

BW Integrated Margin (WBP)

- [Objectives](#)
- [Infoarea & Infoprovider](#)
- [DataFlow Overview](#)
- [Integrated Margin for EP & PI](#)
 - [Data Update Conditions for DSO_PCP1 from DTS_KEKO_CKIS](#)
 - [Data Update Conditions for DSO_PCP3 from 0CO_PC_PCP_02](#)
 - [Data Update Conditions for DSO_PCP2 from 6DB_GL_COST_PURCHASE](#)
 - [Correction of IM costs - flat file dataflow \(ODS_PCP4 from DTS_COPC_02\)](#)
- [Read more information in : <https://drive.google.com/drive/folders/1GlrwkOgG51yYFJYIUVRbBVLGPZV0XrUU>](#)
- [Update with project 8612 - Solstice \(CPX\)](#)

Objectives

This page contains all the documentations of subjects related to Integrated Margin

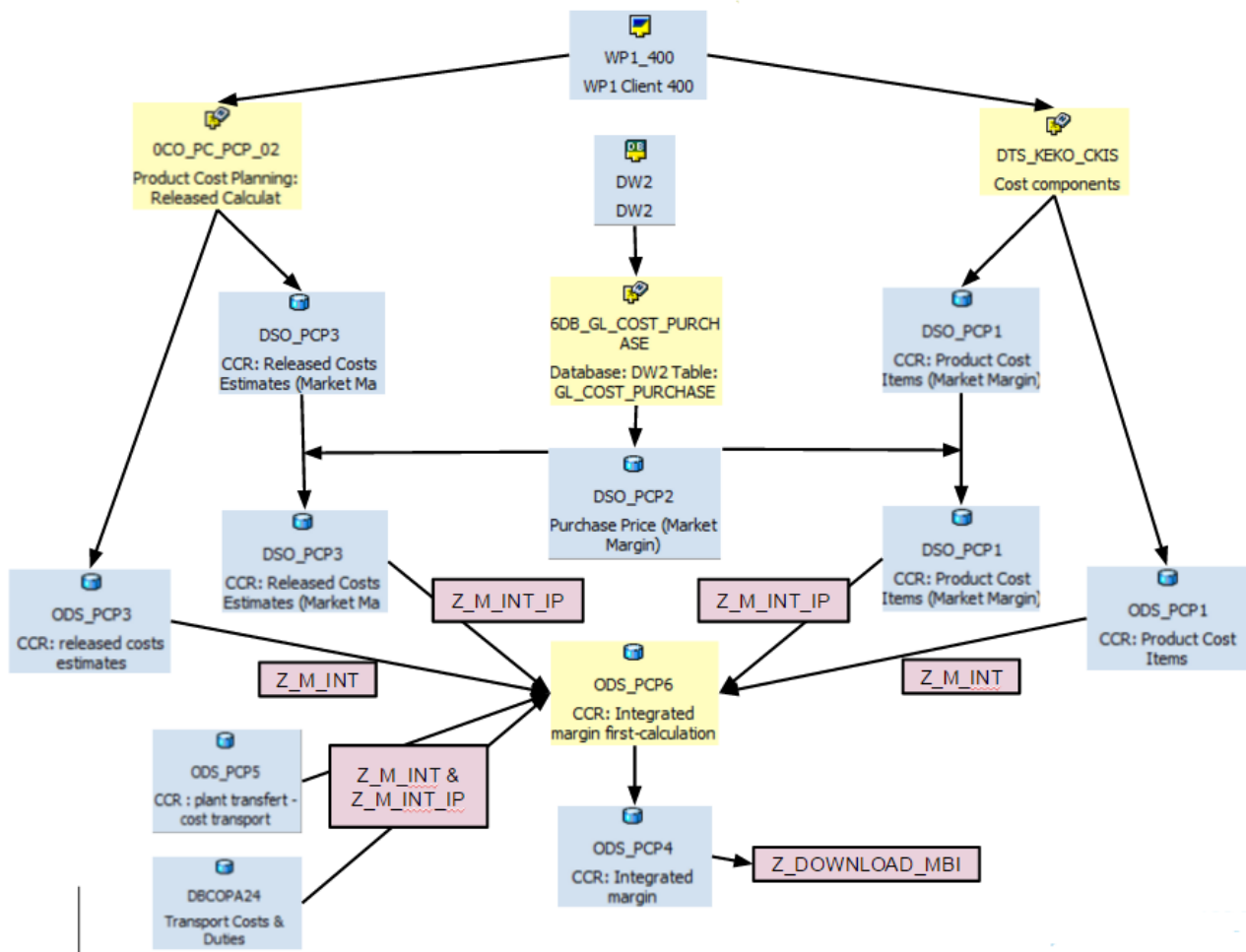
Infoarea & Infoprovider

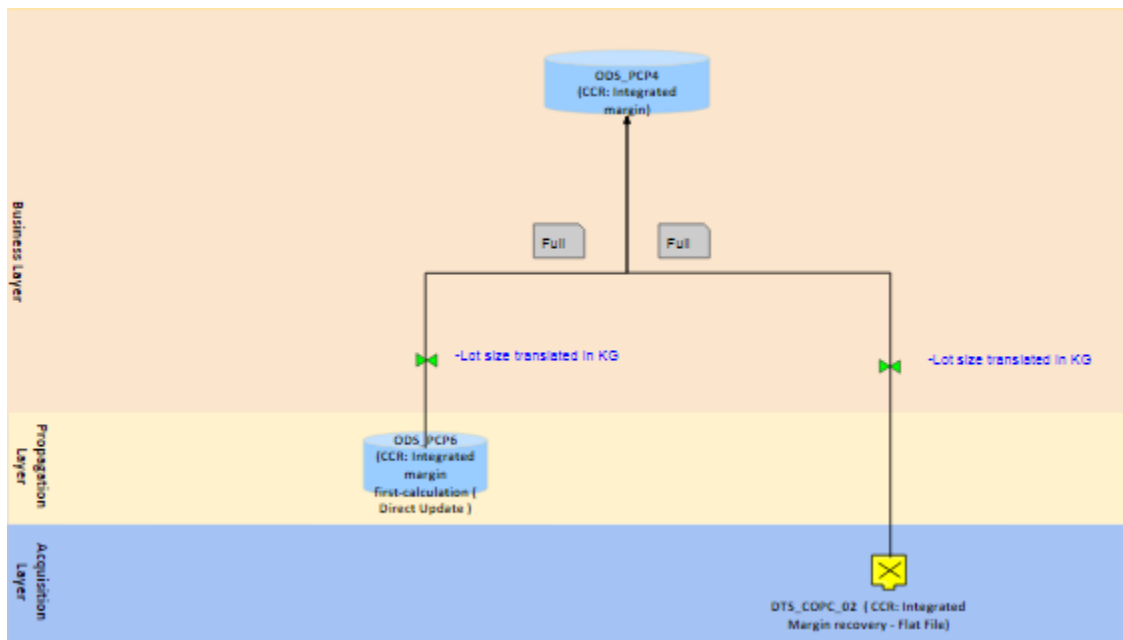
Infoarea Name	Infoarea Techname
Product Cost Controlling	AREA_F_COPC
ODS: Product Cost Controlling	AREA_F_COPC_ODS
CCR: Market Margin	AREA_F_COPC_MM

Infoprovider Name	Infoprovider Techname
Material Units	IST_PCMA
