

# 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 OCO\_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 OCO\_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

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.

Technical rules not linked to Integrated Margin providers (not with IM in red in the data flows) are in [Technical Documentation - P&L](#) wiki

### 1.1 ODS\_PCP4 flow

#### 6DB\_GL\_COST\_PURCHASE DW2 DSO\_PCP2

##### Fields routines

There are 3 individual routines 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_GLBFLT	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-COSTELMNT, we take the COSTCOMP from gt\_matrix (ODS\_PCP2).

In all cases, if 0COSTCOMP from source or from ODS\_PCP2 is equal to '', the default 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\_GLBFLT).

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\_GLBFLT) for STREAM = F\_COPC and RULE = ODS\_PCP1

SSSS										
/BIC/C_STREAM	/BIC/C_RULE	/BIC/C_GLBFLT	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 assignments:

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 = ''.

## **OCO\_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-ELEMENT (Cost Component) = '100'.

RESULT = SOURCE\_FIELDS-WERTF (Fixed value) + SOURCE\_FIELDS-WERTV (Variable value).

K\_CCR\_110 (110 Toll Manufacturing):

IF SOURCE\_FIELDS-ELEMENT = '110'.

RESULT = SOURCE\_FIELDS-WERTF + SOURCE\_FIELDS-WERTV.

K\_CCR\_120 (120 Utilities - VC):

IF SOURCE\_FIELDS-ELEMENT = '120'.

RESULT = SOURCE\_FIELDS-WERTF + SOURCE\_FIELDS-WERTV.

K\_CCR\_200 (200 Labor Costs):

IF SOURCE\_FIELDS-ELEMENT = '200'.

RESULT = SOURCE\_FIELDS-WERTF + SOURCE\_FIELDS-WERTV.

K\_CCR\_210 (210 Supplies & Misc.):

IF SOURCE\_FIELDS-ELEMENT = '210'.

RESULT = SOURCE\_FIELDS-WERTF + SOURCE\_FIELDS-WERTV.

K\_CCR\_220 (220 Maintenance):

IF SOURCE\_FIELDS-ELEMENT = '220'.

RESULT = SOURCE\_FIELDS-WERTF + SOURCE\_FIELDS-WERTV.

K\_CCR\_230 (230 Overheads):

IF SOURCE\_FIELDS-ELEMENT = '230'.

RESULT = SOURCE\_FIELDS-WERTF + SOURCE\_FIELDS-WERTV.

K\_CCR\_240 (240 Utilities -FC):

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

K\_CCR\_300 (300 Depreciation):

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

K\_CCR\_310 (310 N/A):

```
IF SOURCE_FIELDS-ELEMENT = '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.
```

## **OCO\_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-ELEMENT = '100'.

RESULT = SOURCE\_FIELDS-WERTF (Fixed value) + SOURCE\_FIELDS-WERTV (Variable value).

K\_CCR\_105 (105 Packagings):

IF SOURCE\_FIELDS-ELEMENT = '105'.

RESULT = SOURCE\_FIELDS-WERTF + SOURCE\_FIELDS-WERTV.

K\_CCR\_110 (110 Toll Manufacturing):

IF SOURCE\_FIELDS-ELEMENT = '110'.

RESULT = SOURCE\_FIELDS-WERTF + SOURCE\_FIELDS-WERTV.

K\_CCR\_120 (120 Utilities - VC):

IF SOURCE\_FIELDS-ELEMENT = '120'.

RESULT = SOURCE\_FIELDS-WERTF + SOURCE\_FIELDS-WERTV.

K\_CCR\_200 (200 Labor Costs):

IF SOURCE\_FIELDS-ELEMENT = '200'.

RESULT = SOURCE\_FIELDS-WERTF + SOURCE\_FIELDS-WERTV.

K\_CCR\_210 (210 Supplies & Misc.):

IF SOURCE\_FIELDS-ELEMENT = '210'.

RESULT = SOURCE\_FIELDS-WERTF + SOURCE\_FIELDS-WERTV.

