HOW TO ENHANCE STANDARD BW/4HANA ODP EXTRACTORS – BACKEND SOLUTION

Introduction:

What is ODP:

Operational Data Provisioning provides a technical infrastructure that you can use to support two different application scenarios. The first of these is Operational Analytics for decision making in operative business processes

The other very prominent application scenario is data extraction and replication: Operational Data Provisioning supports extraction and replication scenarios for various target applications and supports delta mechanisms in these scenarios. In case of a delta procedure, the data from a source (the so called ODP Provider) is automatically written to a delta queue (the Operational Delta Queue - ODQ) using an update process or passed to the delta queue using an extractor interface. The target applications (referred to as ODQ 'subscribers' or more generally 'ODP Consumers') retrieve the data from the delta queue and continue processing the data.

With SAP BW/4HANA, Operational Data Provisioning (ODP) now becomes the central infrastructure for data extraction and replication from SAP (ABAP) applications to a SAP BW/4HANA Data Warehouse. SAP recommend customers to use ODP for the implementation of new extraction and replication scenarios from SAP (ABAP) applications.

In this document, i try to explain some techniques especially needed for support level projects.

How to enhance standard ODP datasource just adding standard or Z custom fields to extractor structure.

No need any abap select fetch statements required.

I share the necessary steps for commonly used Master Data Datasource ODP object and do the modifications, consume the places in backend, then verificatio steps respectively.

For this scenario, SAP ERP Logistics Processes, Material Master Data, 0MATERIAL_ATTR

Definition of ODP extractor:

Description	Material Number
Datasource Type	Master Data Attributes Extractor
Application	Logistics - Logistics - General
Delta Process	Only New Records (Inserts) Via Extractor (Cube-Compatible)
Extraction Method	Based on Function Module (Complete Reference)
Extractor	MDEX_MATERIAL_MD
Extractor Structure	BIW_MARA_S
Source Tables (Function Module)	MDEX_MATERIAL_MD (Function Module)

Scenario:

This ODP extractor fields corresponds to SAP ERP Logistics MARA master data table fields and gather commonly used some fields of MARA table fields to this ODP extractor.

MARA is a standard SAP Table which is used to store General Material Data data and is available within Source SAP system.

Steps:

Fort this scenario, i need one standard field and one Z custom field to be added 0MATERIAL_ATTR material master data.

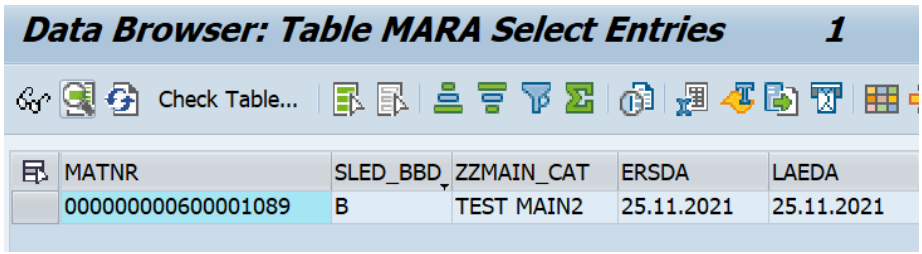
I first do the modifications and add the fields then in the end explain verification steps from program layer.

I add 2 fields, the first one standard MARA-SLED_BBD and the second one Z custom field of MARA table MARA-ZZMAIN_CAT.

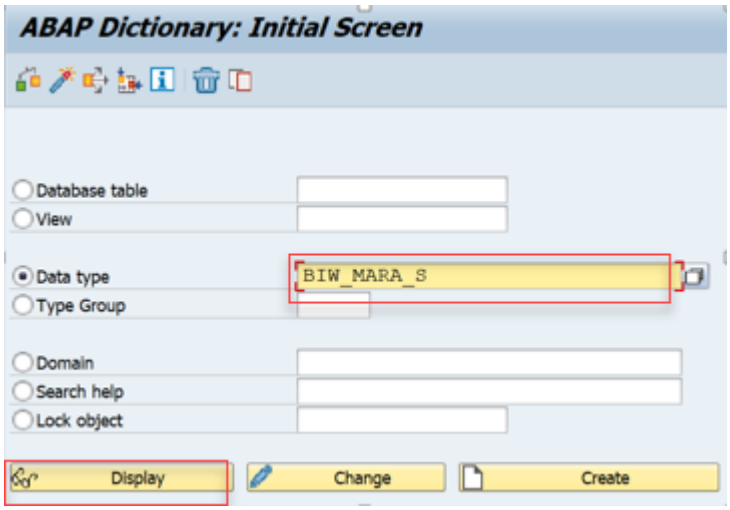
Sample screenshot of MARA table and related fields for one material number.

