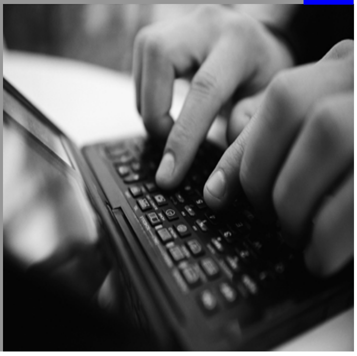
How-to Guide
SAP NetWeaver 2004s



How To... use BPS and BI Integrated Planning in parallel – Data slices and Characteristic Relations

Version 1.00 – February 2006

Applicable Releases:
SAP NetWeaver 2004s

Data contained in this document serves informational purposes only. National product specifications may vary.

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.
SAP NetWeaver “How-to” Guides are intended to simplify the product implementation. While specific product features and procedures typically are explained in a practical business context, it is not implied that those features and procedures are the only approach in solving a specific business problem using SAP NetWeaver. Should you wish to receive additional information, clarification or support, please refer to SAP Consulting.
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.

1 Scenario

You are planning to use BPS and Integrated Planning in parallel. When these two applications access or manipulate the same set of data (e.g. use the same real time InfoCube) it is strongly recommended to define the same rules for validation (characteristic relationships) and data protection (data slices) in both applications.

When changing rules in one application the corresponding rules in the other application has to be adjusted as well. This document describes how this adjustment could be done automatically without the necessity of several manual steps but just by executing an ABAP program.

Please note that the scope is running the applications in parallel keeping the rules consistent. The migration of rules from BPS to BI Integrated Planning is out of scope.

2 Introduction

The solution described assumes that the data slices and characteristic relations are defined in BPS properly. By implementing exits located in Integrated Planning these definitions are exposed in the context of BI Integrated Planning.

In order achieve a high degree of compatibility the exits call the BPS framework directly. This means that in the end the same source code is used for data slices checks and characteristic relationships as in BPS.

Known Differences:
The derivation and combination check of standard time characteristics is done automatically in BI Integrated Planning and hence implicitly added to the rules defined in BPS.

Limitations:
The exit implementations rely on a generic mapping from BI Integrated Planning to BPS parameters for calling the BPS framework. Problems might occur when e.g. exit implementations on BPS side use runtime information of BPS that is not available.

The exit implementations are delivered in order to simplify the implementation of this How to Guide and not as a standard solution with full support of SAP.

3 The Step By Step Solution

The following chapters show how to set up the data slice and characteristic relations customizing in BI Integrated Planning. This has to be done once per planning area.

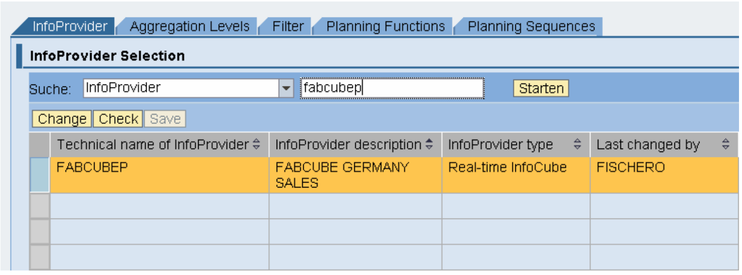
3.1 Data Slices

1. Determine the name of the InfoProvider used in the Planning Area.

   Example: *FABCUBEP*

   Start the planning workbench (Transaction BPS0) and select the planning area. The name of the InfoProvider is located on the first tab strip.

2. Start the planning modeler (e.g. by using the transaction RSPLAN) and select the InfoProvider *FABCUBEP*.

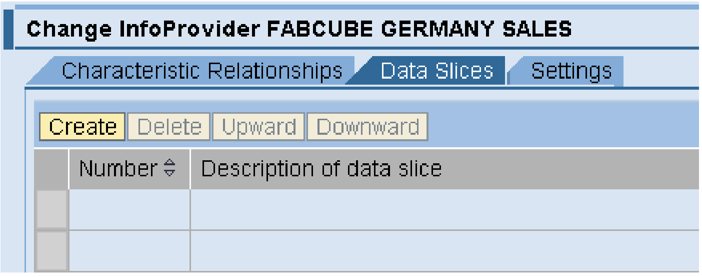


InfoProvider Selection

Suche: InfoProvider fabcubep

Technical name of InfoProvider	InfoProvider description	InfoProvider type	Last changed by
FABCUBEP	FABCUBE GERMANY SALES	Real-time InfoCube	FISCHERO

