Technical Documentation - Bill of Material

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

Authorization Objects

List of authorization objects mandatory for the application.

Authorization object	Explanation

DataFlow

Overview

https://drive.google.com/drive/folders/18VbVRgRTsredyh89a_qSECY7E0yzSKJJ

Dataflow overview :

Objective of the application

The objective of the application is to extract from **WP1 & PF1 SAP systems** Bill of Material data and explode BOM level by level with components data.

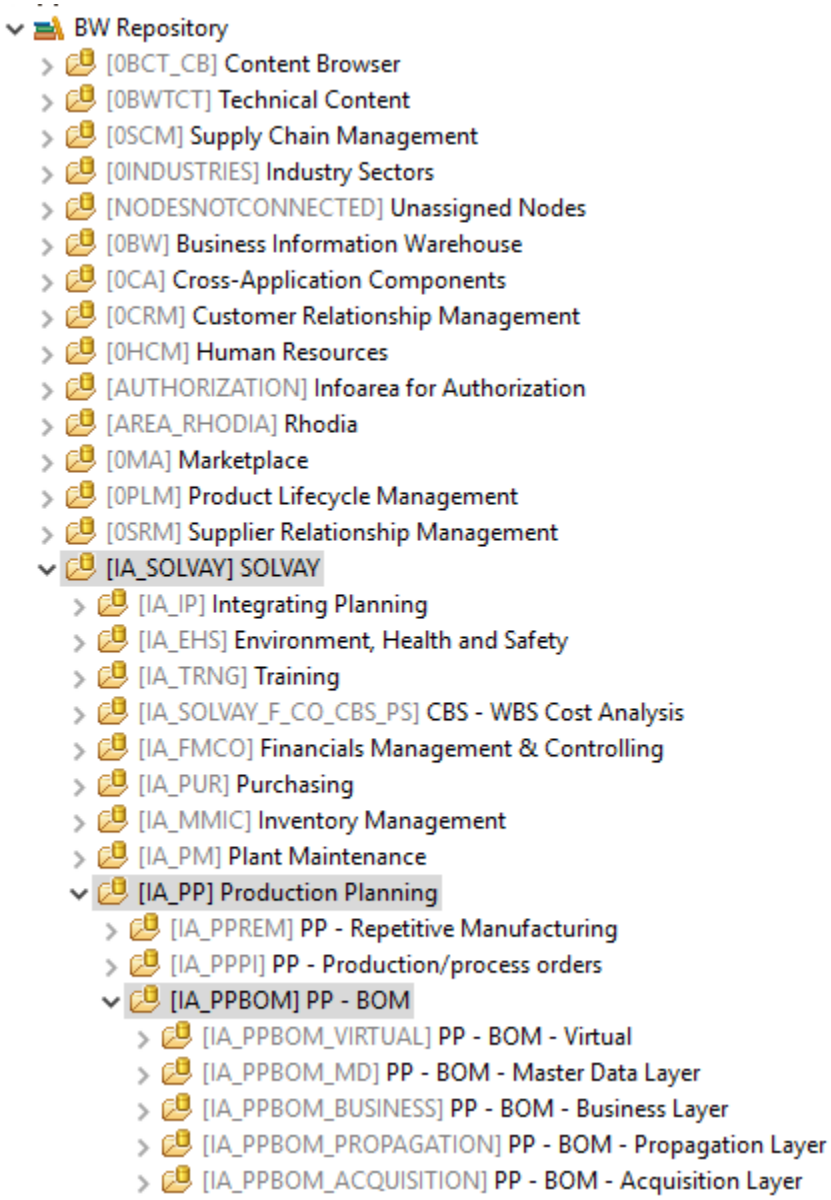
Then data are consumed by Dynasys application by generated flat file.

Tool Leader: Meire Santos

IT leader of the application: Meire Santos (PP), Craig Wanamaker (BW)

Reporting Coordinator: Meire Santos

InfoArea:



History

This application is linked to a need of Dynasy Tool. Every month a manually extract is done of BOM Data from ERP SAP system. The goal is to extract these data from SAP to BW and generate automaticaly BOM data level by level by flat file and send this file to Dynasys for import.

