the companies in the parameter PL_TVARVC_COMPANY_GM and the month in PL_TVARVC_CALMONTH_GM.

The GM flow works with expert routines:

There is a simple ABAP to determine Source system, /BIC/C_PCOMPAN and /BIC/C_COMPPRS and a more complex ABAP to transpose Source Key Figure into Z* P&L Elements:

- Read table c_keyfigr and loop on each value
- Uses field symbol to assign Key Figure source field and Unit source field
- Depending on c_ratioim (Ratio Type) value, we have 3 different cases:

- X means that key figure is assigned to 0AMOUNT and will be used in main P&L query
- Q means that key figure is assigne to invoice quantity (G_QVVA01)
- Empty value means that corresponding key figure will be loaded in K_AMNTDC (Amount in DC)

PCA Flow : DPCOPA11 -> DBCOPA15

There is a ABAP Rule to determine Sign,C_PCOMPRS and Logical System (simple rules)
Be aware that data are filtered on PQ1 side to load only profit center:

- S*
- D*
- SCHEF* (redundant)
- Note that a security filter has been added to DTP on WBP side

PCA Flow : DBCOPA15 -> DBCOPA14

There is a start Routine used to exclude value. It was a functional need to have all value for reporting in one DSO DBCOPA15 and filtered value in DSO DBCOPA14.

Integrated Margin (IM)

LER Determination

LER stands for Legal Entity Reduction. It allows to identify companies that works both in WP1 and PF1 systems.

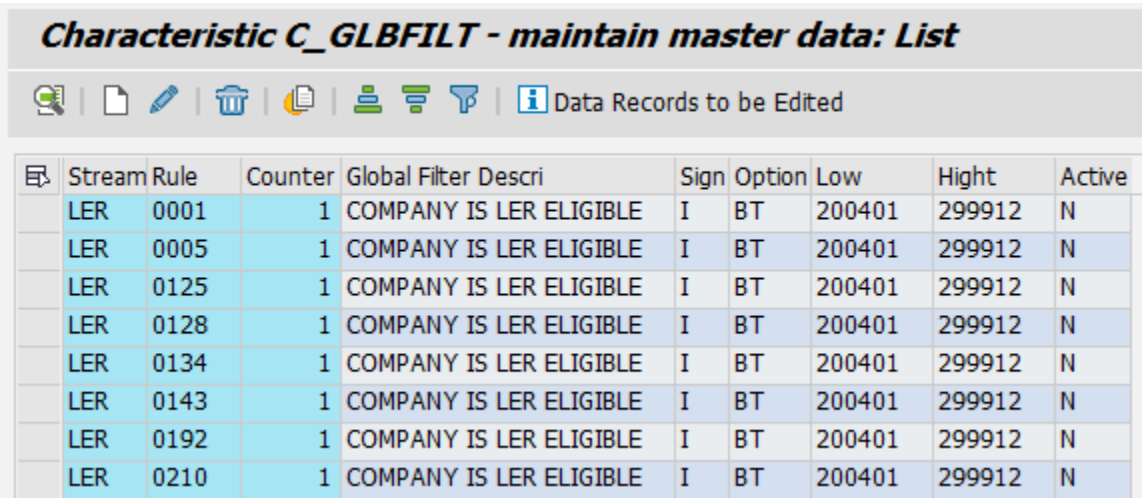
We have 3 cases for LER

1: Company is not LER and Plant is not Trading Plant

2: Company is not LER and Plant is Trading Plant

3: Company is LER

The list of companies is managed in Global filter master data **C_GLBFILT** manually or by flat file:



Stream	Rule	Counter	Global Filter Descri	Sign	Option	Low	Hight	Active
LER	0001	1	COMPANY IS LER ELIGIBLE	I	BT	200401	299912	N
LER	0005	1	COMPANY IS LER ELIGIBLE	I	BT	200401	299912	N
LER	0125	1	COMPANY IS LER ELIGIBLE	I	BT	200401	299912	N
LER	0128	1	COMPANY IS LER ELIGIBLE	I	BT	200401	299912	N
LER	0134	1	COMPANY IS LER ELIGIBLE	I	BT	200401	299912	N
LER	0143	1	COMPANY IS LER ELIGIBLE	I	BT	200401	299912	N
LER	0192	1	COMPANY IS LER ELIGIBLE	I	BT	200401	299912	N
LER	0210	1	COMPANY IS LER ELIGIBLE	I	BT	200401	299912	N

The Trading plant is managed in **C_PLANT** master data through attribute **C_SORT2.** If C_SORT2 = 'NDIR', it means the plant is a trading plant.

In transactional data flow, we determine the LER case using function module **ZDETERMINE_LER**

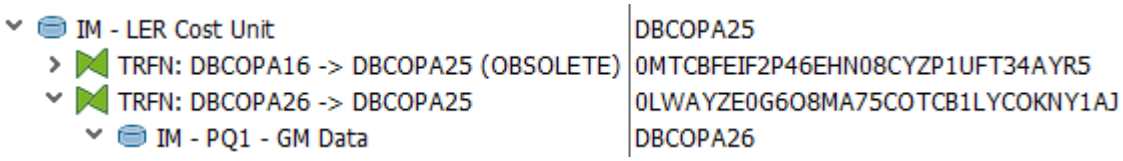
```
CLEAR w_lerflag.
CALL FUNCTION 'ZDETERMINE_LER'
  EXPORTING
    IP_CALMONTH       = <result_fields>-calmonth
    IP_LOGSYS         = <result_fields>-logsys
    IP_PLANT          = <result_fields>-/bic/c_plant
    IP_COMPCDE        = <result_fields>-/bic/c_compcde
  IMPORTING
    EP_LER_FLAG       = w_lerflag
  TABLES
    IT_PLANT          = ITB_PLANT
    IT_GLBFILT        = ITB_GLBFILT.
IF w_lerflag IS NOT INITIAL.
  <result_fields>-/bic/c_lerflag = w_lerflag.
ENDIF.
```

itb_plant is the list of plant with C_PLANT__C_SORT2 = NDIR and **itb_glbfilt** is the list extracted from C_GLBFILT master data

In the function module, we check if the company is in itb_glbfilt for the period.
If yes, LER value = 3. If not, we check if the plant is in itb_plant. If yes, LER value= 2 else LER value = 1

IM Unit cost determination

Basically, LER companies are selling in WP1 products made in PF1. So to calculate integrated margin we need to have the cost unit price from PF1 system.
To do so, we use IM from PQ1 DSO (explained after in case 4)



In the DSO, the key is the material, the origin plant in PF1, the calendar month and the value field

IM - LER Cost Unit (DBCOPA25) Rule Group: Standard Group

Rule	Rule Name	Posi	Key	InfoObject	Ico	Descript.	Int
▶	0LOGSYS	1		0LOGSYS		Source System	
▶	C_MATNR2	2		C_MATNR2		Material	
▶	C_ZZWWE41	3		C_ZZWWE41		Origin Plant	
▶	0CALMONTH	4		0CALMONTH		Calendar Year/Month	
▶	C_FIELDN	5		C_FIELDN		Value Field	
▶	Take data from C_KEYFIGR__C_LERFLD	7		C_PLELMN2		P&L Element	
▶	0G_QVVA04	8		0G_QVVA04		Qty unit base	
▶	0AMOUNT	9		0AMOUNT		Amount	
▶	0G_UVVA04	10		0G_UVVA04		Un Qty unit base	
▶	0CURRENCY	11		0CURRENCY		Currency key	
	Calculated in end routine	12		K_INTVCU		Integrated Variable cost unit	
	Calculated in end routine	13		K_INTNPCU		Integrated Non prop cost unit	
	Calculated in end routine	14		K_INTDUT		Integrated Duty unit cost	
	Calculated in end routine	15		K_INTOFC		Integrated Other Fixed Costs	

Example of results in DBCOPA25

Data Browser: Table /BIC/ADBCOPA2500: 3 of 3 Hits

Source ...	Material	Origin Plant	Calendar ...	Value Field	R	P&L Element	Qty unit b...	Amount	Un Qty uni	Currency	Integ Var	Integ NP cost unit	Integ Duty un...	Integrated Ot...
PF1_020	41476	LHC	05.2016	VVD0C		VVD0C	17.940	130.221,08-	KG	EUR	7,259-	0,000	0,00	0,00
PF1_020	41476	LHC	05.2016	VVE0C		VVE0C	17.940	6.895,06-	KG	EUR	0,000	0,384-	0,00	0,00
PF1_020	41476	LHC	05.2016	VVE2C		VVE2C	17.940	87.309,32-	KG	EUR	0,000	0,000	0,00	4,87-

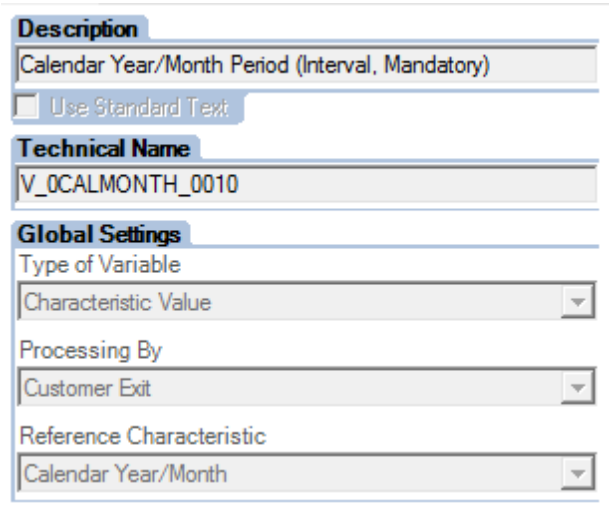
Note: there is very important semantic group in the DTP

CALMONTH
C_MATNR2
C_MATGR1

Transport Costs & Duties

To calculate costs and duties costs, we use TIERS application.
The easiest and most efficient for us was to use APD.