3. Switch to change mode and select the tab strip for data slices



Change InfoProvider FABCUBE GERMANY SALES

Number	Description of data slice

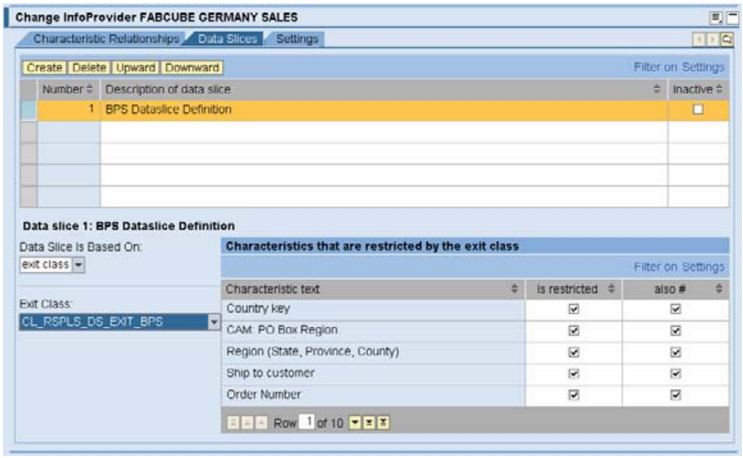
4. If other data slices exist already they have to be deleted (Exception: data slice for STS access).

5. Create a new data slice with the properties:

   Data Slice is Based on: *Exit*

   Exit Class : *CL_RSPLS_DS_EXIT_BPS*

   Please note that only one data slice is needed even if there are several ones defined in BPS.



Change InfoProvider FABCUBE GERMANY SALES

Number	Description of data slice	Inactive
1	BPS Dataslice Definition	□

Data slice 1: BPS Dataslice Definition

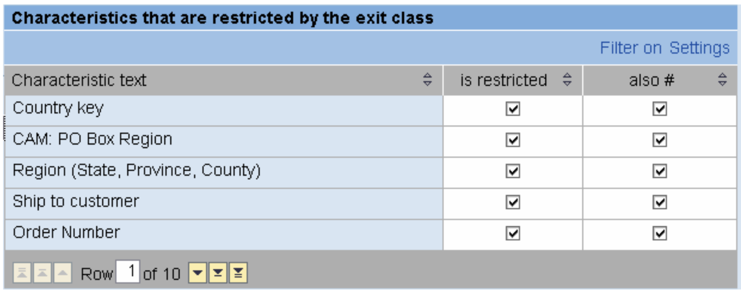
Data Slice Is Based On: exit class

Exit Class: CL_RSPLS_DS_EXIT_BPS

Characteristics that are restricted by the exit class

Characteristic text	is restricted	also #
Country key	☑	☑
CAM: PO Box Region	☑	☑
Region (State, Province, County)	☑	☑
Ship to customer	☑	☑
Order Number	☑	☑

6. Confirm that the check boxes for ‘is restricted’ and ‘also #’ are ticked for all characteristics.



Characteristics that are restricted by the exit class

Filter on Settings

Characteristic text	is restricted	also #
Country key	☑	☑
CAM: PO Box Region	☑	☑
Region (State, Province, County)	☑	☑
Ship to customer	☑	☑
Order Number	☑	☑

Row 1 of 10

7. Save the data slices

3.2 Characteristic Relations

For each characteristic relation defined in BPS a corresponding relation has to be defined in BI Integrated Planning. These definitions are done by the program listed in the appendix of this document. Just execute the report per planning area and tick the check box ‘update’.

Attention: the report has to be re-executed as soon as the number or order of steps have been changed. Examples for such changes are

- adding or deleting of steps or
- changing the order.

In order to understand what the program does and to verify the results, the following description explains how to do it manually for one relation. After the execution of the program all settings described below should be available in the planning modeler.

