

# Virtual Characteristics and Key Figures Revisited: the Missing Part Debugger



## Applies to:

SAP BW 3.5, SAP BI 7.0. For more information, visit the [EDW homepage](#).

## Summary

This document is extension of already submitted many documents on Virtual Characteristics and Key figures, here the main intention is to explain the missing part of implementing VCKF i.e. When, why and how to use debugger while implementing Virtual Characteristics and Key figures.

**Author:** Vipul Goyal

**Company:** Accenture

**Created on:** 1<sup>st</sup> June 2010

## Author Bio



Vipul Goyal is a certified SAP BI Consultant working for Accenture Services Pvt. Ltd. He has more than 4 years of experince in SAP BI/BW

## Table of Contents

Business Scenario .....	3
Background .....	3
Step by Step Guide .....	3
Related Content.....	13
Disclaimer and Liability Notice.....	14

## Business Scenario

An organization has many contractors working for it with planned Start date and End date for their contract. Now the managers, need to know how many days are left for expiry of their contract, so that they can plan whether to extend the contract, or release him from his duties.

Now in this example, the use of Virtual Key figure would be shown to calculate number of Days for Expiry using End Date of Contract as Reference Characteristic.

## Background

A virtual characteristic or a virtual key figure is an InfoObject, which is defined within the InfoProvider as metadata without having any data stored physically.

Virtual Key Figures (KF)/ Characteristics can be used to create the Query more dynamic. The value of Virtual KF/Characteristics does not store in the Data Target, it is calculated at runtime (At Query Execution time).

Complex logic, database access is possible by using Virtual Key Figure and Characteristics.

The implementation part includes following steps, which also include Debugging part, which is used to identify position of referred Characteristic and Virtual Key Figure without which I was not able to implement Virtual concept.

- Create InfoProvider with virtual Key figure
- Create a query and add virtual Key figure and other relevant Characteristics like Contract End Date
- Implement RSR\_OLAP\_BADI
- Identify position of referred Characteristic and Virtual Key Figure using debugger.

## Step by Step Guide

1. Create virtual KF in RSA1

### Display Key Figure ZKF\_VIR: Details

The screenshot displays the SAP Business Content interface for the virtual key figure 'ZKF\_VIR'. The main details section shows the key figure name, descriptions, and status. The 'Additional Properties' section is expanded to show the 'Type/unit' configuration, where 'Number' is selected as the data type, and the 'Data Type' is set to 'DEC - Counter or amount field with comma a'. There are also fields for 'Fixed currency' and 'Fixed Unit of Meas.'.

2. Include the char/kf in the cube/dso

InfoCube	Techn. name / value	Fu...	O...	Data...	L	Key Fi...	C...	N...	Ag...	Ex...	Reference...	Unit	Alias Name
Test Cube	ZTST_CUB												
Object Information													
Version	In Process												
Save	Saved												
Revised Version	Active Version												
Object Status	Active, executable												
Settings													
Dimensions													
Data Package	ZTST_CUBP												
Change Run ID	0CHNGID			NUMC	14							0CHNGID	
Record type	0RECORDTP			NUMC	01							0RECORD...	
Request ID	0REQUID			CHAR	30							0REQUID	
Time	ZTST_CUBT												
Calendar Day	0CALDAY			DATS	08							0DATE	
Unit	ZTST_CUBU												
Dimension 1	ZTST_CUB1												
Valid from	0DATEFROM			DATS	08							0DATE	
Valid to	0DATETO			DATS	08							0DATE	
Dimension 2	ZTST_CUB2												
SAP Cap Employee	ZEMPLOYEE			NUMC	10							ZEMPLOYEE	
Employee Name	ZEMP_NAM			CHAR	50							ZEMP_NAM	
Employee Level	ZLEVEL			CHAR	03							ZLEVEL	
Project Definition	0PROJECT			CHAR	24							0PROJECT	
Dimension 3	ZTST_CUB3												
Coex Lead	COEX_LEAD			CHAR	25							COEX_LEAD	
Dimension 4	ZTST_CUB4												
Loaned Status	LOAN_STA			CHAR	10							LOAN_STA	
Navigation Attributes													
Applicant	0PROJECT_OPS...		<input type="checkbox"/>	NUMC	08							0PS_APPL...	
Person Responsible	0PROJECT_OPS...		<input type="checkbox"/>	NUMC	08							0PS_RES...	
Key Figures													
Vir Kf	ZKF_VIR			DEC	09	Numb...	<input checked="" type="checkbox"/>	<input type="checkbox"/>	SUM	SUM			ZKF_VIR

