Transaction Integrated Margin - Z_INT_COST

Access Management

Roles & Access

List of application role + menu role and explanation if we have several applications role with specials rules.

Role Code	Role Description	Explanation
ZR_RCS_CA_M122	P&L - Integrated Cost	Access to transaction Z_INT_COST
ZBI_RCS_PS_A02	Project Costs and Structures Analysis - End User role	Application Role: ○ End User rights ○ gives access to infoproviders of Infoarea "AREA_PSCO" ○ BI Analysis Authorization "ZBI_PS" ("*" for authorization objects not relevant for the application)
ZBI_RCS_CO_A21	CBS Project costs Reporting Analysis - End User role	Application Role: ○ End User rights ○ gives access to infoproviders of Infoarea "AREA_F_CO_CBS_PS" ○ Gives access to multiprovider:MPR_PS014 ○ BI Analysis Authorization "ZBI_CBS_PC"

Authorization Objects

List of authorization objects mandatory for the application.

Authorization object	Explanation
CPFCTR1_2	Read GBU CPFCTR1_2 from C_MATPNT2 with MATERIAL and PLANT
C_AUTHMA	Read Scope C_AUTHMA from C_MATPNT2 with MATERIAL and PLANT

General presentation

Objective of the application

This transaction is used to displau the detail of structure of product cost.