During this scenario I use material number: 600001089

MARA table screenshot,

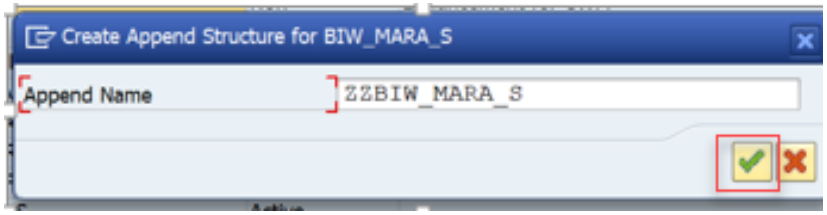


In ERP system goto SE11 and paste BIW_MARA_S and press 'display'



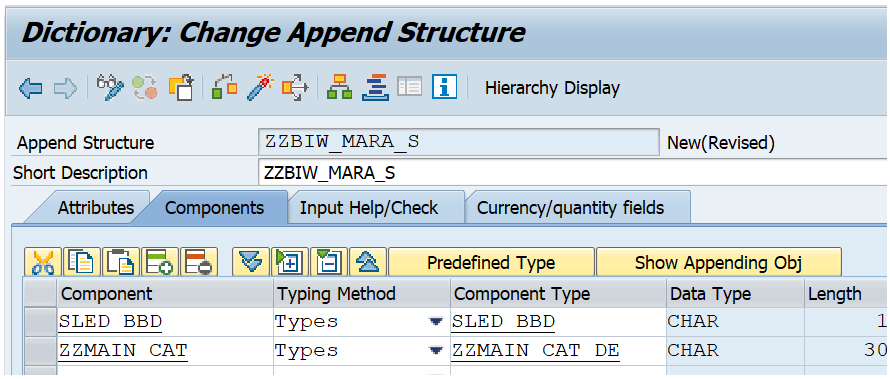
And press ‘Append Structure’.

Now, a popup apears and give a name, for instance ZZBIW_MARA_S

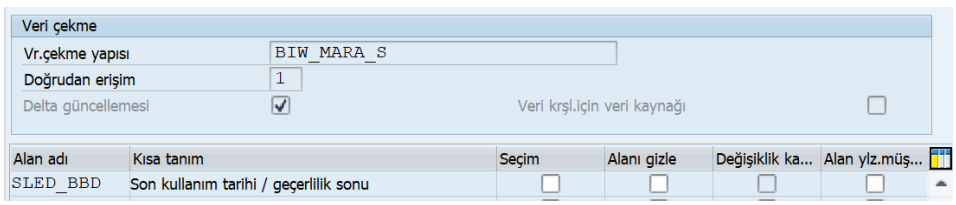


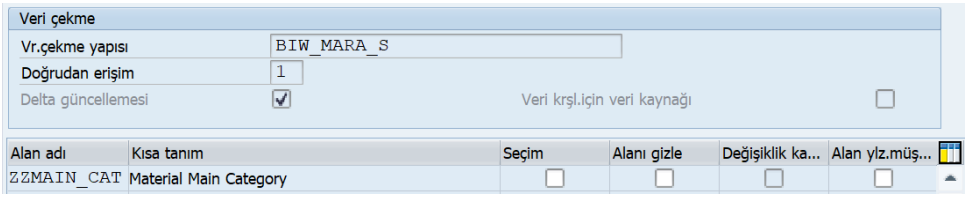
I input two fields and related component type.

The fields that we add to structure must be same name and same component type with MARA-SLED_BBD and MARA-ZZMAIN_CAT respectively.