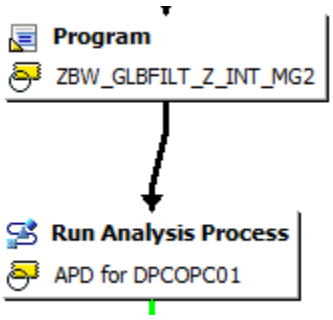
We have created 2 APD using the same query APD_DPCOPC01_0001 and APD_DPCOPC01_0002.
We need to run APD 2 times in the same process chain, for current and previous months. It was not possible to run the same APD 2 times so we had to create 2 identical APD



The customer exit is updated in the process chain

Program ZBW_GLBFILT_Z_INT_MG2 setup global filter Stream = 'Z_INT_MG' and Rule = 'CALMONTH' to be M-2 to M-1

Program ZBW_GLBFILT_Z_INT_MG setup global filter Stream = 'Z_INT_MG' and Rule = 'CALMONTH' to be M-1 to M



APD will use query BW_QRY_MVSDTR01_9999.
In the query, the month is determined by a customer exit (class ZCL_BIU001_V_0CALMONTH_0010 - After variable)

Query will select Stream = 'Z_INT_MG' and Rule = 'TIERS_MTH' for the high value = period and minus low value (normally last 6 month)

Stream	Rule	Counter	V	Change flag (I inserted / D deleted...	Global Filter Descri	Sign	Opti...	Low	Hight	Active
Z_INT_MG	TIERS_MTH	001	A		CALMONTH REFERENCE TO USE FOR DBCOPA24 (TIERS QUERY)	I	EQ	6	202306	Y

BW_QRY_MVSDTR01_9999 (APD_DPCOPC01_0001) -> DPCOPC01 -> DBCOPA24

⚠️ The date is using 0RSTT_RMNTH while Tiers query using C_SHIPNU2__C_COMPMON. It is correct from t icket WO00000000137785 (May 2022) to change from 0CALMONTH to 0RSTT_RMNTH

We store result of the APD in DPCOPC01.

Then DPCOPC01 is loaded into DBCOPA24.
After that we load data from DBCOPA01.

The order is important, DPCOPC01 then DBCOPA24 because we don't fill K_LOTSZE key figures when data exists in DPCOPC01 and DBCOPA01 (it would double the key figure)

Becareful : there is an exclusion of GBU

The authorizerd GBU are managed in Global Filter

Stream : Z_INT_MG

Rule GBU

Active : Y

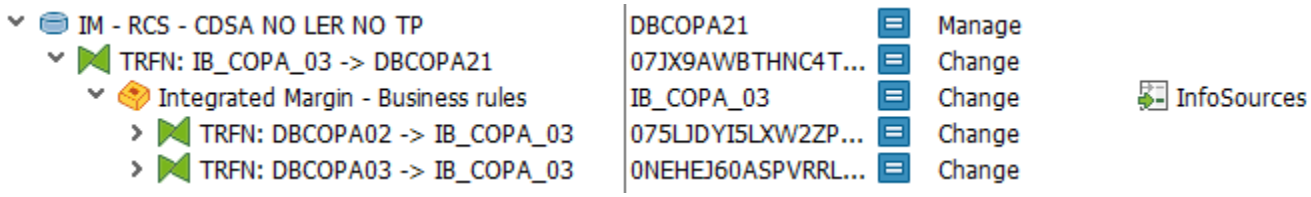
Data Browser: Table /BIC/PC_GLBFILT Select Entries 5

Stream	Rule	Counter	V	Change flag (I inserted / D deleted)	Global Filter Descri	Sign	Option	Low	Hight	Active
Z_INT_MG	GBU	001	A		GBU TO KEEP IN DBCOPA24	I	EQ	CS		Y
Z_INT_MG	GBU	002	A		GBU TO KEEP IN DBCOPA24	I	EQ	PM		N
Z_INT_MG	GBU	003	A		GBU to benefit automatisation of freight in IM calculation	I	EQ	TS		Y
Z_INT_MG	GBU	004	A		GBU SILICA to benefit automatisation of freight in IM calc	I	EQ	SI		Y
Z_INT_MG	GBU	005	A		GBU AROMA to benifit automatisation of freight in IM calc	I	EQ	PA		Y

Case 1: WP1 - No LER No Trading Plant

For WP1 flow, we have 2 sub-flows, 1 for CDSA and 1 for IECRA. And for each sub-flow, we split in 3 cases. So at the end, from 2 source DSO we have 6 target DSO.

To have a BW model as simple as possible to maintain, we have decided to use an Info source between sources and targets:



From source DSO to Info source we only have the LER determination rule to apply (function module **ZDETERMINE_LER**). In term of maintenance, it's easier because if we need to modify the rule, we only have the change the function module and not the transformation.

ZDETERMINE_LER is used in:

TRFN: DBCOPA02 -> IB_COPA_03
TRSF: ABCOPA04 -> IB_COPA_03
TRSF: ABCOPA03 -> IB_COPA_03
TRFN: DBCOPA03 -> IB_COPA_03
TRSF: ABCOPA01 -> IB_COPA_07

From Info source to target DSO, we have a start routine calling program **Z_LER_RULES**

```
* DELETE RECORDS
    DELETE SOURCE_PACKAGE WHERE /bic/c_lerflag <> '1'.
    DELETE SOURCE_PACKAGE WHERE /bic/c_fieldn <> 'QUANTITY'
                            AND /bic/c_fieldn <> 'VVD00'
                            AND /bic/c_fieldn <> 'VVE00'
                            AND /bic/c_fieldn <> 'VVF00'.

* CALL PROGRAM FOR LER
    INCLUDE Z_LER_RULES.

* DELETE RECORDS WITH VALUE FIELD QUANTITY
    DELETE SOURCE_PACKAGE WHERE /bic/c_fieldn = 'QUANTITY'
                             OR /bic/c_fieldn = 'VVD00'
                             OR /bic/c_fieldn = 'VVE00'
                             OR /bic/c_fieldn = 'VVF00'.
```

The first row with LER flag is using the function module **ZDETERMINE_LER** and we keep useful data.
The following rows are to be sure that we keep only useful value fieds. This filter is also set in the DTP.

After the program, as we generated new value fields, we delete the original ones.

ZDETERMINE_LER is used in transformations:

TRFN: IB_COPA_03 -> DBCOPA22
TRFN: IB_COPA_03 -> DBCOPA21
TRSF: IB_COPA_07 -> ABCOPA02
TRFN: IB_COPA_03 -> DBCOPA18
TRFN: IB_COPA_03 -> DBCOPA20
TRFN: IB_COPA_03 -> DBCOPA19
TRFN: IB_COPA_03 -> DBCOPA23

To have more details about **Z_LER_RULES**, there is a dedicated document about it:

https://drive.google.com/file/d/1AXQaMXmP4AdN9FeP8eByaRVQqDgs59BJBPL3NxSAhfc/view

1) determine where the program is sued to be able to select the data from corresponding dso:

```
639          WHEN 'DBCOPA03'.
640    * FOR AMOUNT
641            SELECT /bic/c_pabelnr /bic/c_paposnr co_area /bic/c_versn2
642                 vtype /bic/c_fieldn amount
643            INTO TABLE itb_dbcopa
644            FROM /bic/adbcopa0300
645            FOR ALL ENTRIES IN itb_hashed_dbcopa
646            WHERE /bic/c_pabelnr = itb_hashed_dbcopa-/bic/c_pabelnr
647            AND   /bic/c_paposnr = itb_hashed_dbcopa-/bic/c_paposnr
648            AND   co_area        = itb_hashed_dbcopa-co_area
649            AND   /bic/c_versn2  = itb_hashed_dbcopa-/bic/c_versn2
650            AND   curtype        = itb_hashed_dbcopa-curtype "CBE02++
651            AND   vtype          = itb_hashed_dbcopa-vtype
652            AND  ( /bic/c_fieldn  = 'VVD00'
653            OR     /bic/c_fieldn  = 'VVE00'
654            OR     /bic/c_fieldn  = 'VVF00' ).
655    * FOR QUANTITY
656            SELECT /bic/c_pabelnr /bic/c_paposnr co_area /bic/c_versn2
657                 vtype /bic/c_fieldn g_qvva04
658            INTO TABLE itb_qvva04
659            FROM /bic/adbcopa0300
660            FOR ALL ENTRIES IN itb_hashed_dbcopa
661            WHERE /bic/c_pabelnr = itb_hashed_dbcopa-/bic/c_pabelnr
662            AND   /bic/c_paposnr = itb_hashed_dbcopa-/bic/c_paposnr
663            AND   co_area        = itb_hashed_dbcopa-co_area
664            AND   /bic/c_versn2  = itb_hashed_dbcopa-/bic/c_versn2
665            AND   curtype        = itb_hashed_dbcopa-curtype "CBE02++
666            AND   vtype          = itb_hashed_dbcopa-vtype
667            AND  /bic/c_fieldn   = 'QUANTITY'.
668          WHEN 'ABCOPA03'.
669    * FOR AMOUNT
670            SELECT /bic/c_pabelnr /bic/c_paposnr co_area /bic/c_versn2
671                 vtype /bic/c_fieldn amount
672            INTO TABLE itb_dbcopa
673            FROM /bic/aabcopa032
674            FOR ALL ENTRIES IN itb_hashed_dbcopa
675            WHERE /bic/c_pabelnr = itb_hashed_dbcopa-/bic/c_pabelnr
676            AND   /bic/c_paposnr = itb_hashed_dbcopa-/bic/c_paposnr
677            AND   co_area        = itb_hashed_dbcopa-co_area
678            AND   /bic/c_versn2  = itb_hashed_dbcopa-/bic/c_versn2
679            AND   curtype        = itb_hashed_dbcopa-curtype "CBE02++
680            AND   vtype          = itb_hashed_dbcopa-vtype
```

2) Select data from DBCOPA25:

```
* LOADING OF DBCOPA25
      SELECT logsys /bic/c_matnr2 /bic/c_zzwwe41 calmonth
            /bic/c_fieldn g_qvva04 amount g_uvva04 currency
            /bic/k_intvcu /bic/k_intnpcu /bic/k_intdut /bic/k_intofc
        INTO TABLE itb_dbcopa25
        FROM /bic/adbcopa2500
        FOR ALL ENTRIES IN itb_hashed_dbcopa25
        WHERE /bic/c_zzwwe41 = itb_hashed_dbcopa25-/bic/c_zzwwe41
          AND /bic/c_zzwwe41 <> ''
          AND calmonth <= w_lastmth.
      SORT itb_dbcopa25 DESCENDING BY logsys /bic/c_matnr2 /bic/c_zzwwe41
      calmonth.
```