Tool Leader + IT leader of the application: Gregory Rigal et Ludovic Depoix

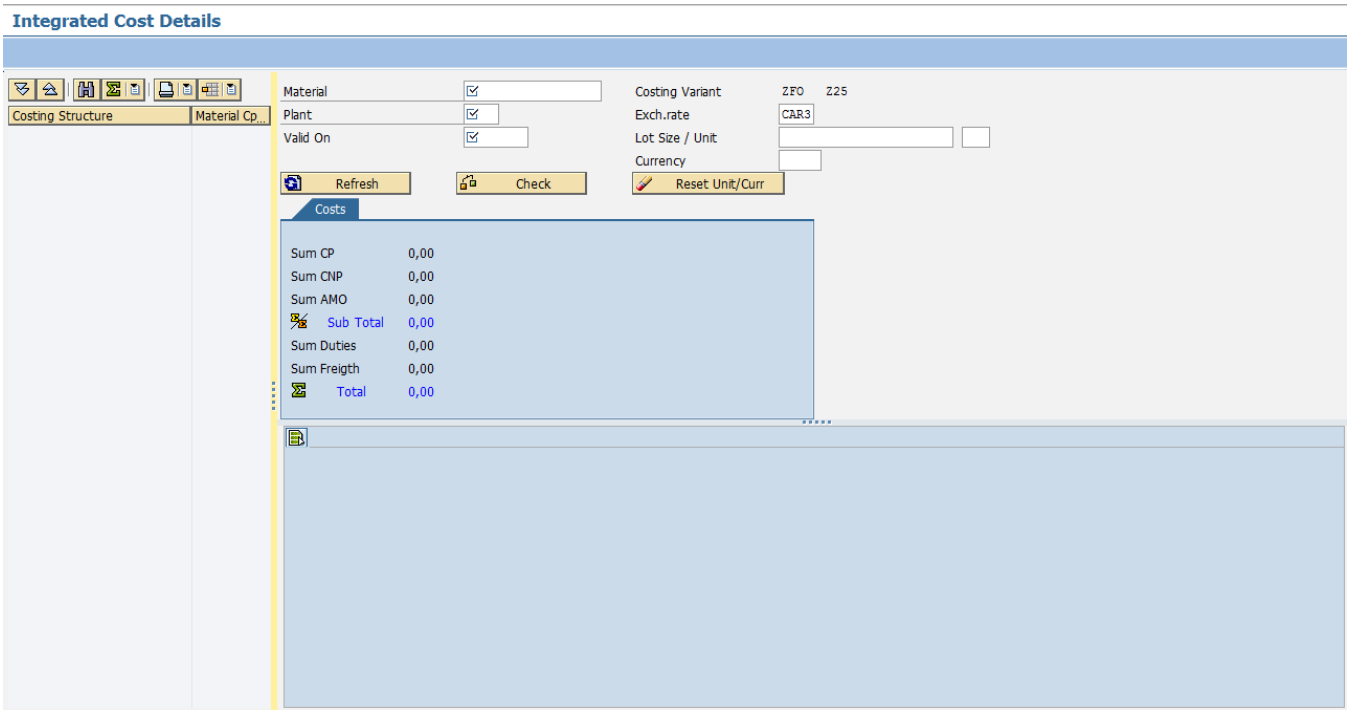
Program associated: ZBW_M_INT_ANALYSIS

Transaction overview

How to use and read

https://drive.google.com/file/d/1KY1WrLy18W05jZVtWxeVaBldqGPdvUxk_4bb52xiC4s/view

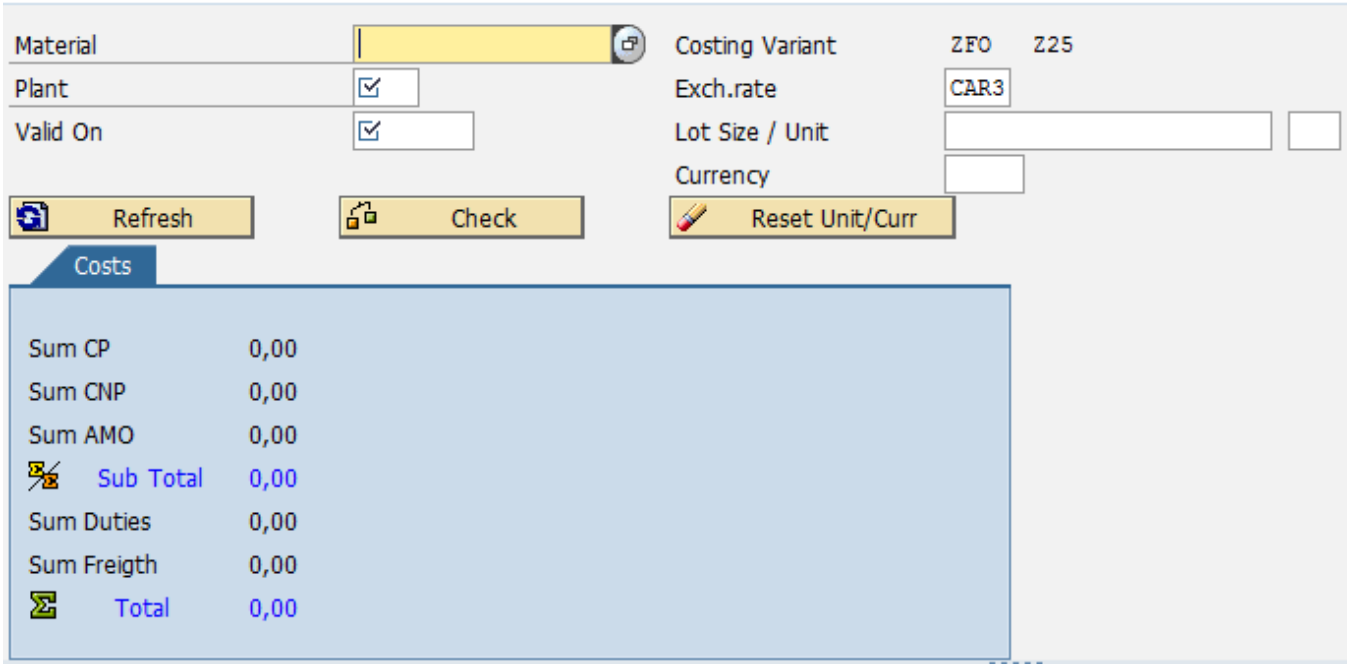
General Overview (default display)



