IM - Transformations rules

- 1 Technical Rules on Workbench
  - 1.1 General informations
  - 1.2 1.1 ODS_PCP4 flow
    - 1.2.1 6DB_GL_COST_PURCHASE DW2 DSO_PCP2
      - 1.2.1.1 Fields routines
    - 1.2.2 DSO_PCP1
      - 1.2.2.1 1) DTS_KEKO_CKIS IFS_KEKO_CKIS
        - 1.2.2.1.1 Fields routines
      - 1.2.2.2 2) FS_KEKO_CKIS ODS_PCP1
        - 1.2.2.2.1 Start routine
        - 1.2.2.2.2 Fields routines
      - 1.2.2.3 ODS_PCP1 ODS_PCP1
      - 1.2.2.4 DTS_KEKO_CKIS DSO_PCP1
        - 1.2.2.4.1 Start routine
      - 1.2.2.5 DSO_PCP1 DSO_PCP1
        - 1.2.2.5.1 Start routine
      - 1.2.2.6 0CO_PC_PCP_02 DSO_PCP3
        - 1.2.2.6.1 Start routine
        - 1.2.2.6.2 End routine
      - 1.2.2.7 DSO_PCP3 DSO_PCP3
        - 1.2.2.7.1 Start routine
      - 1.2.2.8 0CO_PC_PCP_02 ODS_PCP3
        - 1.2.2.8.1 Fields routines
        - 1.2.2.8.2 End routine
      - 1.2.2.9 DBCOPA01 DBCOPA24
        - 1.2.2.9.1 Start routine
        - 1.2.2.9.2 Field routines
        - 1.2.2.9.3 End routine
      - 1.2.2.10 ODS_PCP6
      - 1.2.2.11 ODS_PCP6 ODS_PCP4
        - 1.2.2.11.1 End routine
  - 1.3 1.2 CPX system flow
  - 1.4 1.3 MVCOPA01 flow
    - 1.4.1 Non ERP
      - 1.4.1.1 DBCOPA09 -> DBCOPA27 (IM)
        - 1.4.1.1.1 START ROUTINE
        - 1.4.1.1.2 FIELD ROUTINE
        - 1.4.1.1.3 END ROUTINE
    - 1.4.2 Solvay (PF1)
      - 1.4.2.1 DPCOPA23 IB_COPA_04
        - 1.4.2.1.1 Field routine
      - 1.4.2.2 IB_COPA_04 DBCOPA34
        - 1.4.2.2.1 Expert routine
      - 1.4.2.3 IB_COPA_06 DBCOPA35
        - 1.4.2.3.1 Start routine
        - 1.4.2.3.2 End routine
      - 1.4.2.4 DBCOPA35 -> DBCOPA41 (IM)
        - 1.4.2.4.1 Expert routine
      - 1.4.2.5 DBCOPA35 -> DBCOPA34
        - 1.4.2.5.1 Expert routine
      - 1.4.2.6 DBCOPA29 -> DBCOPA36
        - 1.4.2.6.1 Field routine
        - 1.4.2.6.2 End routine
    - 1.4.3 Rhodia
      - 1.4.3.1 DBCOPA03 -> IB_COPA_03
        - 1.4.3.1.1 START ROUTINE
        - 1.4.3.1.2 END ROUTINE
      - 1.4.3.2 IB_COPA_03 DBCOPA20 & IB_COPA_03 DBCOPA23
        - 1.4.3.2.1 START ROUTINE
      - 1.4.3.3 IB_COPA_03 DBCOPA18 & IB_COPA_03 DBCOPA21
        - 1.4.3.3.1 START ROUTINE
      - 1.4.3.4 IB_COPA_03 DBCOPA19 & IB_COPA_03 DBCOPA22 & IB_COPA_07 -> ABCOPA02
        - 1.4.3.4.1 START ROUTINE
      - 1.4.3.5 ABCOPA02 -> CRCOPA32 & DBCOPA22 -> CRCOPA14
      - 1.4.3.6 Start routine
        - 1.4.3.6.1 End routine
  - 1.5 List of queries

Authorization Objects

List of authorization objects mandatory for the application.

Technical Rules on Workbench

General informations

PQ1 providers are not loaded since end of 2018. Only historical data.

Technicals rules not linked to Integrated Margin providers (not with IM in red in the data flows) are inTechnical Documentation - P&L wiki

1.1 ODS_PCP4 flow

6DB_GL_COST_PURCHASE DW2 DSO_PCP2

Fields routines

There are 3 individual rountines in this transformation to divide the following key figure values that are of currencies Korean Won (KRW) and Japanese Yen (JPY) by 100.

- C_AMT_CP
- C_AMT_CNP
- C_AMT_AMO

DSO_PCP1

1) DTS_KEKO_CKIS IFS_KEKO_CKIS

Fields routines

Fiscal Variant 0FISCVARNT is fixed to K4.

Cost Component Structure 0CCOMPSTRUC is fixed to Z1.

Fiscal Year/Period required some ABAP codings to convert the incoming data into the correct format to be updated.

For the key figure Variable Amount, abap code is required to take the total amount minus the fixed amount to get just the variable amount. (WRTFW_KPF - WRTFW_KFX).

2) FS_KEKO_CKIS ODS_PCP1

Start routine

A select is done on master data global filter in internal table ITB_PLANT_REPLACE_PLANT

/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
M_UNIT_COS	VENDOR	001	A		Variable internal toller vendor code	I	BT	0000091157	7525	Y

A select is done to take all records from ODS_PCP2 into table gt_matrix.

Delete all records where the amount and quantity is equal to zero.

Fields routines

0COSTCOMP (Cost Component):

> If source record for 0COSTCOMP is equal to ' ' (space = null / empty), we make a loop on gt_matrix (Else we keep 0COSTCOMP from source).
>
> > If gs_matrix-/BIC/C_COSTEL1 <= SOURCE_FIELDS-COSTELMNT and gs_matrix-/BIC/C_COSTEL2 >= SOURCE_FIELDS-COST ELMNT, we take the COSTCOMP from gt_matrix (ODS_PCP2).

