

Generate BPC Hierarchy from BW

BPC Classic (Standard) 10.1 on Netweaver platform uses objects in SAP BW, however in many cases the level of integration with BW requires custom development. In the post below I will review steps needed for a BPC dimension hierarchy to be updated automatically from infoobject attributes in SAP BW.

Table of Contents

Generate BPC Hierarchy from BW.....	1
Business Case	1
BW Model for Profit Center InfoObject	1
Convert Attributes to a hierarchy using Expert Routine.....	2
Upload BW infoobject hierarchy in BPC.....	6
Review Profit Center hierarchy in BPC.....	7

Business Case

Let us review a business case where users maintain Profit Center hierarchy in SAP BPC for planning purposes. Once a certain planning scenario is approved Profit Centers are being created in the ERP system. Profit Centers are linked to Business Units in ERP and eventually have to be integrated to SAP BPC as actual values under proper (existing) BU nodes. Technically this requires loading Profit Center – Business Unit mapping from ERP to BW, converting this mapping into a BW hierarchy utilizing existing BPC hierarchy structure, and finally updating BPC hierarchy with the new nodes.

BW Model for Profit Center InfoObject

The approach described below requires 3 infoobjects in BW for Profit Centers:

- **oprofit_ctr** – a standard Business Content object for loading master data from ERP
- **zprofctr** – a custom infoobject with only relevant records for BPC and generated hierarchy, to be updated from **oprofit_ctr**
- **/CPMB/IEDUUVV** – a BPC dimension based infoobject generated and maintained in SAP BPC

Profit Center	OPROFIT_CTR
Profit Center (Attributes)	ATTRIBUTES OPROFIT...
RSDS OPROFIT_CTR_ATTR_ERQCLNT600 -> IOBJ OPR	048B3NSWYT2MS6SFL...
Profit center	OPROFIT_CTR_ATTR
OPROFIT_CTR_ATTR Full	ZPAK_B90WRKE0MDXP...
Data Transfer Processes	ATTRIBUTES OPROFIT...
Profit Center (Hierarchies)	HIERARCHIES OPROFI...
Profit Center (Texts)	TEXTS OPROFIT_CTR
Profit Center	ZPROFCTR
Profit Center (Attributes)	ATTRIBUTES ZPROFCTR
Profit Center (Hierarchies)	HIERARCHIES ZPROFCTR
IOBJ ZPROFCTR -> IOBJ ZPROFCTR	OGZ31HH32VGABO1ZL...
Profit Center (Attributes)	ATTRIBUTES ZPROFCTR
IOBJ OPROFIT_CTR -> IOBJ ZPROFCTR	OQAJL50A7R7LEXHLE13...
Data Transfer Processes	ATTRIBUTES ZPROFCTR
OPROFIT_CTR -> ZPROFCTR	DTP_8C7VF7NIGF4Y0V...
Data Transfer Processes	HIERARCHIES ZPROFCTR
Profit Center (Texts)	TEXTS ZPROFCTR
PROFITCENTER	/CPMB/IEDUUWV
PROFITCENTER (Attributes)	ATTRIBUTES /CPMB/I...
PROFITCENTER (Hierarchies)	HIERARCHIES /CPMB/I...
PROFITCENTER (Texts)	TEXTS /CPMB/IEDUUWV

Convert Attributes to a hierarchy using Expert Routine

In BW 7.4 (and up) on HANA we can generate a hierarchy via a transformation. In this case I propose to use **oprofit_ctr** attributes as a source as it contains the necessary mapping of Profit Centers to Business Units that have to be integrated in the existing BPC hierarchy. We can load attributes to the **zprofctr** infoobject and after that set up a transformation from attributes to the hierarchy as outlined below:

The screenshot shows the SAP BW Expert Routine configuration for a transformation. The source is 'Profit Center (ZPROFCTR)' and the target is 'Profit Center (ZPROFCTR)'. The transformation is active and executable. The 'Hierarchy Header Profit Center (ZPROFCTR)' table is visible, showing the mapping of attributes to hierarchy nodes.

