How To... Delete EAN Assignments from 0RPA_MEAN

Applicable Releases:

SAP BW >= Release 3.5 and above

BI Content >= Release 3.52 and above

Plug-In >= 2004.1 and above

Version 1.01

December 2010

THE BEST-RUN BUSINESSES RUN SAP™

Document History

Document Version	Description
1.00	First official release of this guide
1.01	First correction of this guide

Typographic Conventions

Type Style	Description
Example Text	Words or characters quoted from the screen. These include field names, screen titles, pushbuttons labels, menu names, menu paths, and menu options.
	Cross-references to other documentation
Example text	Emphasized words or phrases in body text, graphic titles, and table titles
`Example text`	File and directory names and their paths, messages, names of variables and parameters, source text, and names of installation, upgrade and database tools.
`Example text`	User entry texts. These are words or characters that you enter in the system exactly as they appear in the documentation.
`<Example text>`	Variable user entry. Angle brackets indicate that you replace these words and characters with appropriate entries to make entries in the system.
`EXAMPLE TEXT`	Keys on the keyboard, for example, `F2` or `ENTER`.

Icons

Icon	Description
🔺	Caution
💡	Note or Important
⚙	Example
⬆	Recommendation or Tip

Table of Contents

1. Business Scenario

You are extracting EANs (European Article Numbers) from an SAP Retail system to SAP Netweaver BW into the InfoObject 0RPA_MEAN or a similar customer InfoObject following the instructions of

SAP Note 835111

Delta extraction MARM und MEAN for POS Analytics Content

You use the InfoObject that contains the assignments of SAP article numbers to EANs to execute material master data checks and enrichments in POS DM (PIPE).

If you are reusing EANs in your material master maintenance for the same EAN number type in SAP ERP, you are faced with the issue that the deletions of EAN article assignments are not uploaded to the BW InfoObjects by the extractor plug-in. Only changed or newly created entries of the MEAN table in ECC Retail are extracted by the generic extractor. As a consequence, it can happen that your 0RPA_MEAN or corresponding customer InfoObject contains entries with multiple assignments of the same EAN to different SAP article numbers.

Characteristic 0RPA_MEAN - maintain master data: List

	Sequence N	UoM for Di	EAN to Article	EAN/UPC	EAN Number	Ind.: Main
	2	PC	000000000000001000	4022652000203	HE	
	2	PC	000000000000001001	4022652000203	HE	

Figure: Multiple EAN to article assignment in 0RPA_MEAN

2. Background Information

The solution proposed is containing:

- A generic extractor in SAP Retail that selects the deleted EAN article assignments of the MEAN table as delta (using a function module)

- The corresponding extract structure and a dedicated DataSource that can be replicated to the BW system

- Some update logic in the BW data flow into the InfoObject that is reassigning the EAN article combinations that are deleted in SAP Retail

3. Prerequisites

- SAP Netweaver BW with BI Content >= Release 3.52

- Plug-In >= 2004.1

- Extractor for MEAN data from SAP Retail into SAP BW 0RPA_MEAN as described in SAP Note 835111

4. Step-by-Step Procedure

To extract the deleted EAN article assignments from SAP Retail to BW, it is necessary to execute different development steps in the following sequence:

SAP Retail

- A generic extractor in SAP Retail that selects the deleted EAN article assignments in MEAN table as delta (using a function module). An additional ALE change pointer message type is created to be able to extract the deleted EAN article assignments.

- The corresponding extract structure and a dedicated DataSource that can be replicated to SAP BW

SAP BW

- Some update logic in the BW data flow into the InfoObject 0RPA_MEAN that is reassigning the EAN article combinations that are deleted in SAP Retail

4.1 Creation of generic extractor in SAP Retail

1. Logon to SAP Retail

2. Create a new logical message type

   Go to transaction WE81

Change View "EDI: Logical Message Types": Overview

	Message Type	Short text
	/BEV1/NEMAT	Beverage Material Master Enhancements for Mass Mainter
	/CWM/MATMAS_BAPI	Create and Change Material Master Data
	/CWM/MATMAS_MASS_BAPI	BAPI for Mass Maintenance of Material Data
	/CWM/MBGMCR	Post Goods Movements with MB_CREATE_GOODS_MOVE
	/CWM/SHP_IBDLV_CHANGE	Change Inbound Delivery
	/CWM/SHP_IBDLV_CONFIRM_DEC	BAPI Confirmation (CWM Save Replica Inbound)
	/CWM/SHP_IBDLV_SAVE_REPLICA	BAPI Function Module for Replication of Inbound Deliveries
	/CWM/SHP_OBDLV_CHANGE	BAPI for Change to Outbound Delivery
	/CWM/SHP_OBDLV_CONFIRM_DEC	BAPI Confirmation (CWM Save Replica)
	/CWM/SHP_OBDLV_SAVE_REPLICA	BAPI Function Module for Replication of Outbound Deliverie
	/CWM/SHP_OBDLV_SPLIT_DECENTRAL	BAPI Delivery Split (CWM Save Replica)
	/CWM/STPPOD	CWM Acknowledgement of Receipt (POD)
	/DSD/HH_CONTROL	DSD Control Output (eod, del, drq, drp, eou)
	/DSD/HH_CREDITDATA	DSD Credit Exposure and Credit ILmit
	/DSD/HH_CUSTMASTEXT	DSD Customer Master Additions

Create a new entry for message type 'Z_EX_MEAN_DEL' with a short text and save the entry

New Entries: Overview of Added Entries

EDI: Logical Message Types	
Message Type	Short text
Z_EX_MEAN_DEL	extract deleted EAN article assignments
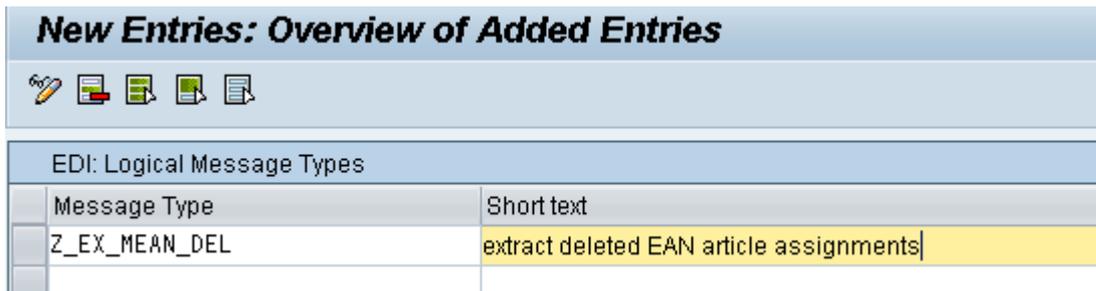
3. Create document items for newly created message type

Go to transaction BD52 to assign the MEAN table fields to the new message type Z_EX_MEAN_DEL

