The page features a decorative graphic consisting of three blue circles of varying sizes, each composed of concentric rings of different shades of blue. These circles are arranged in a vertical line, with the largest at the top, a medium one in the middle, and the largest at the bottom. Two thin blue lines intersect at the top left and extend diagonally across the page, framing the circles.

BCV for TM

Create and maintain BCV Content in SAP TM
8.0 and SAP TM 8.1

Contents

- Introduction..... 3
- Prerequisites..... 3
- Overview about BCV in SAP TM 3
 - General information about the BCV 3
 - Some technical Information about BCV 4
- Create BCV Content for SAP TM..... 5
 - Activate the side panel link for BCV 5
 - Create a Context Key 6
 - Send the Context Key to the Business Context Viewer 7
 - Configuration of the BCV..... 8
 - Search Connector 8
 - Search Connector Type BAPI 9
- BCV Customizing..... 11
 - Search Connector 11
- BCV Query 12
 - Query View 13
 - Overview Page..... 16
- Appendix..... 17
 - The Side Panel link..... 17
- FAQ's 19
 - I try to open the Side Panel and a shortdump occurs. 19
 - I open the Side Panel and my created query will not be displayed. 19
 - One of my queries gets a Timeout. 19

Introduction

This document describes the creation and the maintenance of Business Context Viewer (BCV) Content for SAP TM 8.0 and SAP TM 8.1.

After reading this document you are able to define new BCV Content.

This document is not an introduction to the BCV technology. It describes only the implementation of the BCV in SAP TM.

For further details, please take a look into the BCV wiki or in the SAP Netweaver Knowledge Warehouse.

<https://wiki.wdf.sap.corp/wiki/display/BCV/Business+Context+Viewer+%28BCV%29>

http://aiokeh.wdf.sap.corp:50000/SAPIKS2/contentShow.sap?TMP_IWB_TASK=PREVIEW2&CLASS=BCO_COMMON&LOIO=31CB210DD15C45CAA2DA97CCD6730566&SLOIO=CF56D773309B4100BAB8709AF7C8D48C&LANGUAGE=EN&RELEASE=7015&SCLASS=XDP_STRUCT

Prerequisites

- A SAP TM 8.0 or SAP TM 8.1 System
- A user with development authorization
- The development user must be assigned to the roles:
 - o SAP_QAP_BC_SHOW
 - o SAP_QAP_BCV_ADMIN

Overview about BCV in SAP TM

General information about the BCV

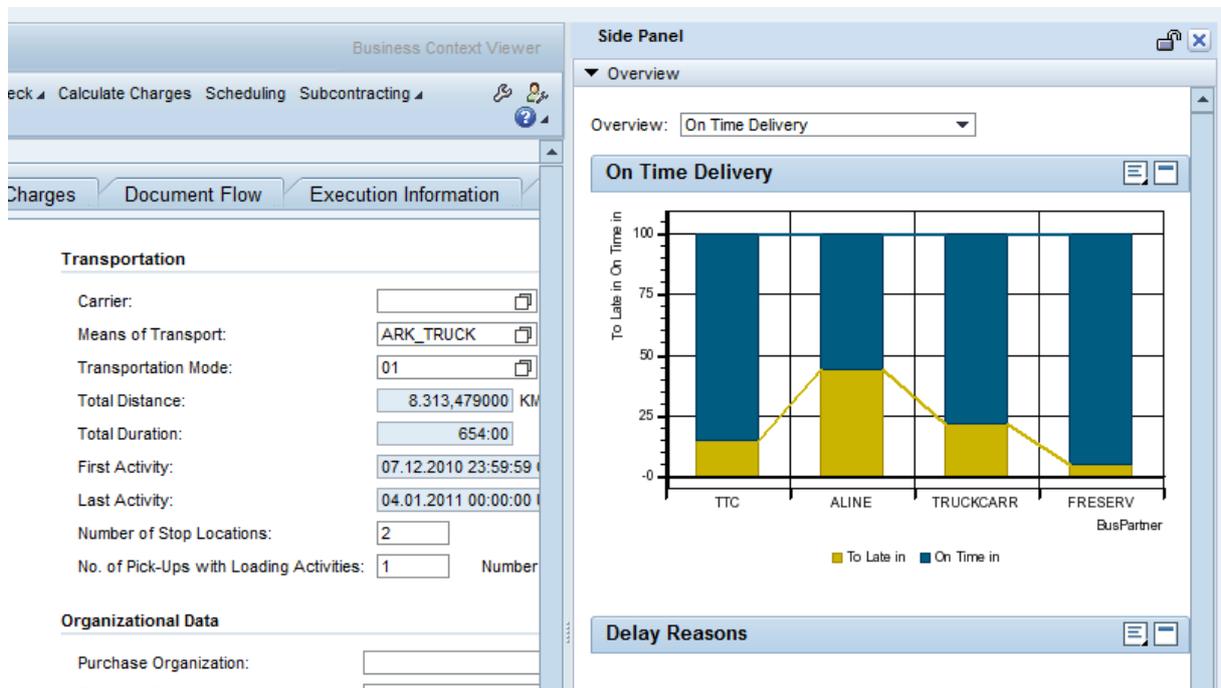
Business Context Viewer (BCV) is a framework that allows all SAP Business Suite applications to integrate different kinds of additional information into the context of their applications. This information can then be analyzed according to the user's business needs.