K\_CCR\_220 (220 Maintenance):

IF SOURCE\_FIELDS-ELEMENT = '220'.

RESULT = SOURCE\_FIELDS-WERTF + SOURCE\_FIELDS-WERTV.

K\_CCR\_230 (230 Overheads):

IF SOURCE\_FIELDS-ELEMENT = '230'.

RESULT = SOURCE\_FIELDS-WERTF + SOURCE\_FIELDS-WERTV.

K\_CCR\_240 (240 Utilities -FC):

IF SOURCE\_FIELDS-ELEMENT = '240'.

RESULT = SOURCE\_FIELDS-WERTF + SOURCE\_FIELDS-WERTV.

K\_CCR\_300 (300 Depreciation):

IF SOURCE\_FIELDS-ELEMENT = '300'.

RESULT = SOURCE\_FIELDS-WERTF + SOURCE\_FIELDS-WERTV.

K\_CCR\_310 (310 N/A):

IF SOURCE\_FIELDS-ELEMENT = '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
- Master 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\_GLBFLT (stream Z\_INT\_MG, rule VFIELD) itb\_fieldn

/BIC/C_STREAM	/BIC/C_RU..	/BIC/C_GLBFLT	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\_GLBFLT (stream Z\_INT\_MG, rule CALMONTH) itb\_fieldn

/BIC/C_STREAM	/BIC/C_RU..	/BIC/C_GLBFLT	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 correpondance 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 correpondance 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 in WBP 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\_GLBFLT (rule = COMPCDE, stream = CO\_NONERP).

/BIC/C_STREA...	/BIC/C_RULE	/BIC/C_GLBFLT	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:

- o  $C\_MGN\_ACC = R1540C / 0AMOUNT = Qty VV04 * Integrated\ variable\ Cost\ Unit (C\_AMT\_CP) * -1 / Lot\ Size$
- o  $C\_MGN\_ACC = R1290C / 0AMOUNT = Qty VV04 * Integrated\ Duty\ costs (K\_INTDUT) * -1 / Lot\ Size$
- o  $C\_MGN\_ACC = R1291C / 0AMOUNT = Qty VV04 * KeyFigure\ Integrated\ Transport\ Cost K\_INTFRG * -1 / Lot\ Size$
- o  $C\_MGN\_ACC = R2549C / 0AMOUNT = Qty VV04 * Integrated\ CNP\ part (C\_AMT\_CNP) * -1 / Lot\ Size$

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 attribute from master data C\_SOLDID (Sold-to party)

C\_DIVISN (Division) is an attribute 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\_plelmt (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\_plelmt = 'ZN8110BQTY'.

A loop is done on this table:

- to populate c\_plelmt ( PF1/P&L Element ) + c\_plelmt2 (P&L Element) and c\_bfsign (sign) with c\_plelmt 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\_plelmt, 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\_collect\_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 + 01).

Determine w\_lcostdate last day of copa delivery month (with function module SLS\_MISC\_GET\_LAST\_DAY\_OF\_MONTH + field w\_costdate).

Search key figures 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 multiply 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 figures 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 multiply 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 =  $-1 * (\text{source package g\_qvva01} * \text{DSO\_PCP4 w\_c\_amt\_raw}) / \text{DSO\_PCP4 w\_lotsize}$

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 =  $-1 * (\text{source package g\_qvva01} * \text{DSO\_PCP4 w\_c\_amt\_pck}) / \text{DSO\_PCP4 w\_lotsize}$

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 =  $-1 * (\text{source package g\_qvva01} * \text{DSO\_PCP4 w\_c\_amt\_uti}) / \text{DSO\_PCP4 w\_lotsize}$

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 =  $-1 * (\text{source package g\_qvva01} * \text{DSO\_PCP4 k\_intfrg}) / \text{DSO\_PCP4 w\_lotsize}$