Maintain Table Views: Initial Screen

Find Maintenance Dialog

Determine Work Area: Entry

	Work Area
Message Type	Z_EX_MEAN_DEL

Further select cond.   Append

Change View "Change document items for message

New Entries   BC Set: Change Field Values

Message Type	Z_EX_MEAN_DEL

Change document items for message type		
Object	Table Name	Field Name
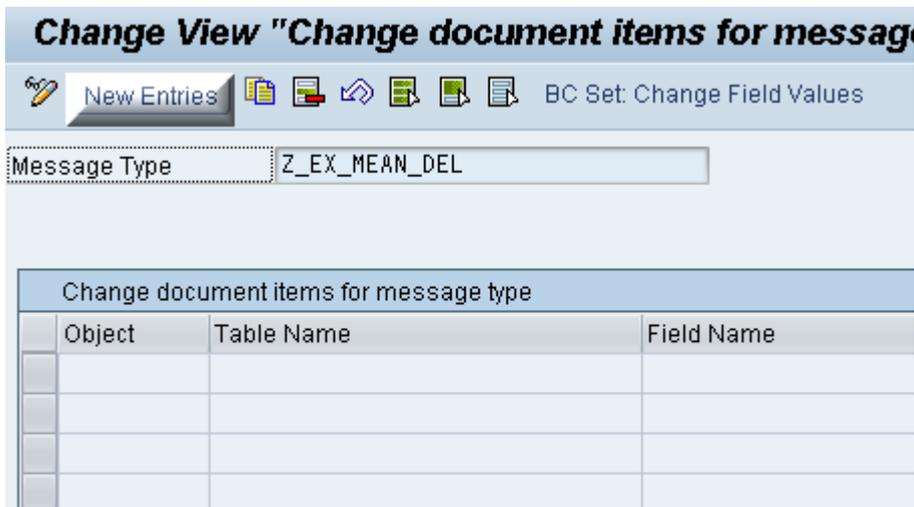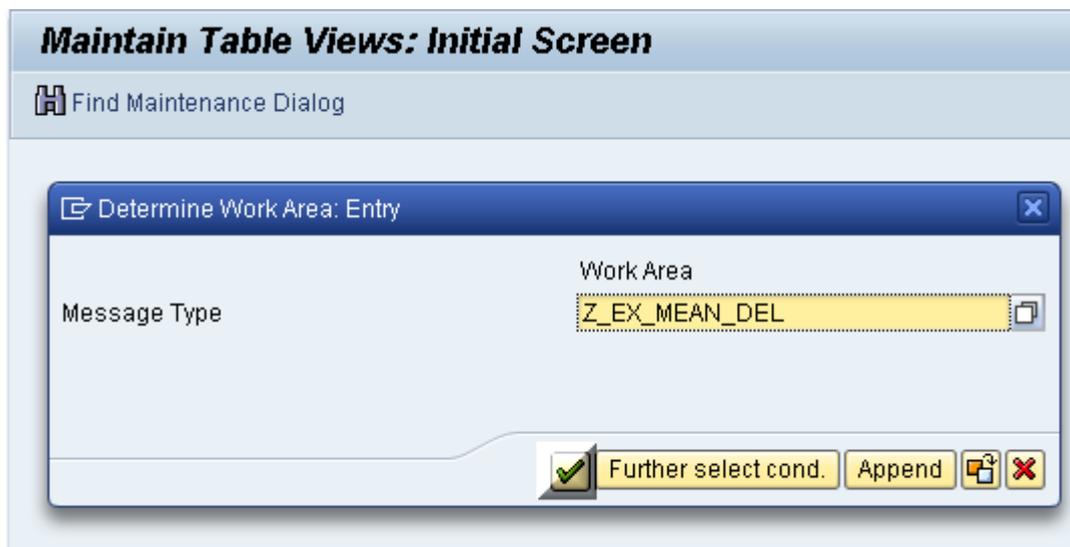
Enter the following entries as change relevant fields for message type Z_EX_MEAN_DEL

Object	Table Name	Field Name
MAT_FULL	DMEAN	MEINH
MAT_FULL	DMEAN	EAN11
MAT_FULL	DMEAN	EANTP

Save the entries
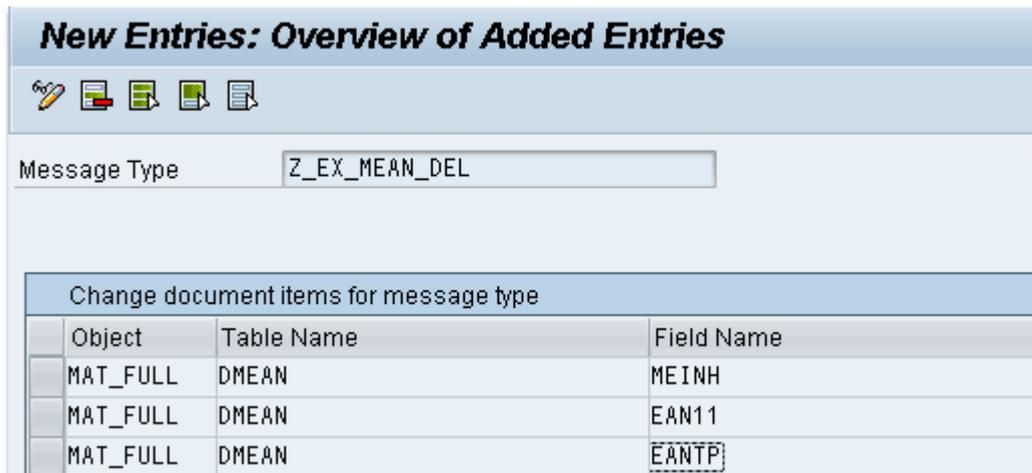
4. Activate the newly created change pointer message type

    Go to transaction BD50 to activate message type Z_EX_MEAN_DEL
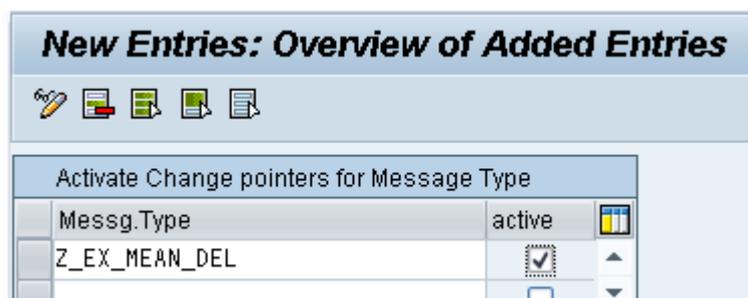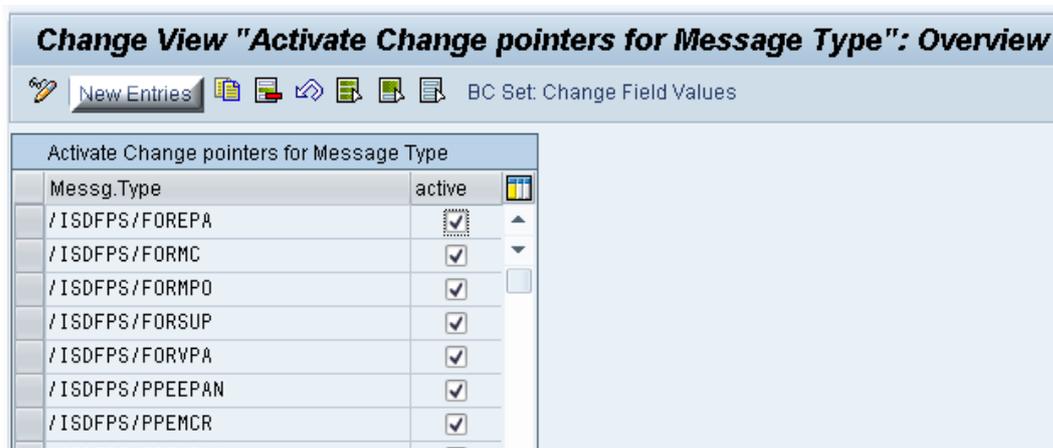
    Save the new entry

5. Create a new data structure

    Go to transaction SE11 and create a new structure 'ZEAN_EX_DELETED_ENTRIES' with following entries:

Component	Component type
MATNR	MATNR
MEINH	MEINH
EAN11	EAN11

ABAP Dictionary: Initial Screen

○ Database table
○ View   V_TBDA2

⦿ Data type   ZEAN_EX_DELETED_ENTRIES
○ Type Group

○ Domain
○ Search help
○ Lock object

🔍 Display   ✎ Change   ☐ Create
                                    Create

ABAP Dictionary: Initial Screen

⌷ Create Type ZEAN_EX_DELET...

○ Data element
⦿ Structure
○ Table type

✓ ✗
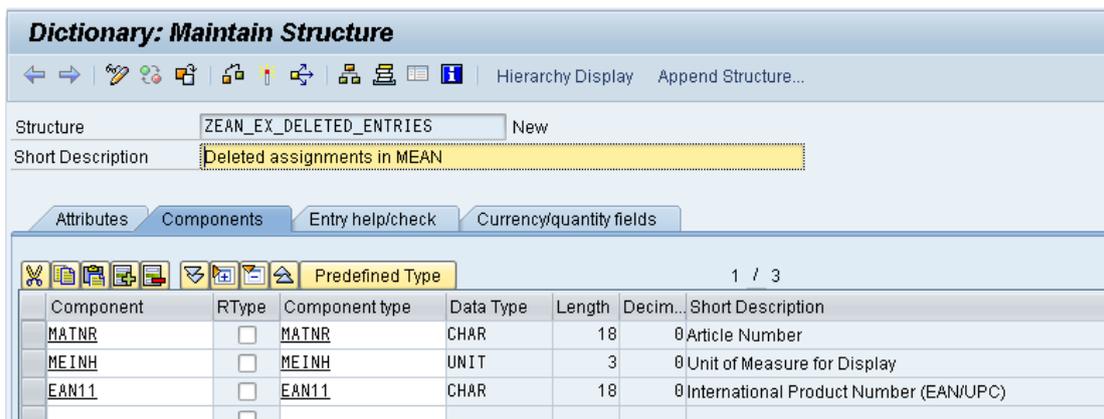
Activate the structure ZEAN_EX_DELETED_ENTRIES