3) Determine the last day of COPA delivery month:

```
      CALL FUNCTION 'SLS_MISC_GET_LAST_DAY_OF_MONTH'
        EXPORTING
          day_in            = w_costdate
        IMPORTING
          last_day_of_month = w_lcostdate
        EXCEPTIONS
          day_in_not_valid  = 1
          OTHERS            = 2.
      .
      IF sy-subrc <> 0.
*       Implement suitable error handling here
      ENDIF.
```

4) Determine the LER FLag with help of DSO DBCOPA25 or master data C_PLANT:

```
            READ TABLE itb_dbcopa25 ASSIGNING <fs_dbcopa25>
            WITH KEY logsys         = w_logsys2
                     /bic/c_matnr2  = w_bismt
                     /bic/c_zzwwe41 = wa_source_package_tmp-/bic/c_zzwwe41
            BINARY SEARCH.

            IF sy-subrc = 0.
              LOOP AT itb_dbcopa25 FROM sy-tabix ASSIGNING <fs_dbcopa25>
                WHERE logsys = w_logsys2
                  AND /bic/c_matnr2 = w_bismt
                  AND /bic/c_zzwwe41 = wa_source_package_tmp-/bic/c_zzwwe41.
                IF <fs_dbcopa25>-calmonth  <= wa_source_package_tmp-calmonth.
                  w_dbcopa25_flg = 1.
                  EXIT.
                ENDIF.
              ENDLOOP.
            ENDIF.

* IF C_MATNR2__C_BISMT/C_ZZWWE41 IN DBCOPA25
            IF w_dbcopa25_flg = 1.
*End of Insertion - CBE01

* IF C_MATNR2__C_BISMT/C_ZZWWE41 NOT IN DBCOPA25 => NOT REAL LER
            ELSEIF wa_source_package_tmp-/bic/c_plant IS NOT INITIAL.
* CHECK IF C_PLANT IS 'NDIR' (TRADING PLANT)
              READ TABLE itb_plant ASSIGNING <fs_plant>
                WITH TABLE KEY logsys       = wa_source_package_tmp-logsys
                               /bic/c_plant = wa_source_package_tmp-/bic/c_plant.
```

```
*********************************************************************
* CASE OF LER FLAG = 3 : LER BUT NOT FOUND IN LER TABLE
* SUB-CASE PLANT IS TRADING PLANT, SAME CODE AS LER FLAG = 2, ELSE LER FLAG = 1
*********************************************************************
              IF sy-subrc = 0." AND wa_source_package_tmp-/bic/c_fieldn <> 'QUANTITY'.
                w_ler = 2.
*********************************************************************
* CASE OF LER FLAG = 3 : LER BUT NOT FOUND IN LER TABLE
* SUB-CASE PLANT IS NOT TRADING PLANT, SAME CODE AS LER FLAG = 1
*********************************************************************
              ELSE."IF wa_source_package_tmp-/bic/c_fieldn <> 'VVD00'.
                w_ler = 1.
              ENDIF.
*********************************************************************
* CASE OF LER FLAG = 3 : LER BUT NOT FOUND IN LER TABLE
* SUB-CASE PLANT IS NULL, SAME CODE AS LER FLAG = 1
*********************************************************************
*             ELSEIF wa_source_package_tmp-/bic/c_fieldn <> 'VVD00'.
            ELSE.
              w_ler = 1.
            ENDIF.
*********************************************************************
* CASE OF LER FLAG = 2 : NO LER TRADING PLANT
*********************************************************************
          WHEN '2'.
            w_ler = 2.
*********************************************************************
* CASE OF LER FLAG = 1 : NO LER NO TRADING PLANT
*********************************************************************
          WHEN '1'.
            w_ler = 1.
          WHEN OTHERS.
        ENDCASE.
```

In function of LER Flag:

Case 1 & 2: 1) No LER no Trading Plant 2) No LER Trading Plant

If c_fieldn = VVD00 or VVE00 or VVF00 and transaction Type = B and Z:

Read data from COPA table and if g_qvva04 is empty + if amount is not empty Create a new line with fieldn VVD0C:

```
          WHEN 1 or 2.  "RI235773 - 27/12/2022 - UC - 4000003116
*           CLEAR: w_flagpcp4.
            IF wa_source_package_tmp-/bic/c_fieldn = 'VVD00' OR
               wa_source_package_tmp-/bic/c_fieldn = 'VVE00' OR
               wa_source_package_tmp-/bic/c_fieldn = 'VVF00'.
* SPECIAL CASE FOR REC TYPE B OR Z AND QUANTITY EMPTY
              IF ( wa_source_package_tmp-rec_type = 'B' OR wa_source_package_tmp-rec_type = '
* VALUE FIELD VVD0C
                READ TABLE itb_qvva04 ASSIGNING <fs_qvva04>
                  WITH TABLE KEY
                  /bic/c_pabelnr = wa_source_package_tmp-/bic/c_pabelnr
                  /bic/c_paposnr = wa_source_package_tmp-/bic/c_paposnr
                  co_area        = wa_source_package_tmp-co_area
                  /bic/c_versn2  = wa_source_package_tmp-/bic/c_versn2
                  vtype          = wa_source_package_tmp-vtype
                  /bic/c_fieldn  = 'QUANTITY'.
                IF ( sy-subrc = 0 AND (
                  <fs_qvva04>-g_qvva04 IS INITIAL OR <fs_qvva04>-g_qvva04 = 0 OR
                  <fs_qvva04>-g_qvva04 = '0.000' ) OR sy-subrc <> 0 ).

*Start of Insertion - CBE02
* We do not create line if amount 0 => no risk with delta as the COPA line can't change
                  IF wa_source_package_tmp-amount IS NOT INITIAL AND
                    wa_source_package_tmp-amount <> 0 AND
                    wa_source_package_tmp-amount <> '0.00'.

*Generate line VVD0C (resp. VVE0C, VVF0C) = VVD00 (resp. VVE00, VVF00)
                    wa_source_package_tmp-/bic/c_fieldn+4(1)  = 'C'.
                    COLLECT wa_source_package_tmp INTO it_source_package.
*Start of Deletion - CBE03
**Generate line VVD0M (resp. VVE0M, VVF0M) = VVD00 (resp. VVE00, VVF00)
*                    wa_source_package_tmp-/bic/c_fieldn+4(1)  = 'M'.
*                    COLLECT wa_source_package_tmp INTO it_source_package.
*End of Deletion - CBE03

                  ENDIF.
*End of Insertion - CBE02
```

If c_fieldn = QUANTITY (standard case): we pick data from ODS_PCP4

```
* CBI01 - 20191128  - UC 4000021782  - END
* GET AMOUNT CP FROM ODS_PCP4 WITH COSTVAR 'ZFO'
              UNASSIGN <fs_pcp4>.
*              READ TABLE itb_pcp4 ASSIGNING <fs_pcp4>    "CBE02--
              READ TABLE itb_pcp4_zfo ASSIGNING <fs_pcp4> "CBE02++
                WITH TABLE KEY costdate     = w_costdat_shft
*                               costvar      = 'ZFO'       "CBE02--
                               plant        =
                               wa_source_package_tmp-/bic/c_plant
                               /bic/c_matnr =
                               wa_source_package_tmp-/bic/c_matnr2.
* IF NO RECORD FOUND IN ODS_PCP4 WITH ZFO, TRY WITHOUT ZFO and ZIP
              IF sy-subrc <> 0.
                UNASSIGN <fs_pcp4>.
*                READ TABLE itb_pcp4_zfo ASSIGNING <fs_pcp4> "CBE02--
                READ TABLE itb_pcp4 ASSIGNING <fs_pcp4>      "CBE02++
                   WITH TABLE KEY costdate     = w_costdat_shft
                                  plant        =
                                  wa_source_package_tmp-/bic/c_plant
                                  /bic/c_matnr =
                                  wa_source_package_tmp-/bic/c_matnr2.
* IF NO RECORD FOUND IN ODS_PCP4, NO ENTRY
                IF sy-subrc <> 0.
                ENDIF.
              ENDIF.
```

```
                IF <fs_pcp4>-currency <> wa_source_package_tmp-currency.
* NEED TO MULTIPLY AMOUNT BY 100 TO AVOID DECIMALS
                  w_c_amt_cp  = <fs_pcp4>-/bic/c_amt_cp * 100.
                  w_c_amt_cnp = <fs_pcp4>-/bic/c_amt_cnp * 100.
                  w_c_amt_amo = <fs_pcp4>-/bic/c_amt_amo * 100.
                  w_k_intfrg  = <fs_pcp4>-/bic/k_intfrg *
                                100.
                  w_k_intdut  = <fs_pcp4>-/bic/k_intdut *
                                100.
*  JBU 20/07/2021  UC 4000029887 - START
                  w_c_amt_raw = <fs_pcp4>-/bic/c_amt_raw * 100.
                  w_c_amt_pck = <fs_pcp4>-/bic/c_amt_pck * 100.
                  w_c_amt_tol = <fs_pcp4>-/bic/c_amt_tol * 100.
                  w_c_amt_uti = <fs_pcp4>-/bic/c_amt_uti * 100.
*  JBU 20/07/2021  UC 4000029887 - END
```

If necessary, we convert quantitites and amounts with module functions: 'Z_CONVERT_CURRENCY' with type of rate CAR3
+ 'Z_MD_CONVERT_MATERIAL_UNIT_F'