The User has different possibilities to get his data and bring them to the UI. The data can result from:

- ABAP Coding (Selects from DB, Retrieves from BOPF, ...)
- Business Warehouse Queries
- Infosets
- Web Services
- Workflow
- Enterprise Search

The analyzed data will be shown in a side panel in the form of:

- Tables
- Web Dynpro Business Graphics
- XCelsius Dashboards.

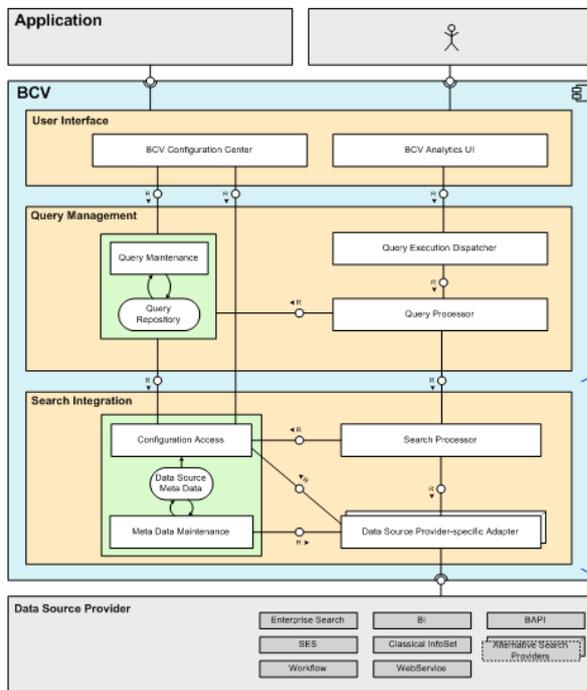


1-1 Integration of the BCV Side panel in SAP TM

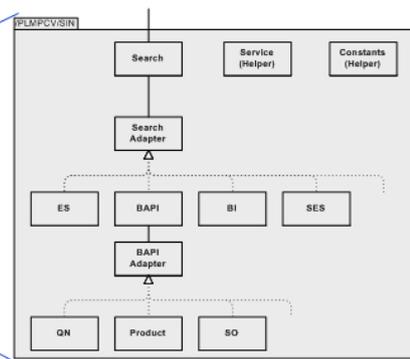
Figure 1-1 shows an Example how the BCV can be embedded in the main application.

Some technical Information about BCV

Following picture shows the resulting three main layers of BCV for data retrieval, data processing and visualization.



Extensibility by implementing a centrally defined BCV Interface thru ABAP-OO classes (Search Adapter)



The further chapters will give an overview about the layer.

Create BCV Content for SAP TM

Activate the side panel link for BCV

The first step is the activation of the BCV Side panel Link in the main application.

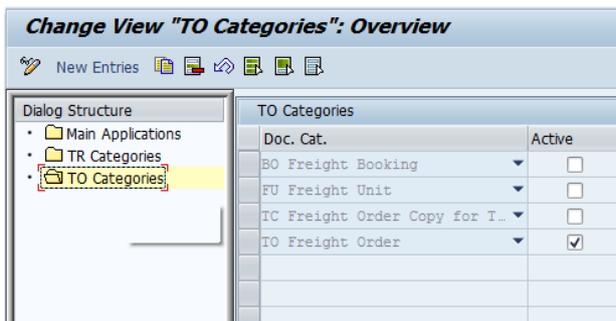
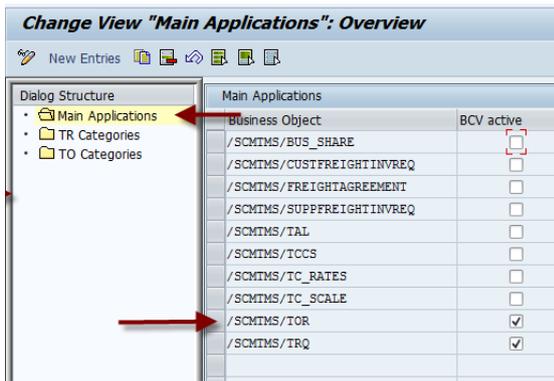
In the TM Shipment Status all coding for activation is available. The only thing the user has to do is the activation of the BCV Link for the related Business Object where BCV Content will be shown.