CCR: Cost elements matrix	ODS_PCP2
CCR: released costs estimates	ODS_PCP3
CO-PC: list of materials of utilities (10)	ODS_PC1A
CO-PC: list of materials of utilities (30)	ODS_PC2A
CO-PC: order header quantity (10)	ODS_PC1B
CO-PC: order header quantity (30)	ODS_PC2B
CCR : Product Cost Planning - All Cost Estimates	CUB_PCP01
CCR Exploded Itemization	CUB_PCP02
CO-PC: Manufacturing Costs per Process Order (10)	CUB_PC001
CO-PC: Manufacturing Costs per Process Order (30)	CUB_PC002
CO-PC: Manufacturing Costs per Process Order	MPR_PC001
Stock valorisation by costing	MPR_ICPC1
CCR : plant transfert - cost transport	ODS_PCP5
Transport Costs & Duties	DBCOPA24
CCR: Integrated margin	ODS_PCP4
CCR: Integrated margin first-calculation (Direct Update)	ODS_PCP6
CCR: Product Cost Items	ODS_PCP1
Cost Components for Cost of Goods Mfd	ODS_PCP7
Resource's Unit Price	ODS_PCP8
CCR: Product Cost Items (Market Margin)	DSO_PCP1

Purchase Price (Market Margin)	DSO_PCP2
CCR: Released Costs Estimates (Market Margin)	DSO_PCP3