And we calculate new value fields:

```
1451  * CREATE NEW VALUE FIELDS
1452  *  JBU 20/07/2021  UC 4000029887 - START
1453                  CLEAR: w_c_amt_total.
1454                  w_c_amt_total = <FS_PCP4>-/BIC/C_AMT_RAW + <FS_PCP4>-/BIC/C_AMT_PCK + <FS_P
1455
1456                  IF w_c_amt_total = 0 OR w_c_amt_total = '0.00'.
1457  * VALUE FIELD VVD0C - OLD CP Detail if no split available
1458                      IF <fs_pcp4>-/bic/c_amt_cp IS NOT INITIAL AND
1459                      <fs_pcp4>-/bic/c_amt_cp <> 0               AND
1460                      <fs_pcp4>-/bic/c_amt_cp <> '0.00'          AND
1461                      <fs_pcp4>-/bic/k_lotsize <> 0.
1462                        wa_source_package_tmp-/bic/c_fieldn  = 'VVD0C'.
1463                        wa_source_package_tmp-amount         = -1 *
1464                         ( wa_source_package_tmp-g_qvva04 * w_c_amt_cp ) / w_lotsize.
1465                        COLLECT wa_source_package_tmp INTO it_source_package.
1466                      ENDIF.
1467                   ELSE.
1468  * VALUE FIELD VVD0C - CP SPLIT: RAW
1469                      IF <FS_PCP4>-/BIC/C_AMT_RAW IS NOT INITIAL AND
1470                      <FS_PCP4>-/BIC/C_AMT_RAW <> 0              AND
1471                      <FS_PCP4>-/BIC/C_AMT_RAW <> '0.00'         AND
1472                      <FS_PCP4>-/BIC/K_LOTSIZE <> 0.
1473                        WA_SOURCE_PACKAGE_TMP-/BIC/C_FIELDN   = 'VVD0C'.
1474                        WA_SOURCE_PACKAGE_TMP-AMOUNT          = -1 *
1475                        ( WA_SOURCE_PACKAGE_TMP-G_QVVA04 * W_C_AMT_RAW ) / W_LOTSIZE
1476                        .
1477                        COLLECT WA_SOURCE_PACKAGE_TMP INTO IT_SOURCE_PACKAGE.
1478                      ENDIF.
1479  * VALUE FIELD VPD0C - CP SPLIT: PCK
1480                       IF <FS_PCP4>-/BIC/C_AMT_PCK IS NOT INITIAL AND
```

If no records are found in ODS_PCP4, we determine value fields from source DSO

```
* IF WE DIDN'T FIND RECORDS IN ODS_PCP4, WE CHECK IN DBCOPA02
            ELSE.
* VALUE FIELD VVD0C
              READ TABLE itb_dbcopa ASSIGNING <fs_dbcopa>
                WITH TABLE KEY
                /bic/c_pabelnr = wa_source_package_tmp-/bic/c_pabelnr
                /bic/c_paposnr = wa_source_package_tmp-/bic/c_paposnr
                co_area        = wa_source_package_tmp-co_area
                /bic/c_versn2  = wa_source_package_tmp-/bic/c_versn2
                vtype          = wa_source_package_tmp-vtype
                /bic/c_fieldn  = 'VVD00'.
              IF sy-subrc = 0 AND <fs_dbcopa>-amount IS NOT INITIAL AND <fs_dbcopa>-amount <> 0 AND <fs_dbcopa>-amount <> '0.00'.
                wa_source_package_tmp-/bic/c_fieldn  = 'VVD0C'.
                wa_source_package_tmp-amount         =
                <fs_dbcopa>-amount.
                COLLECT wa_source_package_tmp INTO it_source_package.
              ENDIF.
* VALUE FIELD VVE0C
              READ TABLE itb_dbcopa ASSIGNING <fs_dbcopa>
                WITH TABLE KEY
                /bic/c_pabelnr = wa_source_package_tmp-/bic/c_pabelnr
                /bic/c_paposnr = wa_source_package_tmp-/bic/c_paposnr
                co_area        = wa_source_package_tmp-co_area
                /bic/c_versn2  = wa_source_package_tmp-/bic/c_versn2
                vtype          = wa_source_package_tmp-vtype
                /bic/c_fieldn  = 'VVE00'.
              IF sy-subrc = 0.
                wa_source_package_tmp-/bic/c_fieldn  = 'VVE0C'.
                wa_source_package_tmp-amount         =
                <fs_dbcopa>-amount.
                COLLECT wa_source_package_tmp INTO it_source_package.
              ENDIF.
* VALUE FIELD VVF0C
              READ TABLE itb_dbcopa ASSIGNING <fs_dbcopa>
                WITH TABLE KEY
                /bic/c_pabelnr = wa_source_package_tmp-/bic/c_pabelnr
                /bic/c_paposnr = wa_source_package_tmp-/bic/c_paposnr
                co_area        = wa_source_package_tmp-co_area
                /bic/c_versn2  = wa_source_package_tmp-/bic/c_versn2
                vtype          = wa_source_package_tmp-vtype
                /bic/c_fieldn  = 'VVF00'.
              IF sy-subrc = 0.
                wa_source_package_tmp-/bic/c_fieldn  = 'VVF0C'.
                wa_source_package_tmp-amount         =
                <fs_dbcopa>-amount.
                COLLECT wa_source_package_tmp INTO it_source_package.
              ENDIF.
```

Case 3: WP1 - LER

This case works only with quantity value field. We read COPA data.

```
          WHEN 3.
*********************************************************************
* CASE OF LER FLAG = 3 : LER
*********************************************************************
********************
* THIS CASE WORKS ONLY WITH QUANTITY
            IF wa_source_package_tmp-/bic/c_fieldn  = 'QUANTITY'.
* RECOVER QUANTITY
              READ TABLE itb_qvva04 ASSIGNING <fs_qvva04>
                WITH TABLE KEY
                /bic/c_pabelnr = wa_source_package_tmp-/bic/c_pabelnr
                /bic/c_paposnr = wa_source_package_tmp-/bic/c_paposnr
                co_area        = wa_source_package_tmp-co_area
                /bic/c_versn2  = wa_source_package_tmp-/bic/c_versn2
                vtype          = wa_source_package_tmp-vtype
                /bic/c_fieldn  = 'QUANTITY'.
********************
```

We get the value field VVD0C from DBCOPA25 (IM Cost Unit) to calculate new value field

```
1908  *Start of Insertion - CBE01
1909              CLEAR w_dbcopa25_flg.
1910
1911              READ TABLE itb_dbcopa25_2 ASSIGNING <fs_dbcopa25_2>
1912              WITH KEY logsys         = w_logsys2
1913                       /bic/c_matnr2  = w_bismt
1914                       /bic/c_zzwwe41 = wa_source_package_tmp-/bic/c_zzwwe41
1915                       /bic/c_fieldn  = 'VVD0C'
1916              BINARY SEARCH.
1917
1918              IF sy-subrc = 0.
1919                LOOP AT itb_dbcopa25_2 FROM sy-tabix ASSIGNING <fs_dbcopa25_2>
1920                  WHERE logsys = w_logsys2
1921                    AND /bic/c_matnr2 = w_bismt
1922                    AND /bic/c_zzwwe41 = wa_source_package_tmp-/bic/c_zzwwe41
1923                    AND /bic/c_fieldn  = 'VVD0C'.
1924
1925                  IF <fs_dbcopa25_2>-calmonth  <= wa_source_package_tmp-calmonth.
1926                    w_dbcopa25_flg = 1.
1927                    EXIT.
1928                  ENDIF.
1929                ENDLOOP.
1930              ENDIF.
1931
1932              IF w_dbcopa25_flg = 1.
1933  *End of Insertion - CBE01
```

If necessary, we convert quantitites with module function 'Z_MD_CONVERT_MATERIAL_UNIT_F'

And we calculate new value fields:

```
2006                IF <fs_dbcopa25_2>-/bic/k_intvcu IS NOT INITIAL AND
2007                <fs_dbcopa25_2>-/bic/k_intvcu <> 0.
2008                  wa_source_package_tmp-/bic/c_fieldn  = 'VVD0C'.
2009                  wa_source_package_tmp-amount         =
2010                  <fs_qvva04>-g_qvva04 * w_k_intvcu.
2011                  COLLECT wa_source_package_tmp INTO it_source_package.
2012                ENDIF.
2013              ENDIF.
2014  ********************
```

It's the same logic for value fields VVE0C and VVE2C.

At the beginning of the project, we determined also VVC3C but it was dropped later.
The code is in comments so it's easy to add it.

Case 4: PF1

For PF1 system, we only use the GM flow (concerning the PQ1 flox[22]w).

Expert routine is the same except for the selection of key figures (C_KEYFIGR info object)

In IM, we keep key figures with the flag as X and ZN8110BQTY P&L element. In the normal flow (concerning the PQ1 flow) , we exclude key figures with flag as X

Characteristic C_KEYFIGR - maintain master data: List

Data Records to be Edited

Key Figure	PF1/P&L Element	Magnitude	Ratio Type	Sign	Unit InfoObjet	Flag for I	Srce Sys.	LER Value	Long Description
K_SALQTYK	ZN8110BQTY	N8110	Q		BASE_UOM			QUANTITY	Sales Quantity in KG
K_ZCNGDEP	ZDEPC	R2549C	X	-	/BIC/C_CNCRCDV	X		VVE0C	Cdv conso depreciations
K_ZCNGOFC	ZFIXCOSTC	R2549C	X	-	/BIC/C_CNCRCDV	X		VVE2C	Cdv conso other fix costs
K_ZCNGRMT	ZMATC	R1540C	X	-	/BIC/C_CNCRCDV	X		VVD0C	Cdv conso raw materials
K_ZCNGTOL	ZTOLLINGC	R1540C	X	-	/BIC/C_CNCRCDV	X		VVD0C	Cdv conso tollings
K_ZCNGUTL	ZUTILC	R1540C	X	-	/BIC/C_CNCRCDV	X		VVD0C	Cdv conso utilities
K_ZDISC	ZR10000DIL	R10000	X		LOC_CURRCY				Discounts & surcharges (invoiced)
K_ZDISCOC	ZR10000DID	R10000			DOC_CURRCY				Discounts in transaction currency
K_ZDUTY	ZDUTYC	R1290C	X	-	LOC_CURRCY	X		VVC3C	Duty
K_ZGROSS	ZR10000GRL	R10000	X		LOC_CURRCY				Invoiced Gross amount
K_ZGROSSC	ZR10000GAD	R10000			DOC_CURRCY				Gross amount in transaction currency (invoice)
K_ZLEGDLT	ZDCOPAL	R15400	X	-	LOC_CURRCY				Cdv legal delta COPCA
K_ZLEGOFC	ZFIXCOSTL	R25490	X	-	LOC_CURRCY				Cdv legal other fix costs
K_ZLEGRMT	ZMATL	R15400	X	-	LOC_CURRCY				Cdv legal raw materials
K_ZLEGTOL	ZTOLLINGL	R15400	X	-	LOC_CURRCY				Cdv legal tollings
K_ZLEGUTL	ZUTILL	R15400	X	-	LOC_CURRCY				Cdv legal utilities
K_ZLGDEP	ZDEPL	R25490	X	-	LOC_CURRCY				Cdv legal depreciations
K_ZREBAT	ZR10000REL	R10000	X		LOC_CURRCY				Rebates
K_ZREBATC	ZR10000RED	R10000			DOC_CURRCY				Rebates in transaction currency
K_ZTRNSPT	ZR12910	R12910	X	-	LOC_CURRCY				Transport

PF1 IM flow particularities:

ZZCONACT table in PF1 is the source for IM key figures in P&L model.

