



**How-to Guide
SAP NetWeaver '04**

How To... Call a BPS Web Interface with Predefined Selections

Version 1.00 – November 2004

**Applicable Releases:
SAP NetWeaver '04
(BW 3.5, SEM 3.1B – SEM3.5)**

© Copyright 2004 SAP AG. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP AG. The information contained herein may be changed without prior notice.

Some software products marketed by SAP AG and its distributors contain proprietary software components of other software vendors.

Microsoft, Windows, Outlook, and PowerPoint are registered trademarks of Microsoft Corporation.

IBM, DB2, DB2 Universal Database, OS/2, Parallel Sysplex, MVS/ESA, AIX, S/390, AS/400, OS/390, OS/400, iSeries, pSeries, xSeries, zSeries, z/OS, AFP, Intelligent Miner, WebSphere, Netfinity, Tivoli, and Informix are trademarks or registered trademarks of IBM Corporation in the United States and/or other countries.

Oracle is a registered trademark of Oracle Corporation.

UNIX, X/Open, OSF/1, and Motif are registered trademarks of the Open Group.

Citrix, ICA, Program Neighborhood, MetaFrame, WinFrame, VideoFrame, and MultiWin are trademarks or registered trademarks of Citrix Systems, Inc.

HTML, XML, XHTML and W3C are trademarks or registered trademarks of W3C[®], World Wide Web Consortium, Massachusetts Institute of Technology.

Java is a registered trademark of Sun Microsystems, Inc.

JavaScript is a registered trademark of Sun Microsystems, Inc., used under license for technology invented and implemented by Netscape.

MaxDB is a trademark of MySQL AB, Sweden.

SAP, R/3, mySAP, mySAP.com, xApps, xApp, SAP NetWeaver, and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP AG in Germany and in several other countries all over the world. All other product and service names mentioned are the trademarks of their respective companies. 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

The planning process in a company never is an isolated process. Usually planning and reporting go hand in hand in a closed loop. Thus a close link between planning and reporting is necessary. In BW web reporting it is possible to call up a web report with preconfigured selections by transporting selection values via the URL. Thus a web report can be called from within a given context (say we want to see data for a specific cost center only) and will show exactly the data that matches within the context. In BPS this is not possible in the standard and the user has to set the selections when he/she enters the planning. The aim of this how to paper is to show an easy workaround that allows the setting of variables in a BPS web interface via a URL.

2 Introduction

In BPS variables are used for interactively setting selections. The user itself usually does this. On the other hand the BPS provides an API that can be used for setting variables. In order to get the desired functionality we have to automate the setting of the variables. We solve this problem by using a BPS application that sets variable values that are transferred by the URL and then automatically starts the web interface. As we do not want to create a BSP application for each web interface where we want to use with this functionality, we also specify the name of the web interface in the URL.

3 The Step By Step Solution

3.1 Introduction

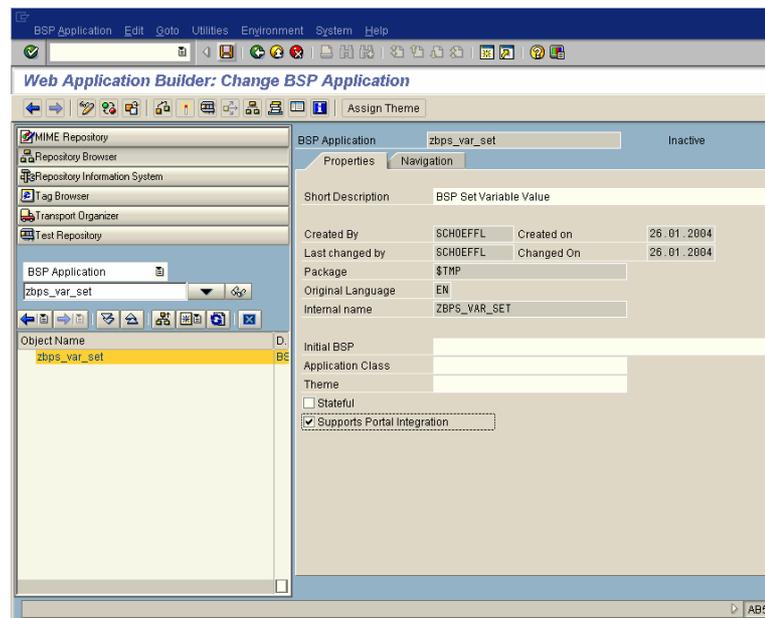
We assume that the use of variables and of web interfaces is known. For setting up a scenario create a web interface in the web interface builder and use variables for those values that should be transferred from the source (maybe a BW report) to the web interface via the URL. If the users should be able to change the selection a variable selector can be used to display the value of the variable. If the user should not be able to change the variable value then the variable could either be displayed as text or not at all. In any case the variable must be used in the planning level. Note that the variable cannot be a variable of type attribute, as this type cannot be set using the API. Note that the replacement type for the variable must be chosen correctly so the variable can be set by the API. The easiest way to assure this is to set the replacement type to “user-defined” and select “user input allowed”. Even if for a given user there are no values yet in the variable the web interface can be started with the chosen values.

