Creating BI Applications in Visual Composer

Applies to:

Visual Composer for SAP enhancement package 1 of SAP NetWeaver Composition Environment 7.1

Summary

This document explains how to create BI applications using Visual Composer and how to use Visual Composer's BI capabilities.

Company: SAP

Created on: 22 November 2009

Authors

Netanel Slomianski is a Visual Composer solution expert working for the Visual Composer Solution Office group.

Talia Gurevich is a Senior Software Developer working for the Visual Composer Development Support group.

Table of Contents

Purpose

This document describes the BI capabilities in SAP NetWeaver Visual Composer and explains how to use them.

BI backend systems that can be consumed in Visual Composer

Visual Composer can connect to an SAP BI system using its RFC connector, and to any JDBC system using its BI JDBC connector. For modeling with RFC-based BI queries, Visual Composer can connect to and consume only those BI data sources on SAP NetWeaver 7.0.

Accessing SAP BI systems

Visual Composer uses the portal system landscape as its repository for BI systems. Therefore, you should configure your BI system as a portal system in the portal. For more information, use this article.

Searching and using BI data services

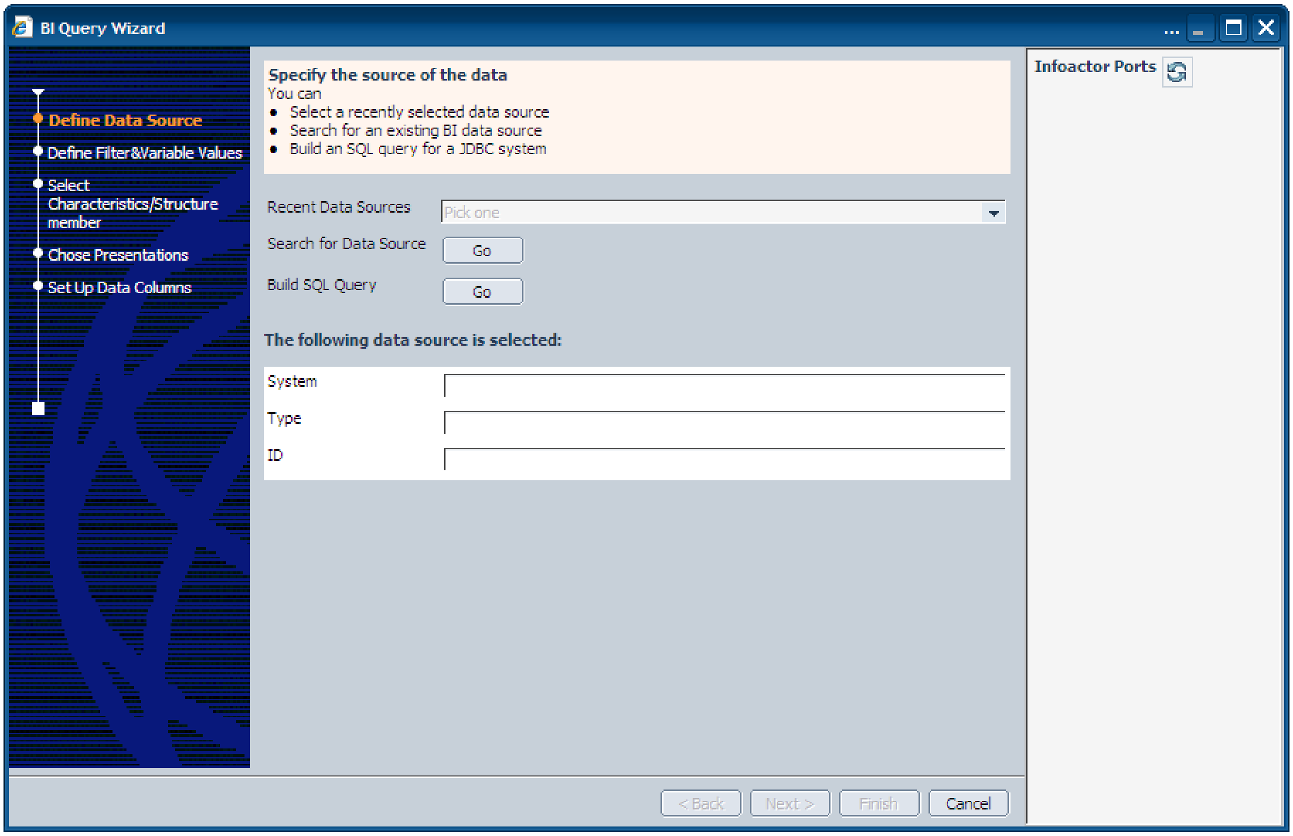
Once you have configured your BI system as a portal system, you can use its data in Visual Composer Storyboard in two ways:

- By selecting a BI system, searching for a certain BI Query / View / InfoProvider / Characteristic / Filter in the Search task panel and dragging it to the model
- By choosing *BI Services* → *Query Wizard* from the main menu. This launches a wizard in which you select a certain system, select a BI Query / View / InfoProvider / Characteristic / Filter and then customize it.