ZZCONACT is fully emptied each time a recalculation occurs. A recalculation is done for **only one month** and a list of companies (not always all the companies) => the solution should be adapted if several months are calculated at the same time.

ZZCONACT is loaded in DPCOPA23.

In WBP, a selective deletion on the recalculated perimeter (reading the PSA content) is done at each level of DSO/cube before loading. In consequence, no need to activate the data in DSO, so it was chosen to use write optimised DSO instead of standard one.

Selectives deletions are done in PC_COPA_PL_30 and PC_COPA_PL_33.

As the selective deletion is done according to PSA content, and only one month is allowed, please think to delete request in the PSA after a manual update!

All characteristics should part of the key but BW do not permit more than 16 Infoobjects, in consequence, technical objects were created to concatenate the value of several characteristic into one objet (similarly to the PQ1 solution) in dso *DPCOPA23 and DBCOPA34*.

If there is duplicated data in the DSO, it could be due to an incorrect recovery procedure or because a new field was added to the ZZCONACT table and should be part of the key.

Allocation of internal logistic costs in IM (DBCOPA28 and DBCOPA36):

Percentage of allocation:

The DSO DPCOPA08 contains the result of the APD ADP_DSO_DPCOPA08. This APD retrieve, from the query BW_QRY_MVCOPA01_0011, the internal logistic costs (Euros) and the external sold quantity (VKG) at level industrial origin / business area / country of destination for the period [M-2 to M-1]:

- M = month defined in c_glbfilt__c_high stream = "PL_ELEMENT", rule = "INT_MARG", counter = "1", it is equal to the month being uploaded from ZZCONACT, derived from TVARVC parameter PL_TVARVC_CALMONTH_GM - 1.
- The number of months to take into account for the calculation, currently 2, can be changed using c_glbfilt__c_low stream = "PL_ELEMENT", rule = "INT_MARG", counter = "1" (if c_low ="1" => calculation is done on M-2 to M-1, if c_low ="2" => calculation is done on M-3 to M-1,...)
- The percentage of allocation is calculated in the APD by the formula : (100 * R12910- Internal Log) / ABS(N8110 - Qty Sold Ext).

Remarks: DPCOPA08 contains only the result of the last APD execution. These data are then loaded into DBCOPA13 which keeps all historical data.

For the moment, the query BW_QRY_MVCOPA01_0011 can't give the result for several GBU (today only one GBU = SD) and it would not be easy to create a unique query that determine the key figure "R12910- Internal Logistic Cost" for several GBU as it should be filtered, for each GBU, by the partner = # or the concerned GBU. If necessary to enhance to several GBU in the future, a solution could be to determine the key figures split by GBU and Partner GBU and to complete the calculation in a transformation.

Reloading procedure:

The following steps must be done once by month to recover (it would be interessant to create a recovery Process Chain, that load for instance one complete year automatically, updating global filter before each APD execution)

1) Change the month in c_glbfilt__c_high for stream = "PL_ELEMENT", rule = "INT_MARG", counter = "1" to the month to reload -1.

2) Execute APD ADP_DSO_DPCOPA08

3) Selective deletion on the month to recover in DBCOPA13

4) Execute DTP: DPCOPA08 -> DBCOPA13 - Full then activate load requests in DBCOPA13

R1291C - Integrated Logistics costs - variable (TIERS)

The original solution (DBCOPA28) was created for PQ1 flow and is kept for historical data (<2018). This solution was reproduced for PF1 flow (DBCOPA36) for data >= 2018, the main difference between historical and new solution is that quantities are not coming from ZZCONACT anymore but from COPA.

PQ1 flow (<2018):

It consists to generate a line on P&L element ZR1291C => it is derived from the quantities on which we **apply the percentage of allocation (= the internal logistic costs / the external sold quantity) in the DSO DBCOPA13** at a level industrial origin / business area / country of destination.

The quantities are determined from ZZCONACT lines with P&L element = "ZN8110BQTY".

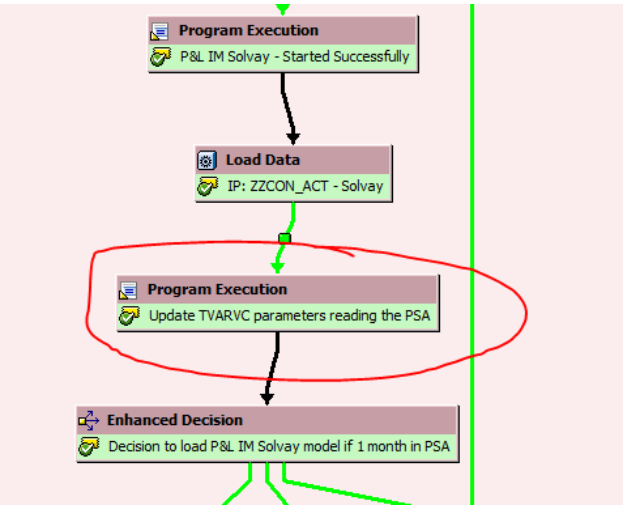
This solution is currently used only for Soda-Ash: DTP and start routine filtered on C_TECHBA__CPFCTR1_2="SD", the GBU is maintained in C_GLBFILT MD: stream = "PL_ELEMENT", rule = " IM_IL_GBU"). In order to enhance to other GBU, it would be necessary to update DBCOPA13 to have the percentage of allocation by GBU.

PF1 flow:

It consists to generate a line on P&L element ZR1291C => it is derived from the quantities on which we **apply the percentage of allocation (= the internal logistic costs / the external sold quantity) in the DSO DBCOPA13** at a level industrial origin / business area / country of destination.

The quantities are determined from COPA lines. Following CROCO project, some quantities are in unity "PRT" (services), these quantities must not be taken into account in the calculation.

The Process Chain is synchronised with ZZCONACT loading as it will determine the month to calculate (as today the percentage of allocation for M-1 is determined using costs on [M-3;M-2] there is no real dependance). This month is updated in TVARVC variable "PL_TVARVC_CALMONTH_IM_SOLV" updated in PF1 IM Process Chain PC_COPA_PL_36:



Deletion of the data on the month to reload is done on the DSO in case the keys are not the same after reload (change on BU, country of destination or restated BA determination or less lines after ZZCONACT recalculation) but it should not be usefull normally.

This solution is currently used only for Soda-Ash: end routine filtered on C_TECPCT2__CPFCTR1_2="SD", the GBU is maintained in C_GLBFILT MD: stream = "PL_ELEMENT", rule = " IM_IL_GBU"). In order to enhance to other GBU, it would be necessary to update DBCOPA13 to have the percentage of allocation by GBU (TRFN: DBCOPA29 -> DBCOPA36)

Case 5: Non ERP

For Non ERP we load data from propagation layer but we only keep some value fields we modify (TRFN: DBCOPA09 -> DBCOPA27)

```
3239
3240          CASE wa_source_package-/bic/c_mgn_acc.
3241            WHEN 'R15400'.
3242              wa_source_package-/bic/c_mgn_acc  = 'R1540C'.
3243              COLLECT wa_source_package INTO it_source_package.
3244  *         WHEN 'R12910'.                                    FD3612850
3245  *           wa_source_package-/bic/c_mgn_acc  = 'R1291C'.
3246  *           COLLECT wa_source_package INTO it_source_package.
3247  *         WHEN 'R12900'.                                    FD3612850
3248  *           wa_source_package-/bic/c_mgn_acc  = 'R1290C'.
3249  *           COLLECT wa_source_package INTO it_source_package.
3250            WHEN 'R25490'.
3251              wa_source_package-/bic/c_mgn_acc  = 'R2549C'.
3252              COLLECT wa_source_package INTO it_source_package.
3253            WHEN OTHERS.
3254          ENDCASE.
3255        ENDIF.
3256      ENDLOOP.
3257
3258  * REFRESH SOURCE PACKAGE AND FILL IT WITH SPLITTED RECORDS
3259      REFRESH: SOURCE_PACKAGE.
3260      IF it_source_package[] IS NOT INITIAL.
3261        SOURCE_PACKAGE[] = it_source_package[].
3262      ENDIF.
3263
3264  * FREE INTERNAL TABLE
3265      REFRESH: it_source_package.
3266      FREE: it_source_package.
3267
3268  ********************START CBE 06/2022*****************
3269  * DELETE SOURCE PACKAGE TO KEEP ONLY R15400, R12910, R12900 AND R25490
3270      DELETE SOURCE_PACKAGE WHERE /bic/c_mgn_acc <> 'R2549C'
3271                              AND /bic/c_mgn_acc <> 'R1540C'
3272                              AND /bic/c_mgn_acc <> 'R1291C'
3273                              AND /bic/c_mgn_acc <> 'R1290C'.
3274  ********************END CBE 06/2022*****************
3275
3276  *    DELETE SOURCE_PACKAGE WHERE /bic/c_mgn_acc = 'R15400'
3277  *                             OR /bic/c_mgn_acc = 'R12910'    FD 3612850
3278  **                             OR /bic/c_mgn_acc = 'R12900'     FD
3279  *3612850
3280  *                             OR /bic/c_mgn_acc = 'R25490'.
3281
3282    ENDMETHOD.                    "start_routine
```

Case 7: Sales in PF1, Production in WP1

In some cases, Integrated Margin needs to be calculated for a sale stored in Solvay system PF1, containing materials produced by a RCS plant and thus stored in WP1 system.