DataFlow Overview





https://drive.google.com/file/d/1fNTKdo_1ePAFojr5e6mYP53za3rDy6r4tQTOtVEh-2w/view

Integrated Margin for EP & PI

Core Query Name	Core Query Techname	Infoprovider Name	Infoprovider Techname
BW - Integrated Margin (Core Query)	BW_QRY_ODS_PCP4_0001	CCR: Integrated margin	ODS_PCP4
BW - Integrated Margin (Core Query)	BW_QRY_ODS_PCP4_2	CCR: Integrated margin first-calculation	ODS_PCP6

Infoprovider Name	Infoprovider Techname	Source System	Source Infoprovider Name	Source Infoprovider Techname
CCR: Integrated margin	ODS_PCP4	WBP_400	CCR: Integrated margin first-calculation (Direct Update)	ODS_PCP6
CCR: Integrated margin first-calculation (Direct Update)	ODS_PCP6	WBP_400	Updated by ABAP Program	1) Z_M_INT 2) Z_M_INT_IP
CCR: Product Cost Items (Market Margin)	DSO_PCP1	WP1_400 WBP_400	Cost components CCR: Product Cost Items (Market Margin)	DTS_KEKO_CKIS DSO_PCP1
CCR: Released Costs Estimates (Market Margin)	DSO_PCP3	WP1_400 WBP_400	Product Cost Planning: Released Calculations CCR: released costs estimates (Markte Margin)	0CO_PC_PCP_02 DSO_PCP3
Purchase Price (Market Margin)	DSO_PCP2	DW2	Database: DW2 Table: GL_COST_PURCHASE	6DB_GL_COST_PURCHASE

Data Update Conditions for DSO_PCP1 from DTS_KEKO_CKIS

Data Source: DTS_KEKO_CKIS

Source System: WP1_400

Data Flow Version: BI 7.0

Transformation: RSDS DTS_KEKO_CKIS WP1_400 -> ODSO DSO_PCP1 (0K1O4SUXOC9DPNTS7R3JSO0DRSOF0BY9)

Start Routine: This start routine is to do 2 main tasks.

- 1) It is to delete all records where the amount and quantity is equal to zero.
- 2) It is to delete all records where the costing variants are not equal to ZFO, Z25 and ZEP.

After the start routine, each individual record will have fields that will be set with constant values or formulas.

Individual Constants:

- 1) Costing Variant is overwritten with the new costing variant value ZIP that is defined for this project.

2) Fiscal Variant is fixed to K4.

3) Cost Component Structure is fixed to Z1.

Individual Routines:

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

2) For the key figure Variable Amount, a formula is required to take the total amount minus the fixed amount to get just the variable amount.

Data Update Conditions for DSO_PCP3 from 0CO_PC_PCP_02

Data Source: 0CO_PC_PCP_02

Source System: WP1_400

Data Flow Version: BI 7.0

Transformation: RSDS 0CO_PC_PCP_02 WP1_400 -> ODSO DSO_PCP3 (0CY1UWMKPVWXUI74VIAPF3V2S12PIYQ1)

Start Routine: This start routine is to do 2 main tasks.

1) It is to delete all records where the currency type is not equal to Company Currency (Value = 10).

2) It is to delete all records where the costing variants are not equal to ZFO, Z25 and ZEP.

After the start routine, each individual record will have fields that will be set with constant values or formulas.

Individual Constants:

1) Costing Variant is overwritten with the new costing variant value ZIP that is defined for this project.

Individual Routines:

1) For the key figure 0AMOUNT, we add the Fixed Amount (WERTF) and Variable Amount (WERTV) together.

2) For the key figure K_CCR_100 (100 Raw Materials), we add the Fixed Amount (WERTF) and Variable Amount (WERTV) together if the Cost Component (ELEMNT) equals to 100.

3) For the key figure K_CCR_105 (105 Packaging), we add the Fixed Amount (WERTF) and Variable Amount (WERTV) together if the Cost Component (ELEMNT) equals to 105.

4) For the key figure K_CCR_110 (110 Toll Manufacturing), we add the Fixed Amount (WERTF) and Variable Amount (WERTV) together if the Cost Component (ELEMNT) equals to 110.

5) For the key figure K_CCR_120 (120 Utilities - VC), we add the Fixed Amount (WERTF) and Variable Amount (WERTV) together if the Cost Component (ELEMNT) equals to 120.

6) For the key figure K_CCR_200 (200 Labor Costs), we add the Fixed Amount (WERTF) and Variable Amount (WERTV) together if the Cost Component (ELEMNT) equals to 200.