Working with the BI Query Wizard

The BI Query Wizard is a powerful tool that enables you to select an existing BI query and customize the data it returns according to your needs. Here is a detailed description of each step in the wizard:

1. Choose *BI Services→Query Wizard* to open the BI Query Wizard.



Specify the source of the data

You can

- Select a recently selected data source
- Search for an existing BI data source
- Build an SQL query for a JDBC system

Recent Data Sources Pick one

The following data source is selected:

System

Type

ID

2. The first step is to define the data source of the query you want to use. You can do one of two things:

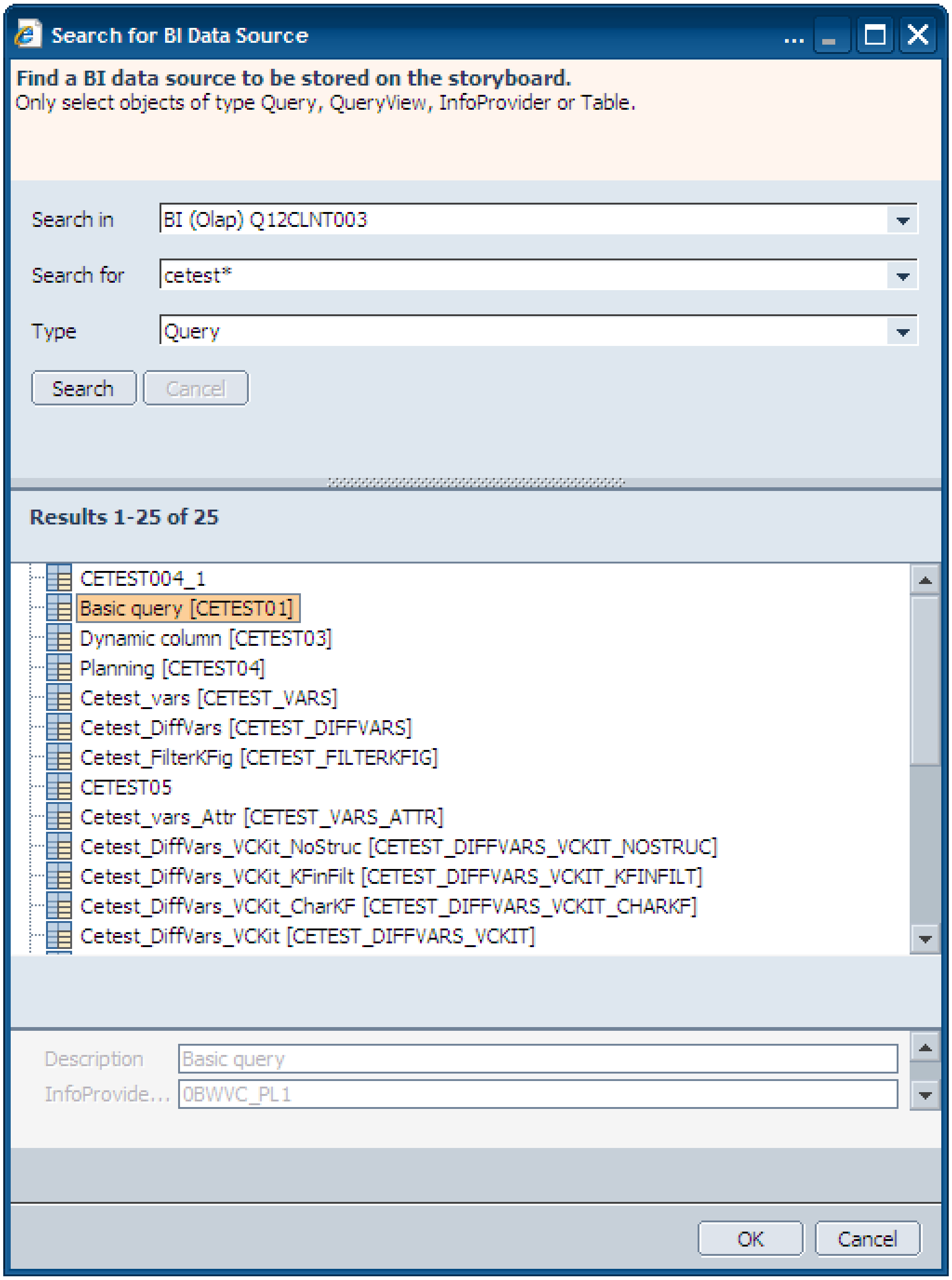
   - Search for a predefined BI query in your OLAP system or select a table from your database:

     i. Click the *Go* button next to *Search for Data Source*.

     ii. In the *Search for BI Data Source* dialog box, select your BI data source, the name of the data source or component you want to use (* wildcard is acceptable) and the type of the data source (*BI Query / View / InfoProvider / Characteristic / Filter* for OLAP systems and *Table* in case it's a BI JDBC system).

     iii. Click *Search* to start searching for the data source.

i. Once you have located your data source, select it and click *OK.*



Search for BI Data Source

Find a BI data source to be stored on the storyboard.
Only select objects of type Query, QueryView, InfoProvider or Table.