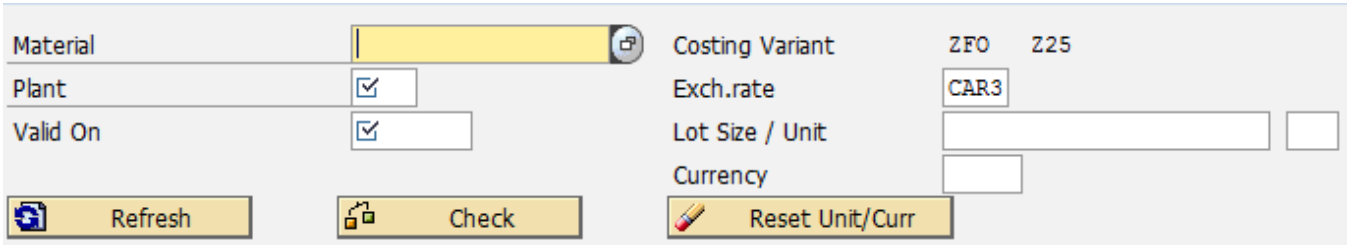
(Target Display)

Screens

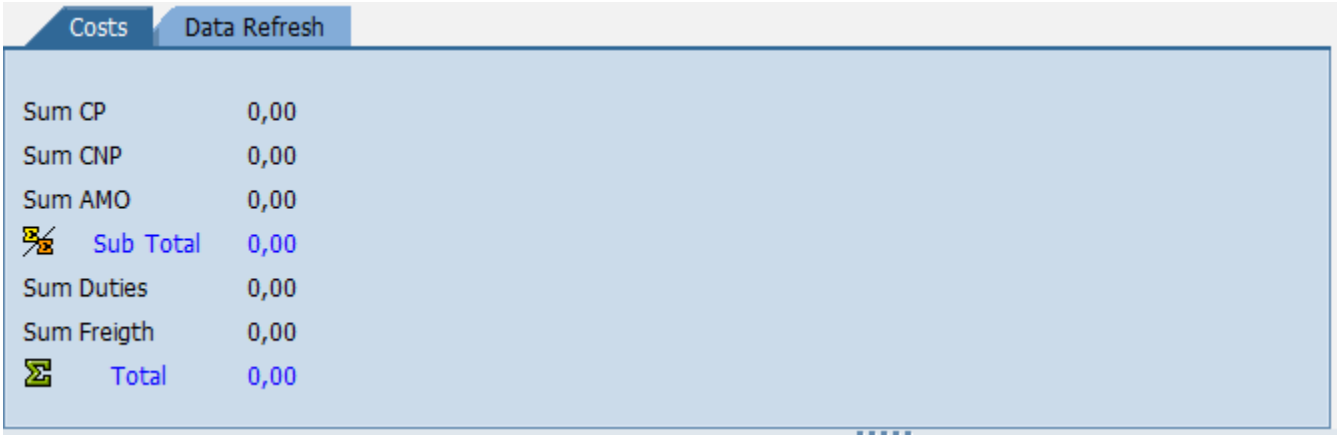
0100 Main Screen



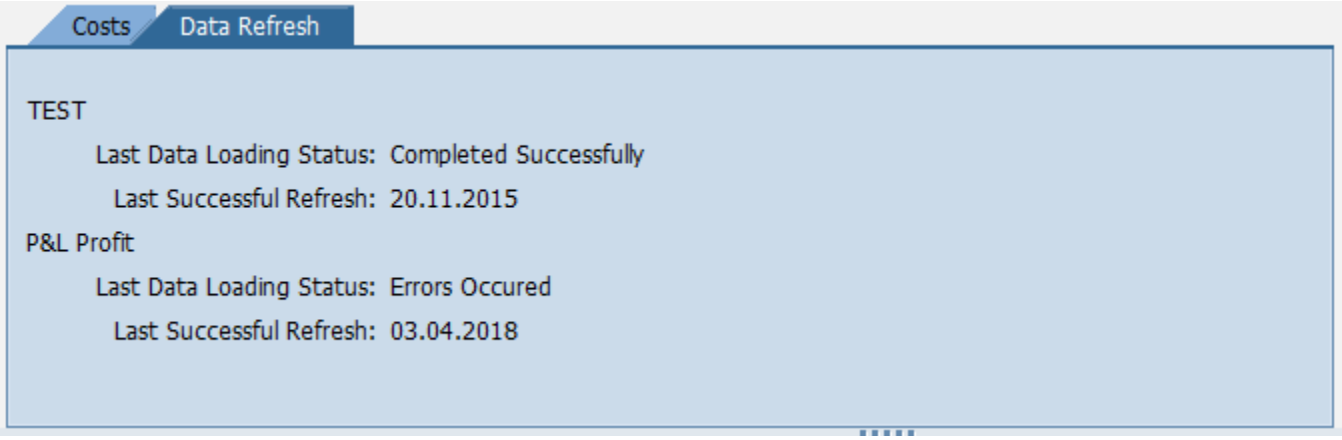
0110 Input Screen



0120 Costs



0130 Data Refresh



Classes

LCL_TREE_EVENT_RECEIVER

Classe standard, permet d'activer des comportements avec le TREE. Chaque action est représentée par une METHOD propre.

Chaque CLASS est définit par une DEFINITION et une IMPLEMENTATION.

Chaque METHOD est présente dans l'une de ces deux classes.

Dans cette transaction, une seule action est activée :

> *METHOD handle_node_double_click : Déplie un noeud de la hiérarchie si on double clique dessus.*

LCL_TOOLBAR_EVENT_RECEIVER

Classe standard, permet d'activer des comportements sur la TOOLBAR du TREE. Chaque action est représentée par une METHOD propre.

Chaque CLASS est définit par une DEFINITION et une IMPLEMENTATION.

Chaque METHOD est présente dans l'une de ces deux classes.

Dans cette transaction, une seule action est activée :

> *METHOD on_function_selected : Récupère les actions sur les boutons Activities et Raw Material*

Includes

- INCLUDE ZBW_M_INT_ANALYSISTOP. " Types / Data declaration
- INCLUDE ZBW_M_INT_ANALYSISCL2. " Class : Handle Tree event
- INCLUDE ZBW_M_INT_ANALYSISO01. " PBO and PAI
- INCLUDE ZBW_M_INT_ANALYSISF01. " Screen Declaration