For activation User must open the IMG activity:

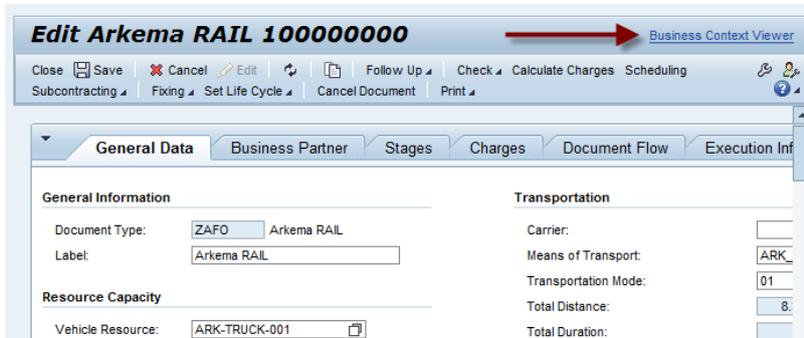
SAP Transportation Management -> Transportation Management -> Basic Functions -> User Interface -> Define Settings for Business Context Viewer. A Maintenance view will open with the available Business Objects for SAP TM.

On the left-hand side all available applications for BCV are displayed.

Double-click on 'Main Applications' and flag column 'BCV active' for the relevant business object. In case you active BCV for transportation request or transportation order, do also flag the relevant document category in 'TR categories' and 'TO categories', respectively.



After press "Save" the Side Panel is available for the Business Objects Freight Order and Forwarding Order. The user can find the Side Panel link in the upper right corner of the Freight Order.



Interested Users can find the Side Panel activation coding in the appendix.

Create a Context Key

BCV is generic tool/framework/add on, which can be used by several applications at the same time.

Therefore it becomes necessary to separate the BCV configuration of the individual applications. The context key identifies the hosting application and the processed business object (BCV for Freight Order should only display Freight Order relevant Queries).

You use this function to specify a combination of application and business object that distinguishes the configuration of BCV in your application from any other configurations that might exist in the system.

The context key identifies at least the hosting application (the application using BCV). In case the application uses BCV in several places, it is recommended to add an identifier for the business object.

It is not easy to explain the Context Key in a few sentences. For further information please take a look in the context key documentation:

http://aiokeh.wdf.sap.corp:50000/SAPIKS2/contentShow.sap?_SCLASS=XDP_STRUCT&_SLOIO=CF56D773309B4100BAB8709AF7C8D48C&TMP_IWB_TASK=PREVIEW2&RELEASE=7015&LANGUAGE=EN&SEQNUM=34&LOIO=71A16114A1D84FC59720C277C1CF2895&CLASS=BCO_COMMON

To define a context key the user must open the maintenance view “/BCV/V_CLF”. Now we create a context key for a Transportation Charge Management Analyze.

We called him /SCMTMS/TCM_MCHA (Merged Charges for Transportation Charge Management).

Classification of Applications Using BCV	
Application	SCM
Object Type	BO
Namespace	/SCM/
Context Name	/SCMTMS/TCM_MCHA
Context Description	TCM Merged Charges

As Application we use “SCM”. The Object Type is a BO (Business Object) and we work in the SCM namespace. ... context name... save

Send the Context Key to the Business Context Viewer

After creating a context key we must “send” this Key to the Side Panel. The BCV uses this Context Key to find relevant Queries which are defined with this key. The method of this key sending is called “tagging”. In our example we want to select all Charge Types in a calculated Forwarding Order and aggregate these data to get an overview which Charge Types has the greatest part of the Transportation Charges.

SAP TM uses the Bootstrap Classes to send the context key for a special BO.

In this example the name of this class is “/SCMTMS/CL_UI_BOOTSTRAP_TRQ” and the method is named “IF_FPM_GUIBB_FORM~PROCESS_EVENT”.

In the last part of this method the user can find the BCV tagging Block.

```
DATA:
lo_fpm_ovp      TYPE REF TO if_fpm_cnr_ovp.

lo_fpm_ovp ?= mo_fpm->get_service( if_fpm_constants=>gc_service_key-cnr_ovp ).
  IF NOT lo_fpm_ovp IS INITIAL.
    lo_fpm_ovp->set_tag_value(
      EXPORTING
        iv_tag = '/BCV/:CONTEXT_KEY'
        i_value = '/SCMTMS/TCM_MCHA' ).

    lo_fpm_ovp->set_tag_value(
      EXPORTING
        iv_tag = '/BCV/:1_ROOT_KEY'
        i_value = mv_key ).

    lo_fpm_ovp->set_tag_value(
```

```

EXPORTING
  iv_tag = '/BCV/:1_BO_KEY'
  i_value = /SCMTMS/IF_TRQ_C=>sc_bo_key ).

lo_fpm_ovp->set_tag_value(
  EXPORTING
    iv_tag = '/BCV/:1_ROOT_NODE'
    i_value = /SCMTMS/IF_TRQ_C=>sc_node-root ).

ENDIF.