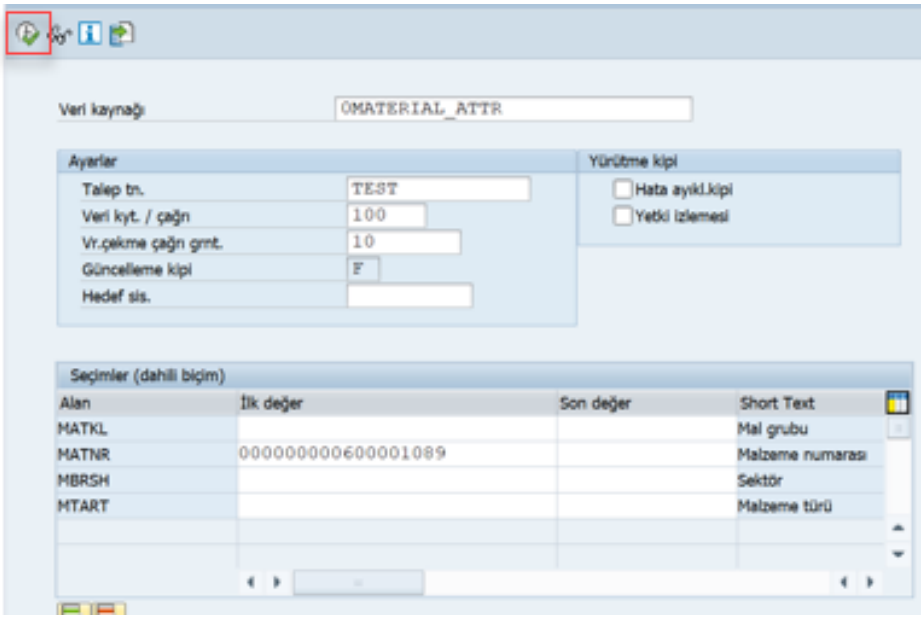


Now go to RSA6 and uncheck all fields of the new two fields to be filled and visible in 0MATERIAL_ATTR.



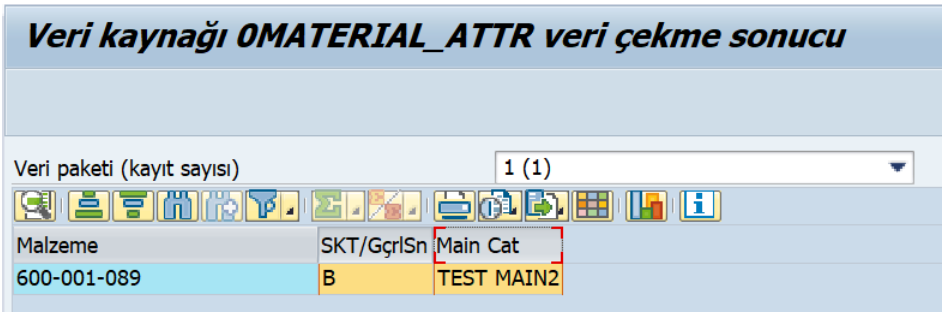


Now , i go to RSA3 and test for material number 600001089



Now , i can see the two newly fields and highlight them, SLED_BBD and ZZMAIN_CAT.

First field Material number has a conversion and output seen with conversion.



During the steps, i didn't write any select fetch abap statements in CMOD or SMOD layer.

So , as a result to add new field to ODP extractor only adding necessary fields to DS structure and uncheck them from RSA6 is enough.

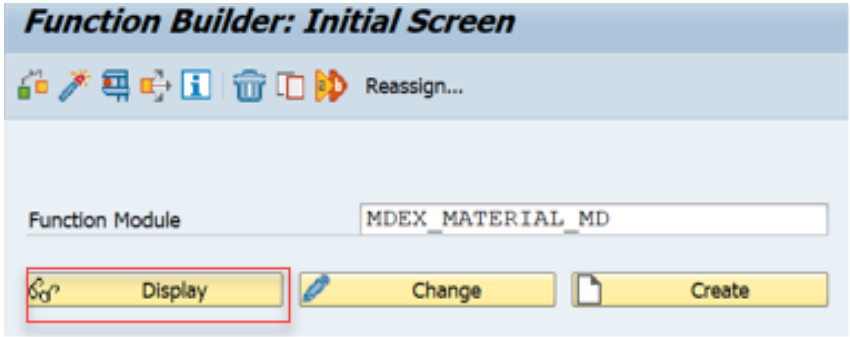
No any data latency , Lock Waits , Deadlocks or performance reduction, zombie DTP requests during data loading processes from ERP to BW4.