In all cases, if 0COSTCOMP from source or from ODS_PCP2 is equal to ' ', the defaul value = 100.

0PLANT_COMP (Issuing Plant):

If C_MATNR = C_COMPNT and PLANT = PLANT_COMP and PCPITEMCAT = 'L' and VENDOR is not empty.

Read table ITB_PLANT_REPLACE_PLANT with help of field VENDOR (VENDOR from source = C_LOW from master data C_GLBFILT).

If something is found, the PLANT_COMP = PLANT from ITB_PLANT_REPLACE_PLANT (field C_HIGH from master data global filter).

Else, keep source PLANT_COMP.

ODS_PCP1 ODS_PCP1

Collect into ITB_REF_DATE value from master data global filter (C_GLBFILT) for STREAM = F_COPC and RULE = ODS_PCP1

/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_ACTIV
F_COPC	ODS_PCP1	001	A		FISCAL PERIOD & DATE FOR DATA REPLICATION	E	EQ	20090901		Y

COSTDATE = low value from ITB_REF_DATE.

DTS_KEKO_CKIS DSO_PCP1

Start routine

A select is done to take all records from ODS_PCP2 into table gt_matrix.

Delete all records where the amount and quantity is equal to zero.

Delete records with Valuation variants (BWVAR) not equal to ZFO, Z25 and ZEP.

Fields routines

Fiscal Variant 0FISCVARNT is fixed to K4.

Cost Component Structure 0CCOMPSTRUC is fixed to Z1.

Costing Variant (0COSTVAR) is fixed to ZIP.

Fiscal Year/Period required some ABAP codings to convert the incoming data into the correct format to be updated.

For the key figure Variable Amount, abap code is required to take the total amount minus the fixed amount to get just the variable amount. (WRTFW_KPF - WRTFW_KFX).

DSO_PCP1 DSO_PCP1

Start routine

1. Load DSO_PCP1 into itself with a look-up at DSO_PCP2 for matching records of the month with the following field value assigments:
    a. Generic Fields:
        - DSO_PCP1-C_PRODVER = ''.
        - DSO_PCP1-ACTTYPE = ''.
        - DSO_PCP1-C_COSTCTR = ''.
        - DSO_PCP1-PCP_RES = ''.
        - DSO_PCP1-WORKCENTER = ''.
        - DSO_PCP1-C_BOMNUM = ''.
        - DSO_PCP1-C_BOMUSAG = ''.
        - DSO_PCP1-C_BOMALT = ''.
        - DSO_PCP1-PLANT_COMP = DSO_PCP2-PLANT.
        - DSO_PCP1-C_COMPNT = DSO_PCP2-/BIC/C_MATPLNT.
        - DSO_PCP1-LOTSIZE_IT = DSO_PCP2-LOTSIZE.
        - DSO_PCP1-CURRENCY = DSO_PCP2-CURRENCY.
        - DSO_PCP1-UNIT = DSO_PCP2-UNIT.
        - DSO_PCP1-BASE_UOM = DSO_PCP1-UNIT.
        - DSO_PCP1-OI_MENGE = DSO_PCP1-LOTSIZE_IT.
        - DSO_PCP1-PRICE_UNIT = DSO_PCP1-LOTSIZE_IT.
    b. Variable Amount (CP):
        - DSO_PCP1-C_KAPOSNR = '1'.

- DSO_PCP1-AMOUNT = DSO_PCP2-C_AMT_CP.
- DSO_PCP1-AMOUNTVR = DSO_PCP1-AMOUNT.
- DSO_PCP1-AMOUNTFX = 0.
- DSO_PCP1-COSTELMNT = '0098150974'.
- DSO_PCP1-COSTCOMP = '100'.

c. Fixed Amount (CNP):

- DSO_PCP1-C_KAPOSNR = '2'.
- DSO_PCP1-AMOUNT = DSO_PCP2-C_AMT_CNP.
- DSO_PCP1-AMOUNTVR = 0.
- DSO_PCP1-AMOUNTFX = DSO_PCP1-AMOUNT.
- DSO_PCP1-COSTELMNT = '0099430120'.
- DSO_PCP1-COSTCOMP = '220'.
- DSO_PCP1-PCPITEMCAT = 'E'.
- DSO_PCP1-C_COMPNT = ''.

d. Depreciation Amount (AMO):

- DSO_PCP1-C_KAPOSNR = '3'.
- DSO_PCP1-AMOUNT = LV_DSO_PCP2-C_AMT_AMO.
- DSO_PCP1-AMOUNTVR = 0.
- DSO_PCP1-AMOUNTFX = WA_PCP1_SP-AMOUNT.
- DSO_PCP1-COSTELMNT = '0099438000'.
- DSO_PCP1-COSTCOMP = '300'.
- DSO_PCP1-PCPITEMCAT = 'E'.
- DSO_PCP1-C_COMPNT = ''.

0CO_PC_PCP_02 DSO_PCP3

Keys figures in summation

Start routine

Delete records with KLVAR (Costing Variant) not equal to ZFO and ZEP and Z25.

Delete records with CURTYP (Currency Type) not equal to 10.

Fields routines

Costing Variant (COSTVAR) = ZIP.

AMOUNT = WERTF (Fixed value) + WERTV (Variable value)

K_CCR_100 (100 Raw Materials):

IF SOURCE_FIELDS-ELEMT (Cost Component) = '100'.

RESULT = SOURCE_FIELDS-WERTF (Fixed value) + SOURCE_FIELDS-WERTV (Variable value).

K_CCR_110 (110 Toll Manufacturing):

IF SOURCE_FIELDS-ELEMT = '110'.

RESULT = SOURCE_FIELDS-WERTF + SOURCE_FIELDS-WERTV.