6. Create a new function module

Go to transaction SE37 and create the new function module 'Z_EAN_EX_DELETED_ENTRIES' with a short text and assign it to a function group created for BW extraction developments
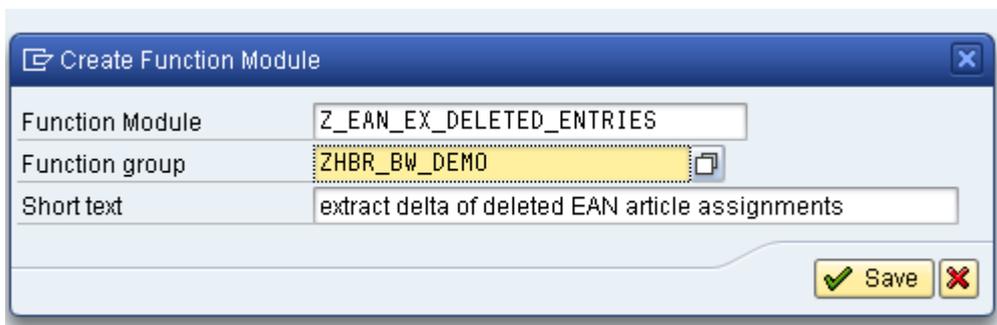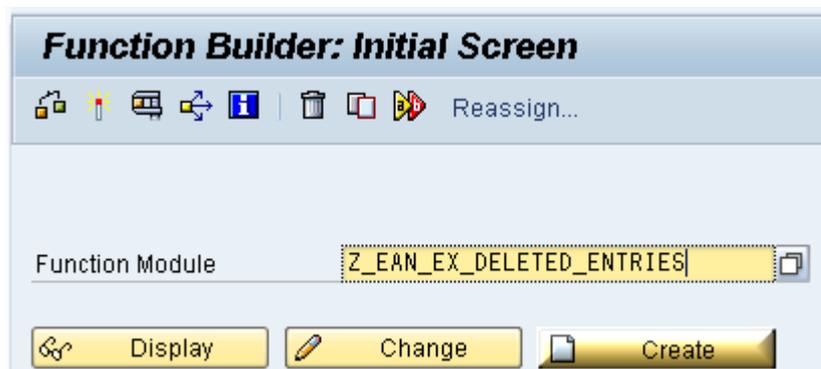
💡 **Important**

To properly create the function module, it is necessary to execute the next steps in the correct sequence

In the function module builder, select in the menu Goto -> Global Data
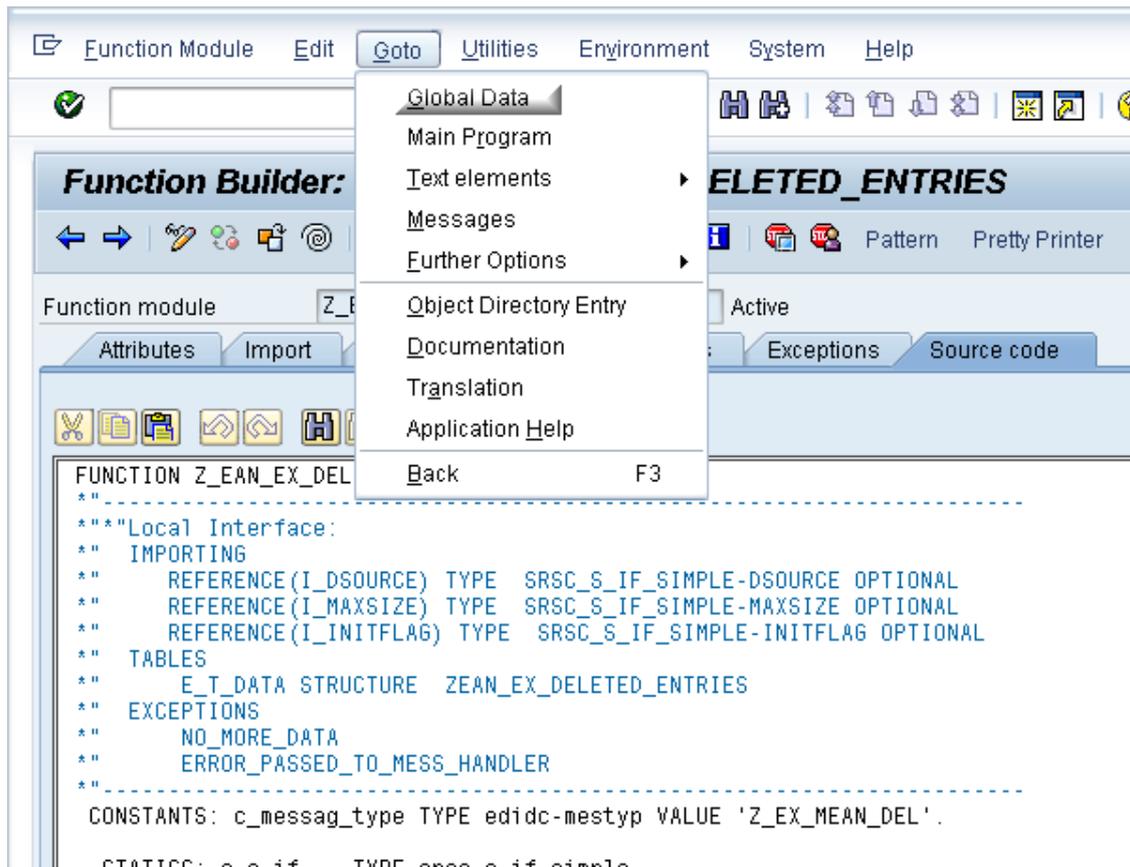
Add the coding:

```
INCLUDE lrsaxd01.
```

Activate the include.
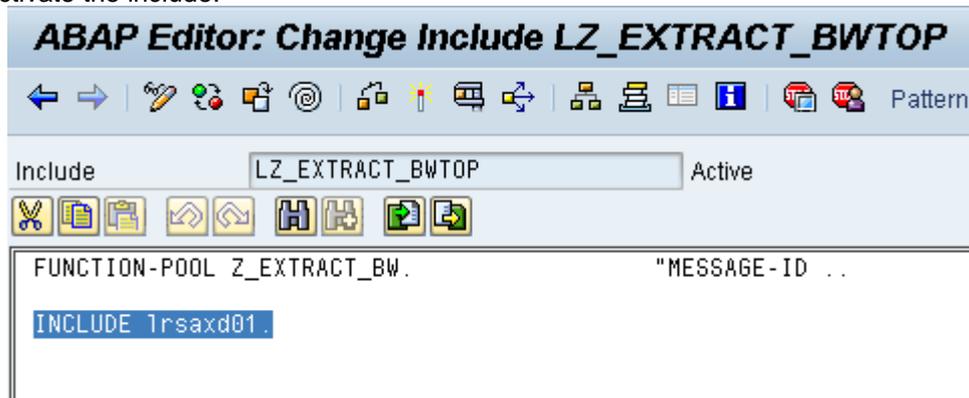
Go 1 step back to the function module builder

Press the 'Import' Tab

Maintain the following entries

Parameter name	Typing	Associated Type	optional	pass value
I_DSOURCE	TYPE	SRSC_S_IF_SIMPLE-DS	x	
I_MAXSIZE	TYPE	SRSC_S_IF_SIMPLE-MA	x	
I_INITFLAG	TYPE	SRSC_S_IF_SIMPLE-IN	x	
L_READ_ONLY	TYPE	SBIWA_FLAG	x	x

Press the 'Tables' Tab

Maintain the following entry

Parameter name	Typing	Associated Type	optional
E_T_DATA	LIKE	ZEAN_EX_DELETED_ENTRIES	x

Press the 'Exceptions' Tab

Maintain the following entries:

Exception

NO_MORE_DATA

ERROR_PASSED_TO_MESS_HANDLER
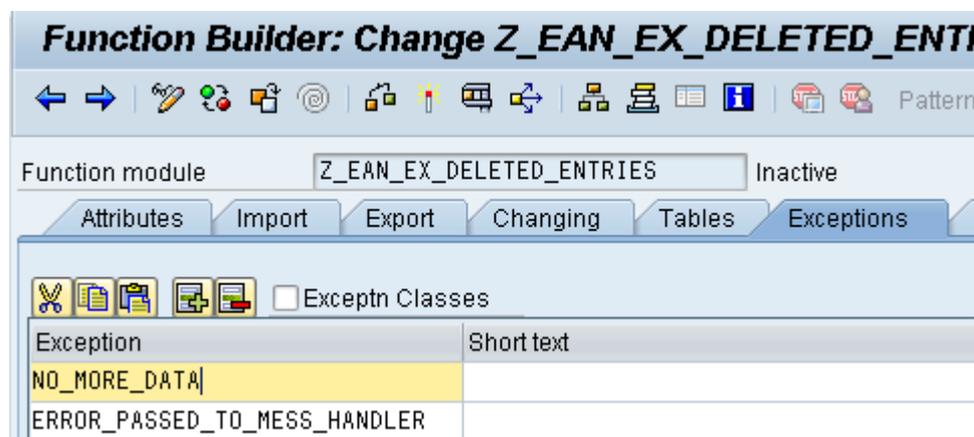
Press the 'Source Code' Tab

Add the coding for function module Z_EAN_EX_DELETED_ENTRIES (see appendix)

Activate the function module

Important

This function module will be the source for the generic extractor in SAP Retail. The function module is extracting the change documents for deleted assignments (in a delta mode). Newly created or changed EAN assignments are not considered at all.