Pos	Key	InfoObject	Info Descript.	Data	Length
1		ZPROFCTR	Profit Center	CHAR	000032
2		ZCLASSIF	Classification	CHAR	000032
3		ZBUSUNIT	Business Unit	CHAR	000032

Here is the ABAP code I have used in the Expert routine to generate the hierarchy:

```
* Update Source table with the existing BPC hierarchy for Profit Centers
DATA: sr TYPE ty_s_SC_1.
```

```
TYPES: BEGIN OF ty_pcbpc,
        NODEID TYPE RSHIENODID,
        NODENAME TYPE RSSHNODENAMESTR,
        PARENTID TYPE RSPARENT,
        END OF ty_pcbpc.
```

```
DATA: pc_bpc TYPE SORTED TABLE OF ty_pcbpc
      WITH NON-UNIQUE KEY NODEID.
FIELD-SYMBOLS: <pc> TYPE ty_pcbpc, <pc1> TYPE ty_pcbpc.
```

```
* Retrieve BPC dimension for PROFITCENTER
DATA pc TYPE c LENGTH 16.
SELECT SINGLE TECH_NAME FROM UJA_DIMENSION INTO pc
WHERE DIMENSION = 'PROFITCENTER'.
* CONCATENATE '/B28/P' pc+6(8) INTO pc.

* Retrieve BPC Hierarchy for Profit Center
DATA hid TYPE RSHIEID.
SELECT SINGLE HIEID FROM RSHIEDIR INTO hid
WHERE IOBJNM = pc AND HIENM = 'PARENTH1'.

CONCATENATE '/B28/H' pc+6(8) INTO pc.
SELECT * FROM (pc) "/B28/HIEDUUWV
INTO CORRESPONDING FIELDS OF TABLE pc_BPC
WHERE HIEID = hid AND OBJVERS = 'A'.

sr-/BIC/ZCLASSIF = 'BPC'.
LOOP AT pc_BPC ASSIGNING <pc>.
* Check if node is already in Source
  READ TABLE SOURCE_PACKAGE TRANSPORTING NO FIELDS
  WITH KEY /BIC/ZPROFCTR = <pc>-NODENAME.
  IF sy-subrc <> 0.
* Add PC line if not already in source
  sr-/BIC/ZPROFCTR = <pc>-NODENAME.
  READ TABLE pc_BPC ASSIGNING <pc1>
  WITH TABLE KEY NODEID = <pc>-PARENTID.
  IF sy-subrc = 0.
* If there is a parent in BPC hier use it
  sr-/BIC/ZBUSUNIT = <pc1>-NODENAME.
  ELSE.
  sr-/BIC/ZBUSUNIT = ''.
  ENDIF.
  INSERT sr INTO TABLE SOURCE_PACKAGE.
  ENDIF.
* ENDIF.
ENDLOOP.

* Remove lowest levels without children originating from BPC
DO 5 TIMES.
  LOOP AT SOURCE_PACKAGE ASSIGNING <SOURCE_FIELDS>
  WHERE /BIC/ZCLASSIF = 'BPC'.
  READ TABLE SOURCE_PACKAGE TRANSPORTING NO FIELDS
  WITH KEY /BIC/ZBUSUNIT = <SOURCE_FIELDS>-/BIC/ZPROFCTR.
  IF sy-subrc <> 0.
```

```
        DELETE SOURCE_PACKAGE.
    ENDIF.
  ENDLLOOP.
ENDDO.

* T1 -----
DATA: rp1 TYPE _ty_s_TG_1.
rp1-H_HIENM = 'BUH'.
rp1-H_STARTLEV = '1'.
INSERT rp1 INTO TABLE RESULT_PACKAGE_1.

* T3 ----- Hierarchy Structure
FIELD-SYMBOLS: <RESULT_FIELDS> TYPE _ty_s_TG_3.
DATA: t_io TYPE SORTED TABLE OF _ty_s_SC_1 WITH NON-UNIQUE KEY
/BIC/ZBUSUNIT.
FIELD-SYMBOLS: <io> TYPE _ty_s_SC_1.
DATA: nid TYPE N LENGTH 8.
DATA: rp TYPE _ty_s_TG_3.
FIELD-SYMBOLS: <rp> TYPE _ty_s_TG_3.

* Remove dummy BUs
DELETE SOURCE_PACKAGE WHERE /BIC/ZBUSUNIT BETWEEN 'Z' AND 'Z_____'.
DELETE SOURCE_PACKAGE WHERE /BIC/ZPROFCTR = ' '.
DELETE SOURCE_PACKAGE WHERE /BIC/ZBUSUNIT = ' ' AND /BIC/ZPROFCTR <>
'TOTBUCC'.

SORT SOURCE_PACKAGE BY /BIC/ZPROFCTR /BIC/ZBUSUNIT /BIC/ZCLASSIF
DESCENDING.
DELETE ADJACENT DUPLICATES FROM SOURCE_PACKAGE COMPARING /BIC/ZPROFCTR.
t_io[] = SOURCE_PACKAGE[].
nid = 1.

* generate lines for all levels
LOOP AT SOURCE_PACKAGE ASSIGNING <SOURCE_FIELDS>.
* check if the node has children
  READ TABLE t_io ASSIGNING <io>
  WITH TABLE KEY /BIC/ZBUSUNIT = <SOURCE_FIELDS>-/BIC/ZPROFCTR.
  IF sy-subrc = 0.
    rp-H_IOBJNM = 'OHIER_NODE'.
  ELSE.
    rp-H_IOBJNM = 'ZPROFCTR'.
  ENDIF.
  rp-H_NODEID = nid.
  nid = nid + 1.
DATA a TYPE C LENGTH 32.
DATA an TYPE I. " LENGTH 32.
DATA pv_error TYPE sy-subrc. pv_error = 0.
```

```
TRY.
  an = <SOURCE_FIELDS>-/BIC/ZPROFCTR. "Convert string to number.
  CATCH cx_sy_conversion_no_number.
    pv_error = 1.
  CATCH cx_sy_conversion_overflow.
    pv_error = 2.
ENDTRY.
IF pv_error = 0.
  a = an. SHIFT a RIGHT. TRANSLATE a USING ' 0'.

ELSE. a = <SOURCE_FIELDS>-/BIC/ZPROFCTR.
ENDIF.

rp-H_HIERNODE = a.
rp-/BIC/ZPROFCTR = <SOURCE_FIELDS>-/BIC/ZPROFCTR.
INSERT rp INTO TABLE RESULT_PACKAGE_3.
ENDLOOP.

* assign parent node ids
LOOP AT RESULT_PACKAGE_3 ASSIGNING <RESULT_FIELDS>.
  READ TABLE t_io ASSIGNING <io>
  WITH KEY /BIC/ZPROFCTR = <RESULT_FIELDS>-/BIC/ZPROFCTR.
  IF sy-subrc = 0.
* If parent exists find its Node ID
    READ TABLE RESULT_PACKAGE_3 ASSIGNING <rp>
    WITH KEY /BIC/ZPROFCTR = <io>-/BIC/ZBUSUNIT.
    IF sy-subrc = 0.
      <RESULT_FIELDS>-H_PARENTID = <rp>-H_NODEID.
    ENDIF.
  ENDIF.
ENDLOOP.

SORT RESULT_PACKAGE_3 BY H_NODEID /BIC/ZPROFCTR ASCENDING.
DELETE ADJACENT DUPLICATES FROM RESULT_PACKAGE_3 COMPARING H_NODEID.
```