A set of eligibility rules has been established to qualify the need to check in WP1 system for Integrated Margin, as displayed in the graph below.

To calculate Integrated Margin in these cases, we must:

1. Collect the Material information from WP1, associate the quantity discovered to an entry in the Cost Unit DSO (*ODS_PCP4*).
2. Suppress the existing Integrated Margin value calculated from PF1 entries (these "negative" entries are stored in *DBCOPA41*).
3. Make the data available for Reporting, via a new cube CRCOPA30 added to MultiProvider MVCOPA01.

Every one of these operations are centered around new DataStore Object *IM - Solvay Sales at Rhodia IM costs* (DBCOPA35), which

- Collects data from existing Solvay DataFlow (DSO *Profitability Analysis: COPA - Solvay Legacy* DBCOPA30), **Note:** *There is a filter in transformation between DBCOPA30 and IB_COPA_06 to delete all but Novecare (CS) data.*
- Proceeds with eligibility checks (reading DSOs *Profitability Analysis: COPA - Origin/Plant Match-up* DPCOPA35 and *IM - Solvay PF1 - GM Data (Write-Optimized)* DBCOPA34 to qualify eligibility),
- Calculates Integrated Margin on eligible entries,
- Feeds the list of eligible entries to *IM - Solvay PF1 - GM Data Annulation (Write-Optimized)* DBCOPA41 to negate existing Integrated Margin entries in DBCOPA34,
- Feeds cube *IM - Solvay PF1 - GM Data Annulation* CRCOPA30 for Reporting of the new WP1 Integrated Margins.

Process Chain: PC_COPA_PL_45 - IM: RCS-Produced Solvay Materials

This PC will delete/reload the data on 0CALMONTH = parameter in C_GLBFILTER

Stream = COPA
Rule = DBCOPA35
Counter = 001

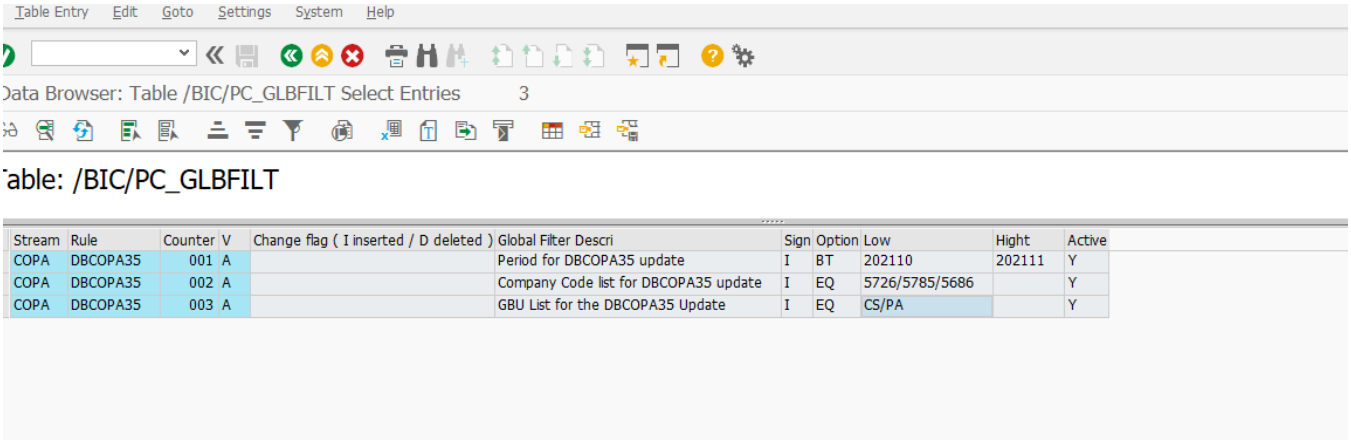
on following infoproviders (program Z_DEL_DBCOPA41 variant Z_MAIN)

DBCOPA41, DBCOPA35, CRCOPA30

Filter data to load from DBCOPA30 to DBCOPA35 on TRSF: ODSO DBCOPA30 -> TRCS IB_COPA_06

Counter 002 by company code

Coutner 003 by GBU

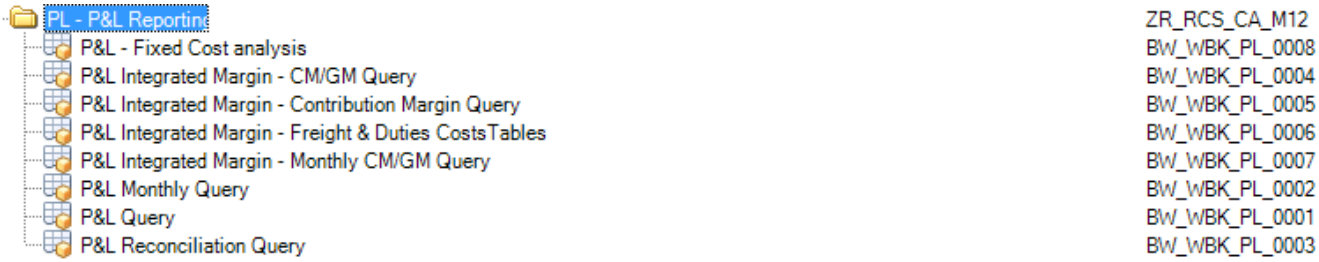


Stream	Rule	Counter	V	Change flag (I inserted / D deleted)	Global Filter Descri	Sign	Option	Low	Hight	Active
COPA	DBCOPA35	001	A		Period for DBCOPA35 update	I	BT	202110	202111	Y
COPA	DBCOPA35	002	A		Company Code list for DBCOPA35 update	I	EQ	5726/5785/5686		Y
COPA	DBCOPA35	003	A		GBU List for the DBCOPA35 Update	I	EQ	CS/PA		Y

Reporting

Queries End User Documentation

All the reporting is available throught workbooks in the role "PL - P&L Reporting" ZR_RCS_CA_M12



Query documentation:

https://drive.google.com/drive/folders/1pFQ_tfTUYK1xEz4-cz0JFel7arKyY7Nn?usp=drive_link

Reporting documentation drive folder:

https://drive.google.com/drive/folders/19pPWUmAkTcW1aYlzlqyHOxvL6JfDdaKe

https://drive.google.com/drive/folders/1zF0862L7nRnu39MXdtVORT-HkmHavs-R

https://drive.google.com/open?id=0B_p_Afe8sjVITEVnc3NjdDV5U0U https://drive.google.com/drive/folders/1miRFX_LJKQ5YGWq-r3-YETCNkZcRhLvy

https://drive.google.com/drive/folders/1xY6QOu8gmrvmyncdpSu5trUA5SQIliRX

https://drive.google.com/drive/folders/1rzUHdVecCOROXOkZBpp7uhbHNJqIc63Y

Main queries

There are several queries but the main ones are:

BW_QRY_MVCOPA01_0001 BW P&L Query

BW_QRY_MVCOPA01_0002 BW P&L - Monthly Query

BW_QRY_MVCOPA01_0003 BW P&L Reconciliation Query

BW_QRY_MVCOPA01_0005 BW Integrated Contribution Margin Query

All queries use structures for key figures. There are many key figures, using calculated and restricted key figures inside.

Key figures logic

Main functionnalities: Jump query available

Dependencies with other applications

List of APD:

APD	Description
APD_PL_0007	ADP P&L Contrib (current month - Conso)
APD_PL_0006	ADP P&L Contrib (current month - Detailed)
APD_PL_0005	ADP P&L Sales Contrib (current and last month)
APD_PL_0004	ADP P&L VC+QTY current month
APD_PL_0001	ADP P&L VC+QTY last month
APD_PL_0003	ADP P&L VSE+QTY current month
APD_PL_0002	ADP P&L VSE+QTY last 2 months
APD_PL_0008	APD MM - SMOG Evolution
APD_PL_0009	APD P&L - Contrib for Global Sales M (Solvay)
APD_PL_0012	APD P&L - Contrib for Global Sales M-1 (Rhodia)

APD_PL_0013	APD P&L - Contrib for Global Sales M-1 (Solvay)
APD_PL_0023	APD P&L Smartfloat
APD_PL_0022	APD P&L: ICM Current month (Rhodia)

Data Loading

Info providers and objects loaded

List on info providers inside the technical cockpit.

Main process chains are:

PC_COPA_PL_GLOBAL : Load WP1 & PF1 COPA, BFC and WP1 IM daily

PC_COPA_PL_22 : Load WP1 & PF1 COPA Extra load during closing period. This chain included too chain PC_COPA_PL_12, which load BFC File in WBP.

PC_COPA_PL_36 : Load PF1 IM and allocation of transportation costs PF1

PC_COPA_PL_09 : Load Non ERP Master data and Transactional data

Other chains:

Process chain	Providers Loaded	Frequency	Time Start	Duration
PC_COPA_PL_03	CRCOPA29 CRCOPA31 CRCOPA02 CRCOPA01	Daily	Arround 1:20am	Arround 5 mins
PC_COPA_PL_12	CRCOPA03 CRCOPA04 CRCOPA05 CRCOPA06 CRCOPA07 CRCOPA08	Daily	Arround 2am	Arround 8 mins
PC_COPA_PL_20	CRCOPA21	Daily	Every hour	Arround 2 mins
PC_COPA_PL_21	CRCOPA18 CRCOPA14 CRCOPA17 CRCOPA13 CRCOPA15 CRCOPA16 CRCOPA20 CRCOPA32	Daily	Arround 2:15am	Arround 6 mins
PC_COPA_PL_32	CRCOPA22	Daily	Arround 1:10am	Arround 3 mins
PC_COPA_PL_33	CRCOPA23	Third and Fourth working day	Several execution between 2am and 4pm	Arround 5 mins
PC_COPA_PL_42	CRCOPA26	Third and Fourth working day	Several execution between 2am and 4pm	Arround 5 mins
PC_COPA_PL_45	CRCOPA30	Third and Fourth working day	Several execution between 2am and 4pm	Arround 8 mins
PC_GBR_14	CRCOPA09 CRCOPA20	Daily	Arround 3:40am	Arround 20 mins

Loading frequency

WP1 and PF1(except for PF1 IM) are loaded daily at 1h30 am (french time) and also 4 other times between working day 2 and 4 (8am, 1pm, 3pm and 8pm, managed in c_glbfilt PL_ELEMENT/CHECK_LOAD).