Lock Waits and Deadlocks help.sap link:

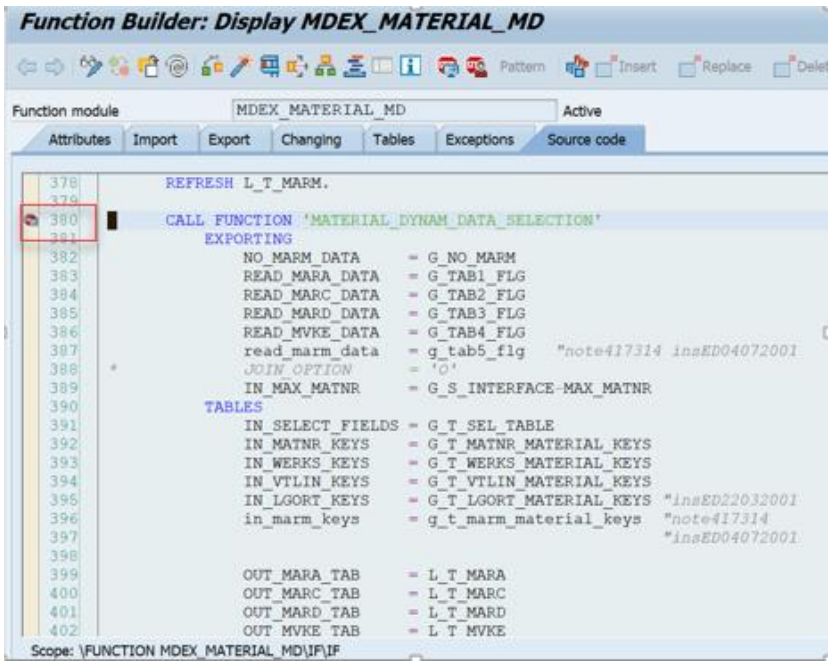
https://help.sap.com/viewer/db6_dbacockpit/4505568cae262461e10000000a1553f7.html

Verification steps:

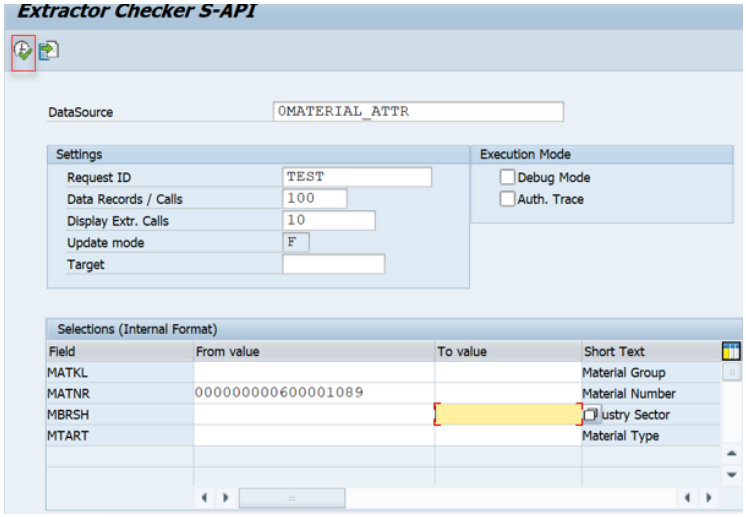
-Go to ERP SE37 program: MDEX_MATERIAL_MD and press ‘display’ button.



