

The Definitive and Complete Method for using Virtual Characteristics and Key Figures in NetWeaver BI



Applies to:

NetWeaver BI 7.0 For more information, visit the [Business Intelligence homepage](#).

Summary

This paper describes and explains step by step on how to create and implement Virtual Characteristics and Key Figures. It also covers various steps omitted from other “How to’s” and provides tips to solve rare problems occurring while running or debugging.

Author: Taroun Napaul

Company: Accenture (Mauritius) Ltd

Created on: 4 April 2010

Author Bio



Taroun Napaul is currently working at Accenture Mauritius. He is a Business Intelligence Technical Developer with 2 years of experience in SAP BI/BW.

Table of Contents

Introduction	3
Scenario	3
Implementation	4
Step 1 – Create BAdi :	4
Step 2 – Edit BAdi :	5
Step 3 – Edit Implementation class :	7
Related Content	14
Copyright	Error! Bookmark not defined.

Introduction

Many times have we come across the case where we needed a Formula / CKF that outputs Alphanumeric values. This is one limitation of Formula / CKF and can be solved without having to remodel the InfoCube using Virtual Chars / KFs.

Virtual Characteristics / Key Figures are normal InfoObjects found in an InfoProvider whose values are not stored in the InfoProvider. The values are being calculated at query execution time.

(Note: Even if values are present for those InfoObjects in the InfoProvider, their values will be overwritten at query runtime.)

Scenario

We will use a simple scenario to illustrate the usage of Virtual Chars / KFs in a query.

- Each Batch of Material produced has an Amount, a Production date and an Expiration date.
- Create a query where
 - o Rows:- Batch, Material, Production date (Attribute of Batch), Expiration date (Attribute of Batch), Status
 - o Columns:- Age of product, Lifetime of product and Amount.
- Conditions:
 - o Age of product in days = Current date – Production date
 - o Lifetime of product in days = Expiration date – Production date
 - o If Lifetime of product < Age of product
 - Status = “Product Expired”
 - o Else
 - Status = “Product Consumable”

The above conditions cannot be formulated with Bex Query Designer. So, to solve this issue, we will use Virtual Chars / KFs.

Characteristic and Key Figures whose values will be calculated at runtime.

Batch Virtual	ZMIR_CUBE1	
Material	A0MATNR	
Batch number	SGBATCH	
Status	A4STATUS	
Navigation Attributes		
Exp. Date	SGBATCH__SGEXPDAT	<input checked="" type="checkbox"/> DATS
Production date	SGBATCH__SGPRDDAT	<input checked="" type="checkbox"/> DATS
Key Figures		
Age of Product	ZAGE_KF	
Lifetime of Product	ZLIF_KF	
Amount	0AMOUNT	

Implementation

Step 1 – Create BAdi :

The first step to build Virtual Chars / KFs is to create a BAdi.

- (i) Transaction SE19
- (ii) Make a copy of the following BAdi **RSR_OLAP_BADI** in the “Create Implementation” section ; “Classic BAdi” ; “BAdi Name” ; “Create Impl.” following the steps 1 → 2 → 3.

BAdi Builder: Initial Screen for Implementations

Step 2 – Edit BAdi :

The second step consists of customizing the newly created BAdi.

- (1) Enter a Description for the BAdi.

- (2) In the “*Defined filters*” section, click on the “*Insert Row*” button to enter an InfoProvider name. This step restricts the Virtual Chars / KFs to InfoProvider level; meaning that only queries built from that InfoProvider containing the Virtual Chars / KFs will execute this BAdi.

Business Add-In Builder: Change Implementation ZTEST_VIRTUAL

Implementation Name: ZTEST_VIRTUAL Inactive

Implementation Short Text: [Yellow Highlighted] 1.

Definition name: RSR_OLAP_BADI

Attributes | Interface

General Data

Package: [Empty]

Language: EN English

Last changed by: [Empty]

Last change: [Empty] 00:00:00

Type

Multiple use

Filter-Dependent Filter type: RSR_OLAP_BADI_FILTER Enhanceable

Filter for Implementing Virtual Characteristics and K

Defined filters

InfoProv	
TEST_CUBE	2.