- INCLUDE ZBW_M_INT_ANALYSISF02. " Data selection
- INCLUDE ZBW_M_INT_ANALYSISF03. " Checks

Subroutine

- ADD_COMPLETE_LINE Add nodes in the TREE (Recursive)
- BUILD_FIELDCATALOG Unique structure of data used in TREE and GRID
- BUILD_HIERARCHY_HEADER TREE header
- CHANGE_TOOLBAR Add buttons to display / hide Activities and Raw Materials
- CHECK_AUTHORITY Perform check on GBU and Scope (AUTHMA)
- CHECK_INPUT Perform check (after **Check** or **Update** button) to verify if fields are filled and if data is available in source before to execute the FM
- CREATE_GRID Filled GRID with DATA from GET_DATA
- CREATE_HIERARCHY Filled TREE with DATA from GET_DATA
- EXIT_PROGRAM Leave transaction
- GET_DATA Execute FM ZBW_M_INT_READ
- INIT_GRID Create GRID Object with FIELDCATALOG
- INIT_TREE Create TREE Object with FIELDCATALOG

- REGISTER_EVENTS Handle some specific action (Only Double click to explode a node on TREE)

Data Quality Control

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

Operational Documentation

Procedures

<Describe the recurring procedures needed to operate the application (eg. start/pause/terminate/restart the app processes, data preparation, data ingestion, ETL, data visualization, data export, other manual activities)>

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