```

We send 1 Context Key (you remember context key /SCMTMS/TCM_MCHA we created in the step before) and 3 Attributes. The BCV Query uses these queries as import parameter (see later chapter).

The relevant part is:

```

lo_fpm_ovp->set_tag_value(
  EXPORTING
    iv_tag = '/BCV/:CONTEXT_KEY'
    i_value = '/SCMTMS/TCM_MCHA' ).

```

The tag name must always be “/BCV/:CONTEXT_KEY”!!!

The tag name of the three other attributes is called “Meanings”. Meanings are explained in a further chapter. But the method to send Context Key (main identifier) and Meanings (Parameter for BCV Queries) is the same.

After set the necessary coding the BCV Configuration can follow.

Configuration of the BCV

3 steps are necessary to configure one BCV Content. The user must create:

- Search Connector
- BCV Query
- Query View

Search Connector

A search connector establishes the connection between the data provision technology and the Business Context Viewer (BCV) by providing search access to a specific data provider at a specific destination (logical system) or list of destinations.

The definition of the search connector includes the description of the selection attributes and the structure of the result data. The search connector delivers result data in a predefined form for the specified selection attributes.

The following types of search connectors are available depending on the type of data provision technology:

- SAP NetWeaver Business Intelligence
- Embedded search
- SAP NetWeaver search engine service
- InfoSet search
- Workflow

- Business Application Programming Interface (BAPI)

In the actual TCM Example we create a search connector with the type BAPI.

Search Connector Type BAPI

In the BAPI the customer can add every ABAP Coding to get data he wished to display.

To create a BAPI Search Connector the user must create a new ABAP Class. The class must implement the interface “/BCV/IF_SIN_SEARCH_BAPI”. After the interface implementation we find 4 methods:

- /BCV/IF_SIN_SEARCH_BAPI~IS_ACTIVE
- /BCV/IF_SIN_SEARCH_BAPI~GET_INSTANCE
- /BCV/IF_SIN_SEARCH_BAPI~GET_DATA_SOURCE_DETAIL
- /BCV/IF_SIN_SEARCH_BAPI~SEARCH

Additionally we create a method called “GET_CHARGES” with following parameters:

IT_ROOT_KEY	Importing	Type	/BOBF/T_FRW_KEY
IV_ROOT_NODE_ID	Importing	Type	/BOBF/CONF_KEY
IV_BO_KEY	Importing	Type	/BOBF/CONF_KEY
ET_MERGED_CHARGES	Exporting	Type	TT_OUTPUT_ATTR

The first three import parameter will be filled later by the tagging service which was explained in the last chapter.

Relevant for this scenario is the method “/BCV/IF_SIN_SEARCH_BAPI~SEARCH”.

If the user opens the Side Panel, the Side Panel checks the given context key and executes the BCV Query and the search method of the relevant search connector.

Before we start with the coding part, we must define an import and an output structure.

The import structure embeds the data given by the application via tagging. The output structure embeds the data which are relevant for the graphical output.

```
types:
  BEGIN OF tys_input_attr,
    bo_key          TYPE /bobf/conf_key,
    root_key        TYPE /bobf/conf_key,
    root_node       TYPE /bobf/conf_key,
  END OF tys_input_attr .

types:
  BEGIN OF tys_output_attr,
    key             TYPE /bobf/conf_key,
    parent_key      TYPE /bobf/conf_key,           " BO Key
    tcet084         TYPE /scmtms/trcharg_elmnt_typecd, " Chargetype
    amount          TYPE /scmtms/amount,           " Calculated Amount
    currcode016     TYPE /scmtms/currency,          " Calculated Amount Currency
    amountlcl       TYPE /scmtms/amount,           " Calculated Amount
    currcode016lcl  TYPE /scmtms/currency,          " Calculated Amount Currency
    item_desc       TYPE /scmtms/description_ul,   " Item Description
    calc_amount     TYPE /scmtms/amount,
    calc_amount_curr TYPE /scmtms/currency,
    rate_amount     TYPE /scmtms/amount,           " Rate
    rate_amount_curr TYPE /scmtms/currency,        " Rate Currency
  END OF tys_output_attr .

types:
```

```

BEGIN OF ts_merged_charges,
    chrg_it_key TYPE /bobf/conf_key,
    chrg_it_desc TYPE /scmtms/description_s,
    it_chrg_el TYPE TABLE OF /scmtms/s_tcc_trchrg_element_k WITH DEFAULT KEY,
END OF ts_merged_charges .
types:
    tt_merged_charges TYPE TABLE OF ts_merged_charges .
types:
    tt_output_attr TYPE TABLE OF tys_output_attr .

data OUTPUT_ATTR type TYS_OUTPUT_ATTR .
constants GC_CLSNAME type SEOCLSNAME value '/SCMTMS/CL_BCV_SC_CHARGES'. "#EC NOTEXT
data INPUT_ATTR type TYS_INPUT_ATTR .