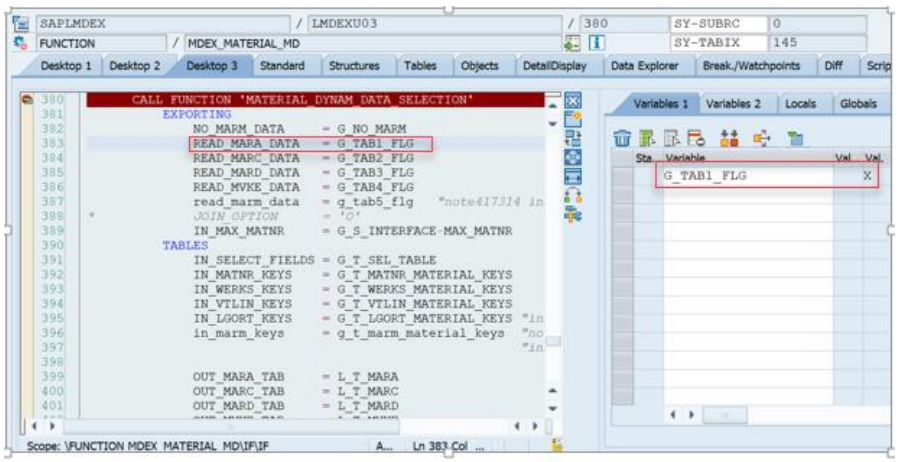
-Now i am adding breakpoint to line 380 , start of inner function modüle : MATERIAL_DYNAM_DATA_SELECTION



-Now, RSA3 and running with sample material number: 600001089

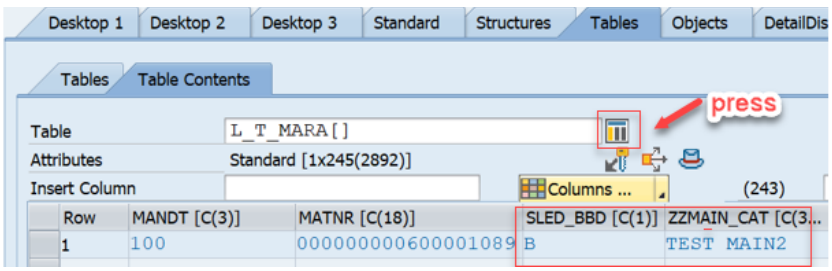


-Now double clicking G_TAB1_FLG which corresponds to read MARA table fields.

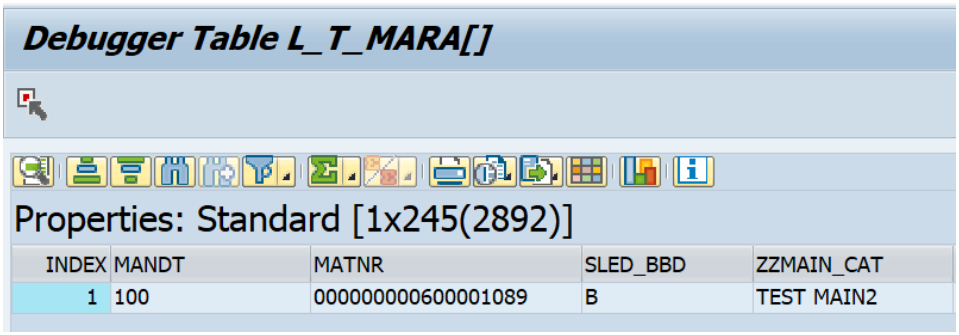


-Press F6

-Now click L_T_MARA, the output values of MATERIAL_DYNAM_DATA_SELECTION now gets the whole fields of MARA in this internal table.