7. Create the generic extractor

Go to transaction RSO2 and create the generic DataSource 'Z_MEAN_DEL_ATTR' for uploading master data attributes

Assign it to the relevant application component (i.e. IS-R-IO for IS retail master data), maintain short, medium and long text and select the newly created function module Z_EAN_EX_DELETED_ENTRIES as source and ZEAN_EX_DELETED_ENTRIES as extract structure.

4.2 Uploading deleted EAN assignments to SAP BW

1. Logon to SAP BW

2. Replicate newly created generic DataSource from SAP Retail source system to SAP BW

    Go to transaction RSA1 and select the relevant SAP source system
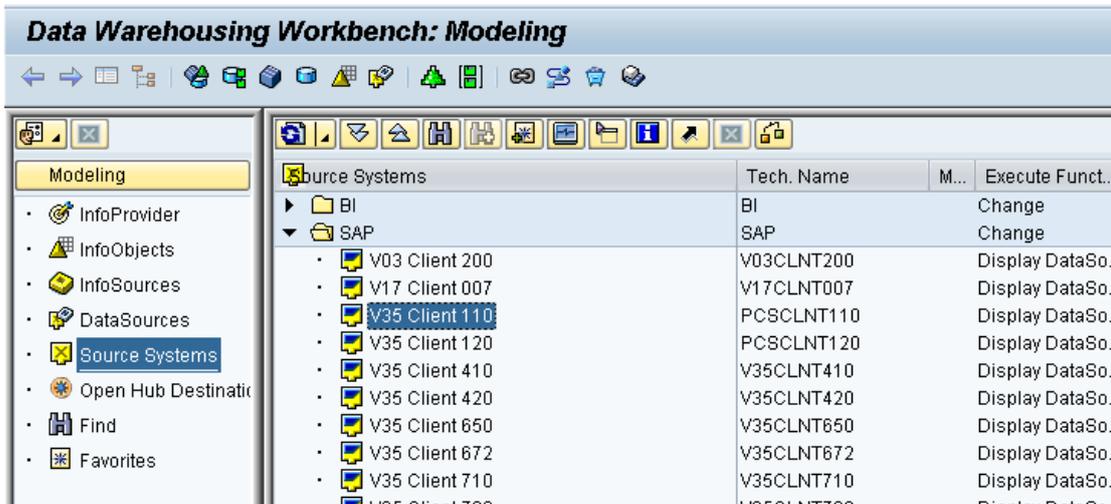
Select the application area defined for the generic DataSource and replicate the metadata
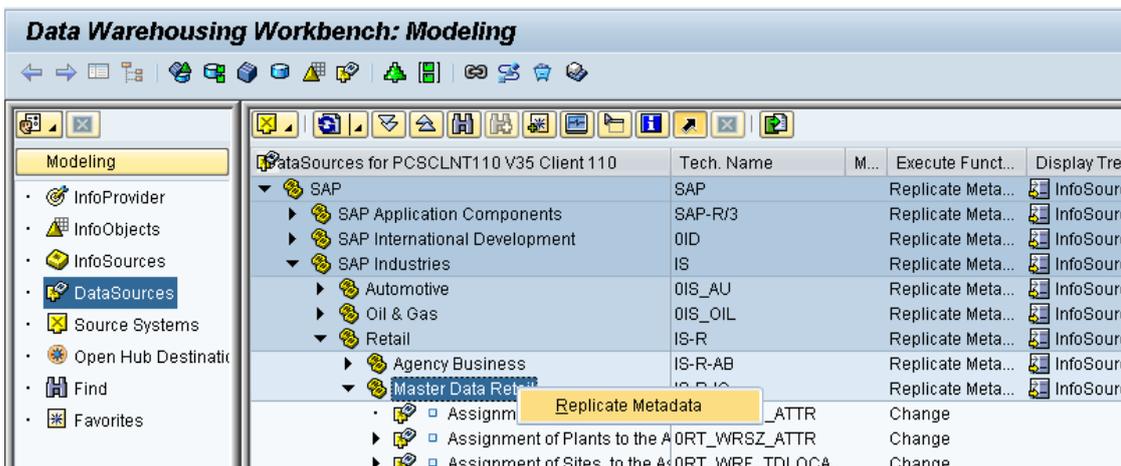
Replicate the DataSource as 7.x DataSource
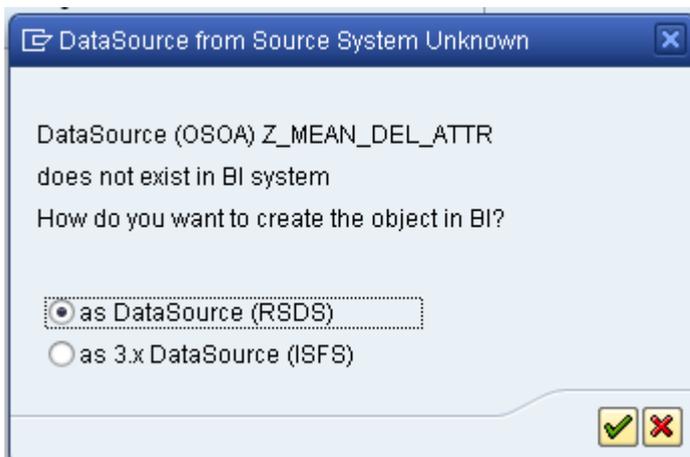
3. create transformation from DataSource to InfoObject