After the transformation execution we get a hierarchy for the infoobject **zprofctr** with the new values Profit Centers from ERP integrated in the existing hierarchy structure from BPC:

Hierarchy 'BUH' Display: 'Active Version'		
BUH	InfoObject	Node Name
▼ [Folder] TOTBUCC	OHIER_NODE	TOTBUCC
▼ [Folder] BU	OHIER_NODE	BU
▶ [Folder] 0000000000000000	OHIER_NODE	00000000000000000000000000000010
▼ [Folder] 0000000000000000	OHIER_NODE	00000000000000000000000000000011
▼ [Folder] 11P	OHIER_NODE	11P
• [Icon] P-0305	ZPROFCTR	P-0305
• [Icon] P-0306	ZPROFCTR	P-0306
• [Icon] P-0307	ZPROFCTR	P-0307
• [Icon] P-0308	ZPROFCTR	P-0308

Upload BW infoobject hierarchy in BPC

Next in the BPC Data Manager we should create Transformation file, Conversion file, and Package for the hierarchy upload.

Transformation File

*OPTIONS

FORMAT = DELIMITED
 HEADER = YES
 DELIMITER = TAB
 AMOUNTDECIMALPOINT = .
 SKIP = 0
 SKIPIF =
 VALIDATERECORDS=YES
 CREDITPOSITIVE=YES
 MAXREJECTCOUNT= -1
 ROUNDAMOUNT=*MAPPING