Search in: BI (Olap) Q12CLNT003

Search for: cetest*

Type: Query

Results 1-25 of 25

- CETEST004_1
- Basic query [CETEST01]
- Dynamic column [CETEST03]
- Planning [CETEST04]
- Cetest_vars [CETEST_VARS]
- Cetest_DiffVars [CETEST_DIFFVARS]
- Cetest_FilterKFig [CETEST_FILTERKFIG]
- CETEST05
- Cetest_vars_Attr [CETEST_VARS_ATTR]
- Cetest_DiffVars_VCKit_NoStruc [CETEST_DIFFVARS_VCKIT_NOSTRUC]
- Cetest_DiffVars_VCKit_KFinFilt [CETEST_DIFFVARS_VCKIT_KFINFILT]
- Cetest_DiffVars_VCKit_CharKF [CETEST_DIFFVARS_VCKIT_CHARKF]
- Cetest_DiffVars_VCKit [CETEST_DIFFVARS_VCKIT]

Description: Basic query

InfoProvide...: 0BWVC_PL1

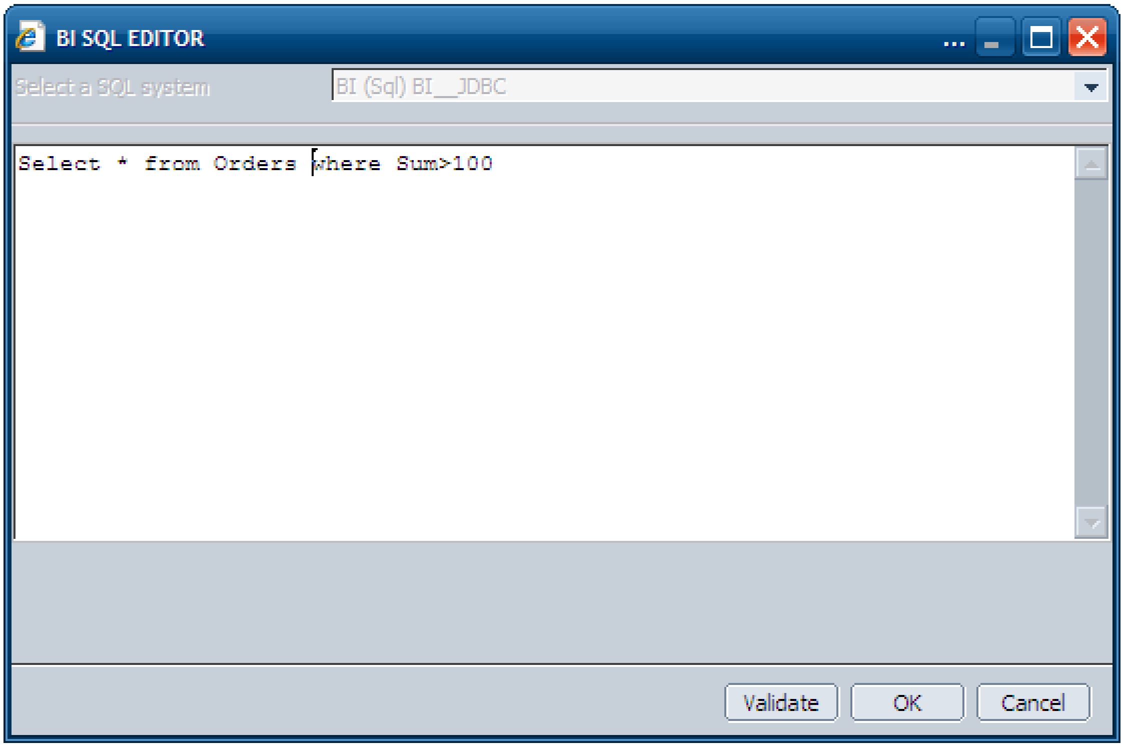
- Write your own SQL statement that will be executed on your database.

  i. Click the *Go* button next to *Build SQL Query*.

  ii. Select an SQL system (BI JDBC) system.

  iii. In the *BI SQL EDITOR* dialog box, enter your SQL statement. You can validate the statement by clicking *Validate*.

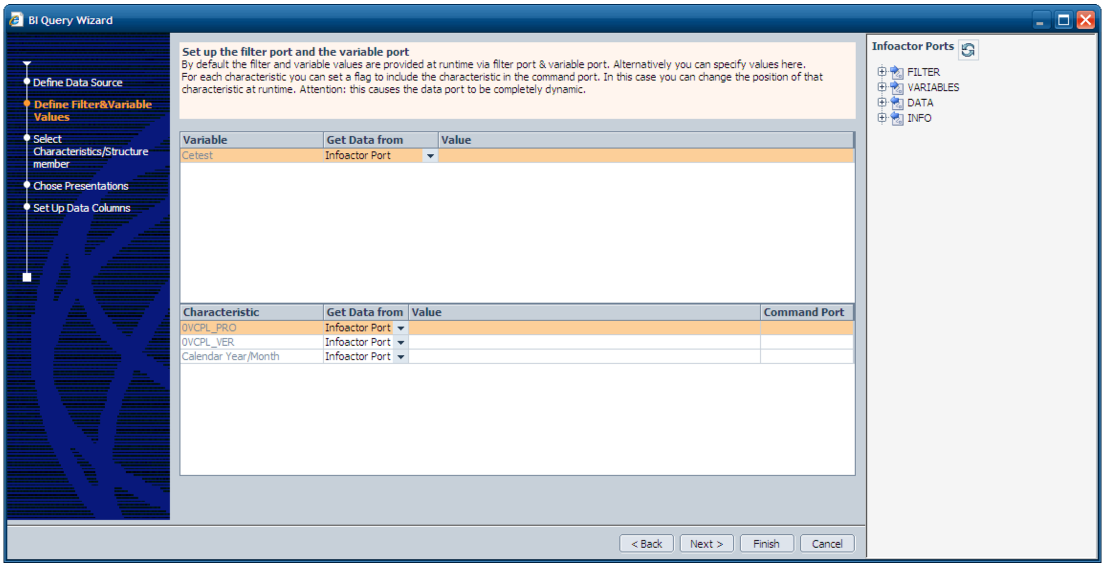
iv. Click *OK* to save the SQL statement.



```
Select * from Orders where Sum>100
```

3. In the wizard, click *Next* to define the filter and variable values.

4. In this step, you can set up the FILTER port and the VARIABLE port. By default, the Filter and Variable values are provided at runtime via the FILTER port and VARIABLE port. Alternatively, you can specify values here. For each characteristic, you can set a flag to include the characteristic in the command port. By doing this, you can change the position of that characteristic at runtime. Note that this causes the DATA port to be fully dynamic.