```

also important is the declaration of the constant GC_CLSNAME. Here we must enter the name of the actual Search Connector class as the initial value:

```

constants GC_CLSNAME type SEOCLSNAME value '/SCMTMS/CL_BCV_SC_CHARGES'. "#EC NOTEXT

```

You see, we create an output type and an output table based on the output type.

After that we can complete the search method. In this example we called a TCM helper method (available in TM standard) with the above mentioned parameter.

```

me->get_charges(
    EXPORTING
        it_root_key      = lt_root_key
        iv_bo_key        = lv_bo_key
        iv_root_node_id  = lv_root_node_id
    IMPORTING
        et_merged_charges = lt_output
).

```

We call our create get_charges method with the import parameter root key, BO Key and the Root Node ID. In the get_charge Method we call the TCM helper method.

METHOD get_charges.

```

DATA:
    ls_ctx          TYPE /scmtms/cl_tcc_do_helper=>ts_ctx,
    lt_root_key     TYPE /bobf/t_frw_key,
    ls_root_key     LIKE LINE OF lt_root_key,

    lt_merged_charges TYPE tt_merged_charges,
    ls_merged_charges LIKE LINE OF lt_merged_charges,

    ls_merged_charge_item LIKE LINE OF ls_merged_charges-it_chrg_el,

    lt_output_table TYPE tt_output_attr,
    ls_output_table LIKE LINE OF lt_output_table.

IF it_root_key IS INITIAL.
    RETURN.
ENDIF.

" update the DO
ls_ctx-host_bo_key      = iv_bo_key.
ls_ctx-host_root_node_key = iv_root_node_id.

/scmtms/cl_tcc_do_helper=>get_merged_charges(
    EXPORTING
        is_ctx          = ls_ctx
        it_root_key     = it_root_key
    IMPORTING

```

```

        et_merged_charges_for_bcv = lt_merged_charges
    ).

    " READ TABLE lt_merged_charges INDEX 1 INTO ls_merged_charges.
LOOP AT lt_merged_charges INTO ls_merged_charges.
    LOOP AT ls_merged_charges-it_chrg_el INTO ls_merged_charge_item.
        MOVE-CORRESPONDING ls_merged_charge_item TO ls_output_table.
        APPEND ls_output_table TO lt_output_table.
    ENDLOOP.

ENDLOOP.

" delete SUM line in the output table.
LOOP AT lt_output_table INTO ls_output_table.
    IF ls_output_table-tcet084 IS INITIAL OR ls_output_table-item_desc = 'Sum'.
        DELETE lt_output_table.
    ENDIF.
ENDLOOP.

et_merged_charges = lt_output_table.

ENDMETHOD.

```

This is only an example. The search Method can embed any coding the user want to have!

The result data will be written in the “et_merged_charges” table and given back to the search method.

After finish the coding we can create the BCV Customizing.

BCV Customizing

Search Connector

In the user Menu on the SAP GUI the user can find a menu entry with the name “Business Context Viewer -> Configuration Center -> Business Context Viewer Homepage”

A worklist window will open. After selection the Search Connector Query we can create a new search connector. The following window must be open:

After entering the search connector ID we use BAPI as Data Provision Technology and press “Apply”.

.... Class name has to be provided ...

The screenshot shows the SAP configuration screen for the search connector '1TCM_MERGED_CHARGES (BAPI)'. The header includes metadata: Created By HANDRICK, Created On 22.02.2011, Changed By HANDRICK, and Changed On 23.02.2011. Below the header is a toolbar with buttons: Save, Create Query, Edit, Close, Read Only, Save As, Refresh, Check Consistency, Test, Where Used, and Delete. The main configuration area contains the following fields:

- Search Connector ID: * 1TCM_MERGED_CHARGE
- Data Provision Technology: * Application BAPI
- Description: Merg
- Data Provider
- Class Name: /SCMTMS/CL_BCV_SC_CHARGES
- Logical system: (empty)
- Additional Systems (0)

An 'Apply' button is located below the configuration fields. At the bottom, there are two tabs: 'Input Fields' and 'Output Fields'. The 'Input Fields' tab is active, displaying a table with the following data:

Field ID	Description	Data Element	Data Type	Length	Deci
BO_KEY	NodeID	/BOBF/CONF_KEY	Byte Sequence	16	
ROOT_KEY	NodeID	/BOBF/CONF_KEY	Byte Sequence	16	
ROOT_NODE	NodeID	/BOBF/CONF_KEY	Byte Sequence	16	

We see that in the tab Input Fields we found the parameter we declared in the input type structure of the search connector class. The same applies to the Output Fields.

Press the “Save Button” and close the window.