Technical Rules on Workbench

TRSF : AAPPBO02 (RCS) -> APPPBO02 (RCS) & TRSF : AAPPBO01 (SLV) -> APPPBO01 (SLV)

Start routine

Internals tables updated from master data c_matpnt3, c_matnr2. Source system logsys defined with method "oref_dynasys_utils ->get_source_systems".

Package is sorted by Material MATNR and Plant WERKS.

End routine

C_CPT_TYP is determined with master data C_MATNR2 (where source logsys = logsys from c_matnr2 and source c_bomcpt = matnr2 from c_matnr2).

MATL_TYPE & C_MMSTA are determined with master data C_MATNR2 (where source logsys = logsys from c_matnr2 and source c_matnr2 = matnr2 from c_matnr2).

C_PFCTR2 is determined with master data C_MATPNT3 (where source logsys = logsys from c_matpnt3 and source c_plant = c_plant from c_matpnt3 and source c_matnr2 = c_matpnt3 from master data c_matpnt3).

C_DYN_052 "Component with System Extension" is a concatenation with c_bomcpt and variable v_rhodia_system_extension (determined with method "oref_dynasys_utils->get_system_extension" with input source system dermined in start routine).

C_DYN_005 "Material with System Extension " is a concatenation with c_matnr2 and variable v_rhodia_system_extension (determined with method "oref_dynasys_utils->get_system_extension" with input source system dermined in start routine).

C_DYN_001 "Plant with System Extension " is a concatenation with c_plant and variable v_rhodia_system_extension (determined with method "oref_dynasys_utils->get_system_extension" with input source system dermined in start routine).

Check in active table if there is a correspondance with record in source package in function of logsys, c_matnr2, c_plant, c_bomcat, c_bomnum and c_bomaltm.

If there is a correspondance, make the same test with in addition C_BOMITM & C_BOMCPT.

If with these two new field there is no correspondance record marked as deleted.

TRSF : APPPBO02 (RCS) -> ABPPBO01 & TRSF : APPPBO01 (SLV) -> ABPPBO01

Quantities conversion and recalculation

BOM quantities and component quantities are converted in KG between propagation and business layer.

Moreover, quantities are recalculated in order to have a **BOM Base quantity equal to 1000 KG (1 TO),** Component quantities are updated accordingly.

Specific rule concerning Alternative Item Group and Usage Probability

On BW side, there is a specific rule applied when the **Alternative Item Group (C_ALPGR infobject) is <> blank** (and only when it's <> blank) :

- We multiply the **component quantity** (K_COMPQTY) **by the Usage probability percentage** (C_EWAHR)
- We delete the record when the **usage probability is null.**

DTP : APPPBO02 (RCS) -> ABPPBO01 - Delta &DTP : APPPBO01 (SLV) -> ABPPBO01 - Delta

Semantic group on:

Key Field	Field Name
☑	LOGSYS
☑	/BIC/C_MATNR2
☑	/BIC/C_PLANT
☑	/BIC/C_BOMCAT
☑	/BIC/C_BOMNUMM
☑	/BIC/C_BOMALTM
☐	/BIC/C_BOMITM

Filters:

Material status <> Z0

Component Type: olap variable V_MATL_TYPE_0002 (class ZCL_BIU001_V_MATL_TYPE_0001):

= ZPIP / Z714 / ZNLA / Z700 / ZMAT / Z707

ERP Group of activ: olap variable V_0G_CWWE03_0001 (class ZCL_BIU001_V_0G_CWWE03_0001):

Plant: olap variable V_0PLANT_0008 (class ZCL_BIU001_V_0PLANT_0008):

Material type: V_MATL_TYPE_0002 (class ZCL_BIU001_V_MATL_TYPE_0001)

= Z714 / Z700 / ZMAT/ ZNLA / ZPIP / Z707.