K_CCR_120 (120 Utilities - VC):

IF SOURCE_FIELDS-ELEMT = '120'.
RESULT = SOURCE_FIELDS-WERTF + SOURCE_FIELDS-WERTV.

K_CCR_200 (200 Labor Costs):

IF SOURCE_FIELDS-ELEMT = '200'.
RESULT = SOURCE_FIELDS-WERTF + SOURCE_FIELDS-WERTV.

K_CCR_210 (210 Supplies & Misc.):

IF SOURCE_FIELDS-ELEMT = '210'.
RESULT = SOURCE_FIELDS-WERTF + SOURCE_FIELDS-WERTV.

K_CCR_220 (220 Maintenance):

IF SOURCE_FIELDS-ELEMT = '220'.
RESULT = SOURCE_FIELDS-WERTF + SOURCE_FIELDS-WERTV.

K_CCR_230 (230 Overheads):

IF SOURCE_FIELDS-ELEMT = '230'.
RESULT = SOURCE_FIELDS-WERTF + SOURCE_FIELDS-WERTV.

K_CCR_240 (240 Utilities -FC):

```
IF SOURCE_FIELDS-ELEMT = '240'.
RESULT = SOURCE_FIELDS-WERTF  + SOURCE_FIELDS-WERTV.
```

K_CCR_300 (300 Depreciation):

```
IF SOURCE_FIELDS-ELEMT = '300'.
RESULT = SOURCE_FIELDS-WERTF  + SOURCE_FIELDS-WERTV.
```

K_CCR_310 (310 N/A):

```
IF SOURCE_FIELDS-ELEMT = '310'.
RESULT = SOURCE_FIELDS-WERTF  + SOURCE_FIELDS-WERTV.
```

End routine

The purpose of this routine is to check if the current incoming record already exists in the DSO_PCP3 from the previous month loading. If it exists, delete the incoming record.

DSO_PCP3 DSO_PCP3

Start routine

In internal table LV_DSO_PCP2, we take records from ODS_PCP2 in function of PLANT, C_MATNR, FISCPER and FISCVAR.

```
DSO_PCP3-AMOUNT = LV_DSO_PCP2-C_AMT_CP + LV_DSO_PCP2-C_AMT_CNP + LV_DSO_PCP2-C_AMT_AMO

DSO_PCP3-AMOUNTVR = LV_DSO_PCP2-C_AMT_CP.

DSO_PCP3-AMOUNTFX = LV_DSO_PCP2-C_AMT_CNP + LV_DSO_PCP2-C_AMT_AMO

DSO_PCP3-LOTSIZE = LV_DSO_PCP2-LOTSIZE.

DSO_PCP3-CURRENCY = LV_DSO_PCP2-CURRENCY.

DSO_PCP3-UNIT = LV_DSO_PCP2-UNIT.

DSO_PCP3-K_CCR_100 = DSO_PCP3-AMOUNTVR.

DSO_PCP3-K_CCR_110 = 0.

DSO_PCP3-K_CCR_120 = 0.

DSO_PCP3-K_CCR_200 = 0.

DSO_PCP3-K_CCR_210 = 0.

DSO_PCP3-K_CCR_220 = LV_DSO_PCP2-C_AMT_CNP.

DSO_PCP3-K_CCR_230 = 0.

DSO_PCP3-K_CCR_240 = 0.

DSO_PCP3-K_CCR_300 = LV_DSO_PCP2-C_AMT_AMO.

DSO_PCP3-K_CCR_310 = 0.

DSO_PCP3-LOTSIZE = LV_DSO_PCP2-LOTSIZE.

DSO_PCP3-CURRENCY = LV_DSO_PCP2-CURRENCY.

DSO_PCP3-UNIT = LV_DSO_PCP2-UNIT.
```

0CO_PC_PCP_02 ODS_PCP3

Keys figures in summation

Fields routines

Costing Variant (COSTVAR) = ZIP.

AMOUNT = WERTF (Fixed value) + WERTV (Variable value)

K_CCR_100 (100 Raw Materials):

IF SOURCE_FIELDS-ELEMT (Cost Component) = '100'.

RESULT = SOURCE_FIELDS-WERTF (Fixed value) + SOURCE_FIELDS-WERTV (Variable value).

K_CCR_105 (105 Packagings):

IF SOURCE_FIELDS-ELEMT = '105'.
RESULT = SOURCE_FIELDS-WERTF + SOURCE_FIELDS-WERTV.

K_CCR_110 (110 Toll Manufacturing):

IF SOURCE_FIELDS-ELEMT = '110'.

RESULT = SOURCE_FIELDS-WERTF + SOURCE_FIELDS-WERTV.

K_CCR_120 (120 Utilities - VC):

IF SOURCE_FIELDS-ELEMT = '120'.
RESULT = SOURCE_FIELDS-WERTF + SOURCE_FIELDS-WERTV.

K_CCR_200 (200 Labor Costs):

IF SOURCE_FIELDS-ELEMT = '200'.
RESULT = SOURCE_FIELDS-WERTF + SOURCE_FIELDS-WERTV.

K_CCR_210 (210 Supplies & Misc.):

IF SOURCE_FIELDS-ELEMT = '210'.
RESULT = SOURCE_FIELDS-WERTF + SOURCE_FIELDS-WERTV.

K_CCR_220 (220 Maintenance):

IF SOURCE_FIELDS-ELEMT = '220'.
RESULT = SOURCE_FIELDS-WERTF + SOURCE_FIELDS-WERTV.

K_CCR_230 (230 Overheads):

IF SOURCE_FIELDS-ELEMT = '230'.
RESULT = SOURCE_FIELDS-WERTF + SOURCE_FIELDS-WERTV.

K_CCR_240 (240 Utilities -FC):

IF SOURCE_FIELDS-ELEMT = '240'.
RESULT = SOURCE_FIELDS-WERTF + SOURCE_FIELDS-WERTV.

K_CCR_300 (300 Depreciation):

IF SOURCE_FIELDS-ELEMT = '300'.
RESULT = SOURCE_FIELDS-WERTF + SOURCE_FIELDS-WERTV.