3.2 Create the BSP Application

1. Create a new BSP Application

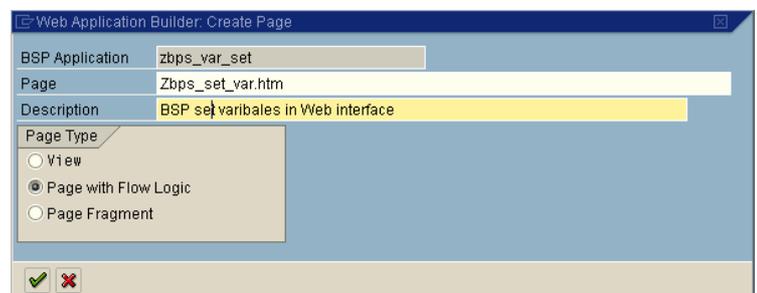
A BSP Application (Business Server Pages) can be created using the development workbench (se80).

Call up transaction se80, select “BSP Application” and enter a name for you application. Set the flags as shown in the screen shot.



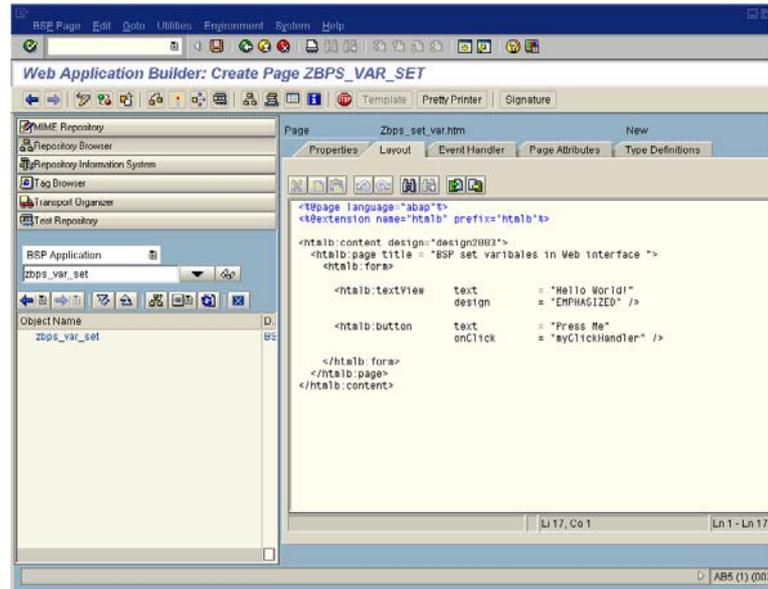
2. Create a new page

Within you application, create a BSP page with flow logic by using the context menu.



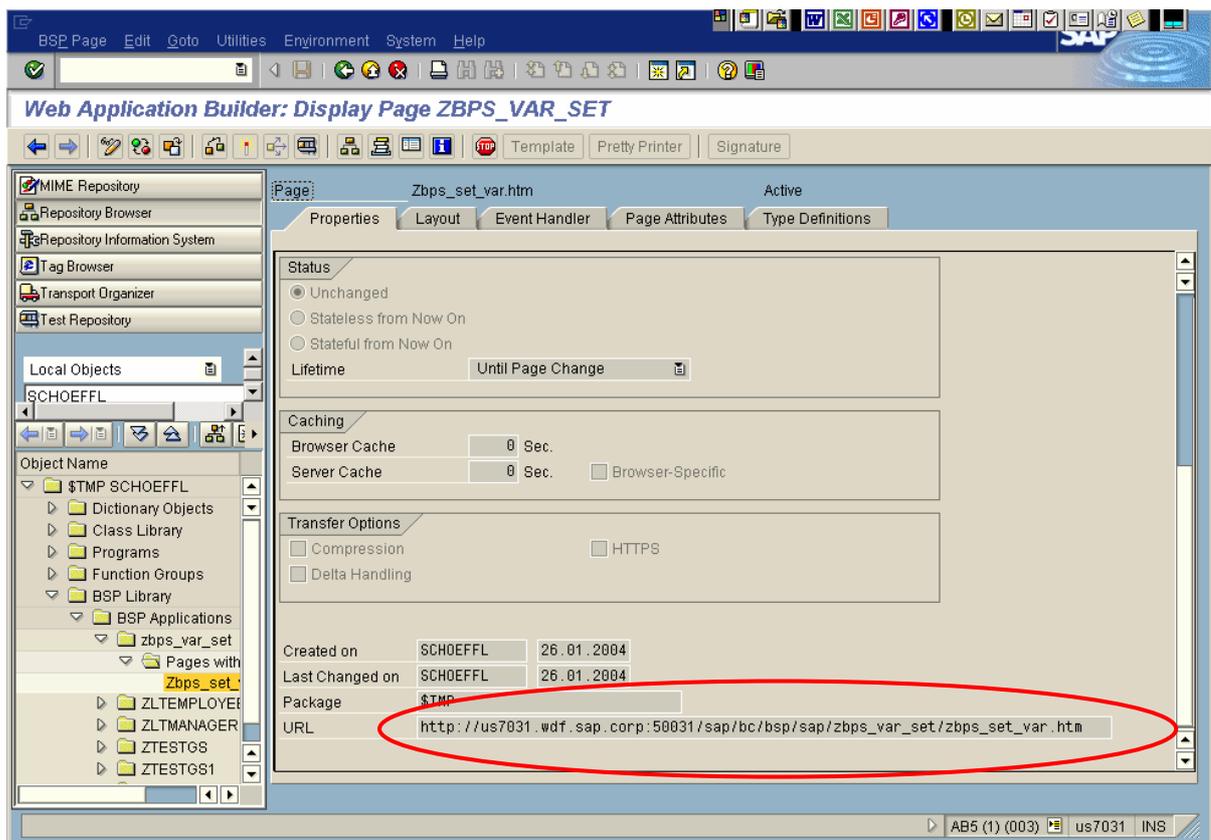
3. Enter the coding contained in the appendix

