SAP BW - Virtual Characteristic (Multiprovider & Infoset) - RSR_OLAP_BADI

Applies to:
SAP BW 3.5 / SAP 7.0 Consultants with ABAP Skills. For more information, visit EDW Homepage.

Summary
Explains how to use BADI RSR_OLAP_BADI to get data for Virtual Characteristic/Key Figure (Multiproviders/Infosets).

Author: Suraj Tigga
Company: Capgemini Consulting
Created on: 17 September 2010

Author Bio
Suraj Tigga is a Senior SAP BI / ABAP consultant at Capgemini Consulting, India. Suraj joined Capgemini Consulting in 2008 and has worked on multiple SAP BI implementation and support Projects.
Table of Contents

Scenario.................................................................................................................................................. 3

Multiprovider (Virtual Characteristic/Key Figure) .............................................................................. 3
  Method - IF_EX_RSR_O LAP_BADI~DEFINE .......................................................................................... 5
  Method IF_EX_RSR_O LAP_BADI~INITIALIZE .................................................................................... 7
  Method IF_EX_RSR_O LAP_BADI~COMPUTE ....................................................................................... 9
  Execute................................................................................................................................................ 9

Infoset (Virtual Characteristic/Key Figure) .......................................................................................... 11
  Method - IF_EX_RSR_O LAP_BADI~DEFINE ...................................................................................... 14
  Method IF_EX_RSR_O LAP_BADI~INITIALIZE ..................................................................................... 16
  Method IF_EX_RSR_O LAP_BADI~COMPUTE ....................................................................................... 18

Related Content ..................................................................................................................................... 20

Disclaimer and Liability Notice ........................................................................................................... 21
Scenario

Virtual Characteristic / Key Figures are used when we want to determine the value of the Infoobject at runtime for a BEx Query. Virtual Key Figure are not populated while data loading to the Cube/DSO (No transformation rule exist for the Virtual Characteristic / Key Figure from the underlying Datasource). Virtual Characteristic/Key Figure values are calculated using either BADI (RSR_OLAP_BADI) or Enhancement (RSR00002). Below the Virtual Characteristic/Key Figure are used for Infoproviders (Multiprovider/Infoset), whose values are determined using BADI (RSR_OLAP_BADI):

- **Multiprovider (Virtual Characteristic/Key Figure):** Multiprovider (ZM_SD_005) built on Infocube (0SD_C03: Sales: Overview) and DSO (0SD_005: Order: Condition Data). DSO 0SD_005 contains the Virtual Key Figure ZVIRTKEY (Virtual Key Figure) whose value would be calculated at query execution.

- **Infoset (Virtual Characteristic/Key Figure):** Infoset (ZSD_005_) built on Infocube (0SD_C03: Sales: Overview), Infoobject 0MATERIAL (Material) and DSO (0SD_005: Order: Condition Data) contains the Virtual Key Figure ZVIRTKEY (Virtual Key Figure) whose value would be calculated at query execution.

**Multiprovider (Virtual Characteristic/Key Figure)**

**Step1:** Create a Virtual Key Figure ‘ZVIRTKEY’ and add the Key Figure to DSO (0SD_005: Order: Condition Data):

Virtual Characteristic / Key Figures are used when we want to determine the value of the Infoobject at runtime for a BEx Query. Virtual Key Figure are not populated while data loading to the Cube/DSO (No transformation rule exist for the Virtual Characteristic / Key Figure from the underlying Datasource). Virtual Characteristic/Key Figure values are calculated using either BADI (RSR_OLAP_BADI) or Enhancement (RSR00002). Below the Virtual Characteristic/Key Figure are used for Infoproviders (Multiprovider/Infoset), whose values are determined using BADI (RSR_OLAP_BADI):

- **Multiprovider (Virtual Characteristic/Key Figure):** Multiprovider (ZM_SD_005) built on Infocube (0SD_C03: Sales: Overview) and DSO (0SD_005: Order: Condition Data). DSO 0SD_005 contains the Virtual Key Figure ZVIRTKEY (Virtual Key Figure) whose value would be calculated at query execution.