Specification Document for Dynasys flat file (Sheet BOM)

BOM Master Data

BOM Number master data (C_BOMNUM) is loaded by the datasource **DTS_ZBW_V_MAST_STKO**.

The Datasource is based on SAP tables MAST (Material to BOM Link) and STKO (BOM Header) => View **ZBW_V_MAST_STKO**

In view ZBW_V_MAST_STKO, we keep only entries from STKO with BOM Status (STLST) is not equal to 2 (inactive BOM) and BOM usage (STLAN from MAST) is equal to '1' (Production).

There are 2 fields used to filter data during extraction and loading on BW :

- AEDAT (STKO)
- ANDAT (MAST)

For more explanations, see part Data Loading.

BOM Item Master Data

BOM Item master data (C_BOMITM) is loaded by the datasource **DTS_ZBW_EXTRACT_BOM_ITEM**.

The datasource is based on specific function module **ZBW_EXTRACT_BOM_ITEM**.

In this function module we call the standard function **CSAP_MAT_BOM_ITEM_SELECT** to extract each item for a given material/plant.

There are 2 fields used to filter data during extraction and loading on BW :

- AEDAT (STPO)
- ANDAT (STPO)

For more explanations, see part Data Loading.

BOM Level determination

The most important need of this dataflow is to extract (and explode) all BOM items and components for a given product.

A specific BW extractor has been developed to build the Bill of Material from SAP WP1 source system : **DTS_ZBW_MAST_BOM_LEVEL**.

There are 2 fields used to filter data during extraction and loading on BW :

- AEDAT (STPO)
- ANDAT (STPO)

For more explanations, see part Data Loading.

The extractor is based on view **ZBW_V_MAST** and in particular from **MAST** table to keep all Bill of material to explode and the class **ZDTS_ZBW_MAST_BOM_LEVEL** is used to extand the datasource adding new fields.

In view ZBW_V_MAST, we keep only entries from STKO with BOM Usage (STLTY) = M (Material) and BOM Status (STLST) is not equal to 2 (inactive BOM).

In the class, we use the standard function module '**CS_BOM_EXPL_MAT_V2**' to build the Bill of material and BOM level. The function module is used as recursive function, as long as BOM exists and have to be explode.

Today we only explode BOM at level 1 (Master material and its components at first level).

We filter records from MAST to have only data with :

- BOM usage (STLAN) is equal to '1' (Production).
- BOM deletion flag (LOEKZ) is not equal to 'X'.
- BOM valid from date (DATUV) is greater than current date.

Additional filter is done during extraction to keep **only one alternative BOM for the material exploded.**

To get this alternative BOM, we select **the first valid production version** (excluding locked production version) from MKAL table.

The equivalent to explode BOM level by level on SAP side is T-CODE **CS11.**

See below, main fields determined or calculated in the class :

- **BMEIN - Base unit of Measure of BOM :** It correponds to the Base unit of measure of the main product exploded (MATNR) linked to the main BOM (STLNR).
- **BMENG - BOM Quantity :** BOM quantity is converted on Base Unit of Measure of product exploded. The quantity is duplicated of each component line linked to this product.

- **ZZMAT - BOM Head :** If there is only 1 Product Level, Bom Head is equal to the main Product (MATNR). If there is several Product Level, BOM Head is equal to the component exploded (from previous level)
- **ZZSTLNR - Bill of Material :** is equal to the BOM linked to the BOM Head material.
- **ZZSTLAL - Alternative BOM :** Alternative BOM linked to ZZSTLNR
- **ZZIDNRK - BOM Component :** Component of the BOM Head Material
- **ZZPOSNR - BOM Item :** BOM Item linked to BOM Component
- **ZZSTUFE - Product Level : _**is equal to 1 when we are at component level of the master product exploded, we increment this level when we are at component level and for each new product (BOM Head) exploded
- **ZZMENGE - Component Quantity :** Component quantity is converted on Base Unit of Measure of the component
- **ZZMMEIN - Component Base Unit of Measure :** Base Unit of measure of the component
- **ZZFMENG - Fixed Quantity :** This field is equal to X when the component is flagged as "Fixed quantity"
- **ZZAUSCH -** Component Scrap Percentage (%)
- **ZZWOPRODVERS - Production Version Flag** :
    1. *PF1 Rule :*

       For each material / plant we read the MAKL table (production version) to get the active (valid from/to have to be valid) production version (sort by alphanumeric) & the alternative BOM linked => Only one version per Material@Plant.