Set up the filter port and the variable port

By default the filter and variable values are provided at runtime via filter port & variable port. Alternatively you can specify values here.
For each characteristic you can set a flag to include the characteristic in the command port. In this case you can change the position of that characteristic at runtime. Attention: this causes the data port to be completely dynamic.

Variable	Get Data from	Value
Cetest	Infoactor Port	

Characteristic	Get Data from	Value	Command Port
0VCPL_PRO	Infoactor Port		
0VCPL_VER	Infoactor Port		
Calendar Year/Month	Infoactor Port		

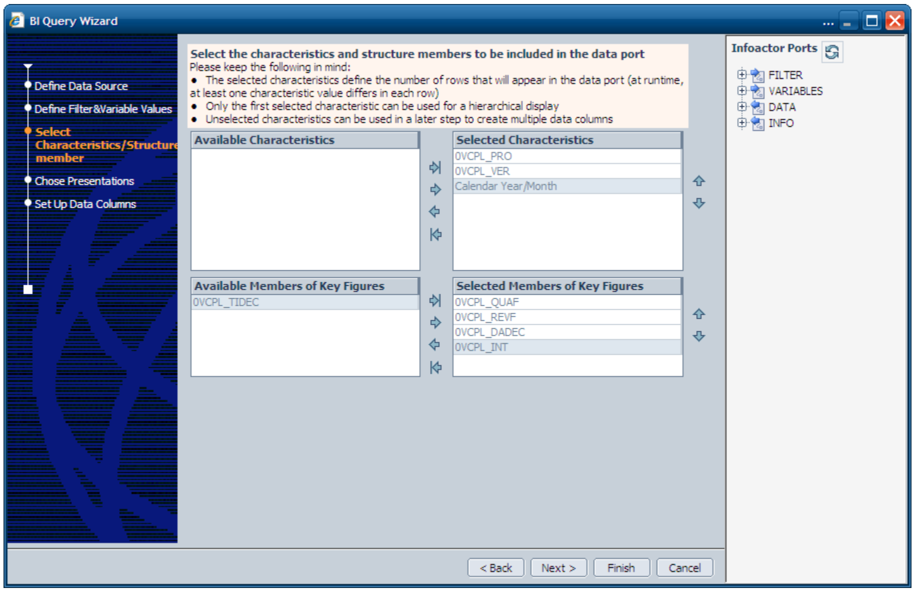
5. Click *Next* to select the characteristics / structure members of the query.

6. In this step, you can select the characteristics and structure members to be included in the DATA port of the query object. You should keep the following in mind:

    - The selected characteristics define the number of rows that will appear in the data port. (At runtime, at least one characteristic value differs in each row.)

    - Only the first selected characteristic can be used for a hierarchical display.

    - Unselected characteristics can be used in a later step to create multiple data columns.



Select the characteristics and structure members to be included in the data port
Please keep the following in mind:
- The selected characteristics define the number of rows that will appear in the data port (at runtime, at least one characteristic value differs in each row)
- Only the first selected characteristic can be used for a hierarchical display
- Unselected characteristics can be used in a later step to create multiple data columns

Available Characteristics	Selected Characteristics
	0VCPL_PRO
	0VCPL_VER
	Calendar Year/Month

Available Members of Key Figures	Selected Members of Key Figures
0VCPL_TIDEC	0VCPL_QUAF
	0VCPL_REVF
	0VCPL_DADEC
	0VCPL_INT

7. Click *Finish* to complete the wizard.

Structure of BI data services in Visual Composer

A BI data service can have up to two input ports and two output ports. Using the FILTER and VARIABLE input ports, you can set the input parameters for the BI query or query view. The results for the BI object are provided in the DATA output port. The INFO output port provides general parameters for the BI object.

- **FILTER input port**
  The FILTER input port contains exactly one field for each characteristic in the rows, columns, or free characteristics of the BI object. Each filter is a field of type String and can contain a filter condition for which you can select a single value, multiple values, intervals, and free selection options:
    - *Single value*

      Enter the value; for example **PROD1**.
    - *Multiple single values*

      Enter the values separated by semicolons; for example **PROD1; PROD2; PROD4**.
    - *Intervals*

      Enter the lower and upper values separated by a minus sign (-); for example, **PROD1-PROD3**.
    - *Free selection options*

      You can enter a combination of single values and intervals; for example, **PROD1; PROD3; PROD5-PROD8.**

  Note
  The values in the filter conditions string must be the display type key and not the display type text.

- **VARIABLE input port**
  The VARIABLE input port contains exactly one field for each input-enabled variable of the BI object. You can specify whether the variables contain one single value, multiple single values, an interval, or a complete selection option. Each variable field is a String type field and can contain selections with the same syntax as that used for filters. The values in the string also have to be the display type key.

  If the BI object does not contain any input-enabled variables, the VARIABLE input port is not generated.