**Infoset (Virtual Characteristic/Key Figure):** Infoset (ZSD_005_) built on Infocube (0SD_C03: Sales: Overview), Infoobject 0MATERIAL (Material) and DSO (0SD_005: Order: Condition Data) contains the Virtual Key Figure ZVIRTKEY (Virtual Key Figure) whose value would be calculated at query execution.

**Multiprovider (Virtual Characteristic/Key Figure)**

**Step1:** Create a Virtual Key Figure ‘ZVIRTKEY’ and add the Key Figure to DSO (0SD_005: Order: Condition Data):

**Step2:** Query (ZM_SD_005_QUERY) built on Multiprovider (ZM_SD_005):
**Step 3:** Go to BADI Definition ‘RSR_OLP_BADI(Virtual Characteristics and Key Figures in Reporting)’ through transaction SE18 and create Implementation ‘ZM_SD_O05_KEY’ assigning Infoprovider (Filter) as ‘ZM_SD_O05’:

**Step 4:** Declare the variables for Characteristic and Key Figures in the Class Interface (ZCL_IM_M_SD_O05_KEY):

- **Key Figure Name:** P_KYF_<Key Figure Infoobject>
- **Characteristic Name:** P_CHA_<Characteristic Infoobject>
- **Level:** Instance Attribute, Visibility: Public
### Attribute | Infobject
--- | ---
P_CHA_0CALMONTH | Characteristic : 0CALMONTH (Calendar Year Month)
P_CHA_0DOC_NUMBER | Characteristic : 0DOC_NUMBER (Sales Document Number)
P_CHA_0S_ORD_ITEM | Characteristic: 0S_ORD_ITEM (Item)
P_CHA_0MATERIAL | Characteristic: 0MATERIAL (Material)
P_CHA_0DIVISION | Characteristic: 0DIVISION (Division)
P_KYF_0KPRICE | Key Figure: 0KPRICE (Condition Price)
P_KYF_ZVIRTKEY | Key Figure: ZVIRTKEY (Virtual Key Figure)

Write the ABAP Codes for methods IF_EX_RSR_OLAP_BADI~DEFINE, IF_EX_RSR_OLAP_BADI~INITIALIZE and IF_EX_RSR_OLAP_BADI~COMPUTE (Explained Below).

**Method - IF_EX_RSR_OLAP_BADI~DEFINE**

Method is used to append the internal tables for Characteristic (c_t-chanm) and Key Figures (c_t_kyfnm).

#### Class Builder: Class ZCL_IM_M_SD_005_KEY Display

<table>
<thead>
<tr>
<th>Ty.</th>
<th>Parameter</th>
<th>Type spec</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>H</td>
<td>L_RKBID</td>
<td>TYPE RSR_OLAP_BADI_FILTER</td>
<td>Filter for implementing Virtual Characteristics and Key Figures for RSR OLAP BADI Processor</td>
</tr>
<tr>
<td>H</td>
<td>L_TH_CHANM_USED</td>
<td>TYPE RRECE_TH_CHANM</td>
<td>Control Bar OLAP Processor</td>
</tr>
<tr>
<td>H</td>
<td>L_TH_KYFNM_USED</td>
<td>TYPE RRECE_TH_KYFNM</td>
<td>Control Bar OLAP Processor</td>
</tr>
<tr>
<td>H</td>
<td>C_T_CHANM</td>
<td>TYPE RRECE_T_CHANM</td>
<td>Control Bar OLAP Processor</td>
</tr>
<tr>
<td>H</td>
<td>C_T_KYFNM</td>
<td>TYPE RRECE_T_KYFNM</td>
<td>Control Bar OLAP Processor</td>
</tr>
<tr>
<td></td>
<td>OK_RS_ERROR</td>
<td>TYPE RRECE_T_KYFNM</td>
<td>BMW: General Error Class</td>
</tr>
</tbody>
</table>