1. Determine the name of the InfoProvider used in the Planning Area.

   Example: *FABCUBEP*

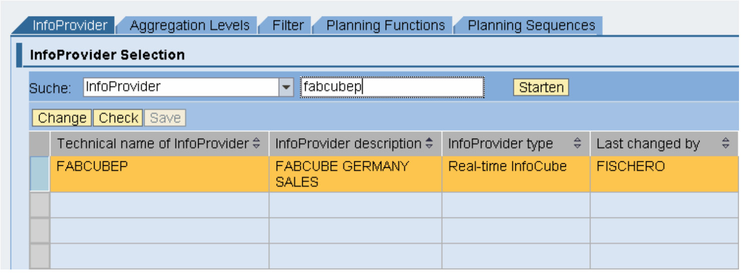
2. From the planning area chosen, get the following information:
   - source characteristics
   - target characteristics
   - Option: with or without derivation

To facilitated this process please implement and run the program Z_GET_SET_INFO_CHAR_RELATION.

It reads the BPS customizing and writes the required information to a simple list.

3. In the following steps we create a corresponding characteristic relation:

4. Start the planning modeler and select the InfoProvider FABCUBEP.

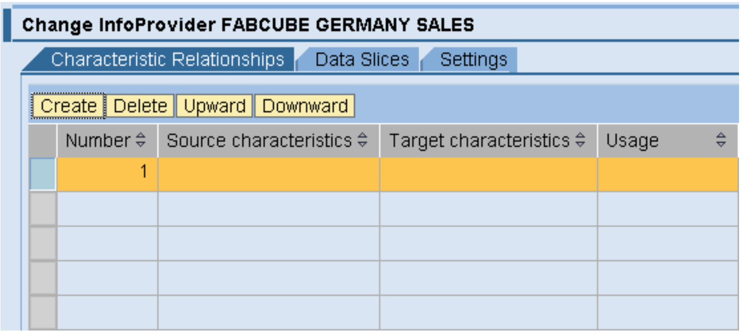


InfoProvider Selection

Suche: InfoProvider fabcubep

Technical name of InfoProvider	InfoProvider description	InfoProvider type	Last changed by
FABCUBEP	FABCUBE GERMANY SALES	Real-time InfoCube	FISCHERO

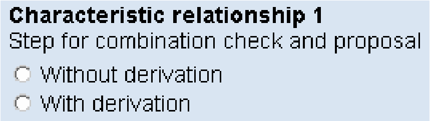
5. Switch to change mode and select the tab strip for characteristic relationships.



Change InfoProvider FABCUBE GERMANY SALES

Number	Source characteristics	Target characteristics	Usage
1			

6. Press ‘Create’. Below the table the information shown on the right appears. Make the same choice as in BPS. In our example: With derivation.



Characteristic relationship 1

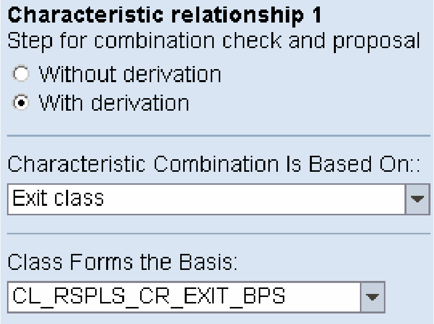
Step for combination check and proposal

○ Without derivation

○ With derivation

7. The characteristic combination could be based on master data attributes, hierarchy, exit class or data store. Select the exit class.

   Select the name of the class as CL_RSPLS_CR_EXIT_BPS.



Characteristic relationship 1

Step for combination check and proposal

○ Without derivation

● With derivation

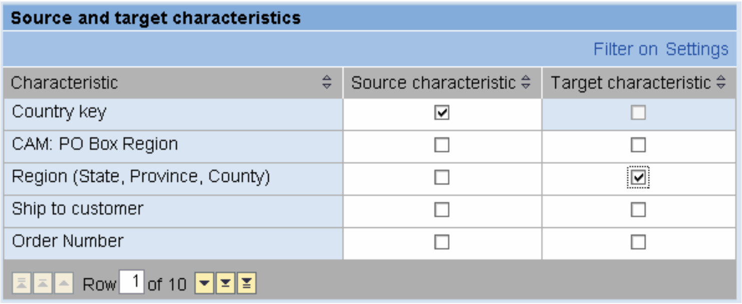
Characteristic Combination Is Based On::