K_CCR_310 (310 N/A):

IF SOURCE_FIELDS-ELEMT = '310'.
RESULT = SOURCE_FIELDS-WERTF + SOURCE_FIELDS-WERTV.

End routine

The purpose of this routine is to check if the current incoming record already exists in the DSO_PCP3 from the previous month loading. If it exists, delete the incoming record.

DBCOPA01 DBCOPA24

Key figures in summation

Start routine

In internal tables we collect data from:

- master data c_custid where logsys and custid are the same as in source package, and plant is not empty itb_shipto

- master data c_plant itb_plant
- DSO UOMC_MAT itb_uomc_mat
- DSO dbcopa24 itb_dbcopa24
- Matser data c_matnr2 in function of logsys and c_matnr from source itb_matnr2
- Master data c_glbfilt in function of interval defined in master data C_GLBFILT (stream Z_INT_MG, rule VFIELD) itb_fieldn

/BIC/C_STREAM	/BIC/C_RU...	/BIC/C_GLBFILT	OBJVERS	CHANGED	/BIC/C_DESC	/BIC/C_SIGN	/BIC/C_OPTION	/BIC/C_LOW	/BIC/C_HIGH	/BIC/C_ACTIVE
Z_INT_MG	VFIELD	001	A		VALUE FIELD TO USE FOR DBCOPA24	I	EQ	QUANTITY		Y
Z_INT_MG	VFIELD	002	A		VALUE FIELD TO USE FOR DBCOPA24	I	BT	VVC30	VVC40	Y

- Master data c_glbfilt in function of interval defined in master data C_GLBFILT (stream Z_INT_MG, rule CALMONTH) itb_fieldn

/BIC/C_STREAM	/BIC/C_RU...	/BIC/C_GLBFILT	OBJVERS	CHANGED	/BIC/C_DESC	/BIC/C_SIGN	/BIC/C_OPTION	/BIC/C_LOW	/BIC/C_HIGH	/BIC/C_ACTIVE
Z_INT_MG	CALMONTH	001	A		CALMONTH TO USE FOR DBCOPA24	I	BT	202306	202307	Y

Then, if source of data are not in ITB_FIELDN or ITB_CALMONTH or ITB_SHIPTO we delete the data.

Field routines

0LOTSIZE_IT = 1000

CPFCTR1_2 = CPFCTR1_2 from master data C_MATPNT2 in function of /BIC/C_PLANT & LOGSYS & /BIC/C_MATPNT2 from source of data.

K_CSTUNIT = empty.

C_DSPLANT = c_plant from itb_shipto.

C_PLANT = c_plant from itb_shipto.

End routine

In function of records in result package. If a correspondance is found in ITB_MATNR2 and unit in ITB_UOMC_MAT:

k_lotsze = k_lotsze * ITB_UOMC_MAT-uomz1d (Quantity - Nominator) / ITB_UOMC_MAT-uomn1d (Quantity -Denominator).

unit = itb_matnr2-base_uom.

In function of records in result package. If a correspondance is found in ITB_PLANT and if the source currency is defferent from currency in ITB_PLANT:

Module function " Z_CONVERT_CURRENCY " is used to convert k_intdut (Integrated Duty unit cost) in plant currency.

K_INTDUT = SUM OF K_INDUT * 1000 / SUM OF LOT_SIZE

In function of records in result package (fields calmonth, logsys, c_dpplant, c_dsplant, c_matnr2) if a correcpondance is found in itb_dbcopa24, k_lotsze = 1000 else k_lotsze = 0.

(LOT_SIZE = 1000 ONLY IF KEY DOESN'T ALREADY EXIST)

In function of records in result package, if a correcpondance is NOT found in itb_matnr2, C_PFCTR1_2 is empty.

Delete data with C_PFCTR1_2 empty.

Delete data with k_intdut = 0.

ODS_PCP6

ODS_PCP6 is loaded by programs Z_M_INT and Z_M_INT_IP in process chain PC_INT_MARGIN.

Programs Z_M_INT & Z_M_INT_P

These program is used to calculate the IM key figures (CP, CNP, AMO, Duty and Freight).

To have more explanations about Z_M_INT (Z_M_INT_IP has same logic) with code associated, check this document:

ODS_PCP6 ODS_PCP4

End routine

If the unit is different from "KG", module function Z_MD_CONVERT_MATERIAL_UNIT_F is used to convert the k_lotsize in KG.

1.2 CPX system flow

In CPX, program ZBW_M_INT is a replication and adaptation (in name of tables for example) to program Z_M_INT in WBP.

We must pay attention to table TVARVC for Z_PERIOD_IM_HIST name: if LOW field is empty, the calculation of integrated margin will be done on current and current period - 1.

If the LOW field is filled (format MMMYYY), the chain will use this period to calculate the integrated margin (to be used in case of recalculation) (same rules inWBP system).

In DTP: ABCOMM06 (Vault) -> OH_COMM01 - Delta, we have a filter to load only data with WX Source of data (to load only data who come from WPX system).

To calculate the Integrated Margin in WPX, we need some informations for WBP. To do that, in WBP we splitted chain RSP_COPCP_MONTHLY in two:

One chain in WBP: RSP_COPCP_MONTHLY load data in WBP and trigger the chain in CPX (PC_CO_PC_MM_04).

The chain PC_CO_PC_MM_04 will load data from WPX and WBP (calculated in chain RSP_COPCP_MONTHLY) and calculate the integrated margin for Vault plant.

After that the data will be sent in WBP with help of webmethods and the chain PC_INT_MARGIN (WBP) will calculate the integrated margin with data from WBP and CPX.

blocked URL

Step 1: RSP_COPCP_MONTHLY (WBP)

Step 2: PC_CO_PC_MM_04 (CPX)

Step 3: PC_INT_MARGIN (WBP)

1.3 MVCOPA01 flow

Non ERP

DBCOPA09 -> DBCOPA27 (IM)

START ROUTINE

For a company list defined in the master data C_GLBFILT (rule = COMPCDE, stream = CO_NONERP).