- **DATA output port**
  The DATA output port is generated as follows:
    - A field is generated for each display type (key, text) of a characteristic in rows or columns; for example, *Product_Text* and *Product_Display_Key*.
    - The display for characteristic attributes is the same as for characteristics.
    - If the BI object contains one key figure structure, three fields are generated for each structure member:
        - Field of type String for formatted values; for example, *Revenue_FMTVALUE*
        - Field of type String for units; for example, *Revenue_UNIT*
        - Numeric field for actual values; for example, *Revenue_VALUE*

      Note
      The field name is generated from the technical name of the structure element in the Query Designer (in the example 'Revenue'). If you do not specify a technical name for the structure element in Query Designer, the GUID is used for the field name.

- If the BI object contains a key figure structure and another structure, these three fields are generated for each member combination of the two structures.

- **INFO output port**
  The INFO output port provides general parameters for the BI object, such as the name of the query, the data, and information about the last change made.

Creating a simple BI application

In this section, we will create a simple BI application that uses a BI data source, an input form and an output table. Make sure that the BI system is configured in the portal system landscape.

1. Create a new Visual Composer composite view.
2. Click the Search button in the task panel toolbar. Select the BI system from the *Search In* dropdown box.
3. Select *Query* from the *Type* dropdown box.
4. Enter the name of the query in the *Search for* field.
5. Click *Search.* You can use the * wildcard.
6. Select the query and drag it onto the Design board.
7. If you have variables or filter defined in your query, drag a form out of the corresponding in-ports.
8. Drag a table/form out of the DATA out port.
9. Deploy the application.

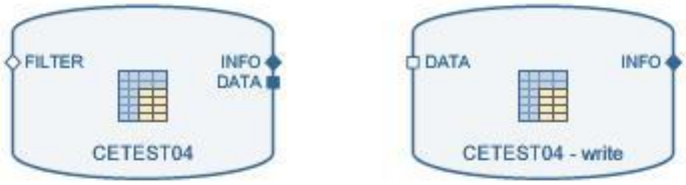
Creating a simple BI Planning application

This section describes how to design a composite view to be used for planning.

In order to use the planning functionality of BI Integrated Planning, you need to select an input-ready query as the data source for your model. You can define and configure the queries in BEx Query Designer. You can also include planning functions and planning sequences created in the BI system.

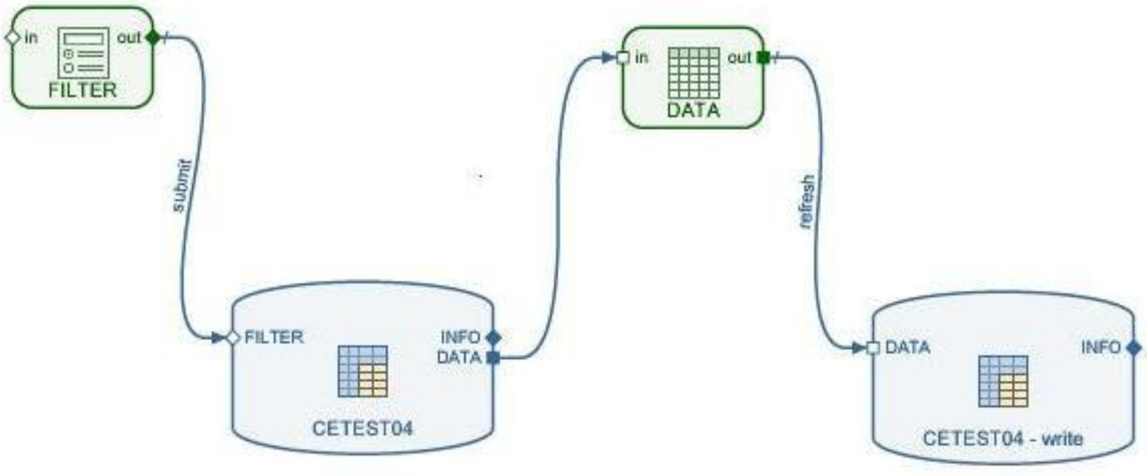
1. Create a new Visual Composer composite view.
2. Select an input-ready query as the data source and add it to your model. For more information, see Searching and using BI data services

   Note
   When you add an input-ready query to the Design board, two data service icons are created. One is responsible for reading the data from the BI system. The other is responsible for writing back the plan data. To distinguish between the two services, the suffix *_write* is added to the name of the second data service.

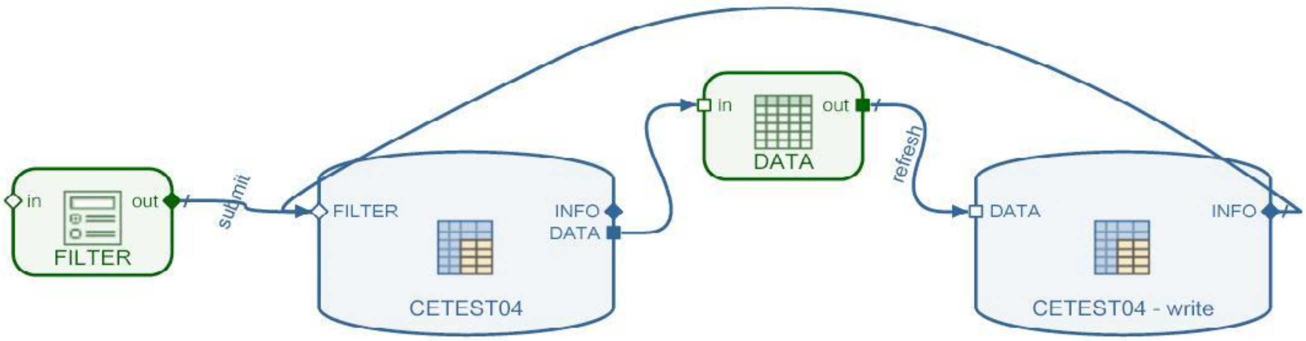


3. Define the data flow.

- The two data service icons need to be connected with a table view. To add a table view to your model, drag a line from the DATA output port of the data service for reading and choose *Table View* from the context menu. In the *Define Data* dialog box, check the fields that you want to be displayed in the table. Connect the OUT output port of the table view with the DATA input port of the data service for writing back.
- Add an action – such as via a button (you can add a button in layout) – that triggers the execution of the writing service. The FILTER and VARIABLE input ports are typically connected to a form view, which enables you to enter values for filters and variables.



