

## HOW TO ENHANCE STANDARD BW/4HANA SERVICE API 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:

For this scenario, I need one standard field and one Z custom field to be added to MATERIAL\_ATTR material master data.

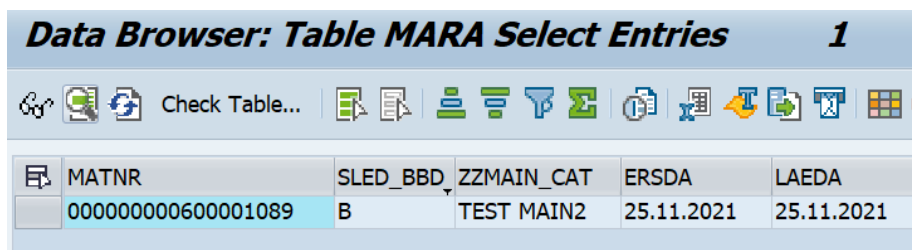
I first do the modifications and add the fields then in the end explain verification steps from program layer.

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.

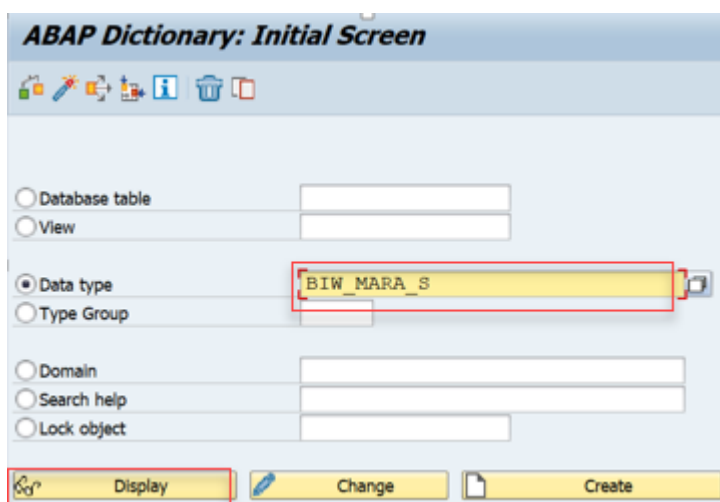
During this scenario I use material number: 600001089

MARA table screenshot,



MATNR	SLED_BBD	ZZMAIN_CAT	ERSDA	LAEDA
000000000600001089	B	TEST MAIN2	25.11.2021	25.11.2021

In ERP system goto SE11 and paste BIW\_MARA\_S and press 'display'



ABAP Dictionary: Initial Screen

☐ Database table  
☐ View  
☒ Data type  
☐ Type Group  
☐ Domain  
☐ Search help  
☐ Lock object

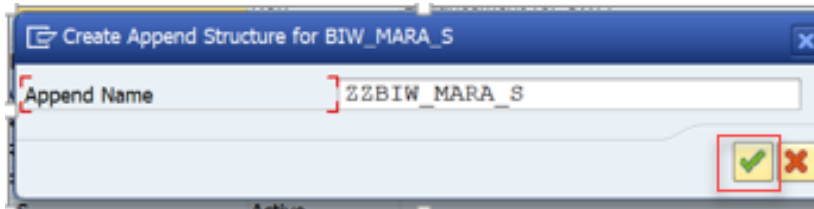
BIW\_MARA\_S

Display Change Create

And press 'Append Structure'.

