**SAP BI - Virtual Infocube based on Function Module (Transport History)**

**Applies to:**
SAP BW 3.5 / BI 7.0 consultants accustomed with SAP ABAP skills. For more information, visit EDW Homepage.

**Summary**
Explains the method to fetch the transport details from relevant database table (E070 - Header of Requests/Tasks, E07T-Short Texts for Requests/Tasks) using Virtual InfoCube with Services.

**Author:** Suraj Tigga
**Company:** Capgemini Consulting Pvt. Ltd.
**Created on:** 05 April 2011

**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

Step-by-Step Solution ............................................................................................................ 3
  Virtual InfoCube ................................................................................................................. 4
  Function Module (Virtual InfoCube) .................................................................................... 5
  Infocube Data Display ....................................................................................................... 7
  Function Module Code (Complete) .................................................................................... 8

Related Content ..................................................................................................................... 12

Disclaimer and Liability Notice ............................................................................................ 13
Scenario

Fetch the transport details from relevant database table (E070 - Header of Requests/Tasks, E07T - Short Texts for Requests/Tasks) using Virtual InfoCube with Services (Using Function Module).

Step-by-Step Solution

Virtual InfoCube can be of three different types:

**SAP Remote Cube**: Remote Cube that allows the definition of queries with direct access to transaction data in other SAP systems.

**Remote Cube**: A Remote Cube is an InfoCube whose transaction data is not managed in the Business Warehouse but externally. Only the structure of the RemoteCube is defined in BW. The data is read for reporting using a BAPI from another System.

**Virtual InfoCube with Services**: A Virtual Infocube with Services is an Infocube that does not have its own physical data storage in BW. A user-defined function module is used as the DataSource. This type of InfoProvider is used primarily in the SAP SEM application. To display data from non-BW datasources in BW Virtual InfoCube with Services can be used without having to copy the data set into the BW structures. Data can be local or remote. Also it provides the flexibility to use our own calculations to change the data before it is passed to the OLAP processor.

This document explains the method to create Virtual Infocube with Services and to fetch data using Function Module. Below are the steps mentioned to populate data to Virtual InfoCube:

**Virtual InfoCube**: Details about the Virtual InfoCube.

**Function Module (Virtual InfoCube)**: Function Module fetches transport details from E070, E07T and feeds while InfoCube data display.

**Infocube Data Display**: Display the Virtual InfoCube data.
Virtual Infocube

**Step 1**: Go to Administrator Workbench and create the Virtual InfoCube ‘YTRAN_VIR’:

![Virtual InfoCube](image)

**Details:**

![Virtual InfoProvider with Function Module](image)
Characteristic in Virtual InfoCube:

<table>
<thead>
<tr>
<th>InfoCube</th>
<th>Techn. name / value</th>
<th>Fu.</th>
<th>O.</th>
<th>Data</th>
<th>L</th>
<th>Key F.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Virtual InfoCube (Transport History)</td>
<td>YTRAN_VIR:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Version</td>
<td>In Process</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Save</td>
<td>Saved</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Revised Version</td>
<td>Active Version</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Object Status</td>
<td>Active, executable</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Settings:
- Type: Virtual InfoProvider
- Subtype: Based on Function ...

Dimensions:
- Data Package: YTRAN_VIRP
- Time: YTRAN_VIRT
- Unit: YTRAN_VIRU
- Transport: YTRAN_VIR1

- Request/Task: YTRKORR
- Request or task category: YKORRDEV
- Owner of a Request or Task: YAS4USER
- Date of Last Change: YAS4DATE
- Last changed at: YAS4TIME
- Short Description of Repository: YAS4TEXT

Function Module (Virtual InfoCube)

Step 1: Go to transaction SE37 and create a Function Module 'Y_VIRT_TRANSPORT_HIST'.