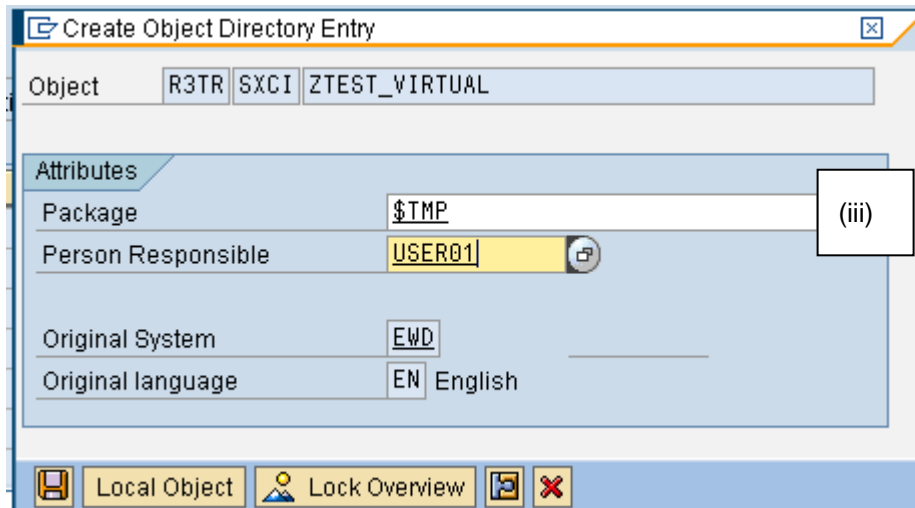
- (3) In the “Interface” tab ; menu “Goto” ; “Sample code” ; (i) “Copy” ; (ii) Over-write codes and (iii) choose a Package for the implementation. In this step, we copy the sample codes for the Methods **Define**, **Initialize** and **Compute** to be used in the implementation.

Save and Activate before proceeding to the next step.

The screenshot shows the SAP NetWeaver BI interface. At the top, there is a menu bar with options: Implementation, Edit, Goto, Utilities(M), Environment, System, Help. Below the menu bar, there is a toolbar with various icons. The main area is titled "Business Add-Ins" and "Implementation ZTEST_VIRTUAL". A context menu is open over the "Sample code" option, with the "Copy" option highlighted. A box labeled "3." points to the menu bar, and a box labeled "(i)" points to the "Copy" option. Below the menu, there are fields for "Implementation Name", "Implementation Short Text", and "Definition name". The "Interface" tab is selected, showing fields for "Interface name" (IF_EX_RSR_OLAP_BADI) and "Name of implementing class" (ZCL_IM_TEST_VIRTUAL). Below these fields is a table with columns "Method", "Implement", and "Description". The table contains three rows: "DEFINE", "INITIALIZE", and "COMPUTE", each with "ABAP Code" in the "Implement" column. Below the table is a field for "Example implementation class" (CL_EXM_IM_RSR_OLAP_BADI).

Method	Implement	Description
DEFINE	ABAP Code	
INITIALIZE	ABAP Code	
COMPUTE	ABAP Code	

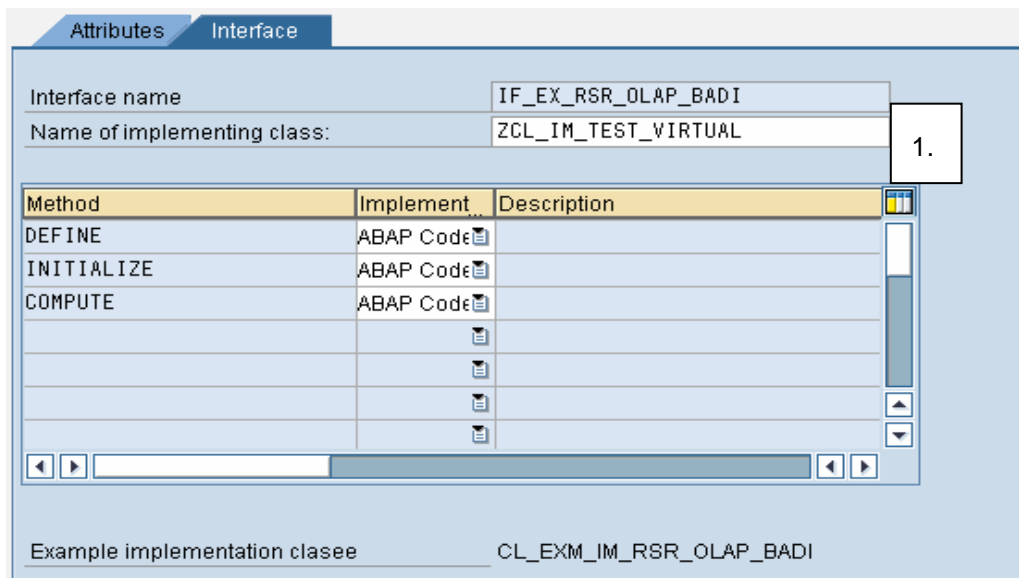
The screenshot shows a dialog box titled "Copy example code". The dialog box contains a question mark icon and the text: "The code for the class to be implemented will be over-written". Below the text are three buttons: "Yes", "No", and "Cancel". A box labeled "(ii)" points to the "Yes" button.