*MAPPING

NODENAME=NODENAME
 HIER_NAME=HIER_NAME
 PARENT=PARENT
 ORDER=ORDER

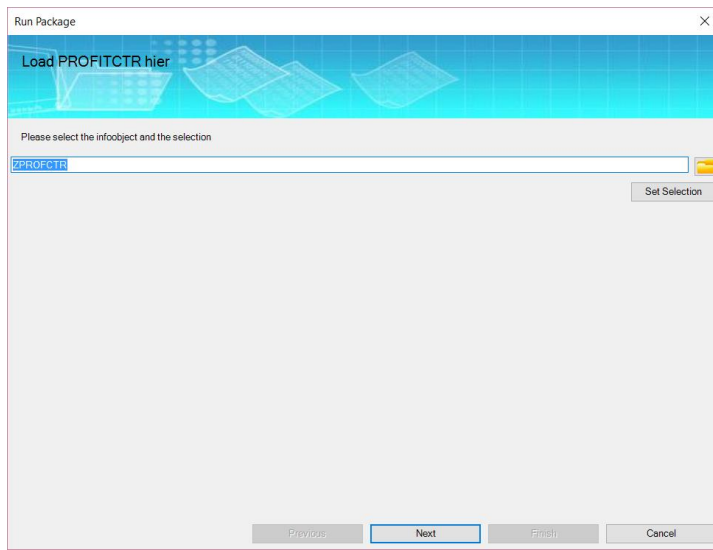
*CONVERSION

HIER_NAME= CONV_HIER.xls

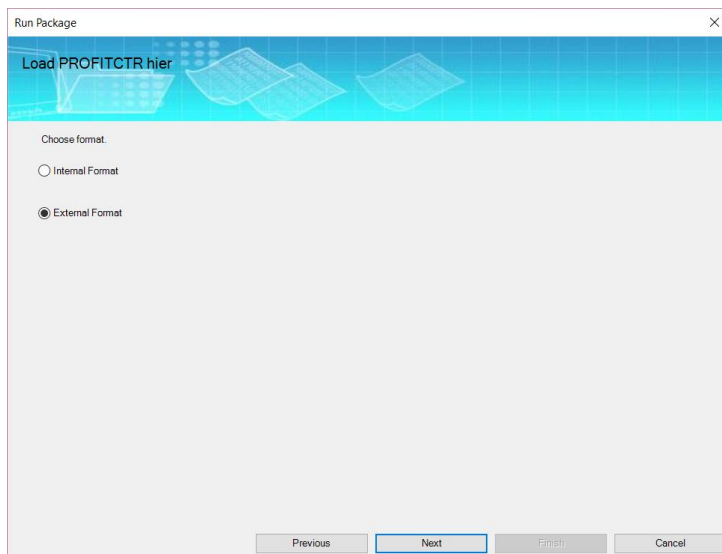
Conversion File

	EXTERNAL	INTERNAL	FORMULA
*		PARENTH1	

Run Package



We have to choose external format:



Review Profit Center hierarchy in BPC

After the upload to BPC our dimension **PROFITCENTER** contains the hierarchy that is partially maintained by business users in SAP BPC and partially in SAP ERP.

EPM - Member Selector - PROFITCENTER

Hierarchy ▾ | Member Properties

Dimension Members

- TOTBUCC - Total BU & CC (948)
 - BU - Business Units (889)
 - 10 - BU Wes (52)
 - 11 - BU ren (150)
 - 11P - Total proje (148)
 - 11PB001 - BU11 Budget Project 1 (0)
 - 11PB002 - BU11 Budget Project 2 (0)
 - 11PB003 - BU11 Budget Project 3 (0)
 - 11PB004 - BU11 Budget Project 4 (0)
 - 11PB005 - BU11 Budget Project 5 (0)
 - 11PB006 - BU11 Budget Project 6 (0)

In the final step I would suggest automating data loads in BW with process chains followed by Package run scheduling in BPC.

Even though the logic for generating the hierarchy is contained in one ABAP script the approach described in this post may sound complex or artificial. I have to admit this complexity won't be necessary in the case of SAP BPC Embedded as BW infoobjects and their hierarchies are directly consumed in the BPC Embedded models.

Contact us for help with SAP BW and BPC



Sergei Peleshuk has over 15 years of experience implementing BI technologies for global clients in retail, distribution, fast-moving consumer goods (FMCG), oil, and gas industries. He has helped clients to design robust BI reporting and planning capabilities, leading them through all project phases: from analysis of requirements to building BI roadmaps, technical architecture, and efficient BI teams. Sergei is an expert in SAP Business Warehouse (SAP BW), SAP HANA, BPC, BusinessObjects, BO

Cloud, and SAP Lumira. You may contact Sergei at peleshuk@biportal.org