7) For the key figure K_CCR_210 (210 Supplies & Misc.), we add the Fixed Amount (WERTF) and Variable Amount (WERTV) together if the Cost Component (ELEMNT) equals to 210.

8) For the key figure K_CCR_220 (220 Maintenance), we add the Fixed Amount (WERTF) and Variable Amount (WERTV) together if the Cost Component (ELEMNT) equals to 220.

9) For the key figure K_CCR_230 (230 Overheads), we add the Fixed Amount (WERTF) and Variable Amount (WERTV) together if the Cost Component (ELEMNT) equals to 230.

10) For the key figure K_CCR_240 (240 Utilities -FC), we add the Fixed Amount (WERTF) and Variable Amount (WERTV) together if the Cost Component (ELEMNT) equals to 240.

11) For the key figure K_CCR_300 (300 Depreciation), we add the Fixed Amount (WERTF) and Variable Amount (WERTV) together if the Cost Component (ELEMNT) equals to 300.

12) For the key figure K_CCR_310 (310 N/A), we add the Fixed Amount (WERTF) and Variable Amount (WERTV) together if the Cost Component (ELEMNT) equals to 310.

End Routine:

1) 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.

Data Update Conditions for DSO_PCP2 from 6DB_GL_COST_PURCHASE

Data Source: 6DB_GL_COST_PURCHASE

Source System: DW2

Data Flow Version: BI 7.0

Transformation: RSDS 6DB_GL_COST_PURCHASE DW2 -> ODSO DSO_PCP2 (07G0MUNSUYM2GSSZ09F9Q6CC9OF3LCOZ)

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

Correction of IM costs - flat file dataflow (ODS_PCP4 from DTS_COPC_02)

This dataflow is used in order to correct directly the costs in ODS_PCP4.

DTS_COPC_02 extract the csv file from BW server, under the following logical path: Z_IM_CORRECTION (/exploit/BW/Integrated_Margin/IM costs correction.csv))

Due to recent archive process the file in the path Z_IM_CORRECTION (/exploit/BW/Integrated_Margin/IM costs correction.csv)) is archived.

We asked Basis team to restore the file and they created new folder and placed the files under the folder.

Folder /exploit/BW/DoNotDelete/

Directory: /exploit/BW/DoNotDelete

Usable	Viewed	Changed	Length	Owner	Lastchange	Lastchange	File Name
			53	root	08.03.2023	17:04:46	.
			4096	exploit	07.03.2023	15:39:48	..
X			12964	exploit	04.09.2021	02:46:22	IM_costs_correction.csv
X			180	exploit	04.09.2021	02:44:49	MARM.csv

we have changed the logical path Z_IM_CORRECTION to the new location /BW/exploit/DoNotDelete/IM_costs_correction.csv

Scheduler (Maintain InfoPackage)

InfoPackage: IP: DTS_COPC_02 - Full - From App Server(ZPAK_SDZ2G9JBSF017G93GGFP7GZL6)

DataSource: CCR: Integrated Margin recovery(DTS_COPC_02)

Data Type: Transaction Dat

Source System: PC Files(PC_FILE)

Last Changed by: FF_BWPCC1_T2 Date: 09.03.2023 Time: 13:59:09

Data Selection Extraction Processing Data Targets Update Schedule

Adapter: Load Text-Type File from Application Server

File Name: /exploit/BW/DoNotDelete/IM_costs_correction.csv

Header Rows to be Ignored: 1

Character Set Settings: Default Setting

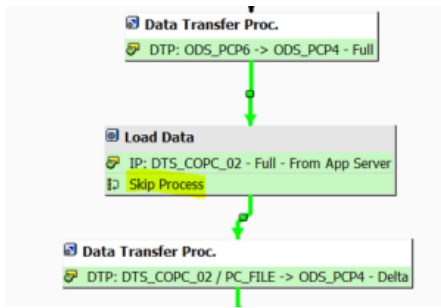
System Codepage: 4103 UTF-16LE Unicode / ISO/IEC 10646

Data Format: Separated with Separator (for Example, CSV)

Data Separator: ; Hex

Escape Sign: " Hex

We have added the skip step for this one as it not required to load every month (will use it when required)



We created the same for WBD, now both WBP and WBD are aligned.