Import Parameters:

Function Builder: Display Y_VIRT_TRANSPORT_HIST
Export Parameters:

**Function Builder: Display Y_VIRT_TRANSPORT_HIST**

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Typing</th>
<th>Associated Type</th>
<th>Pass Val</th>
<th>Short text</th>
<th>Long Text</th>
</tr>
</thead>
<tbody>
<tr>
<td>E_T_DATA</td>
<td>TYPE</td>
<td>STANDARD TABLE</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>E_END_OF_DATA</td>
<td>TYPE</td>
<td>RS_BOOLEAN</td>
<td></td>
<td>Boolean</td>
<td></td>
</tr>
<tr>
<td>E_T_MSG</td>
<td>TYPE</td>
<td>RS_T_MSG</td>
<td></td>
<td>BKY: Table with Messages (Application)</td>
<td></td>
</tr>
</tbody>
</table>

Exceptions:

**Function Builder: Display Y_VIRT_TRANSPORT_HIST**

<table>
<thead>
<tr>
<th>Exception</th>
<th>Short text</th>
<th>Long text</th>
</tr>
</thead>
<tbody>
<tr>
<td>WRONG_INPUT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>READ_ACCESS_ERROR</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Step2:** Fetch the data from transport tables E070 and E07T:

```plaintext
28 CHECK i_infoprov = 'YTRAN_VIR'.
29
30 SELECT a-trkorr
31   a-korrdev
32   a-as4user
33   a-as4date
34   a-as4time
35   b-as4text
36 INTO TABLE i_t_e070
37 FROM e070 AS a
38 INNER JOIN e07t AS b
39 ON a-trkorr EQ b-trkorr.
40 IF sy-subrc EQ 0.
```

Store the values to internal table *i_t_e070*. 
Move the fetched data from internal table \texttt{i\_t\_e070} to the export table \texttt{e\_t\_data} for each infoobject present in the Virtual InfoCube:

```
* get description of components or \texttt{<L\_S\_DATA>}

PERFORM get_type_components(saprlsdrvc_services)
USING \texttt{i\_s\_data}.
CHANGING \texttt{i\_t\_component}.

LOOP AT \texttt{i\_t\_e070} ASSIGNING \texttt{i\_s\_e070}.
PERFORM move_e070_to_data
USING \texttt{i\_s\_e070}.
  i_th_sfc
  i_th_six
  i_t_component
CHANGING \texttt{i\_s\_data}.

* append to output data

APPEND \texttt{i\_s\_data} TO \texttt{e\_t\_data}.
ENDLOOP.

* all data are selected

\texttt{e\_end\_of\_data} = 'X'.
```

Infocube Data Display

**Step1**: Go to administrator workbench and display data for InfoCube `YTRAN_VIR`:

![Infocube Data Display](image)

**InfoCube Data**:

**"YTRAN_VIR", List output**

<table>
<thead>
<tr>
<th>Request Task</th>
<th>YCORRECTOR</th>
<th>YASUSER</th>
<th>YASDATE</th>
<th>YASTIME</th>
<th>Short Description of Repository Object</th>
</tr>
</thead>
<tbody>
<tr>
<td>GC COMPLETE</td>
<td>SYST</td>
<td>J/SAP</td>
<td>10:23:18</td>
<td></td>
<td>Blockliste der Garbage Collections</td>
</tr>
<tr>
<td>ALL_RDIM1</td>
<td>SAP</td>
<td>J/SAP</td>
<td>14:19:23</td>
<td></td>
<td>Blockliste für Massengeneratoren</td>
</tr>
<tr>
<td>ALL_RDIM2</td>
<td>SAP</td>
<td>J/SAP</td>
<td>14:19:24</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ALL_RDIM3</td>
<td>SAP</td>
<td>J/SAP</td>
<td>14:19:24</td>
<td></td>
<td>Andem Tabelle</td>
</tr>
<tr>
<td>B20K990831</td>
<td>SYST</td>
<td>HOFRICH</td>
<td>02:16:01</td>
<td></td>
<td>80EN: Avoid TODIR access with EhPI</td>
</tr>
<tr>
<td>BBP_WV_BASIS</td>
<td>SYST</td>
<td>J/SAP</td>
<td>10:23:17</td>
<td></td>
<td>Blockliste für BBP_WV_BASIS</td>
</tr>
<tr>
<td>BDE_FTD_NWYPN_12030</td>
<td>SYST</td>
<td>J/SAP</td>
<td>23:33:57</td>
<td></td>
<td>Piece List</td>
</tr>
<tr>
<td>BID_P00001</td>
<td>SYST</td>
<td>10055184</td>
<td>15:35:49</td>
<td></td>
<td>Generated Project Piece List</td>
</tr>
<tr>
<td>BID*000002</td>
<td>SYST</td>
<td>SUPPORTPACK</td>
<td>ORT/TPSUPPORT</td>
<td>15:53:40</td>
<td>Define BIDCLNT500 logical system</td>
</tr>
<tr>
<td>BID*000003</td>
<td>SYST</td>
<td>SUPPORTPACK</td>
<td>ORT/TPSUPPORT</td>
<td>18:45:37</td>
<td>Define BIDCLNT500 logical system</td>
</tr>
</tbody>
</table>
**Function Module Code (Complete)**