Step 3 – Edit Implementation class :

In the third step we will be customizing the Implementation class.

(1) Double click on the Implementation class.



(2) In the “Methods” tab, delete Method **GET_FIELD_POSITION_D**. This method is copied in the second step during the Sample codes copying and this method is not needed.

Class Builder: Change Class ZCL_IM_TEST_VIRTUAL

Class Interface: ZCL_IM_TEST_VIRTUAL | Implemented / Active

Properties | Interfaces | Friends | Attributes | **Methods** | Events | Types | Aliases

Parameters | Exceptions | Filter

Method	Level	Visibility	Method type	Description
IF_EX_RSR_OLAP_BADI~DEFINE	Static Method	Public		
IF_EX_RSR_OLAP_BADI~INITIALIZE	Instance Method	Public		
IF_EX_RSR_OLAP_BADI~COMPUTE	Instance Method	Public		
GET_FIELD_POSITION_D	Static Method	Public		

2.

- (3) In the “Attributes” tab we define the Virtual Chars / KFs to be used in the BAdi.
- i. Naming convention for Key Figure: **P_KYF_ < InfoObject Name >**
 - ii. Naming convention for Characteristic: **P_CHA_ < InfoObject Name >**
 - iii. Level: **Instance Attribute**; Visibility: **Public**; Typing: **Type**; Associated Type: **I**.

(Note: -- The attributes are defined with *Associated Type = I* because they are assigned a position number during the *Initialize* method)

Class Interface: ZCL_IM_TEST_VIRTUAL | Implemented / Active

Properties | Interfaces | Friends | **Attributes** | Methods | Events | Types | Aliases

Filter

Attribute	Level	Visibility	Re	Typing	Associated Type	Description	Initial value
			<input type="checkbox"/>	Type			
			<input type="checkbox"/>	Type			
			<input type="checkbox"/>	Type			
			<input type="checkbox"/>	Type			
			<input type="checkbox"/>	Type			
			<input type="checkbox"/>	Type			

Properties | Interfaces | Friends | **Attributes** | Methods | Events | Types | Aliases

Filter

Attribute	Level	Visibility	Re	Typing	Associated Type	Description	Initial value
P_CHA_SGBATCH	Instance Attribute	Public	<input type="checkbox"/>	Type	I	Batch	
P_CHA_ZSTATUS	Instance Attribute	Public	<input type="checkbox"/>	Type	I	Status	
P_KYF_ZLIF_KF	Instance Attribute	Public	<input type="checkbox"/>	Type	I	Lifetime of Product	
P_KYF_ZAGE_KF	Instance Attribute	Public	<input type="checkbox"/>	Type	I	Age of Product	

3.

- (4) Back to the “Method” tab,
- (i) Double click on the “Define” method.

In this section, we will define the InfoObjects (Virtual Chars / KFs) whose values will be calculated virtually. All the attributes added in the “Attributes” tab must be defined here.