PF1 IM are loaded usually once by day from D+2 to D+5 (it is executed automatically on event after ZZCONACT update in PF1)

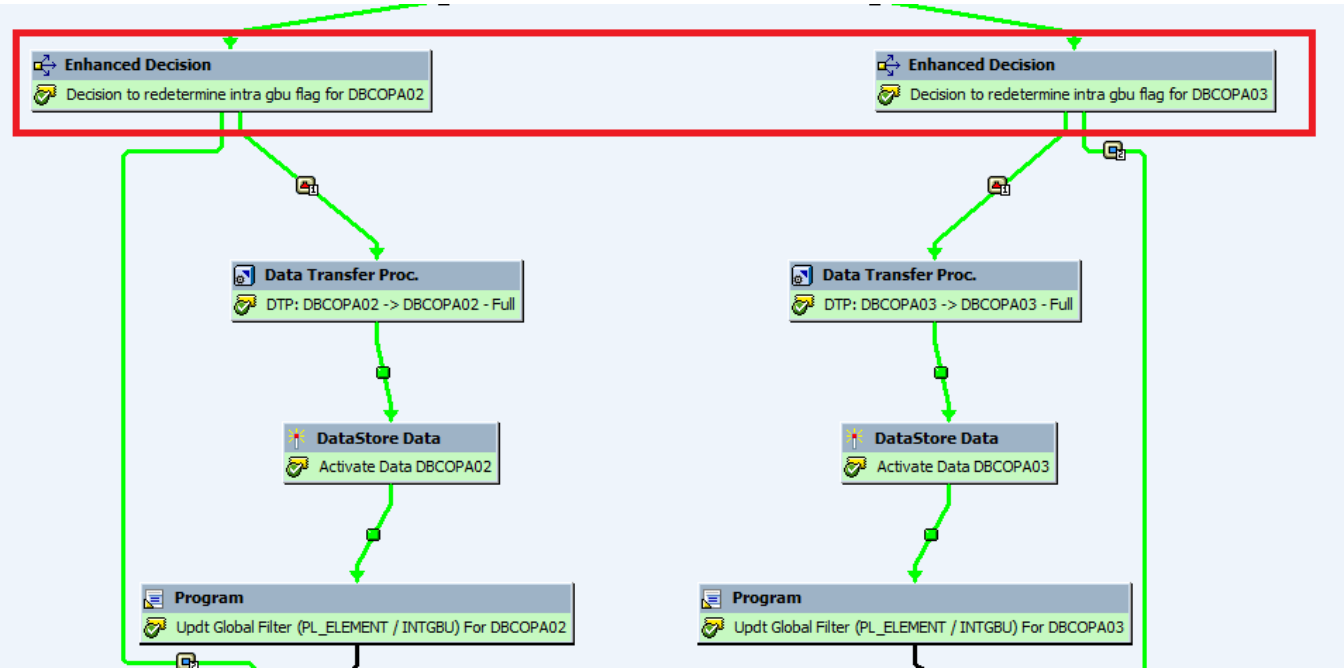
BFC is loaded daily at 1h30 am (french time). On closing period from D+2 to D+5 it is loaded too on 08H, 13H, 15H and 20H.

NON ERP is loaded 4 times a day (7 AM, 13 PM, 18 PM, 24 PM):

PQ1 loading: on demand from PQ1 system => PQ1 is now decommissioned, no more loading

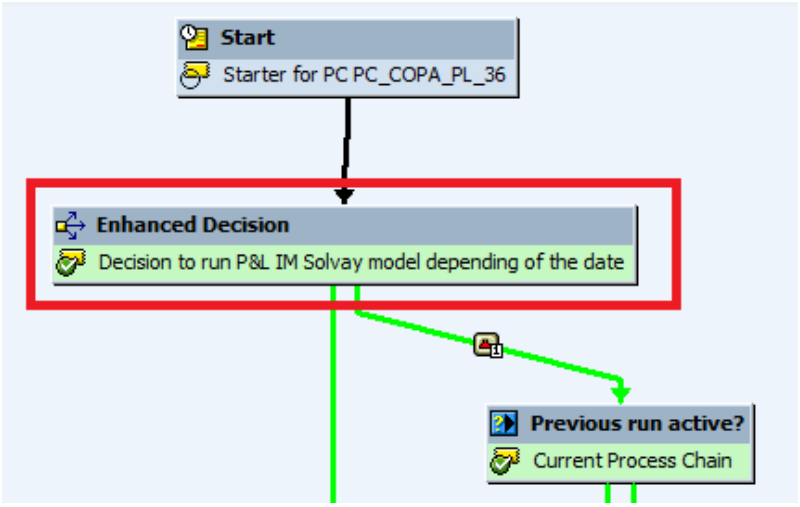
Flag intra gbu decision variant:

Inside the PC PC_COPA_PL_03 (WP1 P&L), the loadings of DBCOPA02 DBCOPA02 and ABCOPA03 ABCOPA03 to redetermine the flag intra GBU are only once a week, determined by enhanced decision variants that read the last week of redetermination using MD c_glbfilt (stream = "PL_ELEMENT", rule = "INTGBU")



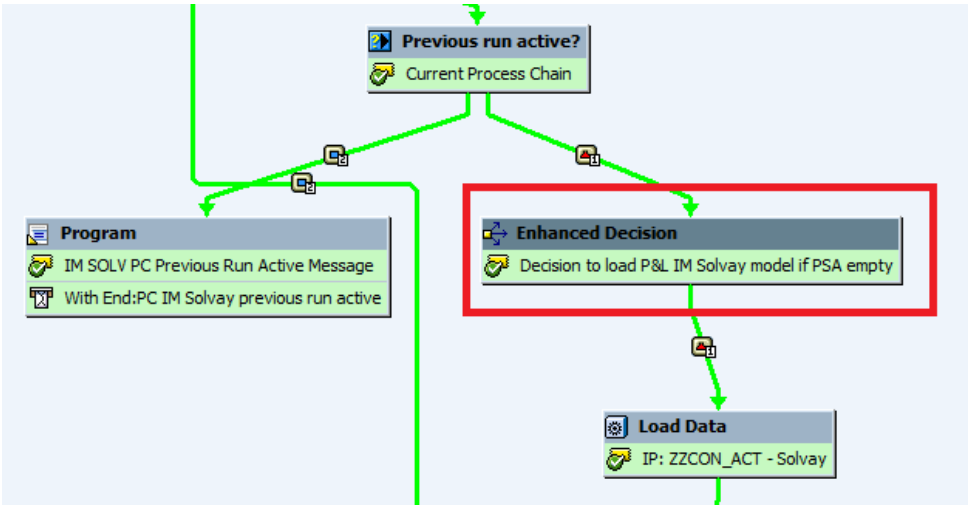
IM PF1 decision variants:

- In PC PC_COPA_PL_36, the first decision variant is used to determine when loading is allowed, depending of the date => it is currently set to run from workday D to D+31 (so no restriction anymore). This interval of workdays is defined in c_glbfilt MD, for stream = "PL_SOLV" and rule = "IM_RUN_PER". The calendar used to determine the workday is also set in c_glbfilt, for stream = "PL_SOLV" and rule = " IM_RUN_CAL".

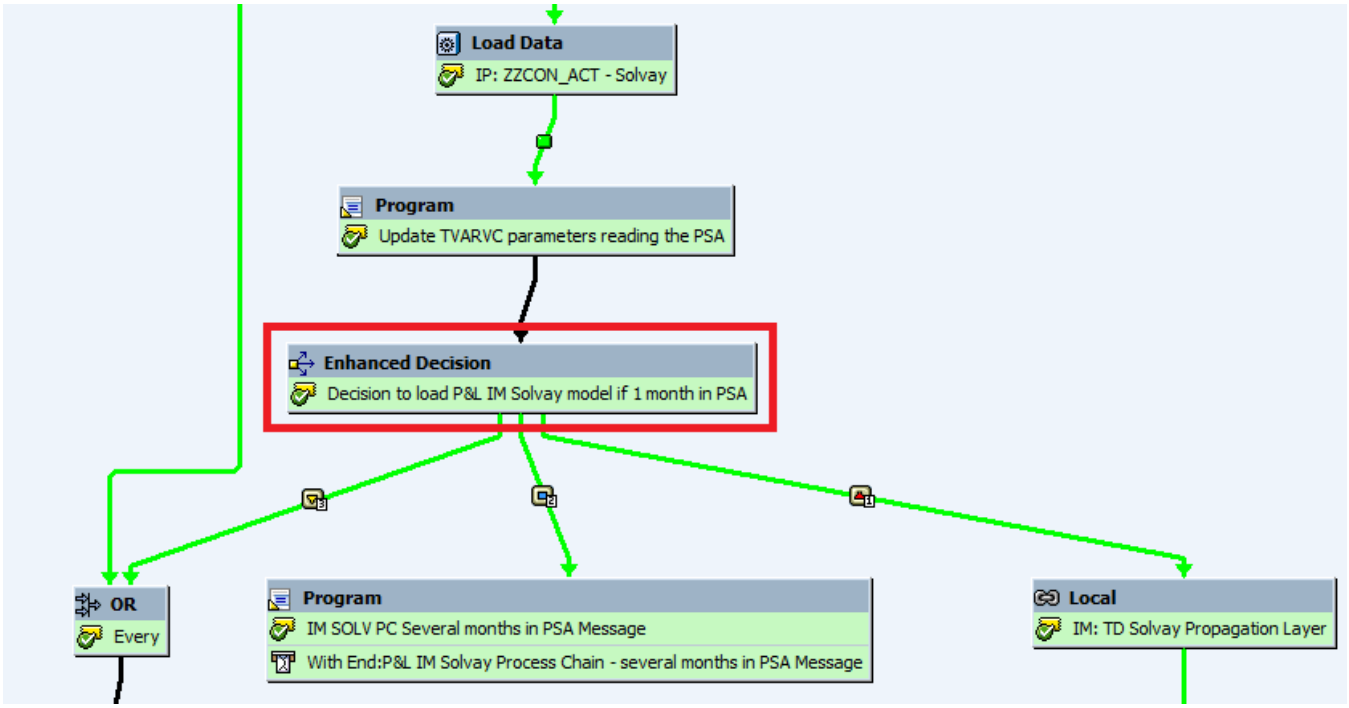


- The second decision variant read the PSA, using PSA table name in table RSTSODS for the datasource set in c_glbfilt MD (stream = "PL_SOLV", rule ="DSNAME"). It ensures that there is no remaining request in the PSA before to start a new loading which could mean a