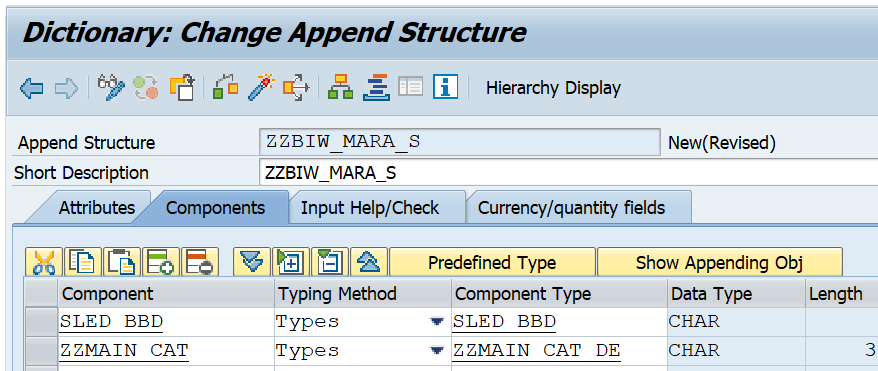


Now, a popup appears and give a name, for instance ZZBIW\_MARA\_S

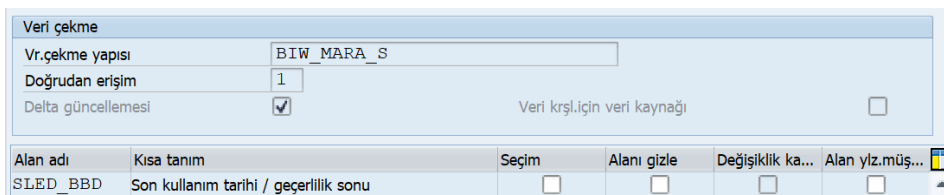


I input two fields and related component type.

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 OMATERIAL\_ATTR.



<b>Veri çekme</b>					
Vr.çekme yapısı	BIW_MARA_S				
Doğrudan erişim	1				
Delta güncellemesi	<input checked="" type="checkbox"/> Veri kışl. için veri kaynağı <input type="checkbox"/>				
Alan adı	Kısa tanım	Seçim	Alanı gizle	Değişiklik ka...	Alan yz. müş...
ZZMAIN_CAT	Material Main Category	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

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

<b>Veri kaynağı</b> OMATERIAL_ATTR			
<b>Ayarlar</b>		<b>Yürütme kipi</b>	
Talep tn.	TEST	<input type="checkbox"/> Hata ayıkl.kipi	
Veri kyt. / çağn	100	<input type="checkbox"/> Yetki izlemesi	
Vr.çekme çağn grnt.	10		
Güncelleme kipi	F		
Hedef sis.			
<b>Seçimler (dahili biçim)</b>			
Alan	İlk değer	Son değer	Short Text
MATKL			Mal grubu
MATNR	0000000000600001089		Malzeme numarası
MBRSH			Sektör
MTART			Malzeme türü

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.

<b>Veri kaynağı OMATERIAL_ATTR veri çekme sonucu</b>		
Veri paketi (kayıt sayısı) 1 (1)		
Malzeme	SKT/GçrlSn	Main Cat
600-001-089	B	TEST MAIN2

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.

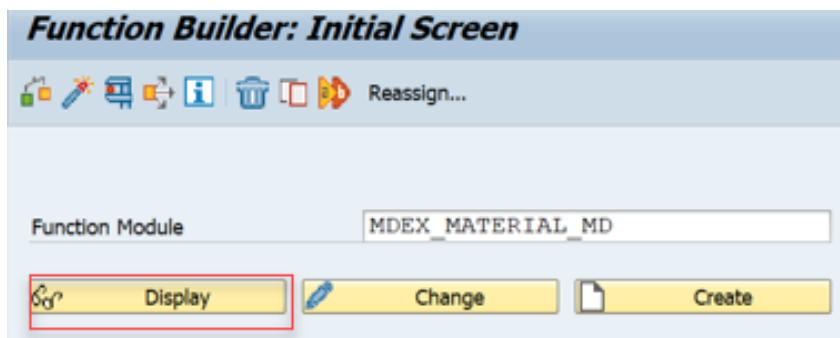
No any data latency , Lock Waits , Deadlocks or performance reduction, zombie DTP requests during data loading processes from ERP to BW4.