**Method: IF_EX_RSR_OLAP_BADI~DEFINE**

```abap
DATA: i_s_chann TYPE rcks_s_chann,
     l_s_chann TYPE rcks_s_chann,
     l_kyfnm TYPE rcks_kyfnm,
     l_t_kyfnm TYPE rcks_t_kyfnm,
     c_t_chanm TYPE rcks_c_chann,
     c_t_kyfnm TYPE rcks_c_kyfnm,
     FIELD-STRINGS.

METHOD IF_EX_RSR_OLAP_BADI~DEFINE.
  IF_RS_ERROR = OK_RS_ERROR.
  DATA: l_s_chann TYPE rcks_s_chann.
    l_kyfnm TYPE rcks_kyfnm.
    l_t_kyfnm TYPE rcks_t_kyfnm.
    FIELD-STRINGS.
    l_s_chann TYPE rcks_s_chann.
    l_s_chann = 'OCALMONTH'.
    l_s_chann to c_t_chann.
    CLEAR l_s_chann.
    l_s_chann = '0DOC_NUMBER'.
    l_s_chann to c_t_chann.
    CLEAR l_s_chann.
    l_s_chann = '0S_ORD_ITEM'.
    l_s_chann to c_t_chann.
    CLEAR l_s_chann.
    l_s_chann = 'O_MATERIAL'.
    l_s_chann to c_t_chann.
    CLEAR l_s_chann.
  ENDIF.
ENDMETHOD.
```
method IF_EX_RSR_OLAP_BADI DEFINE.

DATA: l_s_chanm TYPE rrke_s_chanm,
      l_t_chanm TYPE rrke_t_chanm,
      l_kyfnm TYPE rsd_kyfnm,
      l_t_kyfnm TYPE rsd_t_kyfnm.

FIELD-SYMBOLS:
  <l_s_chanm> TYPE rrke_s_chanm.

CASE i_s_rkb1d-infocube.
  WHEN 'ZM_SD_O05'.
    l_s_chanm-chanm = 'OCALMONTH'.
    l_s_chanm-mode = rrke_c_mode-read.
    append l_s_chanm to c_t_chanm.
    CLEAR l_s_chanm.

    l_s_chanm-chanm = 'ODOC_NUMBER'.
    l_s_chanm-mode = rrke_c_mode-read.
    append l_s_chanm to c_t_chanm.
    CLEAR l_s_chanm.

    l_s_chanm-chanm = 'OS_ORD_ITEM'.
    l_sChanm-mode = rrke_c_mode-read.
    append l_s_chanm to c_t_chanm.
    CLEAR l_s_chanm.

    l_s_chanm-chanm = 'OMATERIAL'.
    l_sChanm-mode = rrke_c_mode-read.
    append l_s_chanm to c_t_chanm.
    CLEAR l_s_chanm.

    l_s_chanm-chanm = 'ODIVISION'.
    l_sChanm-mode = rrke_c_mode-read.
    append l_s_chanm to c_t_chanm.
    CLEAR l_sChanm.

  APPEND 'ZVIRTKEY' TO c_t_kyfnm.
  APPEND 'OKPRICE' TO c_t_kyfnm.
ENDCASE.

endmethod.
Method IF_EX_RSR_OLAP_BADI~INITIALIZE

Method is used to dynamically assign random numbers to the Characteristic and Key Figure Variables.

### Class Builder: Class ZCL_IM_M_SD_005_KEY Display

<table>
<thead>
<tr>
<th>Ty</th>
<th>Parameter</th>
<th>Type spec</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FILT_VAL</td>
<td>TYPE RSR_OLAP_BADI_FILTER</td>
<td>Filter for implementing Virtual Characteristics and Key Figs.</td>
<td></td>
</tr>
<tr>
<td>L_S_RKMD</td>
<td>TYPE RSR_S_RKMD</td>
<td>Control Bar OLAP Processor.</td>
<td></td>
</tr>
<tr>
<td>L_TH_SFC</td>
<td>TYPE RRRG_TH_SFC</td>
<td></td>
<td></td>
</tr>
<tr>
<td>L_TH_SFK</td>
<td>TYPE RRRG_TH_SFK</td>
<td></td>
<td></td>
</tr>
<tr>
<td>L_S_DATA</td>
<td>TYPE ANY</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C_LG_ERROR</td>
<td></td>
<td>BW General Error Class.</td>
<td></td>
</tr>
</tbody>
</table>