    activate the DataSource Z_MEAN_DEL_ATTR
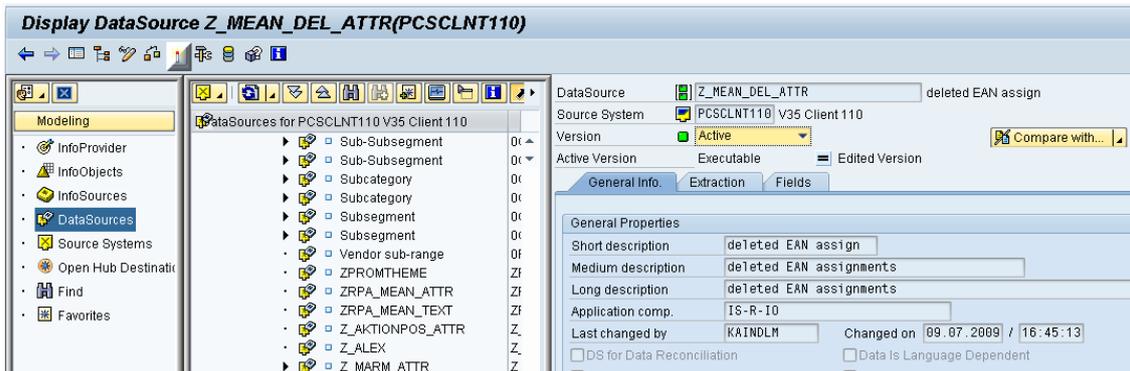
Right click on the DataSource and select 'create transformation'
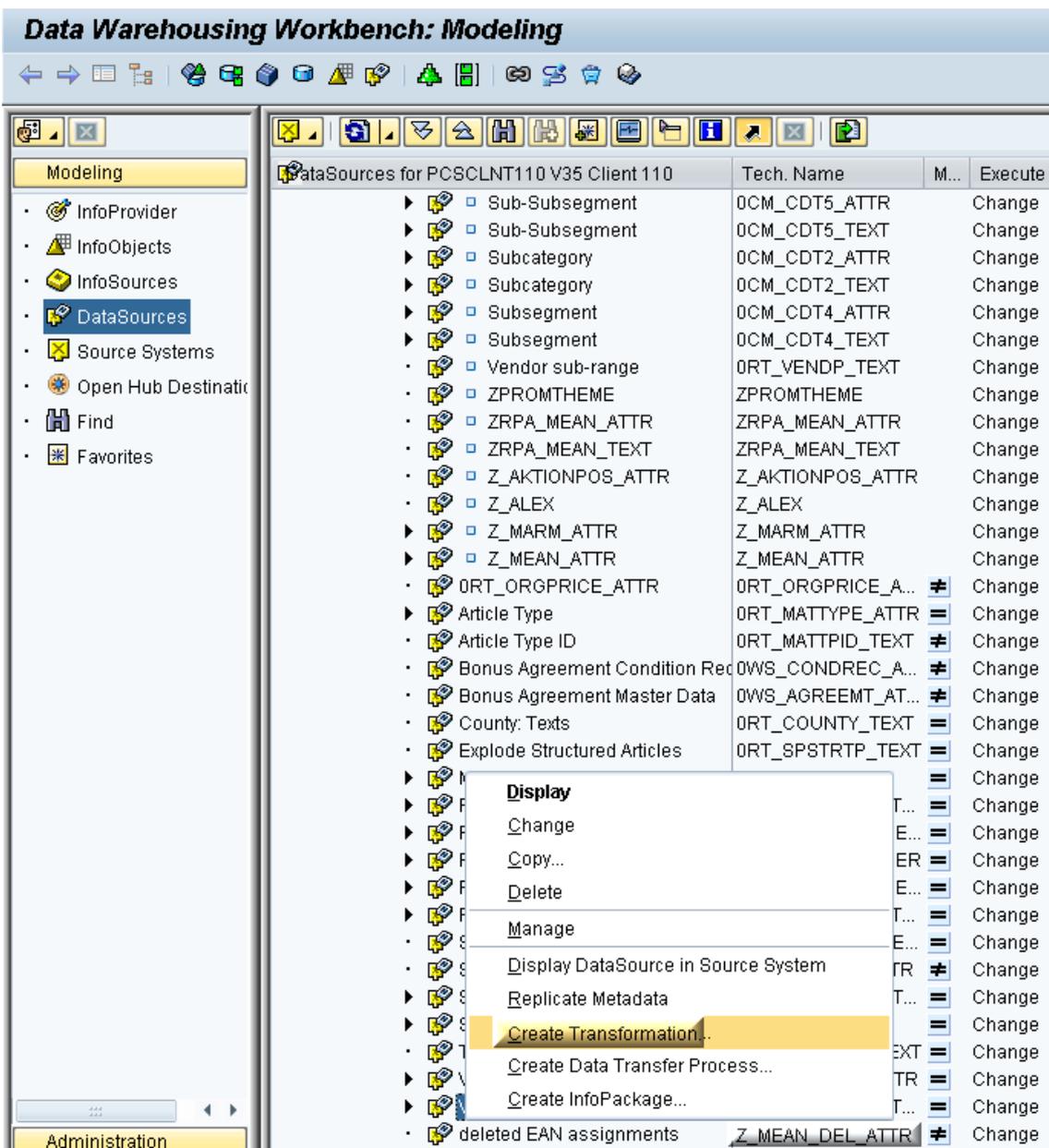
Assign the transformation to the attributes of the InfoObject 0RPA_MEAN

Map the fields from the DataSource Z_MEAN_DEL_ATTR to the InfoObject attributes of 0RPA_MEAN in the following sequence:

MATNR -> 0RPA_MEAN

MEINH -> 0RPA_DISQU

Create a field routine for the field 0RPA_LFNUM

Important

As the extractor from SAP Retail is not containing all relevant key fields of the InfoObject, it is necessary to implement some update logic in this transformation. The unique key of the MEAN table in SAP Retail and of the 0RPA_MEAN InfoObject in SAP BW is MATNR (article number; 0RPA_MEAN), MEINH (unit of measure for display, 0RPA_DISQU) and LFNUM (consecutive sequence number, 0RPA_LFNUM) but the extractor created cannot deliver this LFNUM sequence number. For that reason, it is necessary to identify the current valid EAN article and unit of measure combination stored in the InfoObject. This logic is implemented by the field routine. As the extractor is delivering the deleted article EAN assignments, it is important to set the mapping into the InfoObject field 0EANUPC to initial.

The field routine to 0RPA_LFNUM is containing all fields from the DataSource as source fields.

Copy the ABAP coding into the field routine to 0RPA_LFNUM (see appendix)

Save the routine and the transformation and activate it.