This is all. The Search Connector is finished. Now we create the BCV Query

BCV Query

Create a new BCV Query in the Business Context Viewer Homepage. The following screen opens.

Create Query View [Help](#) [Change Configura](#)

Created By Created On Changed By Changed On

Query View ID: *
 Description: Quick Info Text:
 Table ID: Chart ID:
 Enable Personalization:
 Private: Owner:

Queries

Query ID	Table View	Description

Query ID	Query Input Field ID	Query Input Field Description	Meaning	Required	Lowercase	Fixed Value
				<input type="checkbox"/>	<input type="checkbox"/>	
				<input type="checkbox"/>	<input type="checkbox"/>	
				<input type="checkbox"/>	<input type="checkbox"/>	
				<input type="checkbox"/>	<input type="checkbox"/>	

User must enter a valid Query View ID and assign the query we created before. After import the Input and Output fields press "Save". Actual an error occurs because there is no Table ID or Chart ID assigned.

Consistency check failed

Query View ID: *
 Description:
 Table ID:
 Enable Personalization:
 Private:

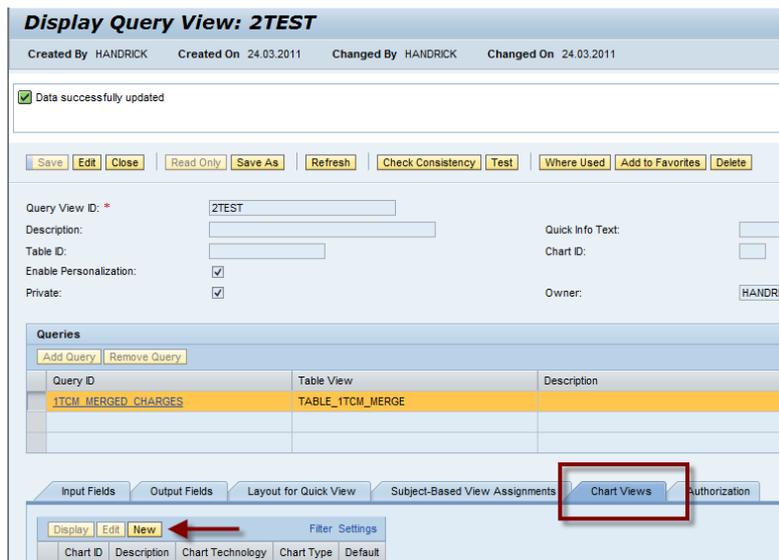
Save Query View 2TEST


 Object data is not consistent
 Do you want to save the data anyway?

Query ID	Table View

User can ignore this message and press "save".

Now you can create a Chart View



A new window will open. In this view the user can create the graphical output of the BCV Query.

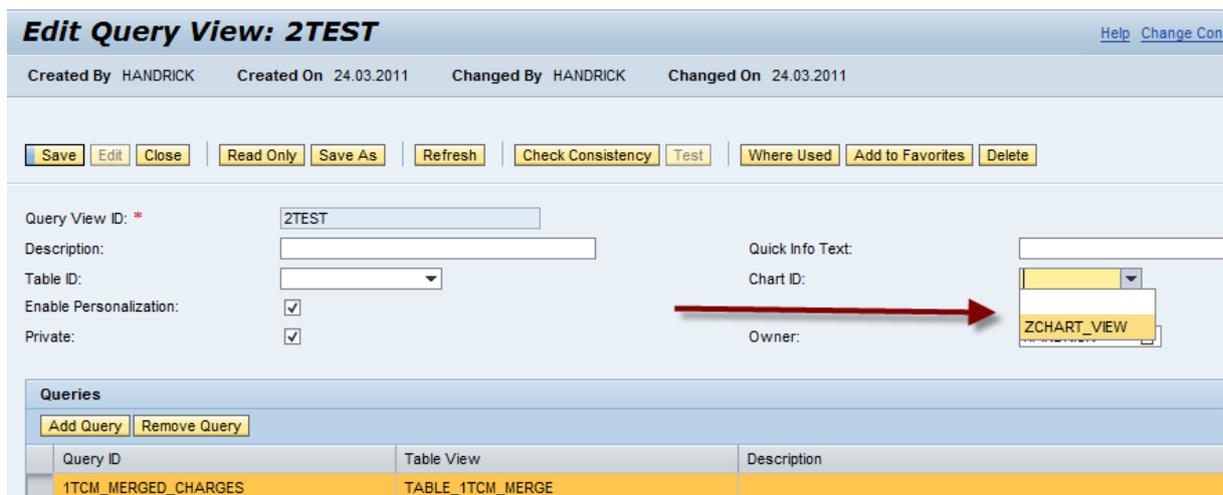
He has the choice between 3 out technologies:

- Business Graphics
- Form (simple table)
- XCelsius Dashboard

In our example we use Business Graphics. The UI will be updated. Now we must select a Chart Type and assign the category and the value to the chart (category and value fields come from the output fields of the BCV Query).