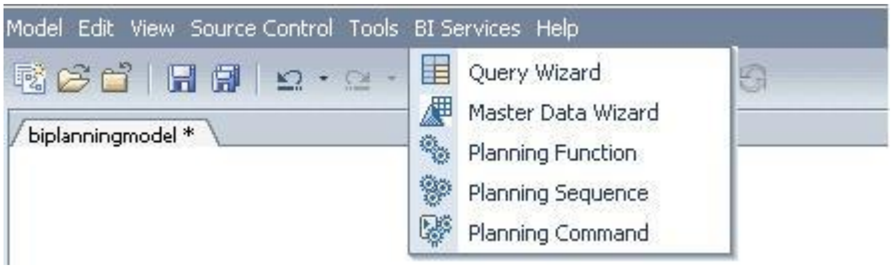
- If the query needs to be refreshed after writing back the data (for example, because a calculated key figure has to be recalculated), connect the INFO output port with one of the input ports of the reading data service.



4. Deploy the application.

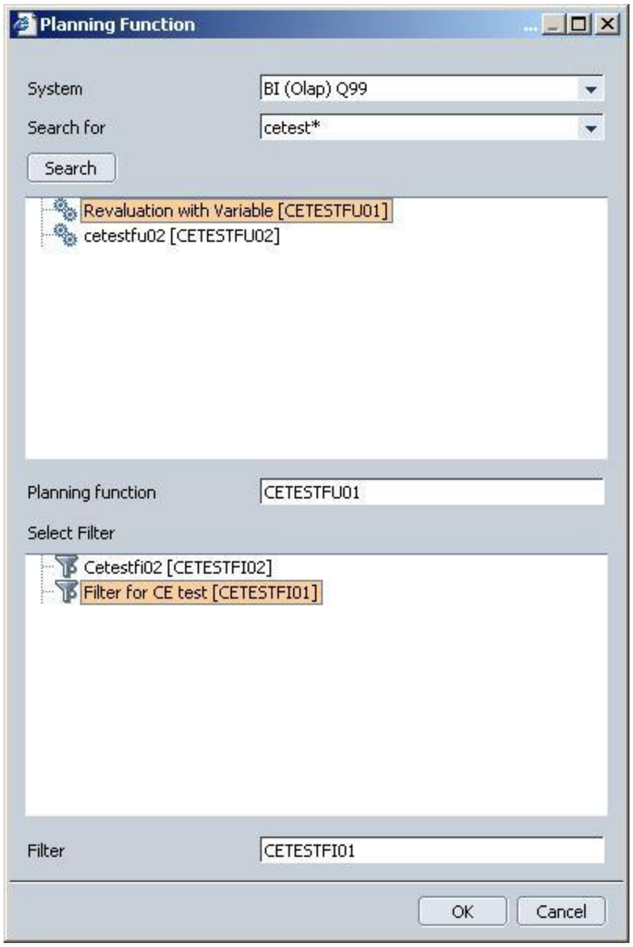
BI Planning data services

This section describes the BI Planning dialog boxes displayed when you choose a menu item from the *BI Services* menu in the Visual Composer main menu bar.



- **Planning Function Dialog Box**

The *Planning Function* dialog box is intended to select a planning function and a filter created in the SAP NetWeaver BI Integrated Planning and to add it to your Visual Composer model.



System: BI (Olap) Q99

Search for: cetest*

Revaluation with Variable [CETESTFU01]
cetestfu02 [CETESTFU02]

Planning function: CETESTFU01

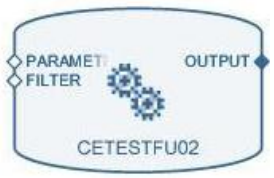
Select Filter

Cetestfi02 [CETESTFI02]
Filter for CE test [CETESTFI01]

Filter: CETESTFI01

The *Planning Function* dialog box contains the following items:

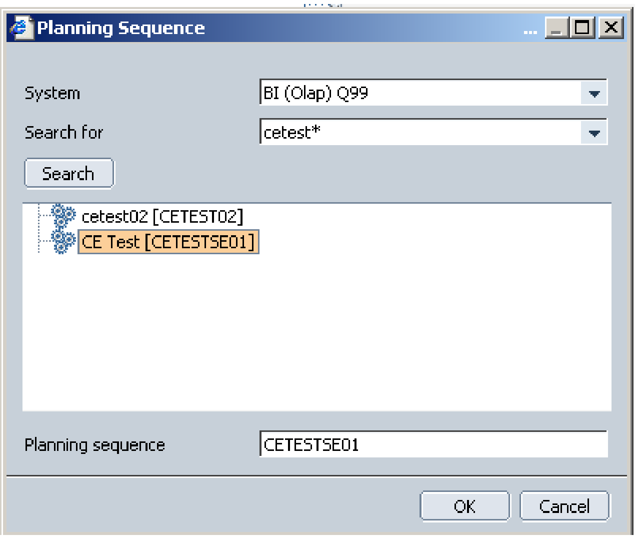
Item	**Description**
System	Select an SAP NetWeaver BI system.
Search for	Enter the name of a planning function you want to locate. You can use an asterisk (*) in the string as a wildcard to search for a partial match.
Planning function	Displays the planning function that you selected in the result list.
Select Filter	Lists the available filters.
Filter	Displays the filter that you selected in the filter list.