### Function Module Code (Y_VIRT_TRANSPORT_HIST):  

```abap
FUNCTION y_virt_transport_hist.
  """---------------------------------------------------------------------
  """Local Interface:
  """IMPORTING
  """  VALUE(I_INFOPROV) TYPE RSINFOPROV
  """  REFERENCE(I_TH_SFC) TYPE RSDRI_TH_SFC
  """  REFERENCE(I_TH_SFK) TYPE RSDRI_TH_SFK
  """  REFERENCE(I_T_RANGE) TYPE RSDRI_T_RANGE
  """  REFERENCE(I_TX_RANGETAB) TYPE RSDRI_TX_RANGETAB
  """  REFERENCE(I_FIRST_CALL) TYPE RS_BOOL
  """  REFERENCE(I_PACKAGESIZE) TYPE I DEFAULT 1000
  """EXPORTING
  """  REFERENCE(E_T_DATA) TYPE STANDARD TABLE
  """  REFERENCE(E_END_OF_DATA) TYPE RS_BOOL
  """  REFERENCE(E_T_MSG) TYPE RS_T_MSG
  """EXCEPTIONS
  """  WRONG_INPUT
  """  READ_ACCESS_ERROR
  """---------------------------------------------------------------------

* initialize
  FIELD-SYMBOLS: <l_s_e070> TYPE typ_e070,
                   <l_s_data> TYPE ANY.
  DATA: l_t_component TYPE abap_compdescr_tab,
        i_t_e070 TYPE t_e070.
  CLEAR: e_t_data, e_t_msg.
* this is specific to infoprovider VIRTICUBE1
  CHECK i_infoprov = 'YTRAN_VIR'.

SELECT a~trkorr
    a~korrdev
    a~as4user
    a~as4date
    a~as4time
    b~as4text
INTO TABLE i_t_e070
FROM e070 AS a
  INNER JOIN e07t AS b
  ON a~trkorr EQ b~trkorr.
  IF sy-subrc EQ 0.
* create a working area
  ASSIGN LOCAL COPY OF INITIAL LINE OF e_t_data TO <l_s_data>.
* get description of components of <L_S_DATA>
  PERFORM get_type_components(saplrsdrc_services)
  USING <l_s_data>
  CHANGING l_t_component.

  LOOP AT i_t_e070 ASSIGNING <l_s_e070>.
    PERFORM move_e070_to_data
    USING <l_s_e070>
    i_th_sfc
    i_th_sfk
    l_t_component
    CHANGING <l_s_data>.
* append to output data
```

---

**SAP BI - Virtual Infocube based on Function Module (Transport History)**

© 2010 SAP AG
APPEND <l_s_data> TO e_t_data.
ENDLOOP.

*all data are selected
  e_end_of_data = 'X'.
ENDIF.
ENDFUNCTION.

Include (LY_BIF01):

**----------------------------------------

***INCLUDE LY_BIF01.
**----------------------------------------

GET the values for Infocube

FORM move_e070_to_data USING i_s_e070 TYPE typ_e070
  i_th_sfc TYPE rsdri_th_sfc
  i_th_sfk TYPE rsdri_th_sfk
  i_t_component TYPE abap_compdescr_tab
CHANGING e_s_data TYPE any.

define FIELD-SYMBOLS
FIELD-SYMBOLS: <l_s_component> TYPE abap_compdescr,
  <l_comp_data> TYPE ANY,
  <l_s_sfc> TYPE rsdri_s_sfc,
  <l_s_sfk> TYPE rsdri_s_sfk.

Datadeclaration
DATA: l_compno TYPE i,
  l_t_component TYPE abap_compdescr_tab,
  l_s_e070 TYPE typ_e070.

initialize
  CLEAR e_s_data.
  l_s_e070 = i_s_e070.