Press Save and Close the Window.

Back in the Query View Window we must assign our Chart ID we created.



Press Save and close the Window.

Last step we have to do is embed the created query view in an overview page.

Overview Page

The Overview Page is that page which will be displayed in the Side Panel. An Overview page can embed one or more Query Views. Another function of the Overview Page is the assignment to the PFCG roles which have the authorization to display the Side Panel content.

Create a new Query View in the Business Context Viewer Homepage. The following screen opens.

The screenshot shows the 'Create Overview' form. At the top, there are fields for 'Created By', 'Created On', 'Changed By', and 'Changed On'. Below these are several action buttons: 'Save', 'Define Layout', 'Edit', 'Close', 'Read Only', 'Refresh', 'Check Consistency', and 'Delete'. The main form area contains several input fields: 'Context Key' with a dropdown arrow, 'Overview ID', 'Description' with a text area, 'Configuration ID', and a 'Private' checkbox. To the right of the 'Description' field is a 'Quick Info T' label, and below the 'Private' checkbox is an 'Owner:' label. At the bottom, there is a section titled 'Roles for which this overview is default' with a 'Filter Settings' button and a table with a 'Role' column and a dropdown arrow.

After assigning the context key, set an Overview ID and the role assignment, save the object, then press button 'define layout'.

The user must define the Layout with the toolbar button "Define Layout".

The screenshot shows the 'Overview Configuration' dialog and the 'CHIP Catalog' window. The 'Overview Configuration' dialog has a toolbar with 'Save', 'Delete', and 'Reset Layout' buttons. It displays the 'Name' as '/SCMTMS/FOW1TCM_MERGED_TOR' and the 'Description' as 'Merged Charges'. Below this, there is a list of items with a width of 390 px. One item, 'Merged Charges', is selected, and a red arrow points to it with the text 'Drag and Drop'. The 'CHIP Catalog' window on the right shows a tree view of query views, with 'Merged Charges on item level' selected.

Via Drag and Drop the user can embed existing query views (queries which have the same context key you assigned to the overview page) in his overview page.

After saving the Layout the user can display the Side Panel with content in his Business Object.

The screenshot displays the SAP Business Context Viewer interface for 'Display Kiran's CSI Logistics'. The main window is divided into several sections:

- Settlement Data:** Includes fields for Invoicing Status (01 - Not Invoiced), Charge Calc. Status (03 - Calculation Error), Total Amount in Local Currency (12,628,000 USD), Total Amount in Document Currency (12,619,000 USD), Rounded Total Amount in Document Currency (12,620,000), and Rounded Amount Difference in Document Currency (1,000).
- Charge Items:** A table listing various charge types such as CSI_BASIC, CSI_TRANS, CSI_BAF, CSI_LOAD, and CSI_LOAD, along with their descriptions and group types.
- Exchange Rates:** A table showing the conversion from EUR to USD with an exchange rate of 1,0638300000000 on 01.03.2011.
- Side Panel:** Contains a 'Merged Charges' pie chart and a 'Merged Charges on item level' bar chart. The pie chart shows the distribution of charges by type, and the bar chart shows the amount in USD for various NodeIDs.

The Business Context Viewer has a lot of further function we which are not explained in this implementation guide. To get an overview about the additional features like drill downs, excel, export, snapshots, ... please have a look into the BCV Wiki:

[https://wiki.wdf.sap.corp/wiki/display/BCV/Business+Context+Viewer+%28BCV%29;jsessionid=gLwmRXou6Ei5ZZrcyXE7sLYWwi7CTtlbKAG-pl0A_SAP:saplb_*\(J2EE9282720\)9282750](https://wiki.wdf.sap.corp/wiki/display/BCV/Business+Context+Viewer+%28BCV%29;jsessionid=gLwmRXou6Ei5ZZrcyXE7sLYWwi7CTtlbKAG-pl0A_SAP:saplb_*(J2EE9282720)9282750)

Appendix

The Side Panel link

The Side Panel link will be activated in the main UI Controller Class `"/SCMTMS/CL_UI_CONTROLLER_CMN"`. For further details please have a look into method `"IWCI_IF_FPM_OVP_CONF_EXIT~OVERRIDE_EVENT_OVP"`.

The last part of this method contains the activation for all Side Panel links in the main Transportation Management BOs

```

case /bofu/if_fbi_controller~ms_object_key-bo.

" TOR
when /scmtms/if_tor_c=>sc_bo_key.
"Check if BCV Content available for the actual BO
lv_bcv_visible = check_bcv_visibility( /scmtms/if_tor_c=>sc_bo_name ).

