Implementing Virtual Key Figure/Characteristics Makes Query More Dynamic

Applies to:
This applies to SAP BI (3.5, 3.0B)

Summary
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).

This paper shows how to create or implement virtual Characteristics / Key Figure.

Author(s): Nooruddin Shaikh
Company: Wipro Technologies Ltd.
Created on: 10 January 2008

Author Bio
Nooruddin Shaikh is a BI certified consultant having more than 8yrs of IT experience including more than 4 yrs in BI. Currently he is working as a BI Architect with Wipro Technologies. You can reach him at nooruddin.shaikh@wipro.com / noor.shaikh@yahoo.co.uk
# Table of Contents

Summary ....................................................................................................................................................... 3  
Creation of InfoObject............................................................................................................................... 4  
Implementation of BADI............................................................................................................................. 5  
Add the InfoObject to the Query................................................................................................................ 9  
Related Content ............................................................................................................................................ 9  
Disclaimer and Liability Notice .................................................................................................................... 10
Summary

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.

This paper shows how to create virtual Characteristics / Key Figure. Example shows about evaluating a value called as 20 Day Value, which requires a complex logic and also require data from existing ODS Characteristics for complex calculation.

The implementation can be divided into the following areas:

1. Create of InfoObject [Key Figure / Characteristics] and attach the InfoObject to the InfoProvider.
2. Implementation of BADI
3. Adding the InfoObject into the Query.
Creation of InfoObject:

1. Create the InfoObject (Key Figure / Characteristics) in the InfoObject Catalogue.

2. Attach the Key Figure to the ODS, for which the value would be calculated at runtime.
Implementation of BADI

3. Create a BADI called as ZVAR_IMPL (for this example) based on the Definition Name (RSR_OLP_BADI). Use Transaction Code (SE19).

Business Add-In Builder: Change Implementation ZVAR_IMPL

You can here define the Filter (The name of the InfoProvider or a Wildcard Value).
4. Go to the next Tab called as Interface to implement your code.

**Business Add-In Builder: Change Implementation ZVAR_IMPL**

![Interface implementation details](image)

<table>
<thead>
<tr>
<th>Method</th>
<th>Implementation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEFINE</td>
<td>ABAP Code</td>
<td></td>
</tr>
<tr>
<td>INITIALIZE</td>
<td>ABAP Code</td>
<td></td>
</tr>
<tr>
<td>COMPUTE</td>
<td>ABAP Code</td>
<td></td>
</tr>
</tbody>
</table>

5. 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_IMPL**

Double Click on the **ZCL_IM_VAR_IMPL**, and add the following attributes.

**Class Builder: Change Class ZCL_IM_VAR_IMPL**

![Class implementation details](image)

As the object ZV_20DV, ZVAR_SHPR, ZVAR_DTD is used in the code in this example, the above attributes are atted. If it is a key figure, use the convention P_KYF_<Key Figure InfoObject> and P_CHA_<Char. InfoObject>.
6. Add the implementation code. To do this go to the methods Tab:

Double click on the method IF_EX_RSR_OLAP_BADI~DEFINE.
Add the following code:

```java
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 'ZV_SIPER'.
      l_s_chanm-chanm = 'ZVAR_SHPR'.
      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 'ZV_20DV' TO c_t_kyfnm.

   ENDCASE.

ENDMETHOD.                   "if_ex_rsr_olap_badi~define"```
Double Click on the method `IF_EX_RSR_OOLAP_BADI~COMPUTE`.

Add the following code:

```
METHOD if_ex_rsr_olap_badi~compute.

FIELD-SYMBOLS <fs_zv_20dv> TYPE ANY.
FIELD-SYMBOLS <fs_zvar_shpr> TYPE ANY.
FIELD-SYMBOLS <fs_zvardt> TYPE ANY.

DATA: l_zv_20dv TYPE curr09.

ASSIGN COMPONENT p_kyf_zv_20dv OF STRUCTURE c_s_data TO <fs_zv_20dv>.
ASSIGN COMPONENT p_cha_zvar_shpr OF STRUCTURE c_s_data TO <fs_zvar_shpr>.
ASSIGN COMPONENT p_cha_zvardt OF STRUCTURE c_s_data TO <fs_zvardt>.

* This is a customized function module which will calculate average daily Value.
* Take input as Loaded Date and Shipper Name
* and use complex algorithm to calculate the required value.

CALL FUNCTION 'ZV_CAL_AVERAGE_DAILY_AGG' 
  EXPORTING 
    loaded_dt = <fs_zvardt> 
    shipper   = <fs_zvar_shpr> 
  IMPORTING 
    avg_val   = l_zv_20dv.


ENDMETHOD.                      "IF_EX_RSR_OOLAP_BADI~COMPUTE

7. Check and Activate the Class.
Add the InfoObject to the Query

8. Now the Virtual Key Figure is ready to use in the Query. Use the Key Figure in the Query:

```
Query Designer: ZV_VAR_Q2 - VAR_Report
```

Note:
- The transaction RSRT is advised to debug the query.
- A infinite loop can be used in the compute method to debug the query at runtime (using Transaction SM50).

Related Content
- [Five Ways to Enhance SAP BI Backend Functionality Using ABAP](#)
- [User Variable used in my Virtual KF](#)
- [Virtual Key Figures](#)
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.