Lock Waits and Deadlocks help.sap link:

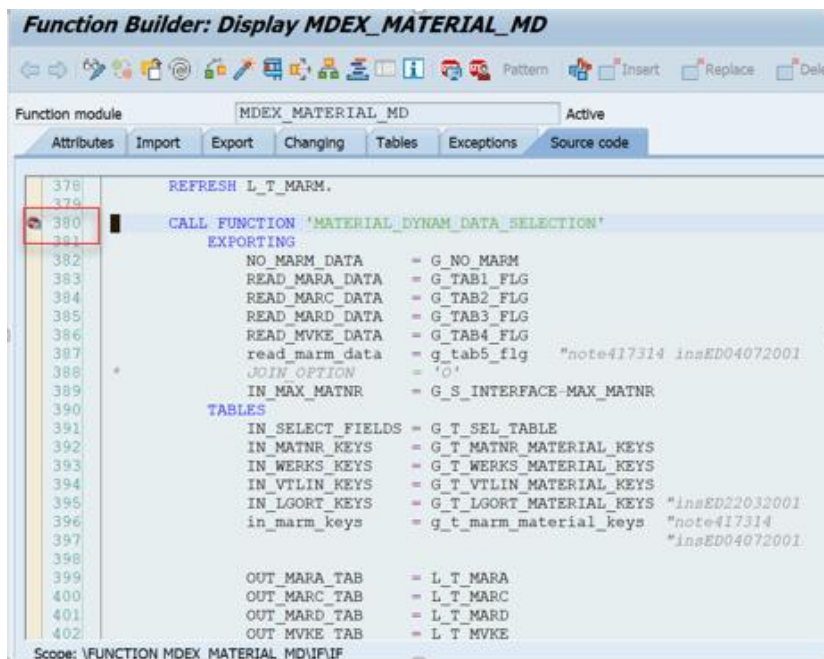
[https://help.sap.com/viewer/db6\\_dbacockpit/4505568cae262461e10000000a1553f7.html](https://help.sap.com/viewer/db6_dbacockpit/4505568cae262461e10000000a1553f7.html)

### 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 module : MATERIAL\_DYNAM\_DATA\_SELECTION



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

**Extractor Checker S-API**

DataSource: OMATERIAL\_ATTR

**Settings**

Request ID: TEST  
 Data Records / Calls: 100  
 Display Extr. Calls: 10  
 Update mode: F  
 Target:

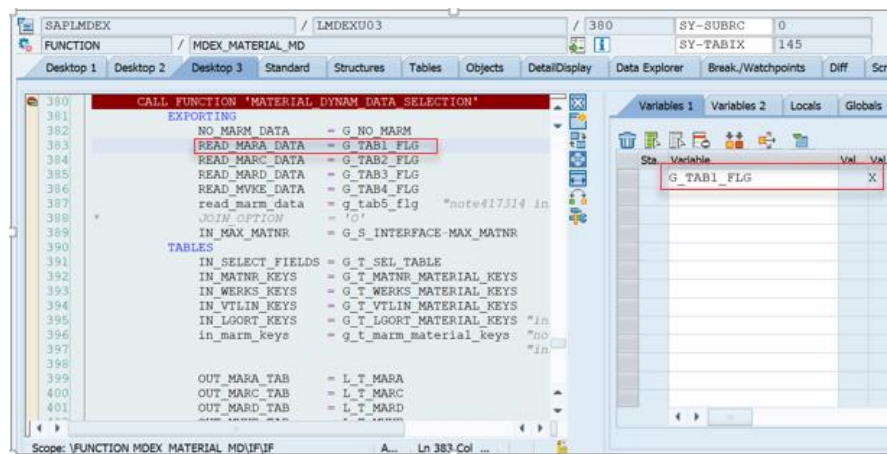
**Execution Mode**

☐ Debug Mode  
☐ Auth. Trace

**Selections (Internal Format)**

Field	From value	To value	Short Text
MATKL			Material Group
MATNR	000000000600001089		Material Number
MBSH			Industry Sector
MTART			Material Type

-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.

Desktop 1 Desktop 2 Desktop 3 Standard Structures Tables Objects DetailDis

Tables Table Contents

Table: L\_T\_MARA [ ]  
 Attributes: Standard [1x245(2892)]  
 Insert Column: Columns ... (243)

Row	MANDT [C(3)]	MATNR [C(18)]	SLED_BBD [C(1)]	ZZMAIN_CAT [C(3)]
1	100	000000000600001089	B	TEST MAIN2

Now we can see the L\_T\_MARA data as ALV display,

Debugger Table L_T_MARA[]				
Properties: Standard [1x245(2892)]				
INDEX	MANDT	MATNR	SLED_BBD	ZZMAIN_CAT
1	100	000000000600001089	B	TEST MAIN2

-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.