3. Go to SE19 Tcode and put RSR\_OLAP\_BADI under Create Implementation->Classical BAdi->BAdi Name

**BADI Builder: Initial Screen for Implementations**

**Edit Implementation**

New BAdi

Enhancement Implementation

Classic BAdi

Implementation

Display Change

---

**Create Implementation**

New BAdi

Enhancement Spot

Classic BAdi

BAdi Name: RSR\_OLAP\_BADI

Create Impl.

#### 4. Enter your Implementation Name

Business Add-In Builder: Create Implementation

Definition Name: RSR\_OLAP\_BADI

Implementation Name: ZVAR\_IMPL

Buttons: [OK] [Cancel]

#### 5. Give Description to your new implementation.

### Business Add-In Builder: Change Implementation ZVAR\_IMPL

Business Add-In Builder: Change Implementation ZVAR\_IMPL

Implementation Name: ZVAR\_IMPL Active

Implementation Short Text: Test virtual kf

Definition name: RSR\_OLAP\_BADI

Attributes | Interface

Interface name: IF\_EX\_RSR\_OLAP\_BADI

Name of implementing class: ZCL\_IM\_VAR\_IMPL

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

Example implementation class: CL\_EXM\_IM\_RSR\_OLAP\_BADI

6. On attribute Tab, give Infoprovider name as filter.

The screenshot shows the SAP configuration interface for a virtual characteristic. The 'Attributes' tab is active, and the 'Interface' sub-tab is selected. The 'General Data' section contains the following fields:

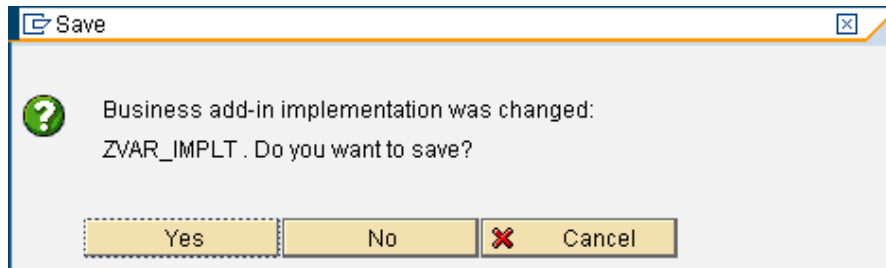
Package	\$TMP		
Language	EN	English	
Last changed by	BI8DEV01	Last activated by:	BI8DEV01
Last change	17.02.2010 13:56:14	Activated on	17.02.2010 13:56:20

The 'Type' section contains the following fields:

- Multiple use
- Filter-Depend. Filter type: RSR\_OLAP\_BADI\_FILTER  Enhanceable
- Filter for Implementing Virtual Characteristics and K: InfoProv

A list of 'Defined filters' is shown below, containing 'InfoProv' and 'ZTST\_CUB'.

7. Go to Interface tab and Double click on "Name of implementing class:" which is ZCL\_IM\_VAR\_IMPLT, and say yes when following screen Pops up to save your implementation.