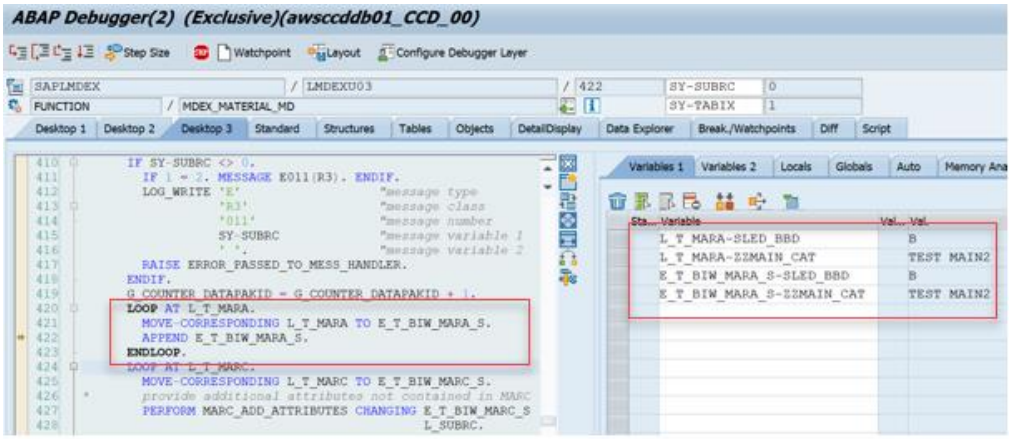


Now we can see the L_T_MARA data as ALV display,

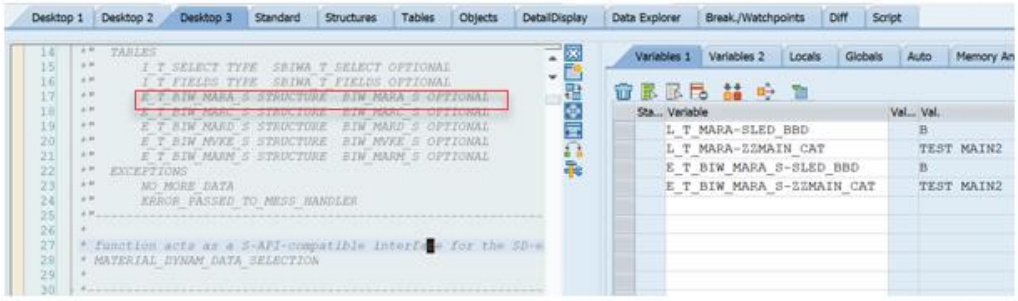


-Still the values in l_t_mara and all fields reside in this internal table.

-Now, in line 421 values pass and append from internal table to DS structure line by line.



E_T_BIW_MARA_S is Datasource internal table which is BIW_MARA_S type, declared in the beginning of MDEX_MATERIAL_MD

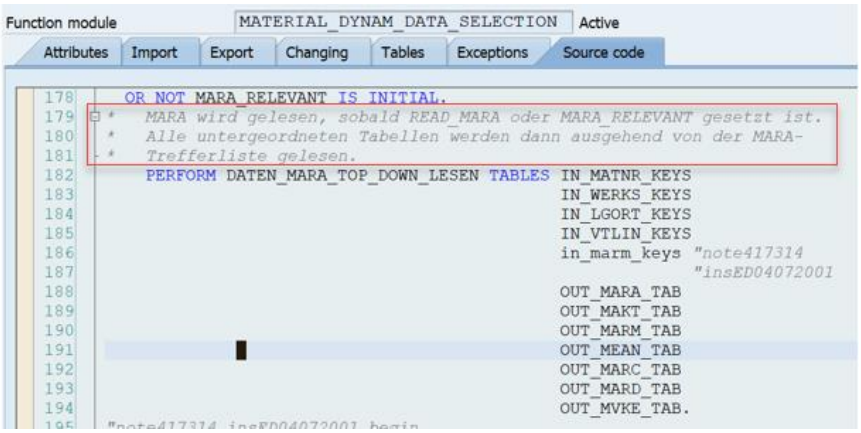


Thats all steps for verification.

In MDEX_MATERIAL_MD

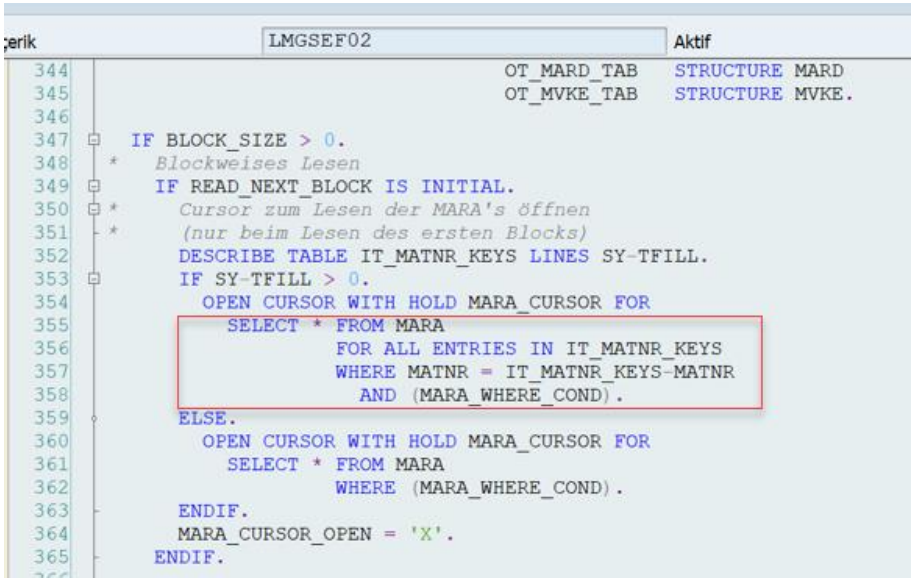
Double click on MATERIAL_DYNAM_DATA_SELECTION