/BIC/C_STREA...	/BIC/C_RULE	/BIC/C_GLBFILT	OBJVERS	CHANGED	/BIC/C_DESC	/BIC/C_SIGN	/BIC/C_OPTION	/BIC/C_LOW
CO_NONERP	COMPCDE	001	A		Company Code	I	EQ	000454
CO_NONERP	COMPCDE	002	A		Company Code	I	EQ	001430
CO_NONERP	COMPCDE	003	A		Company Code	I	EQ	077358

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_AMT_CNP) *-1**/ Lot Size**

C_AMT_CP, K_INTDUT, K_INTFRG and C_AMT_CNP come from ODS_PCP4.

These key figures are multiplied by 100 and converted in currency of the source package with module function 'Z_CONVERT_CURRENCY '.

If data with C_MGN_ACC different of 'R2549C' and 'R1540C' and 'R1291C' and 'R1290C', the data is deleted.

FIELD ROUTINE

C_PROD comes from master data C_MATNR2.

Version = 000

END ROUTINE

doc_date and pstng_date are determined with module function ' SLS_MISC_GET_LAST_DAY_OF_MONTH'

If company = '005720' AND calmonth < '201501', the c_compprs = '5720'.

Else the C_COMPPRS comes from master data C_COMPPRS in function of the company prs.

Else C_COMPPRS = 'NPRS'.

If c_pcompan = '005720' AND calmonth < '201501', the c_compprs = '5720'.

Else the C_COMPPRS comes from master data C_PCOMPAN in function of the partner company.

Else C_COMPPRS = 'NPRS'.

If c_magnitu is not empty. Check in master data C_MAGNITU if a record is found in function of source field C_MAGNITU. If yes, CPFCTR1_2 = CPFCTR1_2 from C_MAGNITU.

If c_pactiv is not empty. Check in master data C_MAGNITU if a record is found in function of source field C_PACTIV. If yes, C_INTGBU2 = CPFCTR1_2 from C_MAGNITU.

If c_zone & c_gbuzone are empty, they are filled with zone and country from master data C_COMPPRS (in function of c_compprs zone).

If c_gzone is empty, she is filled with c_gzone from c_country filled with country from master data COUNTRY (in function of c_compprs country).

Solvay (PF1)

DPCOPA23 IB_COPA_04

Field routine

Fiscal year / posting period / fiscal period are determined from calendar year month.

Currency Type = 10.

Value Type for Reporting = 010.

Version (C_VERSN2) = 000.

C_PCOMPAN (Partner Company is an attribte from master data C_SOLDID (Sold-to party)

C_DIVISN (Division) is an attribte from master data C_MATNR2 (Material)

IB_COPA_04 DBCOPA34

Expert routine

c_shipcom (Ship-to with Company Code) = c_shipid (Ship-to party)

C_WWE41 (Origin Plant) = C_ZZWWE41 (Origin Plant)

If C_PCOMPAN is not empty, C_TECHPAC (Technical Part Activity) = C_COMPCDE (company code).

Internal table itb_c_keyfigr contains fields c_keyfigr (Key Figure), c_plelmnt (PF1/P&L Element), c_ratioim (Ratio Type), c_bfsign (Sign) and c_unitio (Unit InfoObjet) from master data c_keyfigr where c_keyfigr is not empty AND c_intfkf is not empty '' OR c_plelmnt = 'ZN8110BQTY'.

A loop is done on this table:

- to populate c_plelmnt (PF1/P&L Element) + c_plelmn2 (P&L Element) and c_bfsign (sign) with c_plelmnt and c_bfsign from itb_c_keyfigr.
- If c_ratioim = 'X'

    AND c_bfsign = '-'

        amount = -1 * c_keyfigr else amount for DBCOPA34 = c_keyfigr.

    /bic/k_amntdc = 0

    /bic/k_amntndc = 0

    currency = c_unitio

    g_qvva01 (Qty invoice) is empty

- If c_ratioim = 'Q'

    amount = 0

    /bic/k_amntdc = 0

    /bic/k_amntndc = 0

    g_qvva01 (Qty invoice) = c_keyfigr

    g_uvva01 (Un Qty invoice) = c_unitio

- Else

    amount = 0

    k_amntdc = c_keyfigr (in positive sign)

    doc_currcy = c_unitio.

    g_qvva01 is empty.

IB_COPA_06 DBCOPA35

Start routine

Sort source package by c_pabelnr, c_paposnr, co_area, c_versn2, vtype, c_fieldn and record.

In internal table itb_hashed_pcp4 we collect the data from source package where calmonth, c_prodpla, c_matnrc and c_goodiss are not empty. We used c_goodiss + 01 or calmonth + 01 to create the costing date.

In w_lastmth we store the last month (the biggest) in source package.

In internal table itb_collect_pcp5 we collect the data from source package where pstng_date, c_prodpla and c_plant are not empty.

Select in internal table itb_c_keyfigr fields c_keyfigr, c_plelmnt, c_bfsign from master data c_keyfigr where c_keyfigr is not empty.

Select in internal table itb_plant data from master data c_plant in function of logsys in source and if C_SORT2 = 'NDIR'.

Select costdate, costvar, plant, c_matnr, base_uom, UNIT, CURRENCY, c_amt_cp, c_amt_cnp, c_amt_amo, k_intdut, k_intfrg, lotsize_it, c_amt_pck, c_amt_raw, c_amt_tol, c_amt_uti INTO TABLE itb_pcp4 in function of data present in itb_hashed_pcp4 and if c_kkzma <> 'X' and if costvers = '001'.

In internal table ibt_pcp4 we delete the record if costvar = ZIP. If cost_var = ZFO the record is copied in internal table itb_pcp4_zfo.