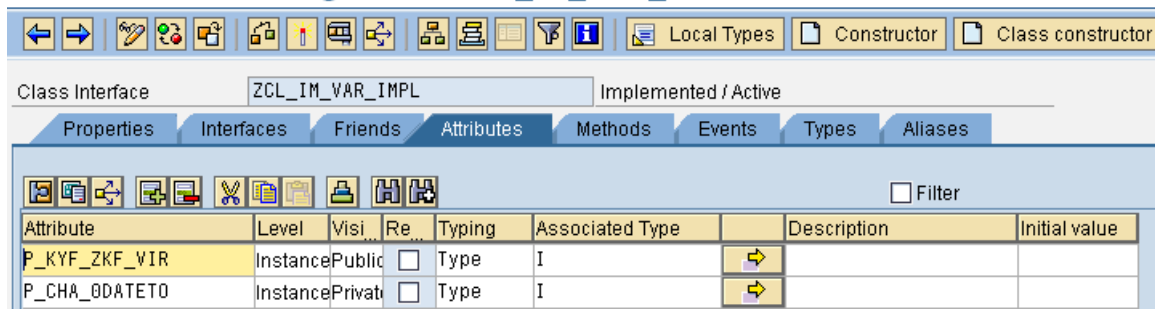
8. Add the attributes to the Implementation Class. The attributes must be of type I and should follow the standard naming convention.

In this example the Name of the implementing class is: ZCL\_IM\_VAR\_IMPLT and add the attributes as shown in Screenshot below

Always use the convention P\_KYF\_<Key Figure InfoObject> if it is Key Figure and if it's characteristic then use P\_CHA\_<Char. InfoObject>.

I have used Key Figure ZKF\_VIR as virtual KF and it would be calculated based on Characteristic ODATE\_TO (End date of Contract)

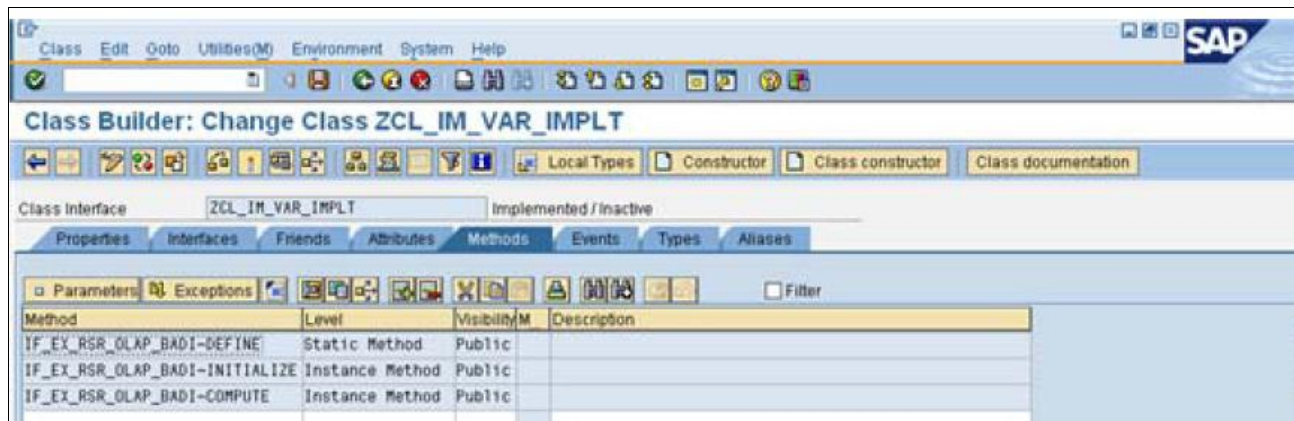
## Class Builder: Change Class ZCL\_IM\_VAR\_IMPL



Attribute	Level	Visi	Re	Typing	Associated Type	Description	Initial value
P_KYF_ZKF_VIR	Instance	Public	<input type="checkbox"/>	Type	I		
P_CHA_ODATE_TO	Instance	Privat	<input type="checkbox"/>	Type	I		

9. After you added the required attributes go to Methods Tab you will find three methods as below:

- IF\_EX\_RSR\_OLAP\_BADI~DEFINE
- IF\_EX\_RSR\_OLAP\_BADI~INITIALIZE
- IF\_EX\_RSR\_OLAP\_BADI~COMPUTE



Method	Level	Visibility	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	

### 10. IF\_EX\_RSR\_OLAP\_BADI~DEFINE

**In define method**, you define of characteristics and KFs involved in the calculations of VKF assign to system internal tables c\_t\_chanm and c\_t\_kyfnm and restrict this assignment in the case statement. to the cube used.

e.g. We are restricting following assignment to the cube 'ZTST\_CUB'

Characteristic ODATE\_TO is assigned to c\_t\_chanm in read mode, as we have read data for this char. during runtime to calculate VKF.