ABAP Debugger(2) (Exclusive)(awscddb01\_CCD\_00)

FUNCTION / MDEX\_MATERIAL\_MD

Desktop 1 Desktop 2 Desktop 3 Standard Structures Tables Objects DetailDisplay Data Explorer Break/Watchpoints Diff Script

```

410 IF SY-SUBRC <> 0.
411 IF 1 = 2. MESSAGE E011(R3). ENDIF.
412 LOG_WRITE 'E' "message type
413 "R3" "message class
414 "011" "message number
415 SY-SUBRC "message variable 1
416 " "message variable 2
417 RAISE ERROR_PASSED_TO_MESS_HANDLER.
418 ENDIF.
419 G COUNTER DATAPAKID = G COUNTER DATAPAKID + 1.
420 LOOP AT L_T_MARA.
421 MOVE-CORRESPONDING L_T_MARA TO E_T_BIW_MARA_S.
422 APPEND E_T_BIW_MARA_S.
423 ENDLOOP.
424 LOOP AT L_T_MARC.
425 MOVE-CORRESPONDING L_T_MARC TO E_T_BIW_MARC_S.
426 "provide additional attributes not contained in MARC
427 PERFORM MARC_ADD_ATTRIBUTES CHANGING E_T_BIW_MARC_S
428 "L_SUBRC.

```

Variables 1 Variables 2 Locals Globals Auto Memory An

Sta.	Variable	Val.	Val.
	L_T_MARA-SLED_BBD	B	
	L_T_MARA-ZZMAIN_CAT	TEST MAIN2	
	E_T_BIW_MARA_S-SLED_BBD	B	
	E_T_BIW_MARA_S-ZZMAIN_CAT	TEST MAIN2	

E\_T\_BIW\_MARA\_S is Datasource internal table which is BIW\_MARA\_S type, declared in the beginning of MDEX\_MATERIAL\_MD

Desktop 1 Desktop 2 Desktop 3 Standard Structures Tables Objects DetailDisplay Data Explorer Break/Watchpoints Diff Script

```

14 ** TABLES
15 ** I_T_SELECT TYPE SBIWA_T_SELECT OPTIONAL
16 ** I_T_FIELDS TYPE SBIWA_T_FIELDS OPTIONAL
17 ** E_T_BIW_MARA_S STRUCTURE BIW_MARA_S OPTIONAL
18 ** E_T_BIW_MARC_S STRUCTURE BIW_MARC_S OPTIONAL
19 ** E_T_BIW_MARD_S STRUCTURE BIW_MARD_S OPTIONAL
20 ** E_T_BIW_MVKE_S STRUCTURE BIW_MVKE_S OPTIONAL
21 ** E_T_BIW_MARH_S STRUCTURE BIW_MARH_S OPTIONAL
22 ** EXCEPTIONS
23 ** NO_MORE_DATA
24 ** ERROR_PASSED_TO_MESS_HANDLER
25 **
26 **
27 ** function acts as a S-API-compatible interface for the SD=
28 ** MATERIAL_DYNAM_DATA_SELECTION
29 **
30 **

```

Variables 1 Variables 2 Locals Globals Auto Memory An

Sta.	Variable	Val.	Val.
	L_T_MARA-SLED_BBD	B	
	L_T_MARA-ZZMAIN_CAT	TEST MAIN2	
	E_T_BIW_MARA_S-SLED_BBD	B	
	E_T_BIW_MARA_S-ZZMAIN_CAT	TEST MAIN2	

Thats all steps for verification.



For other common Master Data ODP extractors,

You can check out ERP SE37 program, since the same technique applied here.

MDEX\_MATERIAL\_MD , line from 101 to 147

MDEX\_CUSTOMER\_MD , line from 99 to 115

MDEX\_VENDOR\_MD, line from 95 to 106

For MDEX\_MATERIAL\_MD, data fetched from MATERIAL\_DYNAM\_DATA\_SELECTION dynamically.

For MDEX\_CUSTOMER\_MD, data fetched from CUSTOMER\_DYNAM\_DATA\_SELECTION dynamically.