Method Inactive

```

1  METHOD IF_EX_RSR_OLAP_BADI-DEFINE .
2
3  DATA: l_s_chanm  TYPE rrke_s_chanm,
4         l_kyfnm   TYPE rsd_kyfnm.
5  FIELD-SYMBOLS:
6     <l_s_chanm> TYPE rrke_s_chanm.
7
8  * CASE i_s_rkbid-infocube.
9
10 *   WHEN 'MYCUBE'.
11
12 *     characteristic
13 *       l_s_chanm-chanm = '...'.
14 *       l_s_chanm-mode  = ???." rrke_c_mode-read / rrke_c_mode-no_selection
15 *       APPEND l_s_chanm TO c_t_chanm.
16
17 *     key figure
18 *       l_kyfnm = '...'.
19 *       APPEND l_kyfnm TO c_t_kyfnm.
20
21 *   ENDCASE.
22
23 ENDMETHOD.

```

(i)

Uncomment the CASE... ENDCASE

Replace 'MYCUBE' by the name of your InfoProvider

"if ex rsr olap badi-define

Uncomment these codes too

Characteristics

```
*      l_s_chanm-chanm = '...'.
*      l_s_chanm-mode   = ???." rrke_c_mode-read / rrke_c_mode-no_selection
*      APPEND l_s_chanm TO c_t_chanm.
```

Key Figures

```
*      l_kyfnm = '...'.
*      APPEND l_kyfnm TO c_t_kyfnm.
```

Replace '...' by the name of your Characteristic or Key Figure

Replace ??? by either **rrke_c_mode-read** or **rrke_c_mode-no_selection**.

Apply **-no_selection** for Characteristics whose values will be calculated and **-read** for Characteristics whose values already exist in the InfoProvider.

The final codes must look like this:

```
CASE i_s_rkb1d-infocube.
  WHEN 'ZTESTCUBE'.
*    characteristic BATCH
    l_s_chanm-chanm = 'SGBATCH'.
    l_s_chanm-mode   = rrke_c_mode-read." rrke_c_mode-read / rrke_c_mode-no_selection
    APPEND l_s_chanm TO c_t_chanm.
*    characteristic STATUS
    l_s_chanm-chanm = 'ZSTATUS'.
    l_s_chanm-mode   = rrke_c_mode-no_selection." rrke_c_mode-read / rrke_c_mode-no_sel
    APPEND l_s_chanm TO c_t_chanm.
*    key figure AGE
    l_kyfnm = 'ZAGE_KF'.
    APPEND l_kyfnm TO c_t_kyfnm.
*    key figure LIFETIME
    l_kyfnm = 'ZLIF_KF'.
    APPEND l_kyfnm TO c_t_kyfnm.

ENDCASE.
```

(ii) Double click on the "Initialize" method.

The codes in the “Initialize” method is already complete. There is no need to modify these codes. The Initialize method gets the positions of the Characteristics and Key Figures.

Method	IF_EX_RSR_OLAP_BADI~INITIALIZE	Inactive
1	METHOD IF_EX_RSR_OLAP_BADI~INITIALIZE .	(ii)
2		
3	DATA: l_global_name TYPE string.	
4	FIELD-SYMBOLS:	
5	<l_global> TYPE i,	
6	<l_s_sfc> TYPE rrkg_s_sfc,	
7	<l_s_sfk> TYPE rrkg_s_sfk.	
8		
9	* there's no need to change this method	
10	* Just create attributes for each characteristics	
11	* with name P_CHA_<characteristic> TYPE i.	
12	* and constants for each key figure with name	
13	* P_KYF_<key figure> TYPE i.	
14		
15	CLASS cl_exm_im_rsr_olap_badi DEFINITION LOAD.	
16		

(iii) Double click on the “Compute” method.

All functionalities of the Virtual Chars /KFs are defined here.

The *IF Statement* checks whether the attributes have been correctly initialized, that is, they are assigned a position in the structure C_S_DATA.

The *Assign Component* assigns the field-symbol to the position of the attribute in the structure C_S_DATA.