And Key Figure 'ZKF\_VIR' is assigned to internal table c\_t\_kyfnm.

You can also define other Local declarations in this Method.

**Class Builder: Class ZCL\_IM\_VAR\_IMPL Change**

Method: IF\_EX\_RSR\_OLAP\_BADI~DEFINE Active

```

METHOD if_ex_rsr_olap_badi~define.

  DATA: l_s_chanm TYPE rrke_s_chanm,
         l_kyfnm  TYPE rsd_kyfnm.
  FIELD-SYMBOLS:
    <l_s_chanm> TYPE rrke_s_chanm.
  ** Insert Code
  CASE i_s_rkb1d~infocube.
    WHEN 'ZTST_CUB'.
      l_s_chanm~chanm = '0DATETO'.
      l_s_chanm~mode = rrke_c_mode~read.
      APPEND l_s_chanm TO c_t_chanm.
      l_s_chanm~chanm = 'ZVARDT'.
      l_s_chanm~mode = rrke_c_mode~read.
      APPEND l_s_chanm TO c_t_chanm.
      APPEND 'ZKF_VIR' TO c_t_kyfnm.
    ENDCASE.
  ENDMETHOD.

```

And then in Initialize Method, you can assign values to the Local Declarations if any, otherwise you can leave it as it, as done in this case.



## 11. IF\_EX\_RSR\_OLAP\_BADI~COMPUTE

Now in compute method all the calculations are done, below is the sample code for the same.

```

Class Builder: Class ZCL_IM_VAR_IMPL Change
Method IF_EX_RSR_OLAP_BADI~COMPUTE
METHOD if_ex_rsr_olap_badi~compute.

  FIELD-SYMBOLS <fs_zkf_vir> TYPE ANY.
  FIELD-SYMBOLS <fs_0dateto> TYPE ANY.
  FIELD-SYMBOLS <fs_zvardt> TYPE ANY.
  DATA: l_date TYPE sy-datum.
  data: l_val type /BIC/OIZKF_VIR.
  break-point.
  p_kyf_zkf_vir = 14.
  p_cha_0dateto = 2.
  ASSIGN COMPONENT p_kyf_zkf_vir OF STRUCTURE c_s_data
  TO <fs_zkf_vir>.
  ASSIGN COMPONENT p_cha_0dateto OF STRUCTURE c_s_data
  TO <fs_0dateto>.
  * ASSIGN COMPONENT p_cha_zvardt OF STRUCTURE c_s_data
  * TO <fs_zvardt>.
  CALL FUNCTION 'ZTST_DIF_DATE'
  EXPORTING
    l_date      = <fs_0dateto>
  IMPORTING
    DATE_DIFF  = l_val
  .

  <fs_zkf_vir> = l_val.

ENDMETHOD.

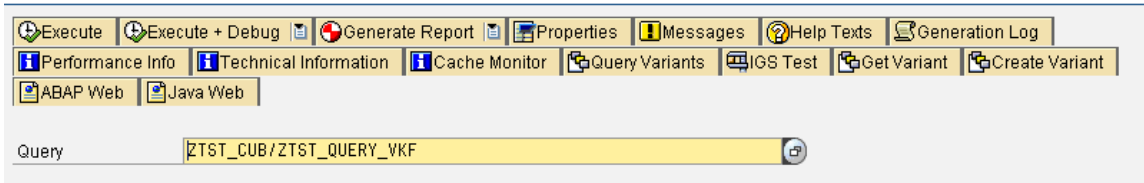
```

Please be aware of the following points while writing this code:

- A.** Define field symbols in the same way as defined below i.e. <FS\_KF name> and <FS\_Char>
- B.** C\_S\_DATA is the table from which you need to read value and assign value of object involved e.g. 0DATE\_TO value is assigned into <FS\_0DATETO> and ZKF\_VIR value assigned to <FS\_ZKF\_VIR>
- C.** Above values are assigned to the Field symbols through components P\_KYF\_ZKF\_VIR and P\_CHA\_0DATETO of C\_S\_DATA defined in the attributes tab.
- D.** But the problem with above assignment, its not returning any value for 0DATE\_TO to respective Field symbol and thus VKF was not giving any value.
- E.** So, to resolve this we identified that the position of the above components is not fixed in the structure C\_S\_DATA, and we need to find the positions of the objects involved. For this we used debugger by inserting the break-point in the code as shown in Red Box above and identified the position as 14 and 2, by running the query in RSRT. I will try to explain this process in detail as this is main missing information which I didn't find in many of the blogs and white papers submitted.