Double click DATEN_MARA_TOP_DOWN_LESEN

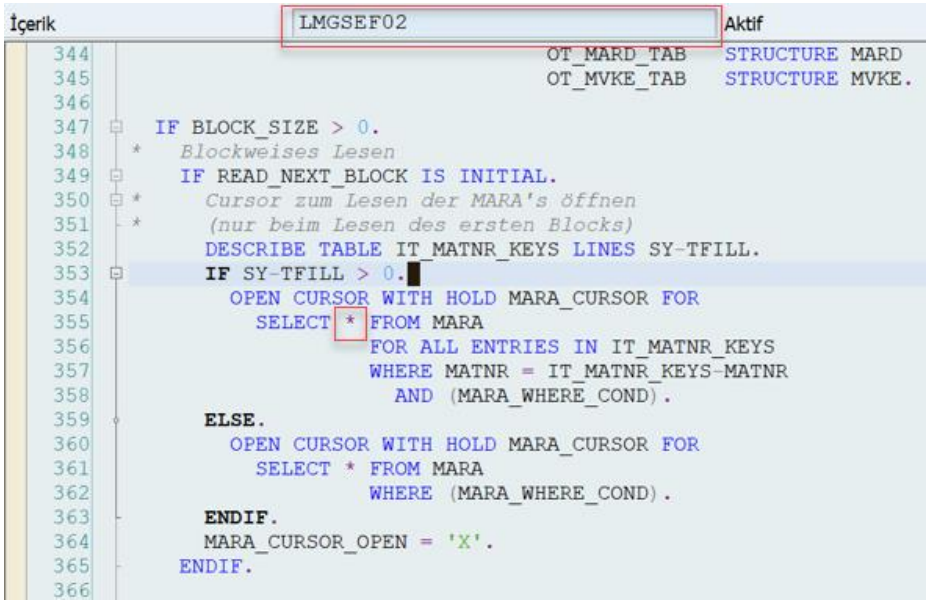



```
*   MARA wird gelesen, sobald READ_MARA oder MARA_RELEVANT gesetzt ist.
*   Alle untergeordneten Tabellen werden dann ausgehend von der MARA-
*   Trefferliste gelesen.
```

```
*MARA is read as soon as READ_MARA or MARA_RELEVANT is set.
*All subordinate tables are then based on the MARA
*Hit list read.
```



Now it is seen that data fetched from MARA with '*' asterisk statement in program LMGSEF02.



Conclusion:

This example is adding fields from one logistics master data to DS ODP extractior, since it is one master table (MARA) , it is flat to apply the implementation steps.

For Standard Transactional Datasources ODP extractors,

A bit more detail code snippets reading and implementation (Badi, Bapi, Std. Program Enhancement) techniques required, since Transactional Datasources fetches necessary datas from many transactional and master data tables.

Abap and debugging techniques can be applied for Troubleshooting, investigating customizing and cross customizing (such as ERP -- IS-Utilities joint scenarios or dumps) and related tables, finding table filters values, exceptions during User Acceptance Test(UAT) Scenarios, BW datasource related ERP Business Functions problems, analyzing data loading dumps from ERP ST22.

SAP Business function help.sap link:

https://help.sap.com/saphelp_ewm900/helpdata/en/97/9bddd4cebe423f9eb5767a275b2d78/content.htm?no_cache=true

Thank you.

Cihan Ekin