```

```

if mv_sidepanel_link is initial and lv_bcv_visible eq 'X'.
    set_sidepanel_link( ).
endif.

" TRQ
when /scmtms/if_trq_c=>sc_bo_key.
    lv_bcv_visible = check_bcv_visibility( /scmtms/if_trq_c=>sc_bo_name ).

if mv_sidepanel_link is initial and lv_bcv_visible eq 'X'.
    set_sidepanel_link( ).
endif.

" /SCMTMS/BUS_SHARE
when /scmtms/if_bus_share_c=>sc_bo_key.
    lv_bcv_visible = check_bcv_visibility( /scmtms/if_bus_share_c=>sc_bo_name ).

if mv_sidepanel_link is initial and lv_bcv_visible eq 'X'.
    set_sidepanel_link( ).
endif.

" /SCMTMS/CUSTFREIGHTINVREQ
when /scmtms/if_custfreightinvreq_c=>sc_bo_key.
    lv_bcv_visible = check_bcv_visibility( /scmtms/if_custfreightinvreq_c=>sc_bo_name ).

if mv_sidepanel_link is initial and lv_bcv_visible eq 'X'.
    set_sidepanel_link( ).
endif.

" /SCMTMS/FREIGHTAGREEMENT
when /scmtms/if_fag_c=>sc_bo_key.
    lv_bcv_visible = check_bcv_visibility( /scmtms/if_fag_c=>sc_bo_name ).

if mv_sidepanel_link is initial and lv_bcv_visible eq 'X'.
    set_sidepanel_link( ).
endif.

" /SCMTMS/SUPPFREIGHTINVREQ
when /scmtms/if_suppfreightinvreq_c=>sc_bo_key.
    lv_bcv_visible = check_bcv_visibility( /scmtms/if_suppfreightinvreq_c=>sc_bo_name ).

if mv_sidepanel_link is initial and lv_bcv_visible eq 'X'.
    set_sidepanel_link( ).
endif.

" /SCMTMS/TAL
when /scmtms/if_tal_c=>sc_bo_key.
    lv_bcv_visible = check_bcv_visibility( /scmtms/if_tal_c=>sc_bo_name ).

if mv_sidepanel_link is initial and lv_bcv_visible eq 'X'.
    set_sidepanel_link( ).
endif.

" /SCMTMS/TCCS
when /scmtms/if_tccs_c=>sc_bo_key.
    lv_bcv_visible = check_bcv_visibility( /scmtms/if_tccs_c=>sc_bo_name ).

if mv_sidepanel_link is initial and lv_bcv_visible eq 'X'.
    set_sidepanel_link( ).
endif.

" /SCMTMS/TC_RATES
when /scmtms/if_tcrates_c=>sc_bo_key.
    lv_bcv_visible = check_bcv_visibility( /scmtms/if_tcrates_c=>sc_bo_name ).

if mv_sidepanel_link is initial and lv_bcv_visible eq 'X'.
    set_sidepanel_link( ).
endif.

```

```

" /SCMTMS/TC_SCALE
when /scmtms/if_tc_scale_c=>sc_bo_key.
  lv_bcv_visible = check_bcv_visibility( /scmtms/if_tc_scale_c=>sc_bo_name ).

  if mv_sidepanel_link is initial and lv_bcv_visible eq 'X'.
    set_sidepanel_link( ).
  endif.

when others.
endcase.

```

At the runtime the Controller Class determine the actual loaded BO and take a look into the BCV Configuration if the Side Panel is available for the BO (method `check_bcv_visibility`).

The method `set_sidepanel_link` create the link in the upper right corner.

```

method set_sidepanel_link.

data: lo_fpm_ovp      type ref to if_fpm_cnr_ovp.

  lo_fpm_ovp ?= mo_fpm->get_service( if_fpm_constants=>gc_service_key-cnr_ovp ).
  if not lo_fpm_ovp is initial.
    lo_fpm_ovp->set_side_panel_link(
      iv_text      = cl_wd_utilities=>get_otr_text_by_alias( '/SCMTMS/UI_CMN/BCV' )
      iv_tooltip   = cl_wd_utilities=>get_otr_text_by_alias( '/SCMTMS/UI_CMN/BCV_DESCR'
    )
      iv_active    = abap_true ).
  endif.
  mv_sidepanel_link = abap_true.

endmethod.

```

FAQ's

I try to open the Side Panel and a shortdump occurs.

In most cases a dump occurs when the BCV has lost the IGS connection. Please check or restart the IGS.

I open the Side Panel and my created query will not be displayed.

There exists an error with the context key. Please check the Bootstrap class if the context key is triggered correct and check the BCV Query and the Overview Page that you use the correct context key.

One of my queries gets a Timeout.

Please set a breakpoint in the Search Connector class (Search method). It looks like that a program error in the data retrieve exists. The second possibility is that the data retrieve needs a long time to get the data (more than 10 seconds). In this case you must get the data from a BW and not from a BAPI Class.