Select destplant, depplant, datefrom, dateto, currency, k_intdut, amount, lotsize, unit, into internal table itb_pcp5 in function of data present in itb_collec t_pcp5.

For each line of source package, **LER rules** are applied:

Determine w_costdate first day of copa delivery month (6 firsts numbers of c_goodiss + 01 or if c_goodiss is empty, 6 firsts numbers of pstng_date + 0 1).

Determine w_lcostdate last day of copa delivery month (with function module SLS_MISC_GET_LAST_DAY_OF_MONTH + field w_costdate).

Search key fidures from ods_pcp4 (itb_pcp4_zfo) with costvar 'ZFO' in function of source package data.

If no record found in ods_pcp4 (itb_pcp4) with zfo, try without 'ZFO' and 'ZIP'.

If something is found in ods_pcp4:

If the currency from ods_pcp4 is not the same of currency from source package:

We multply the key figures from ods_pcp4 (C_AMT_CP, C_AMT_CNP, C_AMT_AMO, K_INTFRG, K_INTDUT, C_AMT_RAW, C_AMT_PCK, C_AMT_TOL, C_AMT_UTI) by 100 to avoid decimals issues.

Used function module Z_CONVERT_CURRENCY with exchange rate CAR3 to convert each key figures (target currency = currency from source package).

We divide the key figures by 100.

Same principle for unit key figure (lotsize) with function module Z_BW_CONVERT_QTY.

If currencies or units are not different, we take it from ods_pcp4 without change.

Search key fidures from ods_pcp5 (itb_pcp5) in function of source package data.

If the currency from ods_pcp5 is not the same of currency from source package:

We multply the key figures from ods_pcp5 (k_intfrg and k_intdut) by 100 to avoid decimals issues.

Used function module Z_CONVERT_CURRENCY with exchange rate CAR3 to convert each key figures (target currency = currency from source package).

We divide the key figures by 100.

Same principle for unit key figure (lotsize) with function module Z_BW_CONVERT_QTY.

If currencies or units are not different, we take it from ods_pcp5 without change.

If C_AMT_RAW from ODS_PCP4 is not equal to 0, fields (from source package) c_fieldn, c_fieldn2 and c_plelmn2 = 'VVD0C'.

source package amount = -1 * (source package g_qvva01 * DSO_PCP4 w_c_amt_**raw**) / DSO_PCP4 w_lotsize

If C_AMT_PCK from ODS_PCP4 is not equal to 0, fields (from source package) c_fieldn, c_fieldn2 and c_plelmn2 = 'VPD0C'.

source package amount = -1 * (source package g_qvva01 * DSO_PCP4 w_c_amt_**pck**) / DSO_PCP4 w_lotsize

If C_AMT_UTI from ODS_PCP4 is not equal to 0, fields (from source package) c_fieldn, c_fieldn2 and c_plelmn2 = 'VUD0C'.

source package amount = -1 * (source package g_qvva01 * DSO_PCP4 w_c_amt_**uti**) / DSO_PCP4 w_lotsize

If K_INTFRG from ODS_PCP4 is not equal to 0, fields (from source package) c_fieldn, c_fieldn2 and c_plelmn2 = 'VVC2C'.

source package amount = -1 * (source package g_qvva01 * DSO_PCP4 **k_intfrg**) / DSO_PCP4 w_lotsize

If K_INTFRG from ODS_PCP5 is not equal to 0, fields (from source package) c_fieldn, c_fieldn2 and c_plelmn2 = 'ZZC2C'.

source package amount = -1 * (source package g_qvva01 * DSO_PCP5 **k_intfrg**) / DSO_PCP5 w_lotsize

If K_INTDUT from ODS_PCP4 is not equal to 0, fields (from source package) c_fieldn, c_fieldn2 and c_plelmn2 = 'VVC3C'.

source package amount = -1 * (source package g_qvva01 * DSO_PCP4 **k_intdut**) / DSO_PCP4 w_lotsize

If K_INTDUT from ODS_PCP5 is not equal to 0, fields (from source package) c_fieldn, c_fieldn2 and c_plelmn2 = 'ZZC3C'.

source package amount = -1 * (source package g_qvva01 * DSO_PCP5 **k_intdut**) / DSO_PCP5 w_lotsize

If C_AMT_CNP from ODS_PCP4 is not equal to 0, fields (from source package) c_fieldn, c_fieldn2 and c_plelmn2 = 'VVE0C'.

source package amount = -1 * (source package g_qvva01 * DSO_PCP4 **c amt_cp**) / DSO_PCP4 w_lotsize

If C_AMT_AMO from ODS_PCP4 is not equal to 0, fields (from source package) c_fieldn, c_fieldn2 and c_plelmn2 = 'VVF0C'.

source package amount = -1 * (source package g_qvva01 * DSO_PCP4 **c amt_amo**) / DSO_PCP4 w_lotsize

Add new records with new key figures in source package.

End routine

Field c_bfsign comes from master data c_keyfigr

DBCOPA35 -> DBCOPA41 (IM)

Expert routine

In internal table ITB_DBCOPA34 we select data from dso DBCOPA34 in function of package fields calmonth, c_matnr2 and c_compcde.

ITB_DBCOPA34 is sorted.

For each records in package, we search in ITB_DBCOPA34 a correspondance (stored in " S_COLLECT_SOLVAY").

If correspondance found, serach in ITB_DBCOPA34 same records but only if P&L Element equal WMAT, WDEPC, WFIXCOSTC or WUTILC) stored in " S_COLLECT_RCS".

If nothing is found, we duplicate the records with negative amount (S_COLLECT_SOLVAY-amount * -1).

If something is found but S_COLLECT_SOLVAY-amount is not equal to S_COLLECT_RCS-amount, we duplicate the record with negative amount (S_COLLECT_SOLVAY-amount * -1).

If package have same records in function of fields: RECORD, CALMONTH, C_ORGKEY2, C_MATKEY2, C_CUSKEY2, C_KEY01, C_KEY02, C_CURCCY, LOGSYS, AMOUNT, we keep only one records and delete the others.