Planning Function icon on Visual Composer Design board

- **Planning Sequence Dialog Box**

The *Planning Sequence* dialog box is intended to select a planning sequence created in the SAP NetWeaver BI Integrated Planning and add it to your Visual Composer model.



System: BI (Olap) Q99

Search for: cetest*

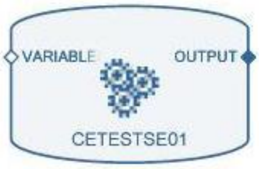
cetest02 [CETEST02]

CE Test [CETESTSE01]

Planning sequence: CETESTSE01

The *Planning Sequence* dialog box contains the following items:

Item	Description
System	Select an SAP NetWeaver BI system.
Search for	Enter the name of a planning sequence you want to locate. You can use an asterisk (*) in the string as a wildcard to search for a partial match.
Planning Sequence	Displays the planning sequence that you selected in the result list.

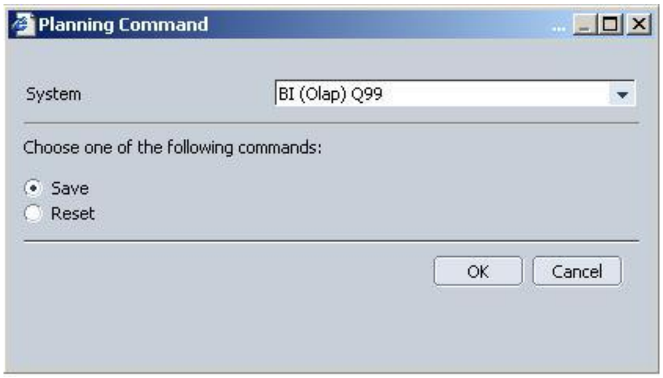


Planning Sequence icon on Visual Composer Design board

- **Planning Command Dialog Box**

The *Planning Command* dialog box is intended to select the planning command *Save* and *Reset* from SAP NetWeaver BI Integrated Planning and add it to your Visual Composer model. A button with the command will be displayed at runtime.

If you change data via the write back service or planning functions and sequences, only the data in the memory is changed. The changes can be persisted to the database with the *Save* command. The *Reset* command undoes all changes and releases the locks. This command is only needed in exceptional cases, as the locks are released anyway when leaving an application.



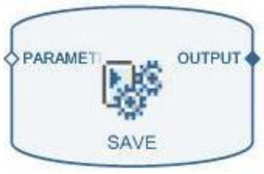
System BI (Olap) Q99

Choose one of the following commands:

● Save
○ Reset

The *Planning Command d*ialog box contains the following items:

Item	Description
System	Select an SAP NetWeaver BI system
Commands	Select Save or Reset.



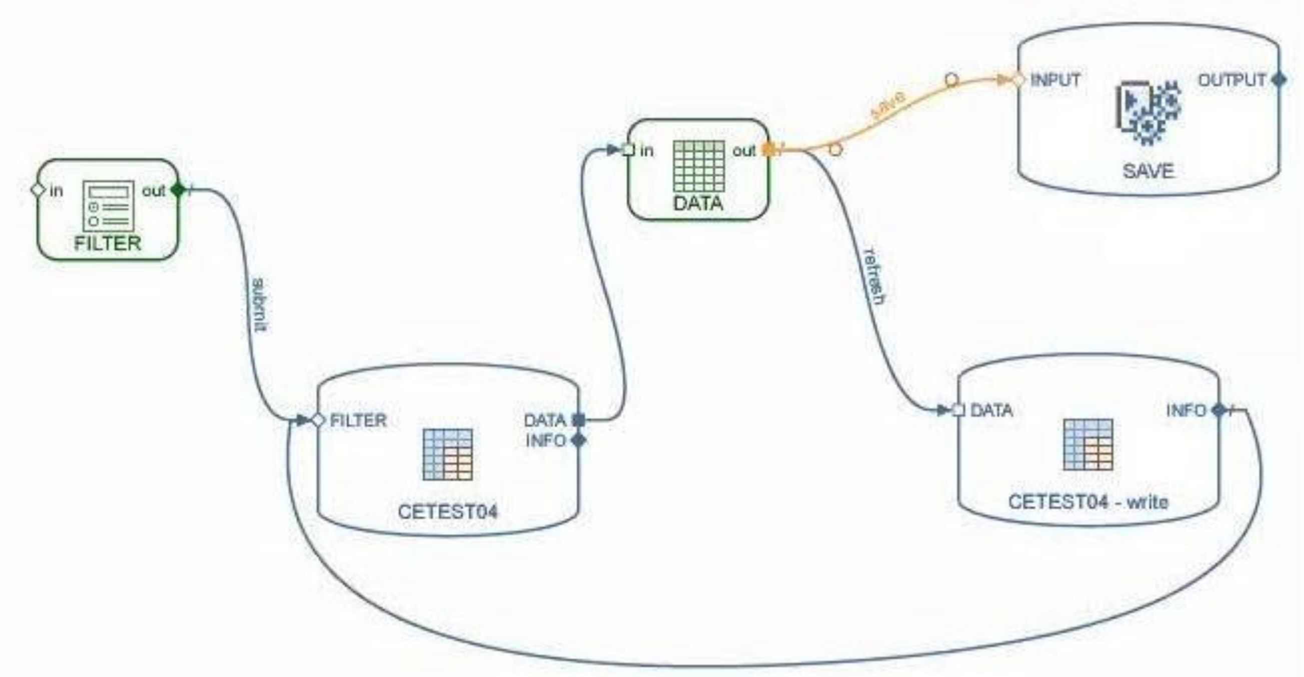
Planning Command icon on Visual Composer Design board