       If no production version is found, the flag is equal to 'X' .

       If the production version is found, the flag is equal to blank and only the alternative BOM linked to the production version is displayed in CSV file.
    2. *WP1 Rule :*

       We get from MKAL table (production version), for a material@plant **only one production version.**
       This version is determined by :

        - Valid date from/to
        - Sort production version (VERID) by alphanumeric to keep the first one.

       Finally we have one production version with material@plant@alternativeBOM got from MKAL table.
        - **Case 1 :** A production version is existing for the record treated, Alternative BOM in the record is the same as the alternative BOM of production version => **record is treated and displayed in the file.**

        - **Case 2 :** If a production version is existing but the alternative BOM is not the same between the record treated and MKAL table or there is a prod version but it is not valid anymore **we delete the record**.
        - **Case 3 :** If no production version is existing in MKAL table, **we flag the record with the field without prod version = 'X'. => Rule to determine co-packing product.**
- **ZZALPGR :** Alternative Item Group
- **ZZEWAHR :** Usage Probability in %

Fields with system extension

There are 3 fields stored in BOM Propagation and Business ADSO with a system extension :

- **C_DYN_001** : Plant with System Extension
- **C_DYN_005** : Material with System Extension
- **C_DYN_052** : Component with System Extension

For more explanations see : BW OTC - Supply Chain- DPS (WBP) /!\ Obsolete /!\#SupplyChain-DPS(WBP)-FieldwithextensionC_DYN_***

Reporting

No queries created in CPPPBO01

Dependencies with other applications

BOM data are used by Dynasys tool (consumed by generated flat file).

aDSO ABDPDY50 is loaded by aDSO ABPPBO01. with help of calculation view **CV_DY_BOM.**

Dynasys flat file is generated by reading ABDPDY50 data.

Data Loading

Info Providers and objects loaded

Detail of process chain, list + link between or special event done for the loading

Main Process Chain	Final Provider Loading	Frequency	Time start	Duration
PC_PP_BOM_01	ABPPBO01 Meta chain	Daily	00 am	1h30 mins

Scope of data loaded

Both master data and acquisition layer ADSO AAPPBO02 are loaded by 2 FULL DTPs.

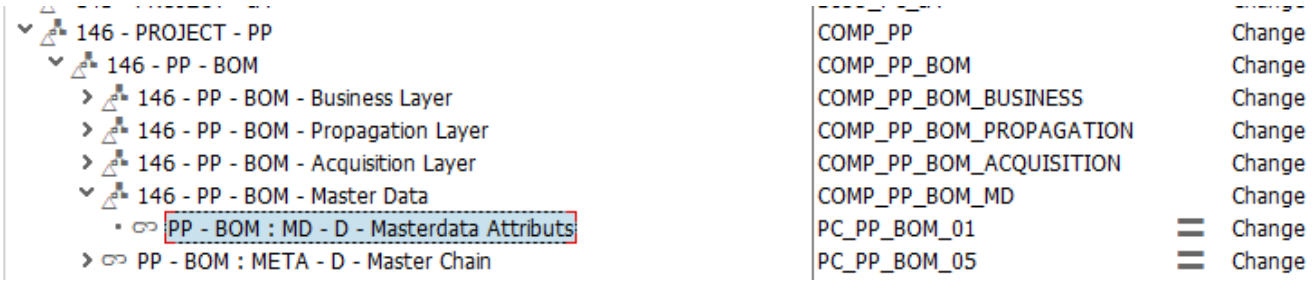
The fisrt one is based on creation date and the second on modification date of BOM.

We load for each of them from current date to current date - 7.

Note that DTP are different for each objects.