### Method IF_EX_RSR_OLAP_BADI~INITIALIZE

```plaintext
DATA: l_global_name TYPE string.
FIELD-SYMBOLS:
  <l_global> TYPE i,
  <l_s_sfc> TYPE rrrg_s_sfc,
  <l_s_sfk> TYPE rrrg_s_sfk.

* there's no need to change this method
* Just create attributes for each characteristic
  * with name P_CHA_<characteristic> TYPE i.
* and constants for each key figure with name
  * P_KVF_<key figure> TYPE i.

CLASS cl_exm_im_rsr_olap_badi DEFINITION LOAD.
```

```plaintext
/* method IF_EX_RSR_OLAP_BADI~INITIALIZE. */
DATA: l_global_name TYPE string.
FIELD-SYMBOLS:
  <l_global> TYPE i,
  <l_s_sfc> TYPE rrrg_s_sfc,
  <l_s_sfk> TYPE rrrg_s_sfk.

* there's no need to change this method
* Just create attributes for each characteristic
  * with name P_CHA_<characteristic> TYPE i.
* and constants for each key figure with name
  * P_KVF_<key figure> TYPE i.

CLASS cl_exm_im_rsr_olap_badi DEFINITION LOAD.
```
* get field positions for characteristics in structure
  LOOP AT i_th_sfc ASSIGNING <l_s_sfc>
    WHERE user_exit NE rrke_c_mode-None.
  *[field name in structure is keyreturnnm]*
  * name of the global variable
    CONCATENATE 'P_CHA' <l_s_sfc>-chanm
    INTO l_global_name
    SEPARATED BY '_'.
  *[fill the global variable]*
    UNASSIGN <l_global>.
    ASSIGN (l_global_name) TO <l_global>.
    CHECK <l_global> IS ASSIGNED.
    <l_global> = cl_exm_im_rsr_olap_badi=>get_field_position_d(    i_fieldnm = <l_s_sfc>-keyreturnnm
    i_s_data = i_s_data ).
  ENDLOOP.

* get field positions for key figures in structure
  LOOP AT i_th_sfk ASSIGNING <l_s_sfk>.
  *[name of the global variable]*
    CONCATENATE 'P_KYF' <l_s_sfk>-kyfnm
    INTO l_global_name
    SEPARATED BY '_'.
  *[fill the global variable]*
    UNASSIGN <l_global>.
    ASSIGN (l_global_name) TO <l_global>.
    CHECK <l_global> IS ASSIGNED.
    <l_global> = cl_exm_im_rsr_olap_badi=>get_field_position_d(    i_fieldnm = <l_s_sfk>-VALUE_RETURNNM
    i_s_data = i_s_data ).
  ENDLOOP.

endmethod.
Assign value ‘10’ to Virtual Key Figure ‘ZVRTKEY’.

**Step 1**: Go to transaction RSRT and execute the query ‘ZM_SD_O05_QUERY’:

![Calendar Year Overview (Virtual) Multiprovider](image)