Here is an exemple of file to upload: https://drive.google.com/drive/folders/1XCCCDxe6eWyeV3SR_vM_ZYgeYgwUhzt

The transformation into ODS_PCP4 is similar to the one from ODS_PCP6: it is direct mapping except the conversion of the lot size in Kilogram.

This file is loaded into ODS_PCP4 in the Process Chain after the loading from ODS_PCP6 so that the correction is not cleared after an historical recovery.

Read more information in :

<https://drive.google.com/drive/folders/1GIrwkOgG51yYFJYIUVRbBVLGPZV0XrUU>

Update with project 8612 - Solstice (CPX)

[BW - CO - Integrated Margin](#)

In scope of project [8612 - Solstice](#) we put in place a link between WBP and CPX system to take into account the vault data in calculation of Integrated Margin.

To do that, we have three steps:

- 1 - Transfert source data from WBP to CPX.
- 2 - Calculation of IM in CPX side with data from WBP and WPX.
- 3 - Transfert the result of IM in CPX to WBP and take it into account in the calculation of IM in WBP.

Details:

To put in place these three steps we had to split the chain RSP_COPCP_MONTHLY in two chains:

RSP_COPCP_MONTHLY to load the source of data and PC_INT_MARGIN to [retrieve result of IM from CPX](#) (in dso ABDSOPCP1) and calcul the IM in WBP side.

More details about the split here:

<https://drive.google.com/file/d/12tcCmr7A5nzHRrX8DNPFh4sDTdy1r73H9JePMjU4b8I/view#>

1 - transfert from WBP to CPX:

We use datasource from WBP dso to load data in CPX side.

At the end of process chain **RSP_COPCP_MONTHLY** there is a process "Remote process chain" to launch process chain PC_CO_PC_MM_04 in CPX.

2 - calculation of IM in CPX:

The principle is similar to what we done in WBP. We load data from WPX and WPB in ABDSOPCP1 and after we execute program ZBW_M_INT to calculate the Integrated Margin. All is done in meta chain **PC_CO_PC_MM_04** (CPX).

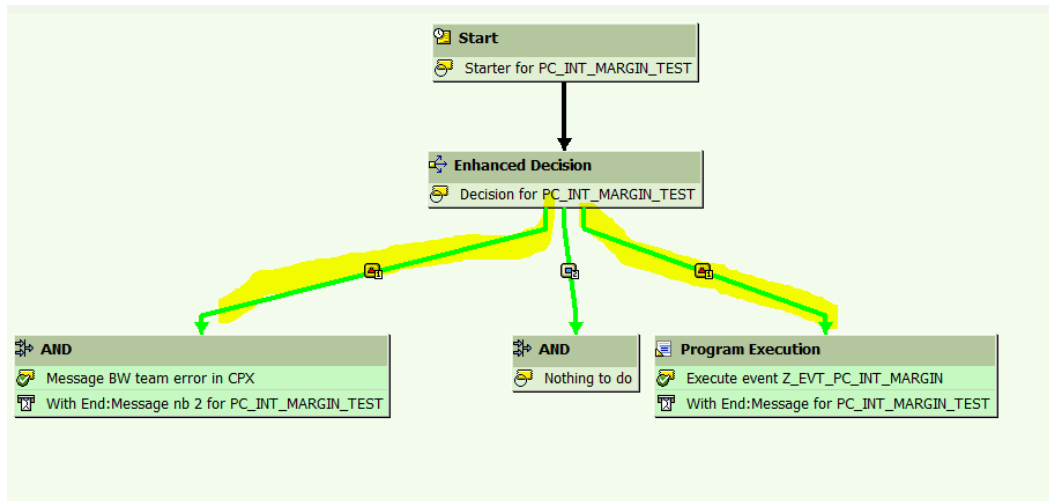
ZBW_M_INT in CPX is a copy Z_M_INT in WBP, so if there is a change in Z_M_INT it may be helpful to make the same change in ZBW_M_INT.