Exit class

Class Forms the Basis:

CL_RSPLS_CR_EXIT_BPS

8. Select the same source and target characteristics as in BPS.



Source and target characteristics

Filter on Settings

Characteristic	Source characteristic	Target characteristic
Country key	☑	□
CAM: PO Box Region	□	□
Region (State, Province, County)	□	☑
Ship to customer	□	□
Order Number	□	□

Row 1 of 10

9. Repeat the steps for the other relationships that are located in the planning area as well.

4 Appendix

```
*&---------------------------------------------------------------------*
*& Report  Z_GET_INFO_CHAR_RELATION
*&
*&---------------------------------------------------------------------*
*&
*&
*&---------------------------------------------------------------------*

REPORT  z_get_set_info_char_relation.

TYPE-POOLS: rs.

CONSTANTS:
  l_c_mtype_exit  TYPE rspls_mtype VALUE 'E',
  l_c_role_source TYPE rspls_cha_role VALUE 'S',
  l_c_role_target TYPE rspls_cha_role VALUE 'T',
  l_c_exit_class  TYPE rspls_cr_exitcl VALUE 'CL_RSPLS_CR_EXIT_BPS'.

DATA: lt_steps TYPE upc_yto_steps,
      ls_steps TYPE upc_ys_steps,
      l_chanm TYPE upc_ys_cha,
      l_info_cube TYPE rsinfocube,
      l_info_prov TYPE rsinfoprov,

      l_error TYPE rs_bool,
      l_exists TYPE rs_bool,
      l_s_cr_all TYPE rspls_s_cr_all_set,
      l_s_cr_all_dt TYPE rspls_s_cr_all_dt,
      l_s_cr_role TYPE rspls_s_cr_role,
      l_s_cr_step TYPE rspls_s_cr_steps,

      l_return TYPE rsplfb_rfc_ret_val,
      l_t_bapiret2 TYPE rsplfb_t_bapiret2,
      l_s_bapiret2 TYPE bapiret2.

DEFINE check_error.
  if l_return <> 0.
    loop at l_t_bapiret2 into l_s_bapiret2.
      write: / l_s_bapiret2-id, l_s_bapiret2-number, l_s_bapiret2-
message.
    endloop.
    exit.
  else.
    write 'succesful' color 3.
  endif.
END-OF-DEFINITION.

PARAMETERS: area LIKE upc_steps-area DEFAULT 'ZADCU01',
            update type c as checkbox.

CLEAR l_error.
CLEAR l_s_cr_all.

* Read definitions from BPS Customizing
CALL FUNCTION 'UPC_STEPS_GET'
  EXPORTING
    i_area    = area
  IMPORTING
    eto_steps = lt_steps.

WRITE: 'Characteristic Relations of Planning Area: ', area.

* Determine corresponding InfoProvider
SELECT SINGLE infocube FROM upc_bw_area INTO l_info_cube
   WHERE area = area.
IF sy-subrc <> 0.
  l_error = rs_c_true.
  l_info_prov ='<Not found>'.
ENDIF.
WRITE: / 'Corresponding InfoProvider: ', l_info_cube.
ULINE.

* Write infos to screen - for information only
WRITE: / 'Overview BPS customizing' color 1.
LOOP AT lt_steps INTO ls_steps.
  NEW-LINE. ULINE.
  WRITE at: 2(8) 'Step no.', 11(2) ls_steps-
step, 20(18) 'Uses derivation = ', 39(1) ls_steps-derive.

  WRITE at: /4(30) 'Source Chars:' color 2. NEW-LINE.
  LOOP AT ls_steps-to_source INTO l_chanm.
    WRITE AT: /6(30) l_chanm.
  ENDLOOP.

  WRITE at: /4(30) 'Target Chars:' color 2. NEW-LINE.
  LOOP AT ls_steps-to_target INTO l_chanm.
    WRITE AT: /6(30) l_chanm.
  ENDLOOP.
ENDLOOP.
ULINE.

IF l_error = rs_c_true or update = rs_c_false.
  EXIT.
ENDIF.

WRITE: / 'Execution log: change/create char-relations in BI Integrated