**Selection Screen Input.**

**Step 2**: DEBUG the code and check the value of Virtual Key figure:
### Report Output

#### BW - output test

<table>
<thead>
<tr>
<th>Calendar Year/Month</th>
<th>Sales document</th>
<th>Item</th>
<th>Material</th>
<th>Division</th>
<th>Condition Rate</th>
<th>Condition Code</th>
<th>Condition Date</th>
<th>Sales Date</th>
<th>Sales Amount</th>
</tr>
</thead>
<tbody>
<tr>
<td>'01-'05</td>
<td>'0270'</td>
<td>'18</td>
<td>'01'</td>
<td>Sunny Sunny 01 187</td>
<td>High Tech 30,1925 EUR</td>
<td>10,000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>'0277'</td>
<td>'0277'</td>
<td>'18</td>
<td>'01'</td>
<td>Sunny Sunny 01 187</td>
<td>High Tech 30,1925 EUR</td>
<td>10,000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>'0292'</td>
<td>'0292'</td>
<td>'18</td>
<td>'01'</td>
<td>Sunny Sunny 01 187</td>
<td>High Tech 30,1925 EUR</td>
<td>10,000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>'0296'</td>
<td>'0296'</td>
<td>'18</td>
<td>'01'</td>
<td>Sunny Sunny 01 187</td>
<td>High Tech 30,1925 EUR</td>
<td>10,000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>'0300'</td>
<td>'0300'</td>
<td>'18</td>
<td>'01'</td>
<td>Sunny Sunny 01 187</td>
<td>High Tech 30,1925 EUR</td>
<td>10,000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>'0304'</td>
<td>'0304'</td>
<td>'18</td>
<td>'01'</td>
<td>Sunny Sunny 01 187</td>
<td>High Tech 30,1925 EUR</td>
<td>10,000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>'0308'</td>
<td>'0308'</td>
<td>'18</td>
<td>'01'</td>
<td>Sunny Sunny 01 187</td>
<td>High Tech 30,1925 EUR</td>
<td>10,000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>'0312'</td>
<td>'0312'</td>
<td>'18</td>
<td>'01'</td>
<td>Sunny Sunny 01 187</td>
<td>High Tech 30,1925 EUR</td>
<td>10,000</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

© 2010 SAP AG
Infoset (Virtual Characteristic/Key Figure)

**Step 1**: Assign the Virtual Key Figure `ZVIRTKEY (Virtual Key Figure)` to DSO (0SD_O05: Order: Condition Data):

Infoset `ZSD_O05_I (Infoset (DSO 0SD_O05))` is built on Infocube (0SD_C03: Sales: Overview), Infoobject 0MATERIAL (Material) and DSO (0SD_O05: Order: Condition Data) which contains the Virtual Key Figure `ZVIRTKEY (Virtual Key Figure)` whose value would be calculated at query execution.

**Step 2**: Create a Query on the Infoset (ZSD_O05_I):
Step 3: Go to BADI Definition ‘RSR_OLPAD_BADI (Virtual Characteristics and Key Figures in Reporting)’ through transaction SE18 and create Implementation ‘ZM_SD_005_KEY’ assigning Infoprovider (Filter) as ‘ZSD_005_I’.

Implementation Name | ZSD_005_I.IMPLE | Active
Implementation Short Text | Virtual Characteristic ZSD_005_I
Definition name | RSR_OLPAD_BADI

Step 4: Declare the Variables for Characteristics and Key Figures:

Class Builder: Display Class ZCL_IM_SD_005_I.IMPLE
Characteristic Variable: P_CHA_<Characteristic Name from Query Designer>
Key Figure Variable: P_KYF_<Key Figure from Query Designer>
Level: Instance Attribute
Visibility: Public
Associated Type: I

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Query Characteristic/Key Figure</th>
<th>Characteristic/Key Figure</th>
</tr>
</thead>
<tbody>
<tr>
<td>P_CHA_ZSD_O05_I___F64</td>
<td>ZSD_O05_I___F64</td>
<td>Characteristic : 0CALMONTH</td>
</tr>
<tr>
<td>P_CHA_ZSD_O05_I___F1</td>
<td>ZSD_O05_I___F1</td>
<td>Characteristic : 0DOC_NUMBER</td>
</tr>
<tr>
<td>P_CHA_ZSD_O05_I___F2</td>
<td>ZSD_O05_I___F2</td>
<td>Characteristic : 0S_ORD_ITEM</td>
</tr>
<tr>
<td>P_CHA_ZSD_O05_I___F6</td>
<td>ZSD_O05_I___F6</td>
<td>Characteristic : 0MATERIAL</td>
</tr>
<tr>
<td>P_CHA_ZSD_O05_I___F5</td>
<td>ZSD_O05_I___F5</td>
<td>Characteristic : 0KNCOUNTER</td>
</tr>
<tr>
<td>P_CHA_ZSD_O05_I___F4</td>
<td>ZSD_O05_I___F4</td>
<td>Characteristic : 0KNART</td>
</tr>
<tr>
<td>P_KYF_ZSD_O05_I___F43</td>
<td>ZSD_O05_I___F43</td>
<td>Key Figure: 0KPRICE</td>
</tr>
<tr>
<td>P_KYF_ZSD_O05_I___F40</td>
<td>ZSD_O05_I___F40</td>
<td>Key Figure: 0EXCHG_RATE</td>
</tr>
</tbody>
</table>