DBCOPA35 -> DBCOPA34

Expert routine

In internal table ITB_DBCOPA34 we select data from dso DBCOPA34 in function of package fields calmonth, c_matnr2 and c_compcde.

ITB_DBCOPA34 is sorted.

For each records in package, we search in ITB_DBCOPA34 a correspondance (stored in " S_COLLECT_SOLVAY").

If correspondance found, serach in ITB_DBCOPA34 same records but only if P&L Element equal WMAT, WDEPC, WFIXCOSTC or WUTILC) stored in " S_COLLECT_RCS".

If nothing is found, we duplicate the records with negative amount (S_COLLECT_SOLVAY-amount * -1).

If something is found but S_COLLECT_SOLVAY-amount is not equal to S_COLLECT_RCS-amount, we duplicate the record with negative amount (S_COLLECT_SOLVAY-amount * -1).

If package have same records in function of fields: RECORD, CALMONTH, C_ORGKEY2, C_MATKEY2, C_CUSKEY2, C_KEY01, C_KEY02, C_CURCCY, LOGSYS, AMOUNT, we keep only one records and delete the others.

DBCOPA29 -> DBCOPA36

Field routine

C_VERSN2 = 000

0COUNTRY comes from master data C_SHIPID.

C_REST_BA (restated business area) comes from C_TECPCT2 (Technical Profit Center (compounded with c_busarea).

CPFCTR1_2 (gbu) comes from C_TECPCT2 (Technical Profit Center (compounded with c_busarea).

C_TECHPAC (Technical Part Activity) is equal to c_compcde (is pcompany is not empty).

End routine

Data is deleted if g_qvva04 (Qty unit base) = 0 AND if cpfctr1_2 (gbu) is NOT present in master data global filter (c_glbfilt) for stream = PL_ELEMENT AND rule = IM_IL_GBU.

/BIC/C_STREA...	/BIC/C_RULE	/BIC/C_GLBFILT	OBJVERS	CHANGED	/BIC/C_DESC	/BIC/C_SIGN	/BIC/C_OPTION	/BIC/C_LOW
PL_ELEME...	IM_IL_GBU	001	A		BFC GBU for ...	I	EQ	SD

c_fieldn = c_fieldn

c_fieldn2 = c_fieldn

c_plelmn2 = c_fieldn

K_PERCENT (P&L-IM-Assignment Percentage) comes from DSO dbcopa13.

If K_PERCENT is equal to 0, data is deleted.

amount = (g_qvva04 * k_percent) / 100.

g_qvva04 = 0.
g_qvva01 = 0.

Rhodia

DBCOPA03 -> IB_COPA_03

Before load data in DBCOPA20, DBCOPA21, DBCOPA22, DBCOPA23 transformation goes through infosource IB_COPA_03.

The idea is to have in one transformation TRSF: DBCOPA03 -> IB_COPA_03 same rules that will applied to the different targets.

TRFN: DBCOPA03 -> IB_COPA_03	0NEHEJ60ASPVRRLZ2LT1JAMNFW5FSJPB
Integrated Margin - Business rules	IB_COPA_03
TRFN: IB_COPA_03 -> DBCOPA18	0AMUKJDUXUQJFIO65G8OI8PAU490700C
IM - RCS - IECRA NO LER NO TP	DBCOPA18
TRFN: IB_COPA_03 -> DBCOPA19	0R4J44YT4RN9C07NQ5ESBKJ84Q52NXCZ
IM - RCS - IECRA NO LER TP	DBCOPA19
TRFN: IB_COPA_03 -> DBCOPA20	0N3QFYMSQHJ7YJ2TMZN61WU1NN2MR1M0
IM - RCS - IECRA LER	DBCOPA20
TRFN: IB_COPA_03 -> DBCOPA21	07JX9AWBTHNC4TRSORE09SLC6VWSA3IS
IM - RCS - CDSA NO LER NO TP	DBCOPA21
TRFN: IB_COPA_03 -> DBCOPA22	00SR0DL0P26X6IQIRABOPND9RLR8WV3O
IM - RCS - CDSA NO LER TP	DBCOPA22
TRFN: IB_COPA_03 -> DBCOPA23	0RJ1JWH0CGI691M0F7O16CCB5FURS3ZO
IM - RCS - CDSA LER	DBCOPA23

START ROUTINE

Internal table itb_glbfilt is created with values from master data global filter (c_glbfilt) for STREAM = 'LER'.

Data Browser: Table /BIC/PC_GLBFILT Select Entries 108

Table: /BIC/PC_GLBFILT

/BIC/C_STREA...	/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
LER	0001	001	A		COMPANY IS...	I	BT	200401	299912	N
LER	0005	001	A		COMPANY IS...	I	BT	200401	299912	N
LER	0125	001	A		COMPANY IS...	I	BT	200401	299912	N
LER	0128	001	A		COMPANY IS...	I	BT	200401	299912	N
LER	0134	001	A		COMPANY IS...	I	BT	200401	299912	N
LER	0143	001	A		COMPANY IS...	I	BT	200401	299912	N
LER	0192	001	A		COMPANY IS...	I	BT	200401	299912	N
LER	0210	001	A		COMPANY IS...	I	BT	200401	299912	N
LER	0212	001	A		COMPANY IS...	I	BT	200401	299912	N

Internal table itb_plant is created with values from master data Plant(c_plant) in function of logsys and where field C_SORT2 (Search Term 2) = 'NDIR'.

END ROUTINE

A module function ZDETERMINE_LER is used to determine C_LERFLAG (Flag for LER) in function of itb_plant and itb_glbfilt.

IB_COPA_03 DBCOPA20 & IB_COPA_03 DBCOPA23

START ROUTINE

Records are deleted when C_LERFLAG (Flag for LER) <> '3'.

Records are deleted when C_FIELDN (Value Field) <> 'QUANTITY' AND <> 'VVD00' AND <> 'VVE00' AND <> 'VVF00'.