4. Create InfoPackage to upload the deleted EAN article assignments

Right click on the DataSource and select 'Create InfoPackage'
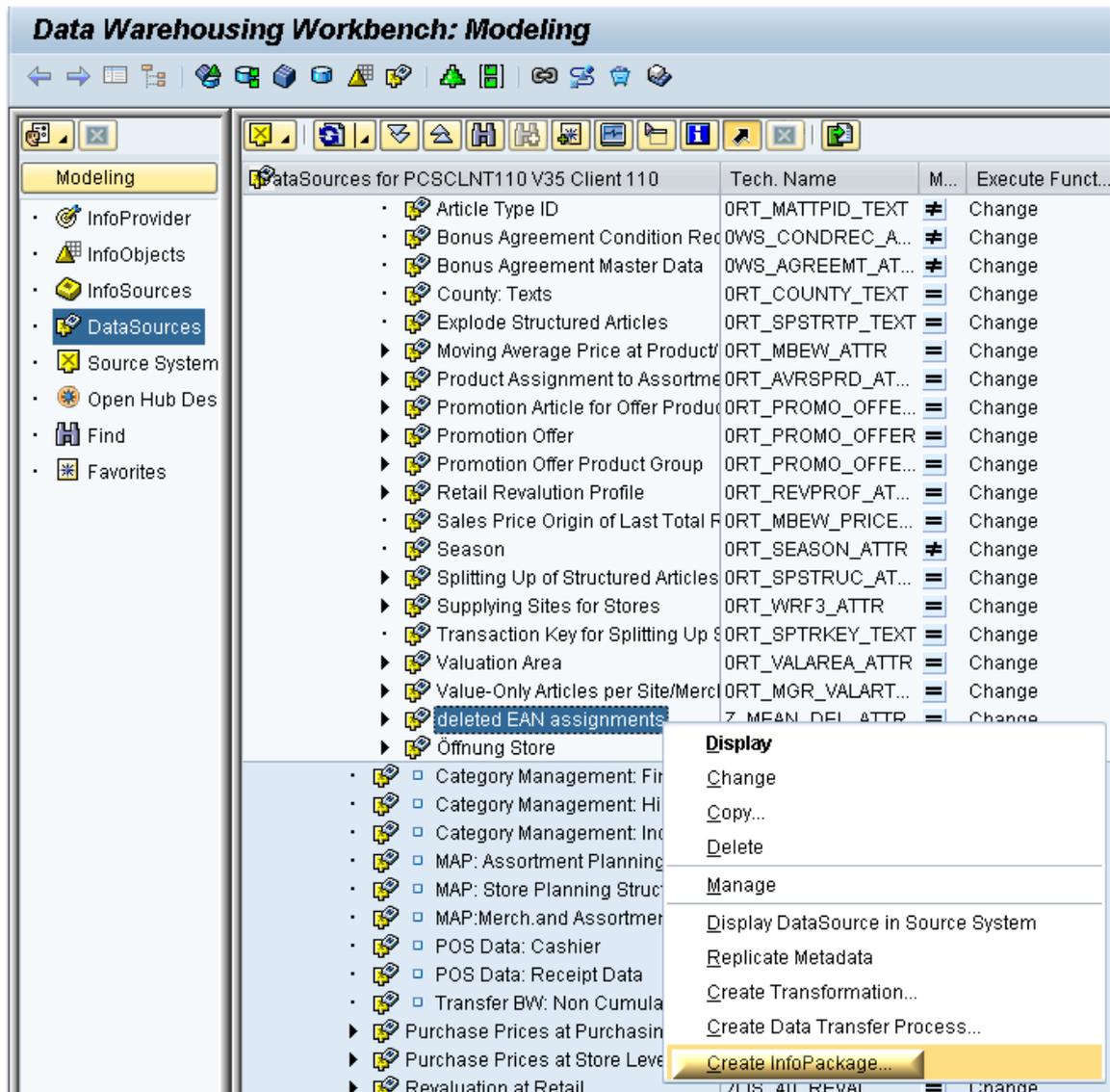
Define a name for the InfoPackage and save it
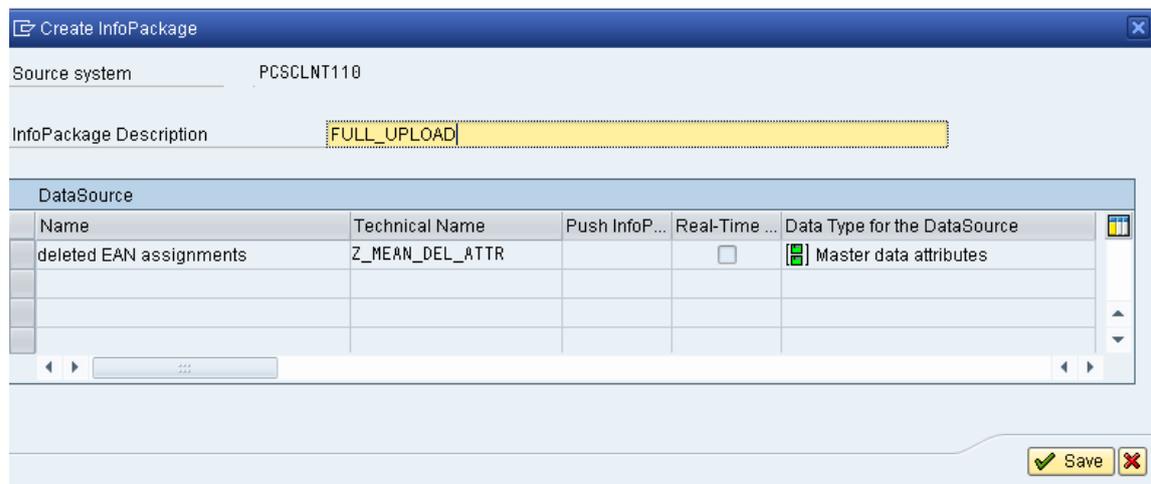
Select the 'update' tab
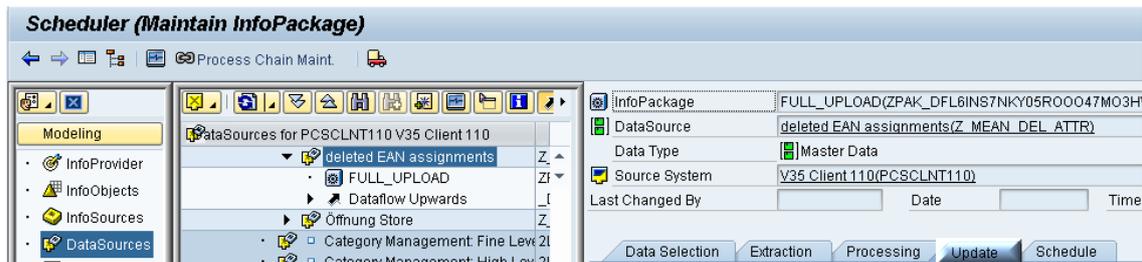
Select full update as update mode
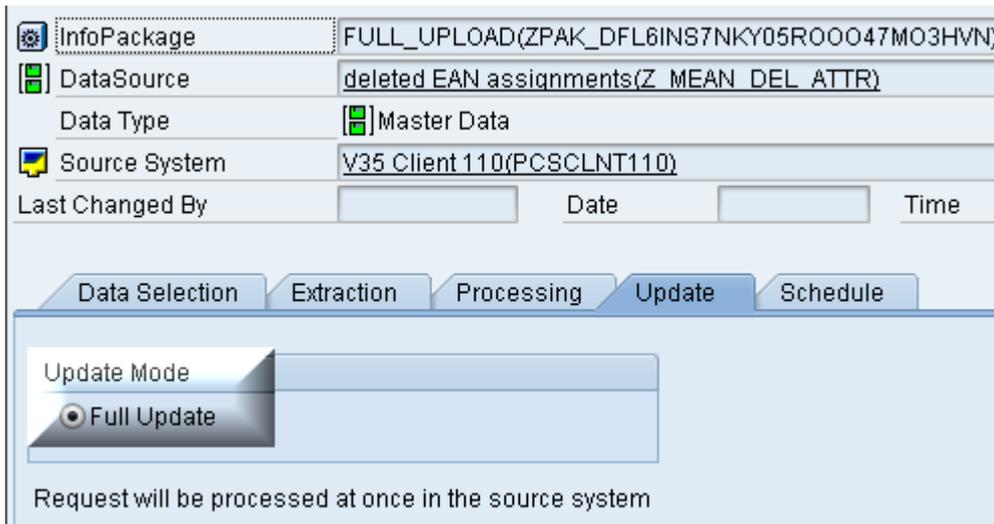
Important

The InfoPackage must be a full update InfoPackage as delta handling is not supported by generic extractors that are based on function modules. The function module that is the source of the generic extractor is delivering the delta updates for the SAP Retail system. The consequence of that fact with regard of what steps has to be executed in case that an upload from SAP Retail to BW is failing for this scenario will be described in the appendix section 'Error handling'.

Save the InfoPackage

5. Create data transfer process

As 7.x DataSources are only updating the corresponding PSA tables, it is necessary to create a DTP to upload the data from the PSA table of the DataSource into the InfoObject. The DTP itself is executed in delta update mode
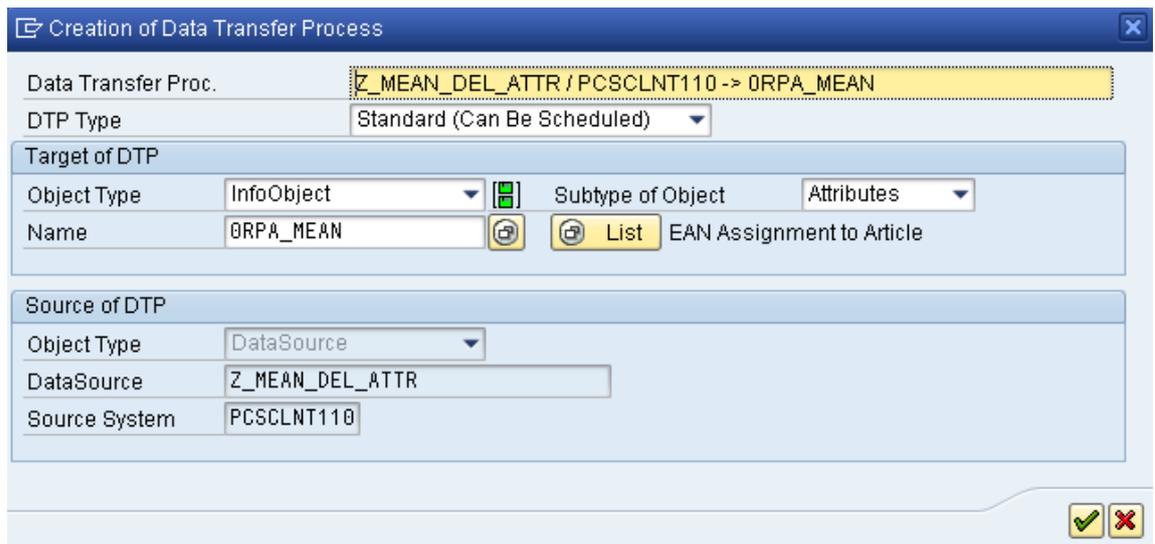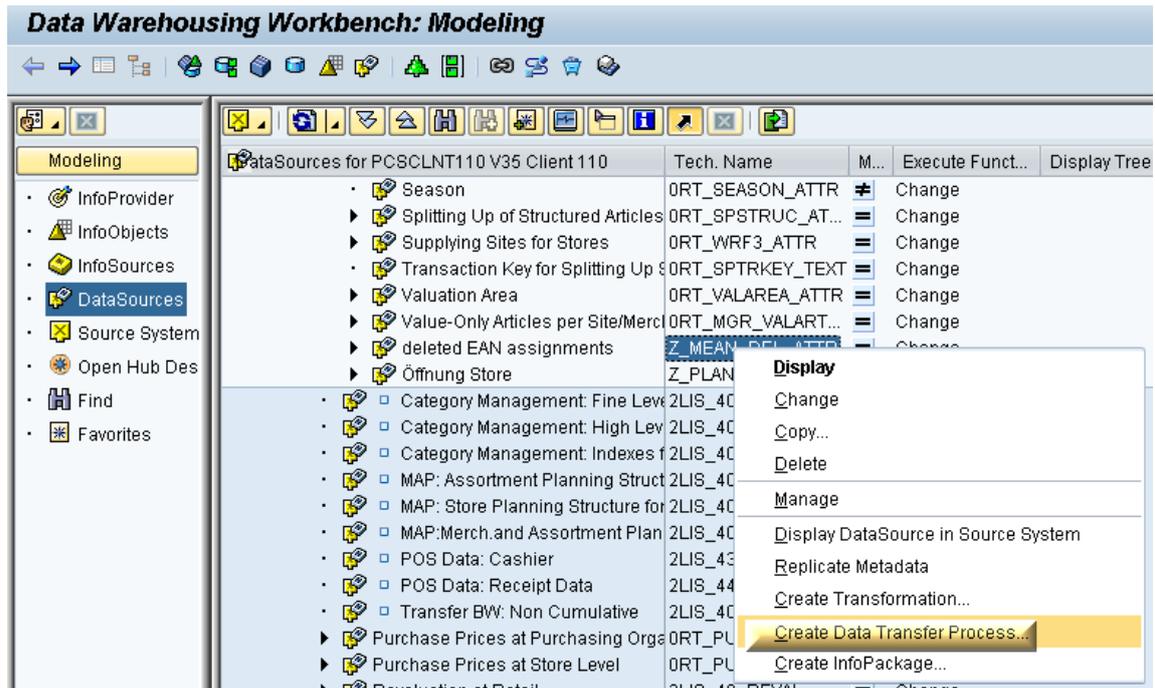
Activate the DTP
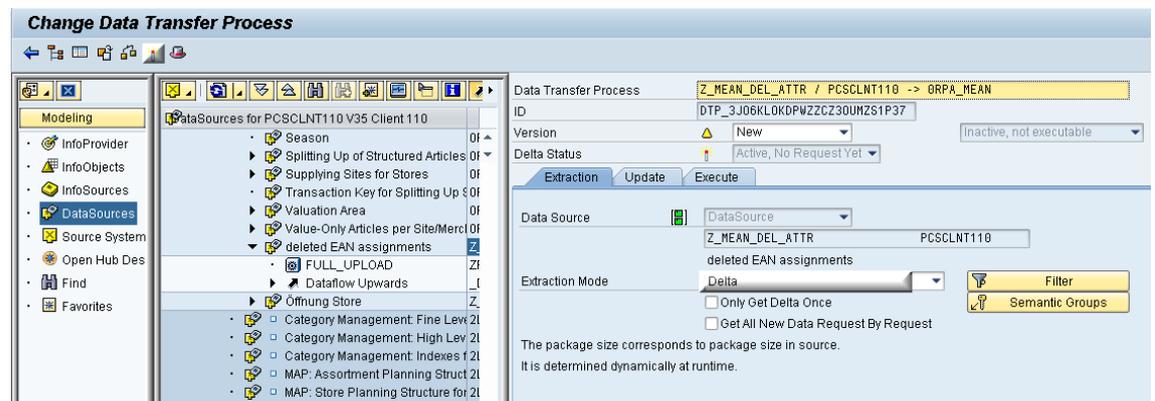
5. Appendix