Write the ABAP Code for methods IF_EX_RSR_OLAP_BADI~DEFINE, IF_EX_RSR_OLAP_BADI~INITIALIZE and IF_EX_RSR_OLAP_BADI~COMPUTE to fill Virtual Key Figure values.
Method - IF_EX_RSR_OOLAP_BADI~DEFINE

Fill the internal tables for characteristic (c_t_chanm) and Key Figures (c_t_kyfnn).

**Class Builder: Class ZCL_IM_SD_005_Iimple Display**

<table>
<thead>
<tr>
<th>Ty</th>
<th>Parameter</th>
<th>Type spec.</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FLT_VAL</td>
<td>FILTER_RSR_OOLAP_BADI</td>
<td>TYPE RSR_OOLAP_BADI_FILTER</td>
<td>Filter for implementing virtual characteristics and Key Figs</td>
</tr>
<tr>
<td>l_s_rkb1d</td>
<td>TYPE RSR_S_RKB1D</td>
<td>Control Bar OLAP Processor</td>
<td></td>
</tr>
<tr>
<td>l_th_channm_used</td>
<td>TYPE RRKE_TH_CHANNM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>l_th_kyfnn_used</td>
<td>TYPE RRKE_TH_KYFNN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>c_t_chanm</td>
<td>TYPE RRKE_T_CHANNM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>c_t_kyfnn</td>
<td>TYPE RRKE_T_KYFNN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CA_RES_ERROR</td>
<td>TYPE BW_GENERAL_ERROR</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Method</th>
<th>IF_EX_RSR_OOLAP_BADI~DEFINE</th>
<th>Active</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>method IF_EX_RSR_OOLAP_BADI~DEFINE.</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>DATA: l_s_chanm TYPE rrke_s_chanm,</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>l_t_chanm TYPE rrke_t_chanm,</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>l_kyfnn TYPE rsd_kyfnn,</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>l_t_kyfnn TYPE rsd_t_kyfnn.</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>FIELD-SYMBOLS:</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>&lt;l_s_chanm&gt; TYPE rrke_s_chanm.</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>CASE i_s_rkb1d-infocube.</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>WHEN 'ZSD_005_I'.</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>* Calendar Year Month</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>l_s_chanm-chanm = 'P_CHA_ZSD_005_I___F64'.</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>l_s_chanm-mode = rrke_c_mode-read.</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>append l_s_chanm to c_t_chanm.</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>CLEAR l_s_chanm.</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td></td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>* Sales Document</td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>l_s_chanm-chanm = 'P_CHA_ZSD_005_I___F1'.</td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>l_s_chanm-mode = rrke_c_mode-read.</td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>append l_s_chanm to c_t_chanm.</td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>CLEAR l_s_chanm.</td>
<td></td>
</tr>
<tr>
<td>26</td>
<td></td>
<td></td>
</tr>
<tr>
<td>27</td>
<td></td>
<td></td>
</tr>
<tr>
<td>28</td>
<td>* Items</td>
<td></td>
</tr>
<tr>
<td>29</td>
<td>l_s_chanm-chanm = 'P_CHA_ZSD_005_I___F2'.</td>
<td></td>
</tr>
<tr>
<td>30</td>
<td>l_s_chanm-mode = rrke_c_mode-read.</td>
<td></td>
</tr>
<tr>
<td>31</td>
<td>append l_s_chanm to c_t_chanm.</td>
<td></td>
</tr>
</tbody>
</table>