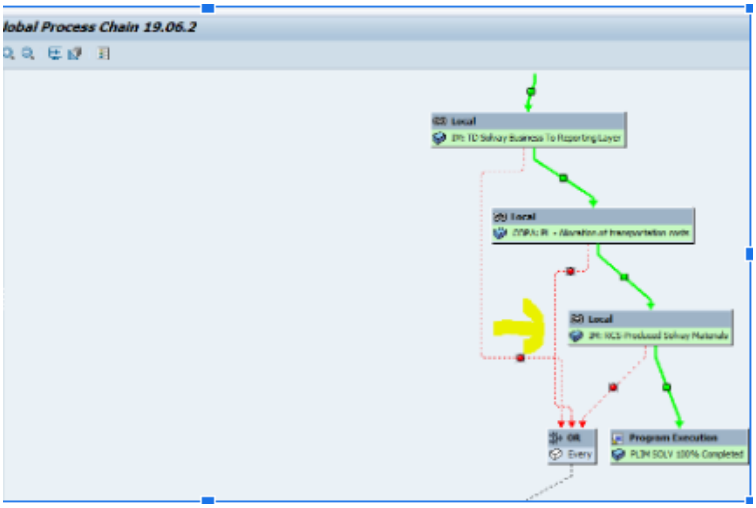
potential problem: that the last run was not completed as PSA is emptied at the end of the Process chain or the PSA was not cleaned after a manual loading.



- The third decision variant read the TVARV parameter "PL_TVARVC_CALMONTH_IM_SOLV" which contains the calmonths that were loaded in the PSA. The TVARV was updated previously in ABAP variant "Z_UPDATE_TVARVC_PL_IM_SOLV" which uses PSA table name in table RSTSODS for the datasource set in c_glbfilt MD (stream = "PL_SOLV", rule ="DSNAME").
  - If there is only one month in the PSA, then it is the normal behaviour and the loading is authorized.
  - If there are several months in the PSA, the PC is aborted and an error message is sent. In that case, an analysis must be done to understand why there are several months before to reload the data.



As part of the change # 4287353 , In the meta chain PC_COPA_PL_36,added the PC_COPA_PL_45 process chain after the execution of PC_COPA_PL_42.



Average performance

WP1: 1h40 for P&L and IM

PF1: 20min for COPA and 20min for IM

BFC: 10 min

Non ERP: 5 min

PQ1: 10min for GM and 1h for PCA

Record Keeping

We keep all records.

Dependencies with other applications

Broadcast

No broadcast

Open Hub

Domain	OH	Name	Query	Target
Finance	OH_IMEP1	IMEP-WP1/PF1dataforPCFOH	BW_QRY_MVCOPP01_0021	Z_BW_PCF_IMEP
Finance	OH_SOL002	Solstice:CPCOPA01	BW_QRY_MVCOPA02_0002	Z_SOLSTICE_OHSOL002_FILE

APD

DOMAIN	PROCESS	NAME	QUERY	TARGET	PC
Finance	APD_PL_0003	ADP P&L VSE+QTY current month	APD_QRY_MVCOPA01_0001	DPSDSO70	PC_GL_SALES_APD0003
Finance	APD_PL_0002	ADP P&L VSE+QTY last 2 months	APD_QRY_MVCOPA01_0002	DPSDSO69	-
Finance	APD_PL_0001	ADP P&L VC+QTY last month	APD_QRY_MVCOPA01_0003	DPSDSO72	-
Finance	APD_PL_0004	ADP P&L VC+QTY current month	APD_QRY_MVCOPA01_0004	DPSDSO73	PC_GL_SALES_APD001
Finance	APD_PL_0005	ADP P&L Sales Contrib (current and last month)	APD_QRY_MVCOPA01_0005	DPSDSO74	PC_GL_SALES_APD_CONTRIB

Finance	APD_PL_0006	ADP P&L Contrib (current month - Detailed)	APD_QRY_MVCOPA01_0007	DPSDSO75	PC_GL_SALES_APD_CONTRIB
Finance	APD_PL_0007	ADP P&L Contrib (current month - Conso)	APD_QRY_MVCOPA01_0007	DPSDSO95	PC_GL_SALES_APD_CONTRIB
Finance	APD_PL_0009	APD P&L - Contrib for Global Sales M (Solvay)	APD_QRY_MVCOPA01_0009	DPSDSO81	-
Finance	APD_PL_0010	SPM - APD P&L (Rhodia) - curr + prev month	APD_QRY_MVCOPA01_0010	DPSDSP10	PC_SPM_TRANS_01
Finance	APD_PL_0011	SPM - APD P&L (Solvay) - curr + prev month	APD_QRY_MVCOPA01_0011	DPSDSP11	PC_SPM_TRANS_02
Finance	APD_PL_0012	APD P&L - Contrib for Global Sales M-1 (Rhodia)	APD_QRY_MVCOPA01_0013	DPSDSO88	-
Finance	APD_PL_0013	APD P&L - Contrib for Global Sales M-1 (Solvay)	APD_QRY_MVCOPA01_0014	DPSDSO89	-
Finance	APD_PL_0022	APD P&L: ICM Current month (Rhodia)	APD_QRY_MVCOPA01_0015	DPSDSO95	-
Finance	APD_PL_0020	SPM - APD P&L (Rhodia) - any period	APD_QRY_MVCOPA01_0020	DPSDSP10	PC_SPM_RELOAD_01
Finance	APD_PL_0021	SPM - APD P&L (Solvay) - any period	APD_QRY_MVCOPA01_0021	DPSDSP11	-
Finance	APD_PL_0023	APD P&L Smartfloat	APD_QRY_MVCOPA01_0023	Z_BW_SMARTFLOAT_EXTRACT	PC_PL_APD0001
Finance	APD_PL_GS_0030	Netback Sales P&L	APD_QRY_MVCOPA01_NBACK_0001	DPSDSO7A	PC_GL_SALES_APD030
Finance	APD_PL_GS_0031	Netback Sales P&L (w/o interval)	APD_QRY_MVCOPA01_NBACK_0002	DPSDSO7A	-
Finance	ADP_DSO_DPCOPA08	APD for DSO DPCOPA08	BW_QRY_MVCOPA01_0011	DPCOPA08	PC_COPA_PL_23 PC_COPA_PL_42 PC_COPA_PL_25
Dynasys	APD_DPS_DYNASYS_0023	Dynasys - ICM COPA	BW_QRY_MVCOPA01_0027	Z_BW_DYNASYS_ICM_COPA	PC_DPS_DYNASYS_09
Dynasys	APD_DPS_DYNASYS_0046	Dynasys - Contribution Margin SKU	BW_QRY_MVCOPA01_0029	Z_BW_DYNASYS_CM_SKU	PC_DPS_DYNASYS_09
EHS	APD_EHS_SVHC_001	EHS - SVHC - Previous Month Sales	BW_QRY_MVCOPA01_EHS_001A	DPEHS024	-
EHS	APD_EHS_SVHC_002	EHS - SVHC - Quarterly Sales	BW_QRY_MVCOPA01_EHS_001B	DPEHS024	-
EHS	APD_EHS_SHVC_003	EHS - SVHC - Last 12 months Sales excluding current month	BW_QRY_MVCOPA01_EHS_001C	DPEHS024	PC_EHS_SVHC_006
Dynasys	APD_DPS_DYNASYS_0068	Dynasys - Netback COPA	BW_QRY_MVCOPA01_NETBACK_0002	Z_BW_DYNASYS_NETBACK	PC_DPS_DYNASYS_09

Data Quality Control

Data come from SAP system. To compare data between BW and sources systems, check propagation layers.

Table KE30 can valid with P&L on non-console view.

Table CE1+ Operating concern in the source should be the same value as Propagation layer of COPA data.

Operational Documentation

Procedures

P&L : Non ERP transaction

https://drive.google.com/file/d/1Mqb9wnFqnp2tl6j4AFKbWjm6BnDGSxPTQJHyOYvwspQ/view

P&L : Time reference for CICC and GM flat files

https://drive.google.com/file/d/1EoZAjLveLXL2wASs1m-l1D8LO8l28y5A_Jze6sO-bS4/view

IM: Delta + full load of current and previous months.

Program for LER determination

https://drive.google.com/file/d/1VnOB0-jeV85c6kJYram-fuIh70U3gWU7LHiTHtB8e0I/view

https://drive.google.com/file/d/12hYsIOncEYpMI6aeOtjZtIHS3o2-GCLnnCZiKzJOUiU/view

Program for LER info providers

https://drive.google.com/file/d/1AXQaMXmP4AdN9FeP8eByaRVQqDgs59BJBPL3NxSAhfc/view

Scheduling

<Describe the scheduling in place for the application (eg. existing jobs, trigger time/event based, dependencies)>

Monitoring

<Describe the monitoring checks to confirm the application is performing well (eg. check the overall status, check performance metrics like runtime/data volume/memory/disk/CPU, maintain and react to alerts/notifications)>

Error Handling

<Describe how to handle errors (eg. error codes, description and respective resolution, alert users)>

Known Bugs

<List the existing bugs, its criticity, workarounds and resolution plan.>

Roadmap

Todo list: https://drive.google.com/file/d/1cvHfPpBByAQnYIZUZpvMkwFQmgmGESkSdrQrBhcTz9s/view