Planning' color 1. uline.

* Check if char-relations already exist
l_info_prov = l_info_cube.
WRITE: / 'Check if char relations already exist ... '.
CALL FUNCTION 'RSPLSCR_PLCR_EXISTS'
  EXPORTING
    i_infoprov   = l_info_prov
  IMPORTING
    e_exists     = l_exists
    e_t_bapiret2 = l_t_bapiret2
    e_return     = l_return.
check_error.

IF l_exists = rs_c_false.

  WRITE: / 'Create new characteristic relations object ... '.
  CALL FUNCTION 'RSPLSCR_PLCR_CREATE'
    EXPORTING
      i_infoprov   = l_info_prov
    IMPORTING
      e_t_bapiret2 = l_t_bapiret2
      e_return     = l_return.
  check_error.

ENDIF.

* Loading current definitions like key-date, max no. of combination etc.
WRITE: / 'Load current definitions of BI Integrated Planning ... '.
CALL FUNCTION 'RSPLSCR_PLCR_LOAD'
  EXPORTING
    i_infoprov   = l_info_prov
    i_readonly   = rs_c_false
  IMPORTING
    e_s_cr_dt    = l_s_cr_all_dt
    e_t_bapiret2 = l_t_bapiret2
    e_return     = l_return.
check_error.

* Restore header information like key-date etc.
MOVE-CORRESPONDING l_s_cr_all_dt-prop TO l_s_cr_all-prop.
MOVE-CORRESPONDING l_s_cr_all_dt-head TO l_s_cr_all-head.

* Create steps in BI Integrated Planning based on BPS definitions
WRITE: / 'Creating new steps for characteristic relations ships in BI In
tegrated Planning ... '.

CLEAR l_s_cr_step.
CLEAR l_s_cr_role.

LOOP AT lt_steps INTO ls_steps.

  l_s_cr_step-step   = ls_steps-step.
  l_s_cr_step-derive = ls_steps-derive.
  l_s_cr_step-mtype  = l_c_mtype_exit.
  l_s_cr_step-exitclass = l_c_exit_class.
  l_s_cr_step-used   = rs_c_true.

  LOOP AT ls_steps-to_source INTO l_chanm.
    l_s_cr_role-step  = ls_steps-step.
    l_s_cr_role-role  = l_c_role_source.
    l_s_cr_role-chanm = l_chanm.
    APPEND l_s_cr_role TO l_s_cr_all-t_role.
  ENDLOOP.

  LOOP AT ls_steps-to_target INTO l_chanm.
    l_s_cr_role-step  = ls_steps-step.
    l_s_cr_role-role  = l_c_role_target.
    l_s_cr_role-chanm = l_chanm.
    APPEND l_s_cr_role TO l_s_cr_all-t_role.
  ENDLOOP.

  APPEND l_s_cr_step TO l_s_cr_all-t_steps.

ENDLOOP.
WRITE 'succesful' color 3.

* Validate the customizing created
WRITE: / 'Check customizing definitions created ... '.
CALL FUNCTION 'RSPLSCR_PLCR_CHECK'
  EXPORTING
    i_infoprov   = l_info_prov
    i_use_cr_all = rs_c_true
    i_s_cr_all   = l_s_cr_all
  IMPORTING
    e_t_bapiret2 = l_t_bapiret2
    e_return     = l_return.
check_error.

* Change the current customizing
WRITE: / 'Apply changes to characteristic relations in BI Integrated
Planning ...'.
CALL FUNCTION 'RSPLSCR_PLCR_CHANGE'
  EXPORTING
    i_infoprov   = l_info_prov
    i_s_cr_all   = l_s_cr_all
  IMPORTING
    e_t_bapiret2 = l_t_bapiret2
    e_return     = l_return.
check_error.

* Save the new customizing
WRITE: / 'Save created customizing ...'.
CALL FUNCTION 'RSPLSCR_PLCR_SAVE'
  EXPORTING
    i_infoprov   = l_info_prov
    i_do_dequeue = rs_c_true
    i_use_cr_all = rs_c_true
    i_s_cr_all   = l_s_cr_all
  IMPORTING
    e_t_bapiret2 = l_t_bapiret2
    e_return     = l_return.
check_error.
```