Request/Task
  IF NOT i_s_e070-trkorr IS INITIAL.
      READ TABLE i_th_sfc ASSIGNING <l_s_sfc> WITH TABLE KEY chanm = 'YTRKORR'.
      IF sy-subrc = 0.
  * get number of target column
      READ TABLE i_t_component ASSIGNING <l_s_component> WITH KEY name = <l_s_sfc>-chaalias.
      IF sy-subrc = 0.
  * number of target data component
      l_compno = sy-tabix.
  * get target data component
      ASSIGN COMPONENT l_compno OF STRUCTURE e_s_data TO <l_comp_data>.
      MOVE i_s_e070-trkorr TO <l_comp_data>.
  ENDIF.
  ENDIF.
  ENDIF.

* Request or task category
  IF NOT i_s_e070-korrdev IS INITIAL.
READ TABLE i_th_sfc ASSIGNING <l_s_sfc> WITH TABLE KEY chanm = 'YKORRDEV'.
IF sy-subrc = 0.
* get number of target column
  READ TABLE i_t_component ASSIGNING <l_s_component> WITH KEY name = <l_s_sf c>-chaalias.
  IF sy-subrc = 0.
  * number of target data component
    l_compno = sy-tabix.
    * get target data component
      ASSIGN COMPONENT l_compno OF STRUCTURE e_s_data TO <l_comp_data>.
      MOVE i_s_e070-korrdev TO <l_comp_data>.
    ENDF.
  ENDF.
ENDIF.
ENDIF.

* Owner of a Request or Task
IF NOT i_s_e070-as4user IS INITIAL.
  READ TABLE i_th_sfc ASSIGNING <l_s_sfc> WITH TABLE KEY chanm = 'YAS4USER'.
  IF sy-subrc = 0.
  * get number of target column
    READ TABLE i_t_component ASSIGNING <l_s_component> WITH KEY name = <l_s_sf c>-chaalias.
    IF sy-subrc = 0.
    * number of target data component
      l_compno = sy-tabix.
      * get target data component
        ASSIGN COMPONENT l_compno OF STRUCTURE e_s_data TO <l_comp_data>.
        MOVE i_s_e070-as4user TO <l_comp_data>.
      ENDF.
    ENDF.
  ENDF.
ENDIF.

* Date of Last Change
IF NOT i_s_e070-as4date IS INITIAL.
  READ TABLE i_th_sfc ASSIGNING <l_s_sfc> WITH TABLE KEY chanm = 'YAS4DATE'.
  IF sy-subrc = 0.
  * get number of target column
    READ TABLE i_t_component ASSIGNING <l_s_component> WITH KEY name = <l_s_sf c>-chaalias.
    IF sy-subrc = 0.
    * number of target data component
      l_compno = sy-tabix.
      * get target data component
        ASSIGN COMPONENT l_compno OF STRUCTURE e_s_data TO <l_comp_data>.
        MOVE i_s_e070-as4user TO <l_comp_data>.
      ENDF.
    ENDF.
  ENDF.
ENDIF.

* Last changed at
IF NOT i_s_e070-as4time IS INITIAL.
  READ TABLE i_th_sfc ASSIGNING <l_s_sfc> WITH TABLE KEY chanm = 'YAS4TIME'.
  IF sy-subrc = 0.
  * get number of target column
    READ TABLE i_t_component ASSIGNING <l_s_component> WITH KEY name = <l_s_sf c>-chaalias.
    IF sy-subrc = 0.
* number of target data component
  l_compno = sy-tabix.
* get target data component
  ASSIGN COMPONENT l_compno OF STRUCTURE e_s_data TO <l_comp_data>.
  MOVE i_s_e070-as4time TO <l_comp_data>.
  ENDSB.
  ENDSB.
ENDIF.
ENDIF.
ENDIF.

* Short Description of Repository Objects
  IF NOT i_s_e070-as4text IS INITIAL.
    READ TABLE i_th_sfc ASSIGNING <l_s_sfc> WITH TABLE KEY chanm = 'YAS4TEXT'.
    IF sy-subrc = 0.
  * get number of target column
    READ TABLE i_t_component ASSIGNING <l_s_component> WITH KEY name = <l_s_sfc>-chaalias.
    IF sy-subrc = 0.
  * number of target data component
    l_compno = sy-tabix.
  * get target data component
    ASSIGN COMPONENT l_compno OF STRUCTURE e_s_data TO <l_comp_data>.
    MOVE i_s_e070-as4text TO <l_comp_data>.
    ENDSB.
    ENDSB.
  ENDSB.
ENDIF.
ENDIF.
ENDIF.

ENDFORM.            " MOVE_E070_TO_DATA
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.