Add data services for planning

You can add planning functions, planning sequences and planning commands using the *BI Services* option in the Visual Composer main menu. A data service can be executed via an action, for example, triggered by a button.

Example 1

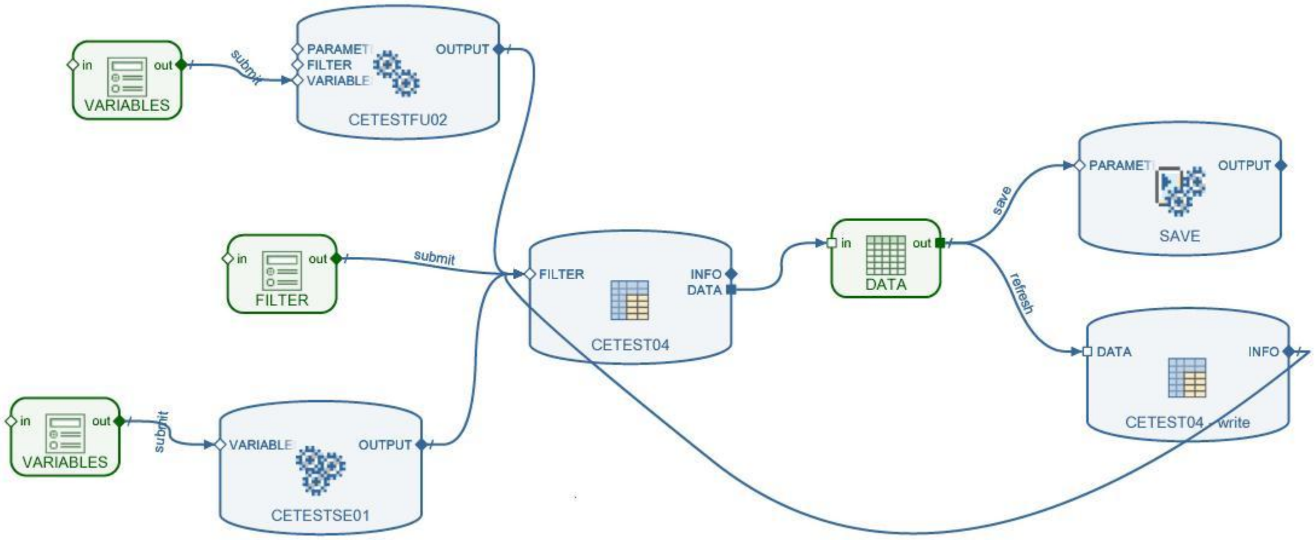
- Add a *Save* action to the table (using a button, for example).
- Add the *Save Planning* command to your model
- Drag a line from the OUT output port of the table view and connect it with PARAMETERS input port of the *Save Planning Command* service.
- Configure the line to trigger the *Save* event that you defined on the table.
- Deploy and run the application.
- You can use the *Save* button of the table in order to save the plan data.



Example 2

In this example, you can see the use of *Planning Function* and *Planning Sequence* services.

Typically, a query within the same model will need to be refreshed in order to show the results of the planning function or sequence. Therefore, connect the OUTPUT out port of the planning function/sequence with one of the input ports of the reading data service - in a way similar to what you did to refresh the query after writing back data.



Related Content

[Visual Composer for SAP NetWeaver CE 7.1]

[SAP NetWeaver Visual Composer FAQ]

[Modeling Composite Views Using BI Data Services]

[BI Data Services]

[Configuring Visual Composer to Consume Data Sources]

Copyright

These materials are subject to change without notice. These materials are provided by SAP AG and its affiliated companies ("SAP Group") for informational purposes only, without representation or warranty of any kind, and SAP Group shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP Group products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.

These materials are provided “as is” without a warranty of any kind, either express or implied, including but not limited to, the implied warranties of merchantability, fitness for a particular purpose, or non-infringement.

SAP shall not be liable for damages of any kind including without limitation direct, special, indirect, or consequential damages that may result from the use of these materials.

SAP does not warrant the accuracy or completeness of the information, text, graphics, links or other items contained within these materials. SAP has no control over the information that you may access through the use of hot links contained in these materials and does not endorse your use of third party web pages nor provide any warranty whatsoever relating to third party web pages.

Any software coding and/or code lines/strings (“Code”) included in this documentation are only examples and are not intended to be used in a productive system environment. The Code is only intended better explain and visualize the syntax and phrasing rules of certain coding. SAP does not warrant the correctness and completeness of the Code given herein, and SAP shall not be liable for errors or damages caused by the usage of the Code, except if such damages were caused by SAP intentionally or grossly negligent.