Appendix A - Error Handling

As the extractor is uploading the delta of deleted EAN article combinations, it is necessary to execute the following steps in case that the InfoPackage is failing during processing because it is not possible to repeat a failed delta upload (even if the InfoPackage is a full upload package, the function module used by the extractor is set up like a delta upload).

Delete all entries in the InfoObject 0RPA_MEAN
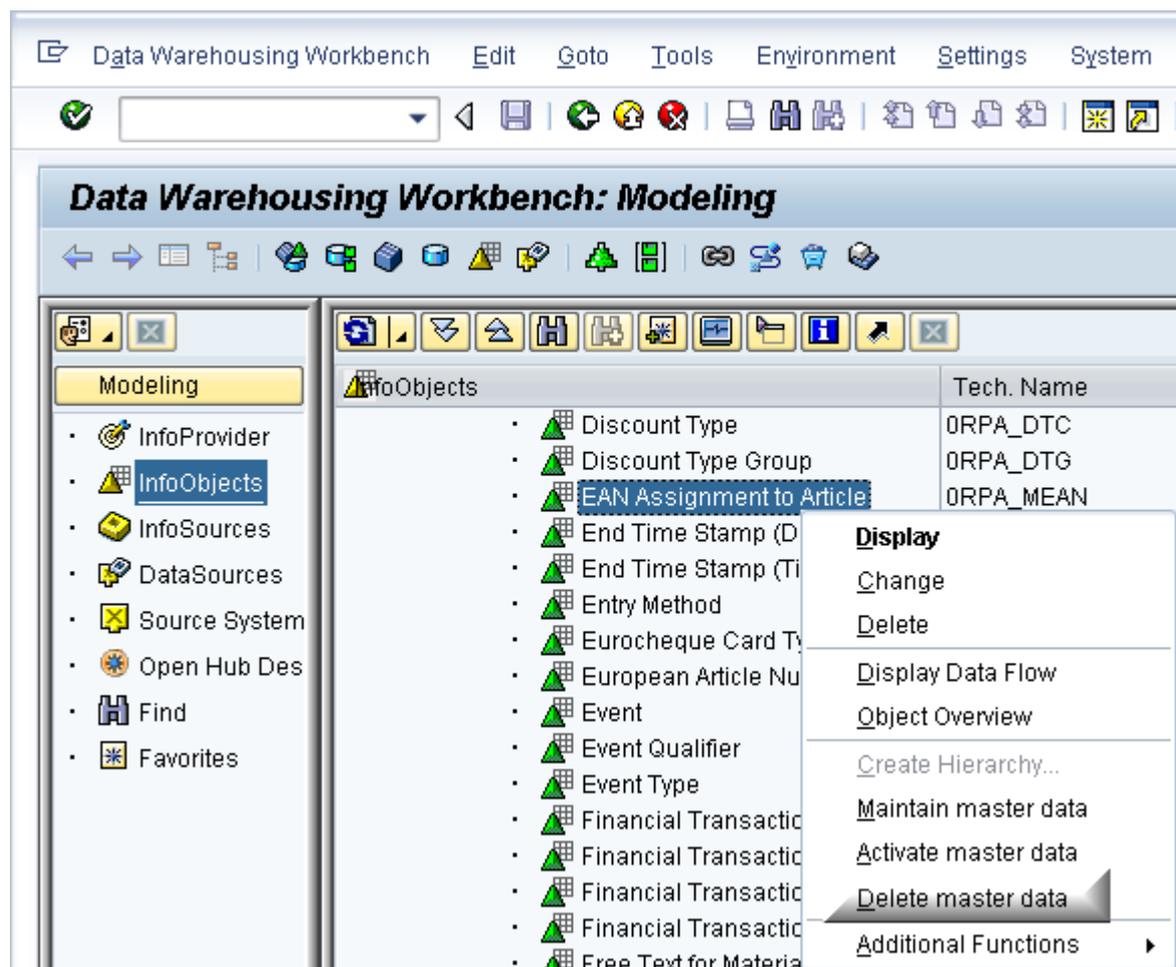
Execute the Full update InfoPackage of the DataSource created following the instructions of note 835111 'Delta extraction MARM und MEAN for POS Analytics Content' that is extracting the attributes for the InfoObject 0RPA_MEAN.
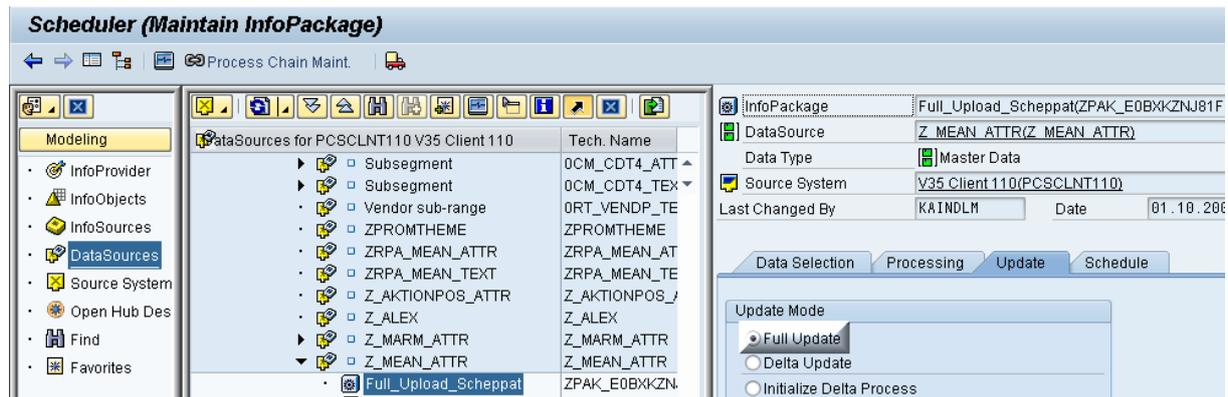
As this extractor is uploading all valid EAN article assignments from SAP Retail to BW, the consistency of the 0RPA_MEAN InfoObject is given.