12. After inserting the break-point in the above code, GOTO RSRT and execute the Report in Debug mode, the query will stop at the break-point just inserted as shown below:

### Query Monitor



### (3) - ABAP Debugger Controls Session 3 (Exclusive)

```

ABAP Source Code Display
└─ Br P Line Srce Code
    1 METHOD if_ex_rsr_olap_badi~compute.
    2
    3 FIELD-SYMBOLS <fs_zkf_vir> TYPE ANY.
    4 FIELD-SYMBOLS <fs_0dateto> TYPE ANY.
    5 FIELD-SYMBOLS <fs_zvardt> TYPE ANY.
    6 DATA: l_date TYPE sy-datum.
    7 data: l_val type /BIC/OIZKF_VIR.
    8 break-point.
    9 p_kyf_zkf_vir = 14.
   10 p_cha_0dateto = 2.
   11 ASSIGN COMPONENT p_kyf_zkf_vir OF STRUCTURE c_s_data
   12 TO <fs_zkf_vir>.
   13 ASSIGN COMPONENT p_cha_0dateto OF STRUCTURE c_s_data
   14 TO <fs_0dateto>.
   15 * ASSIGN COMPONENT p_cha_zvardt OF STRUCTURE c_s_data
   16 * TO <fs_zvardt>.
   17 CALL FUNCTION 'ZTST_DIF_DATE'
   18 EXPORTING
   19 l_date = <fs_0dateto>
   20 IMPORTING
   21 DATE_DIFF = l_val
   22
   23
   24
   25
   26 <fs_zkf_vir> = l_val.
   27
   28
   29 ENDMETHOD.
  
```

13. Double click on the structure C\_S\_DATA and following screen appears. Now, In case of characteristic ODATE\_TO I identified the position by seeing its value and for KF by seeing type as shown below.



15. Run the query and Virtual Key figure is calculated as shown below:

Test Query Last Data Update: 17.02.2010 09:37:41

Rows	Coex Lead	Employee	Loaned Status	Valid from	Valid to	Vir KF
Coex Lead	ANA	Singh	LOANED	22.12.2009	29.10.2010	200,000
Employee		Mahajan	LOANED	10.12.2009	22.01.2010	-80,000
Loaned Status		ajan	LOANED	05.09.2009	31.03.2010	-12,000
Valid from		ajan	LOANED	03.11.2009	31.03.2010	-12,000
Valid to		an Subramanian	READY FOR	16.09.2009	31.10.2009	-163,000
		ek	LOANED	08.09.2009	15.03.2010	-28,000
			READY FOR	16.09.2009	31.10.2009	-163,000
		rakash Pateriya	LOANED	29.09.2008	31.03.2010	-12,000
		mana	LOANED	04.01.2010	19.02.2010	-52,000
		stogi	READY FOR	28.10.2009	27.11.2009	-136,000
		ndan	LOANED	03.11.2009	31.03.2010	-12,000
		'jayakumar Lakkavalli	LOANED	21.12.2009	30.06.2010	79,000
		oukat Nanji	LOANED	18.09.2009	29.01.2010	-73,000
		asmukh Pethad	LOANED	28.01.2009	30.04.2010	18,000
		idal	LOANED	18.08.2009	29.01.2010	-73,000

## Related Content

[Virtual Key Figures Characteristics Makes Query More Dynamic](#)

[Using the Cache Despite Virtual Characteristics/Key Figures](#)

[Step By Step Process for Virtual Key Figures and Characteristics through BAdi](#)

For more information, visit the [EDW 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.