Master Data process chain

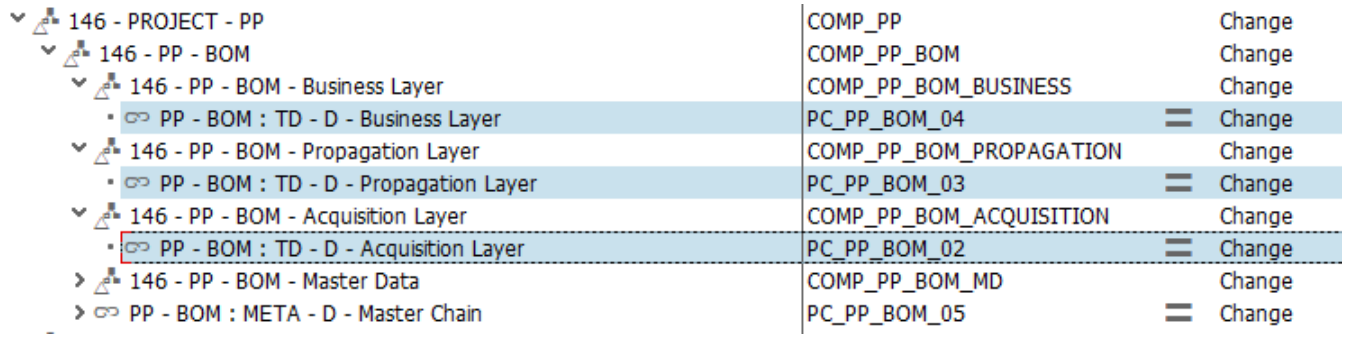
BOM Master Data are loaded by process chain **PC_PP_BOM_01**



BOM Level data process chain

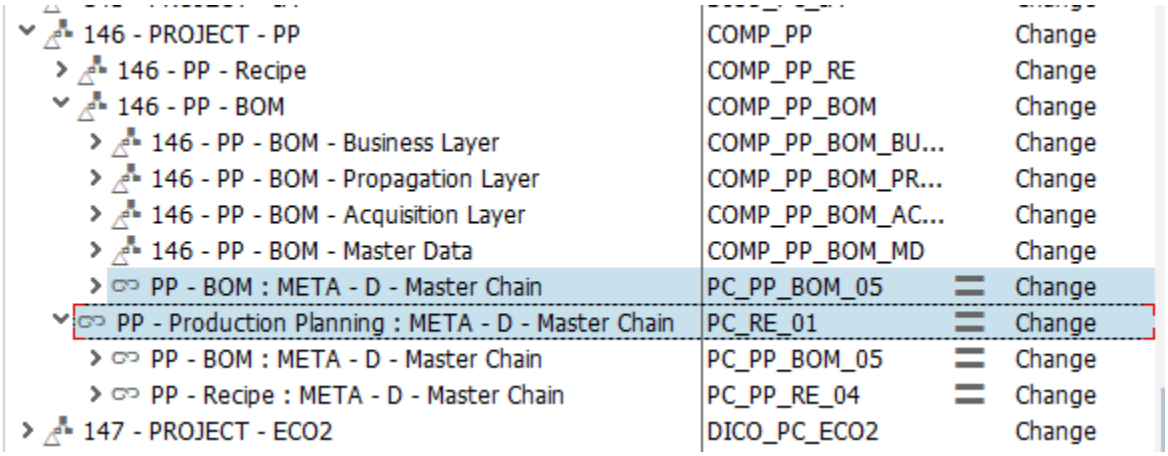
BOM Master Data are loaded by following process chain :

- Acquisition layer : **PC_PP_BOM_02**
- Propagation layer : **PC_PP_BOM_03**
- Business layer : **PC_PP_BOM_04**



A Master process chain load both part : **PC_PP_BOM_05**

This master chain is part of a global chain with **Recipe application : PC_RE_01**



Today the process chain **PC_RE_01** is scheduled at **00:30 (CET).**

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

<List past & future evolutions for the application (including links to MED/FSD/TSD)>