Method	IF_EX_RSR_OLAP_BADI~COMPUTE	Inactive
1	METHOD IF_EX_RSR_OLAP_BADI~COMPUTE .	(iii)
2		
3	* if P_CHA_... > 0.	
4	* assign component P_CHA_... of structure C_S_DATA to <l_...>.	
5	* endif.	
6		
7	ENDMETHOD. "IF EX RSR OLAP BADI~COMPUTE	

(1) First, define a FIELD-SYMBOL for each of the attributes.

```
FIELD-SYMBOLS:
<fs_batch> TYPE ANY,
<fs_status> TYPE ANY,
<fs_age> TYPE ANY,
<fs_lifetime> TYPE ANY.

DATA:
prod_date TYPE sy-datum,
exp_date TYPE sy-datum.
```

(2) Assign the attributes' field-symbols their respective position in the structure.

```

** BATCH
IF p_cha_sgbatch > 0.
  ASSIGN COMPONENT p_cha_sgbatch OF STRUCTURE c_s_data TO <fs_batch>.
ENDIF.

** STATUS
IF p_cha_zstatus > 0.
  ASSIGN COMPONENT p_cha_zstatus OF STRUCTURE c_s_data TO <fs_status>.
ENDIF.

** LIFETIME
IF p_kyf_zlif_kf > 0.
  ASSIGN COMPONENT p_kyf_zlif_kf OF STRUCTURE c_s_data TO <fs_age>.
ENDIF.

** AGE
IF p_kyf_zage_kf > 0.
  ASSIGN COMPONENT p_kyf_zage_kf OF STRUCTURE c_s_data TO <fs_lifetime>.
ENDIF.

```

(3) Write the functionalities for the Virtual Chars / KFs

(Note:-- The “Compute” method functions like a loop. It will loop through the contents of the InfoProvider row by row that will appear in the query.)

```

** Select Production date and Expiration date from Batch
SELECT SINGLE /bic/sgprdat /bic/sgexpdat
FROM /bic/psgbatch
INTO (prod_date, exp_date)
WHERE /bic/sgbatch = <fs_batch>.

** calculating Age
<fs_age> = sy-datum - prod_date.

** calculating Lifetime
<fs_lifetime> = exp_date - prod_date.

** Defining Status
IF <fs_lifetime> < <fs_age>.
  <fs_status> = 'PRODUCT EXPIRED'.
ELSE.
  <fs_status> = 'RODUCT CONSUMABLE'.
ENDIF.

```

(4) Run Query to test the Virtual Chars / KFs.

Batch number	Material	Production date	Exp. Date	Status	Lifetime of Product	Current Age of Product	Amount
139	8 BAGB KEG 1 50L AL RKK	14.05.2007	08.05.2008	PRODUCT EXPIRED	1,099	360	123.00 ERROR
62090	54 BECK OW 24 0,33L BOX PROMO RU	01.01.2007	01.01.2008	PRODUCT EXPIRED	1,232	365	145.00 ERROR
73100	659 BECA ZOOM P18,5	12.12.2009	12.12.2010	RODUCT CONSUMABLE	156	365	432.00 ERROR

Tips:

The values for **Status**, **Lifetime of Product** and **Current Age of Product** are calculated at query runtime.

- Use RSRT to test the Query.
- Place a Break-point in the *Initialize* method to start debugger.
- If Virtual Chars / KFs are not working or Debugger is not starting; regenerate the query, that is, make a save as of the query and run the query_2.

Related Content

<http://www.sdn.sap.com/irj/scn/go/portal/prtroot/docs/library/uuid/e051fda8-71a9-2a10-ac9e-8d17414a8c8c>

<http://www.sdn.sap.com/irj/scn/go/portal/prtroot/docs/library/uuid/60e34f63-f44c-2c10-488e-c89b04e0ca7c>

For more information, visit the [Business Intelligence homepage](#).

Disclaimer and Liability Notice

This document may discuss sample coding or other information that does not include SAP official interfaces and therefore is not supported by SAP. Changes made based on this information are not supported and can be overwritten during an upgrade.

SAP will not be held liable for any damages caused by using or misusing the information, code or methods suggested in this document, and anyone using these methods does so at his/her own risk.

SAP offers no guarantees and assumes no responsibility or liability of any type with respect to the content of this technical article or code sample, including any liability resulting from incompatibility between the content within this document and the materials and services offered by SAP. You agree that you will not hold, or seek to hold, SAP responsible or liable with respect to the content of this document.