3 - Transfert the result of IM in CPX to WBP and take it into account in the calculation of IM in WBP:

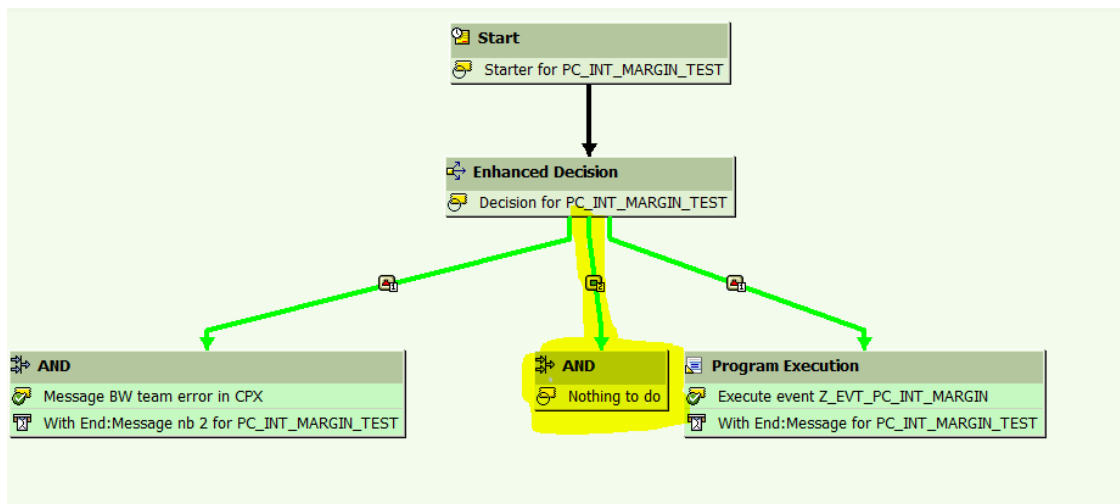
After the calculation of IM in CPX, a webmethods will sent a flat file with results (/exploit/BW/Integrated_Margin/OH_COMM01.CSV) in WBP and execute the chain **PC_INT_MARGIN** to load dso DSO ABDSOPCP1 with data from CPX, and after load this in ODS_PCP1 and do the calculation of IM.

If an error occurs in CPX system or with WebMethods, there is a process chain **PC_INT_MARGIN_TEST (Integrated Margin - test if Integrated Margin 2 is scheduled)** which runs every D+3 at 1 a.m. to check if PC_INT_MARGIN is running or not:

If not, we can suppose there was a problem and we force the execution of process chain PC_INT_MARGIN.



If yes, there is no problem, it's not necessary to force the execution of PC_INT_MARGIN.



The calculation of IM has same principle than before, there are two things that change with CPX data.

- Webmethods load data from CPX with a flat file and this flat file is loaded in DSO AADSOPCP1 APDSOPCP1 ABDSOPCP1 and ABDSOPCP 2 for historical save.

Data from DSO ABDSOPCP1 are loaded in ODS_PCP1, like that we have the source from WBP (WP1) and CPX (WPX) to calculate the IM.

Between DSO APDSOPCP1 and ABDSOPCP1 we transpose the data to adapt the source to the next target (ODS_PCP1). Some values of ABDSOPCP1 are defined in master dat global filter (C_GLBFLT) on stream "DSO_PCP1"

- In program Z_M_INT we calcul the Integrated Duty and the Integrated Freight with data from CPX (assigned in cost component 400 and 500 in loading of DSO ABDSOPCP1) (this calculation is not done in CPX side with prorgam ZBW_M_INT).

ABAP Editor: Display Report Z_M_INT

[←](#)
[→](#)

[Pattern](#)
[Pretty Printer](#)
[Text Elements](#)

Report Z_M_INT Active

Report Active

Report Z_M_INT Active

```
931         ELSEIF itb_temps-costcomp(1) = '4'.
932             str_data-bic/k indut = str_data-bic/k indut +
```

```
932 str_data=/bic/k_intaut - str_data=/bic/k_intaut +
933 itb temps-amountvr.
```

934

```
935 wa_result_sim-/bic/k_intdut = itb_temps-amounttvr + itb_temps-amountfx .
936 wa_result_sim-k_intdut parent = itb_temps-amounttvr + itb_temps-amountfx
```

```

936         wa_result_sim_k_intduc_parent = itb_temps-amountvi + itb_temps-amountix .
937     ELSEIF itb_temps-costcomp(1) = '5'.
938         str_data-/BIC/K_INTFRG = str_data-/BIC/K_INTFRG +

```

```

939                                     itb_temps-amountvr.

```

```
940 wa_result_sim-/BIC/K_INTERG = itb_temps-amountvr + itb_temps-amountfx.
```

```

941 wa_result_sim-k_initkg_parent = tcb_cemps-amountcv1 + tcb_cemps-amountcv1x ;
942 * MGD 14/06/2019 End

```

```
943         ENDIF.
```

0.44	EMITE	WITH 246 LINES 12 LINES 100
------	-------	-----------------------------