**method IF_EX_RSR_OOLAP_BADI~DEFINE.**

DATA: l_s_chanm TYPE rrke_s_chanm, |
| 1_t_chanm TYPE rrke_t_chanm, |
| l_kyfnn TYPE rsd_kyfnn, |
| l_t_kyfnn TYPE rsd_t_kyfnn. |

FIELD-SYMBOLS: |
| <l_s_chanm> TYPE rrke_s_chanm. |

CASE i_s_rkb1d-infocube. |
| WHEN 'ZSD_005_I'. |

* Calendar Year Month |
| l_s_chanm-chanm = 'P_CHA_ZSD_005_I___F64'. |
| l_s_chanm-mode = rrke_c_mode-read. |
| append l_s_chanm to c_t_chanm. |
| CLEAR l_s_chanm. |

* Sales Document |
| l_s_chanm-chanm = 'P_CHA_ZSD_005_I___F1'. |
| l_s_chanm-mode = rrke_c_mode-read. |
| append l_s_chanm to c_t_chanm.
CLEAR l_s_chanm.

* Item
  l_s_chanm-chanm = 'P_CHA_ZSD_O05_I___F2'.
  l_sChanm-mode = rrke_c_mode-read.
  append l_s_chanm to c_t_chanm.
  CLEAR l_s_chanm.

* Material
  l_s_chanm-chanm = 'P_CHA_ZSD_O05_I___F6'.
  l_sChanm-mode = rrke_c_mode-read.
  append l_s_chanm to c_t_chanm.
  CLEAR l_s_chanm.

* Condition Counter
  l_s_chanm-chanm = 'P_CHA_ZSD_O05_I___F5'.
  l_sChanm-mode = rrke_c_mode-read.
  append l_s_chanm to c_t_chanm.
  CLEAR l_s_chanm.

* Condition Type
  l_s_chanm-chanm = 'P_CHA_ZSD_O05_I___F4'.
  l_sChanm-mode = rrke_c_mode-read.
  APPEND l_s_chanm TO c_t_chanm.
  CLEAR l_s_chanm.

* Key Figures
  APPEND 'P_KYF_ZSD_O05_I___F40' TO c_t_kyfnm.

       APPEND 'P_KYF_ZSD_O05_I___F43' TO c_t_kyfnm.
       ENDCASE.
endmethod.
Method IF_EX_RSR OLAP BADI~INITIALIZE

Characteristic and Key Figure variables are populated at query execution run time:

```table
<table>
<thead>
<tr>
<th>Ty</th>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>FLT_VAL</td>
<td>TYPE RSR OLAP BADI_FILTER</td>
<td>Filter for implementing Virtual Characteristics and Key Fig</td>
</tr>
<tr>
<td></td>
<td>L_TH_SFC</td>
<td>TYPE RSR OLAP BADI_TH_SFC</td>
<td>Control for OLAP Procedure</td>
</tr>
<tr>
<td></td>
<td>L_TH_SFK</td>
<td>TYPE RSR OLAP BADI_TH_SFK</td>
<td></td>
</tr>
<tr>
<td></td>
<td>L чха</td>
<td>TYPE ANY</td>
<td></td>
</tr>
<tr>
<td></td>
<td>CN_RG_ERROR</td>
<td>TYPE ANY</td>
<td></td>
</tr>
</tbody>
</table>
```