If K\_INTFRG from ODS\_PCP5 is not equal to 0, fields (from source package) c\_fieldn, c\_fieldn2 and c\_plelmn2 = 'ZCC2C':

source package amount =  $-1 * (\text{source package g\_qvva01} * \text{DSO\_PCP5 k\_intfrg}) / \text{DSO\_PCP5 w\_lotsize}$

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 =  $-1 * (\text{source package g\_qvva01} * \text{DSO\_PCP4 k\_intdut}) / \text{DSO\_PCP4 w\_lotsize}$

If K\_INTDUT from ODS\_PCP5 is not equal to 0, fields (from source package) c\_fieldn, c\_fieldn2 and c\_plelmn2 = 'ZCC3C':

source package amount =  $-1 * (\text{source package g\_qvva01} * \text{DSO\_PCP5 k\_intdut}) / \text{DSO\_PCP5 w\_lotsize}$

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 =  $-1 * (\text{source package g\_qvva01} * \text{DSO\_PCP4 c\_amt\_cp}) / \text{DSO\_PCP4 w\_lotsize}$

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 =  $-1 * (\text{source package g\_qvva01} * \text{DSO\_PCP4 c\_amt\_amo}) / \text{DSO\_PCP4 w\_lotsize}$

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

OCOUNTRY 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.

EX	/BIC/C_STREA...	/BIC/C_RULE	/BIC/C_GLBFLT	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	ONEHEJ60ASPVRRLZ2LT1JAMNFW5FSJPB
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\_GLBFLT Select Entries 108**

Table: /BIC/PC\_GLBFLT

	/BIC/C_STREA...	/BIC/C_RULE	/BIC/C_GLBFLT	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'.

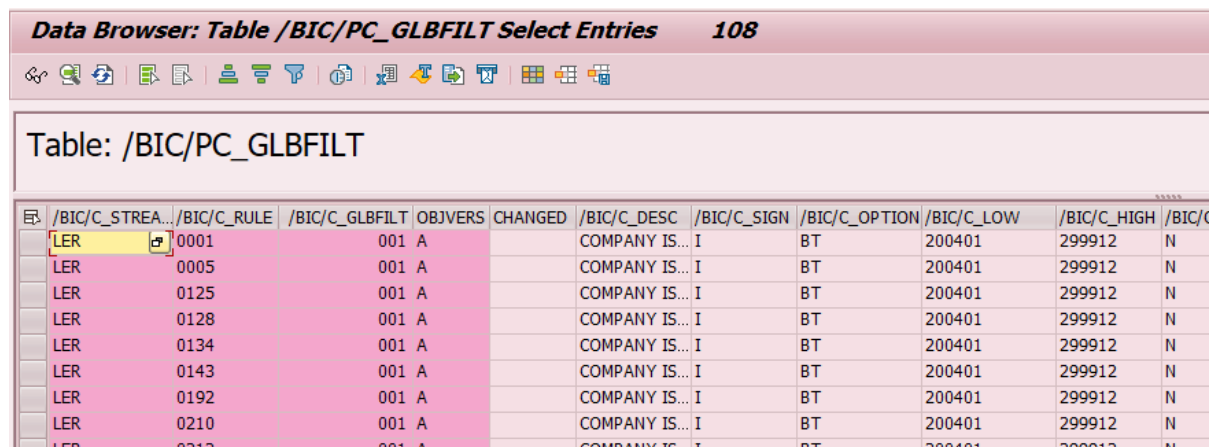
Program Z\_LER\_RULES is executed. (Used for Integrated Margin ? need to be explained).

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

### ABCOPA02 -> CRCOPA32 & DBCOPA22 -> CRCOPA14

#### Start routine

Internal table itb\_glbfilter is created with values from master data global filter ( c\_glbfilter ) for STREAM = 'LER'.



**Data Browser: Table /BIC/PC\_GLBFILTER Select Entries 108**

	/BIC/C_STREA...	/BIC/C_RULE	/BIC/C_GLBFLT	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\_glbfilter.

## List of queries

<b>Query technical name</b>	<b>Query Description</b>
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