After that, it is sufficient to execute again only the Delta update InfoPackage for that DataSource and afterwards the InfoPackage to upload the deleted EAN article assignments described in this how to paper.

Appendix B – ABAP Coding

Function module Z_EAN_EX_DELETD_ENTRIES

```
FUNCTION Z_EAN_EX_DELETED_ENTRIES.
*"----------------------------------------------------------------------
*"*"Local Interface:
*"  IMPORTING
*"     REFERENCE(I_DSOURCE) TYPE  SRSC_S_IF_SIMPLE-DSOURCE OPTIONAL
*"     REFERENCE(I_MAXSIZE) TYPE  SRSC_S_IF_SIMPLE-MAXSIZE OPTIONAL
*"     REFERENCE(I_INITFLAG) TYPE  SRSC_S_IF_SIMPLE-INITFLAG OPTIONAL
*"     VALUE(L_READ_ONLY) TYPE  SBIWA_FLAG OPTIONAL
*"  TABLES
*"     E_T_DATA STRUCTURE  ZEAN_EX_DELETED_ENTRIES OPTIONAL
*"  EXCEPTIONS
*"     NO_MORE_DATA
*"     ERROR_PASSED_TO_MESS_HANDLER
*"----------------------------------------------------------------------
 CONSTANTS: c_messag_type TYPE edidc-mestyp VALUE 'Z_EX_MEAN_DEL'.

  STATICS: s_s_if    TYPE srsc_s_if_simple.

  FIELD-SYMBOLS: <l_s_change_pointers> TYPE bdcp.

  DATA: l_no_records           TYPE i,
        l_s_data               TYPE zean_ex_deleted_entries,
        l_s_chng_pntrs_idents TYPE bdicpident,
        l_t_chng_pntrs_idents TYPE TABLE OF bdicpident,
        l_t_change_pointers   TYPE TABLE OF bdcp.


  IF i_initflag = sbiwa_c_flag_on.

***********************************************************************
* Initialization
***********************************************************************


* Check DataSource validity
    IF i_dsource <> 'Z_MEAN_DEL_ATTR'.
      log_write 'E'         "message type
```

```
                'R3'          "message class
                '009'         "message number
                i_dsource     "message variable 1
                ' '.          "message variable 2
      CLEAR: sy-msgty, sy-msgid, sy-msgno, sy-msgv1, sy-msgv2,
             sy-msgv3, sy-msgv4.
      RAISE error_passed_to_mess_handler.
    ENDIF.

* Fill parameter buffer for data extraction calls
    s_s_if-dsource = i_dsource.
    s_s_if-maxsize = i_maxsize.

* Set blocksize for Change Pointer read
    CALL FUNCTION 'CHANGE_POINTERS_READ_MODE_SET'
      EXPORTING
        message_type = c_messag_type
        block_size   = 1000
      EXCEPTIONS
        OTHERS       = 1.

    IF sy-subrc <> 0.
      log_write_full sy-msgty   "message type
                     sy-msgid   "message class
                     sy-msgno   "message number
                     sy-msgv1   "message variable 1
                     sy-msgv2   "message variable 2
                     sy-msgv3   "message variable 3
                     sy-msgv4.  "message variable 4
      CLEAR: sy-msgty, sy-msgid, sy-msgno, sy-msgv1, sy-msgv2,
             sy-msgv3, sy-msgv4.
      RAISE error_passed_to_mess_handler.
    ENDIF.


  ELSE.

**************************************************************************
* Data transfer: Read Change Pointers and send data
**************************************************************************

    WHILE l_no_records < s_s_if-maxsize.

* Read Change Pointers for message type Z_EX_MEAN_DEL
      CALL FUNCTION 'CHANGE_POINTERS_READ'
        EXPORTING
          message_type          = c_messag_type
        TABLES
          change_pointers       = l_t_change_pointers
        EXCEPTIONS
          error_in_date_interval = 1
          error_in_time_interval = 2
          OTHERS                 = 3.

      IF sy-subrc <> 0.
        log_write_full sy-msgty   "message type
                       sy-msgid   "message class
                       sy-msgno   "message number
                       sy-msgv1   "message variable 1
                       sy-msgv2   "message variable 2
                       sy-msgv3   "message variable 3
                       sy-msgv4.  "message variable 4
        CLEAR: sy-msgty, sy-msgid, sy-msgno, sy-msgv1, sy-msgv2,
               sy-msgv3, sy-msgv4.
        RAISE error_passed_to_mess_handler.
      ENDIF.
```

```
        IF l_t_change_pointers IS INITIAL.
          IF l_no_records = 0.
*         No more data
            RAISE no_more_data.
          ELSE.
*         Data for this call is handed over to SAPI
            EXIT.
          ENDIF.
        ENDIF.

* Make sure that Change Pointers are in the right sequence
        SORT l_t_change_pointers BY cpident.

        LOOP AT l_t_change_pointers ASSIGNING <l_s_change_pointers>.

* Collect processed Change Pointer IDs
          l_s_chng_pntrs_idents-cpident = <l_s_change_pointers>-cpident.
          APPEND l_s_chng_pntrs_idents TO l_t_chng_pntrs_idents.

          IF <l_s_change_pointers>-cdchgid = 'E'.
*         Transfer only deleted entries to BW
            l_s_data-matnr = <l_s_change_pointers>-cdobjid.
            l_s_data-meinh = <l_s_change_pointers>-tabkey(3).
            l_s_data-ean11 = <l_s_change_pointers>-tabkey+3.
            APPEND l_s_data TO e_t_data.
          ENDIF.
        ENDLOOP.

* Write Status of processed Change Pointers
* only for processing in non read-only mode
* Modification 1 of original version

        IMPORT l_read_only FROM MEMORY ID 'RSFH_REO'.
        IF l_read_only NE 'X'.
         CALL FUNCTION 'CHANGE_POINTERS_STATUS_WRITE'
           EXPORTING
             message_type          = c_messag_type
           TABLES
             change_pointers_idents = l_t_chng_pntrs_idents.

         IF sy-subrc <> 0.
           log_write_full sy-msgty    "message type
                          sy-msgid    "message class
                          sy-msgno    "message number
                          sy-msgv1    "message variable 1
                          sy-msgv2    "message variable 2
                          sy-msgv3    "message variable 3
                          sy-msgv4.   "message variable 4
           CLEAR: sy-msgty, sy-msgid, sy-msgno, sy-msgv1, sy-msgv2,
                  sy-msgv3, sy-msgv4.
           RAISE error_passed_to_mess_handler.
         ENDIF.
        ENDIF.

* Delete duplicate records
        SORT e_t_data BY matnr meinh.
        DELETE ADJACENT DUPLICATES FROM e_t_data COMPARING matnr meinh.

* Record counter
        DESCRIBE TABLE e_t_data LINES l_no_records.

      ENDWHILE.

    ENDIF.
```

ENDFUNCTION.

Field routine in transformation for 0RPA_LFNUM


```
* DATA: l_s_errorlog TYPE rssm_s_errorlog_int.

* Get sequence number of existing Material/Unit/EAN combination
  SELECT SINGLE rpa_lfnum FROM /bi0/prpa_mean
    INTO result
    WHERE rpa_disqu = source_fields-meinh
      AND rpa_mean  = source_fields-matnr
      AND eanupc    = source_fields-ean11
      AND objvers   = 'A'.

  IF sy-subrc <> 0.
* Check if 'M' record is available
    SELECT SINGLE rpa_lfnum FROM /bi0/prpa_mean
      INTO result
      WHERE rpa_disqu = source_fields-meinh
        AND rpa_mean  = source_fields-matnr
        AND eanupc    = source_fields-ean11
        AND objvers   = 'M'.

    IF sy-subrc <> 0.
* No record for Material/Unit/EAN combination available
      result = ''.
    ENDIF.
  ENDIF.
```

www.sdn.sap.com/irj/sdn/howtoguides