Method IF_EX_RSR OLAP BADI~INITIALIZE

```abap
DATA: l_global_name TYPE string.
FIELD-SYMBOLES:
   <l_global> TYPE i,
   <l_s_sfc> TYPE rrkg_s_sfc,
   <l_s_sfk> TYPE rrkg_s_sfk.

* there's no need to change this method
* Just create attributes for each characteristic
* with name P_CHA<characteristic> TYPE i.
* and constants for each key figure with name
* P_KYF<key figure> TYPE i.

CLASS cl_exm_im_rsr_olap_badi DEFINITION LOAD.

* get field positions for characteristics in structure
LOOP AT i_th_sfc ASSIGNING <l_s_sfc>
   WHERE user_exit NE rrke_c_mode-none.
* field name in structure is keyreturnnm
* name of the global variable
CONCATENATE 'P_CHA' <l_s_sfc>-chanm INTO l_global_name
   SEPARATED BY '_'.
* fill the global variable
UNASSIGN <l_global>.
ASSIGN (l_global_name) TO <l_global>.
CHECK <l_global> IS ASSIGNED.
<l_global> = cl_exm_im_rsr_olap_badi->get_field_position_d(
   i_fieldnm = <l_s_sfc>-keyreturnnm
   i_data = <l_s_data>).`
i_s_data = i_s_data).

ENDLOOP.

* get field positions for key figures in structure
LOOP AT i_th_sfk ASSIGNING <l_s_sfk>.
* name of the global variable
  CONCATENATE 'P_KYF' <l_s_sfk>-kyfnm
  INTO l_global_name
  SEPARATED BY '_'.
* fill the global variable
  UNASSIGN <l_global>.
  ASSIGN (l_global_name) TO <l_global>.
  CHECK <l_global> IS ASSIGNED.
  <l_global> = cl_exm_im_rsr_olap_badi=>get_field_position_d(
    i_fieldnm = <l_s_sfk>-VALUE_RETURNNM
    i_s_data = i_s_data).
  ENDLOOP.
endmethod.
Method IF_EX_RSR_OLAP_BADI~COMPUTE

Virtual Key Figure values would be calculated based on below conditions:
If Condition Type EQ 'MSWT', then Virtual Key Figure EQ '10'.
If Condition Type EQ 'PR00', then Virtual Key Figure EQ '20'.
If Condition Type EQ 'SKTO', then Virtual Key Figure EQ '30'.
If Condition Type EQ 'VPRS', then Virtual Key Figure EQ '50'.
If Other Condition Type, Virtual Key Figure EQ '50'.

Assign component P_CHA_ZSD_005_I___F64 of structure c_s_data to <FS_0CALMONTH>.
Assign component P_CHA_ZSD_005_I___F1 of structure c_s_data to <FS_0DDOC_NUMBER>.
Assign component P_CHA_ZSD_005_I___F2 of structure c_s_data to <FS_0S_ORD_ITEM>.
Assign component P_CHA_ZSD_005_I___F6 of structure c_s_data to <FS_0MATERIAL>.
Assign component P_CHA_ZSD_005_I___F5 of structure c_s_data to <FS_0KNCOUNTER>.
Assign component P_CHA_ZSD_005_I___F4 of structure c_s_data to <FS_0KNART>.
ASSIGN COMPONENT P_KYF_ZSD_O05_I___F40 OF STRUCTURE c_s_data TO <FS_0KPRICE>.
ASSIGN COMPONENT P_KYF_ZSD_O05_I___F43 OF STRUCTURE c_s_data TO <FS_ZVIRTKEY>.

CASE <FS_OKNART> .
  * IF condition Type EQ 'MWST', then exchange rate EQ 10.
  WHEN 'MWST' .
    <FS_ZVIRTKEY> = '10' .
  WHEN 'PR00' .
    * IF condition Type EQ 'PR00', then exchange rate EQ 20.
    <FS_ZVIRTKEY> = '20' .
  WHEN 'SKTO' .
    * IF condition Type EQ 'SKTO', then exchange rate EQ 30.
    <FS_ZVIRTKEY> = '30' .
  WHEN 'VPRS' .
    * IF condition Type EQ 'SKTO', then exchange rate EQ 50.
    <FS_ZVIRTKEY> = '40' .
  WHEN OTHERS .
    <FS_ZVIRTKEY> = '50' .
ENDCASE.

Note: Check the SAP Support Package (SAP Notes 657690 and 618738) before executing queries for Infoset having Virtual Characteristic/Key Figures.
Related Content
For more information, visit 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.