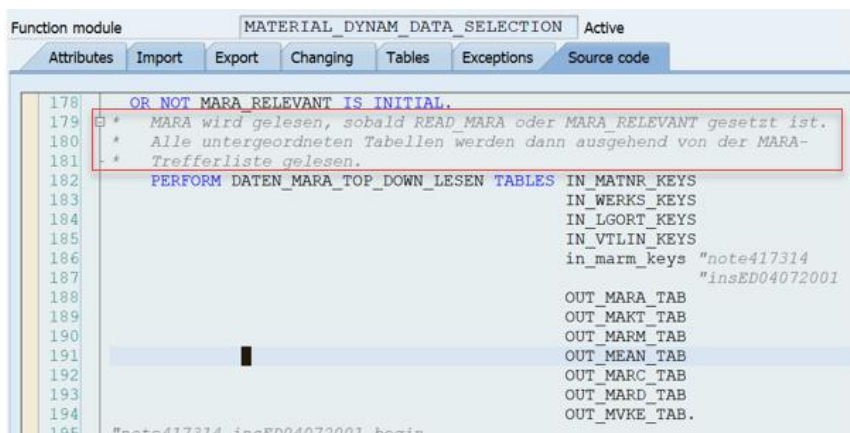
For MDEX\_VENDOR\_MD, data fetched from VENDOR\_DYNAM\_DATA\_SELECTION dynamically.

For our example, 0MATERIAL\_ATTR,

inside of 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.
```



İçerik	LMGSEF02	Aktif
344		OT_MARD_TAB STRUCTURE MARD
345		OT_MVKE_TAB STRUCTURE MVKE.
346		
347	IF BLOCK_SIZE > 0.	
348	* Blockweises Lesen	
349	IF READ_NEXT_BLOCK IS INITIAL.	
350	* Cursor zum Lesen der MARA's öffnen	
351	* (nur beim Lesen des ersten Blocks)	
352	DESCRIBE TABLE IT_MATNR_KEYS LINES SY-TFILL.	
353	IF SY-TFILL > 0.	
354	OPEN CURSOR WITH HOLD MARA_CURSOR FOR	
355	SELECT * FROM MARA	
356	FOR ALL ENTRIES IN IT_MATNR_KEYS	
357	WHERE MATNR = IT_MATNR_KEYS-MATNR	
358	AND (MARA_WHERE_COND).	
359	ELSE.	
360	OPEN CURSOR WITH HOLD MARA_CURSOR FOR	
361	SELECT * FROM MARA	
362	WHERE (MARA_WHERE_COND).	
363	ENDIF.	
364	MARA_CURSOR_OPEN = 'X'.	
365	ENDIF.	

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

İçerik	LMGSEF02	Aktif
344		OT_MARD_TAB STRUCTURE MARD
345		OT_MVKE_TAB STRUCTURE MVKE.
346		
347	IF BLOCK_SIZE > 0.	
348	* Blockweises Lesen	
349	IF READ_NEXT_BLOCK IS INITIAL.	
350	* Cursor zum Lesen der MARA's öffnen	
351	* (nur beim Lesen des ersten Blocks)	
352	DESCRIBE TABLE IT_MATNR_KEYS LINES SY-TFILL.	
353	IF SY-TFILL > 0.	
354	OPEN CURSOR WITH HOLD MARA_CURSOR FOR	
355	SELECT * FROM MARA	
356	FOR ALL ENTRIES IN IT_MATNR_KEYS	
357	WHERE MATNR = IT_MATNR_KEYS-MATNR	
358	AND (MARA_WHERE_COND).	
359	ELSE.	
360	OPEN CURSOR WITH HOLD MARA_CURSOR FOR	
361	SELECT * FROM MARA	
362	WHERE (MARA_WHERE_COND).	
363	ENDIF.	
364	MARA_CURSOR_OPEN = 'X'.	
365	ENDIF.	
366		

## Conclusion:

This example is adding fields from one logistics master data to DSODP extractor, since it is one master table, i mean 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](https://help.sap.com/saphelp_ewm900/helpdata/en/97/9bddd4cebe423f9eb5767a275b2d78/content.htm?no_cache=true)

Thank you.

Cihan Ekin