Program Z_LER_RULES is executed.

Records are deleted when C_FIELDN (Value Field) = QUANTITY OR = 'VVD00' OR = 'VVE00' OR = 'VVF00'.

IB_COPA_03 DBCOPA18 & IB_COPA_03 DBCOPA21

START ROUTINE

Records are deleted when C_LERFLAG (Flag for LER) <> '1'.

Records are deleted when C_FIELDN (Value Field) <> 'QUANTITY' AND <> 'VVD00' AND <> 'VVE00' AND <> 'VVF00'.

Records are deleted when ITEM_CATEG (Sales document item category) = 'ZRES' OR = 'ZKES' OR = 'ZKBS' OR = 'ZKAS' OR = 'ZKRS' OR = 'TAPS'.

Program Z_LER_RULES is executed. (Used for Integrated Margin ? need to be explained).

Records are deleted when C_FIELDN (Value Field) = QUANTITY OR = 'VVD00' OR = 'VVE00' OR = 'VVF00'.

IB_COPA_03 DBCOPA19 & IB_COPA_03 DBCOPA22 & IB_COPA_07 -> ABCOPA02

START ROUTINE

Records are deleted when C_LERFLAG (Flag for LER) <> '2'.

Records are deleted when C_FIELDN (Value Field) <> 'QUANTITY' AND <> 'VVD00' AND <> 'VVE00' AND <> 'VVF00'.

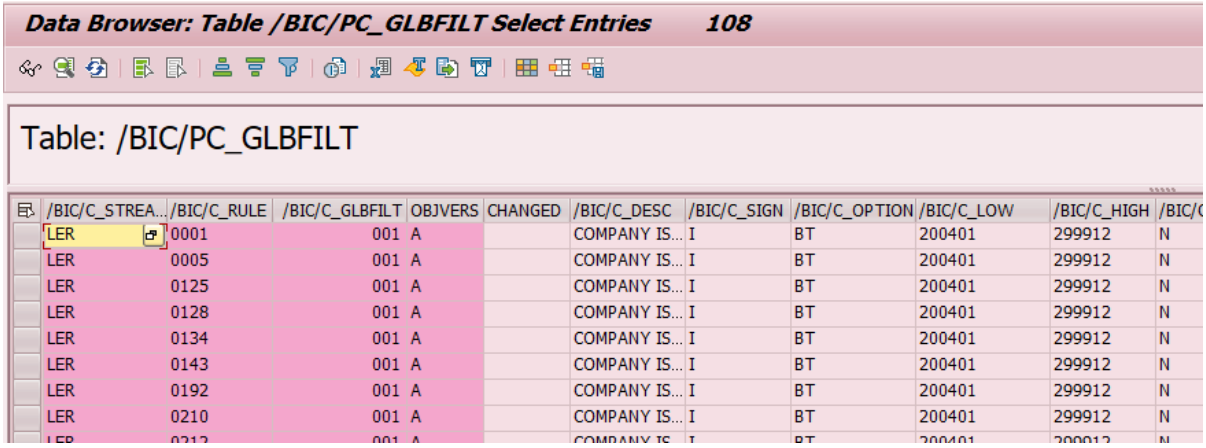
Program Z_LER_RULES is executed. (Used for Integrated Margin ? need to be explained).

Records are deleted when C_FIELDN (Value Field) = QUANTITY OR = 'VVD00' OR = 'VVE00' OR = 'VVF00'.

ABCOPA02 -> CRCOPA32 & DBCOPA22 -> CRCOPA14

Start routine

Internal table itb_glbfilt is created with values from master data global filter (c_glbfilt) for STREAM = 'LER'.



Data Browser: Table /BIC/PC_GLBFILT Select Entries 108

Table: /BIC/PC_GLBFILT

/BIC/C_STREA...	/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
LER	0001	001	A		COMPANY IS...	I	BT	200401	299912	N
LER	0005	001	A		COMPANY IS...	I	BT	200401	299912	N
LER	0125	001	A		COMPANY IS...	I	BT	200401	299912	N
LER	0128	001	A		COMPANY IS...	I	BT	200401	299912	N
LER	0134	001	A		COMPANY IS...	I	BT	200401	299912	N
LER	0143	001	A		COMPANY IS...	I	BT	200401	299912	N
LER	0192	001	A		COMPANY IS...	I	BT	200401	299912	N
LER	0210	001	A		COMPANY IS...	I	BT	200401	299912	N
LER	0212	001	A		COMPANY IS...	I	BT	200401	299912	N

Internal table itb_plant is created with values from master data Plant(c_plant) in function of logsys and where field C_SORT2 (Search Term 2) = 'NDIR'.

End routine

A module function ZDETERMINE_LER is used to determine C_LERFLAG (Flag for LER) in function of itb_plant and itb_glbfilt.

List of queries

Query technical name	Query Description
BW_QRY_MVCOPA01_0004	BW P&L Integrated Margin CM/GM Query
BW_QRY_MVCOPA01_0005	BW P&L Integrated Contribution Margin Query
BW_QRY_MVCOPA01_0007	BW P&L - Monthly Integrated Margin CM/GM Query
BW_QRY_MVCOPA01_0009	BW P&L - Monthly Integrated Margin CM/GM Query(Month Select)
QVSBS_BW_QRY_MVCOPA01_0004	QV - BW P&L & Integrated Margin Query (CAR3)
BW_QRY_MVCOPA01_0013	BW P&L Market Margin CM/GM Query
BW_QRY_DSO_PCP2_0001	Integrated Margin - Query control for Purchase Price
BW_QRY_ODS_PCP3_0001	CCR check at ODS level
BW_QRY_ODS_PCP4_0001	BW - Integrated Margin (Core Query)
BW_QRY_ODS_PCP5_0001	Freight Cost for Integrated Margins (Core Query)
RCS_QRY_ODS_PCP5_001	INM - Integrated Margin - Transport Cost Check