The system will automatically generate some coding on the tab strip layout. Replace this coding by the coding in the appendix and generate the BSP application.



3.3 Use the BSP Application to call a Web Interface

On the properties tab strip you find the URL for the new web page.



When calling the page you can extend the URL with parameters containing the information necessary for setting the variables and calling the proper web interface. In our example we use the following parameter names. The names can easily be changed by adapting the coding:

area	Name of the planning area (Note that in our coding it is assumed that all variables are defined in the same planning area!)
bps-appl	Name of the BSP web interface
bps-page	Name of the page within the web application
var1, var2, ...	Names of the variables
value1_1, value1_2,..., value2_1,...	Values for the variables

Note that for one variable you can have more than one value. This is necessary as a variable can contain more than one characteristic. The sort order of these values is the same as the sort order of the characteristics that are used in that variable.

The BSP application we have produced checks the type of the variables. You can use variables of type characteristic value, hierarchy node, and attribute. As mentioned above you cannot use a variable type attribute, as this type cannot be set using the API.

3.4 Example for a Call of Our BSP Application

We want to call a web interface that is called 'Example'. The web interface contains two pages, called 'Page1' and 'Page2', and we want the web interface to be started with 'Page2' as active page.

In our web interface (or the planning levels used in this web interface) we have two variables: 'var_year' for fiscal year and a variable called 'var_cc' that contains cost center and company code. We want to set the year to '2004', the cost center to '1234' and the company code to '1000'. As the characteristic 0COMPCODE is before 0COSTCNTR (the technical names of the characteristics we use in our planning level) we have to fill the value for company code before we fill the value for the cost center. Both variables are defined in the planning area 'example_area'. Our URL that we use will look like as follows:

```
<URL we get from our BSP application>?area=example_area&bps-appl=Example&bps-page=Page2&var1=var_year&value1_1=2004&var2=var_cc&value2_1=1000&value2_2=1234
```

Note that as the BPS API uses characteristic values in external format you have to provide the values in the URL in external format as well. If for example you are using a variable for the characteristic 0FISCPER the internal characteristic value for January 2004 is '2004001', whereas the external value is '001.2004'.

If we now call the above URL our BSP application will set the variable values and automatically call the web interface within a frame. If any error occurs the system will not call the web interface but will display the error message(s).

4 Appendix (Coding)

```
<<< <%@page language="abap" %>
<%@extension name="htmlb" prefix="htmlb" %>
<html>
<%
data: l_variable type UPC_Y_VARIABLE,
l_value type UPC_Y_CHAVLEXT,
l_num_value type UPC_Y_VAR_NUMBER,
l_name type string,
l_cntr type n,
l_cntr2 type n,
l_page type string,
l_bps_appl type string,
l_url type string.

DATA: ls_varse1 TYPE upc_ys_api_varse1,
lt_varse1 TYPE STANDARD TABLE OF upc_ys_api_varse1.

DATA: l_subrc type sysubrc,
l_type type UPC_Y_VAR_TYPE,
l_error type c,
ls_hiese1 type upc_ys_api_hiese1,
ls_cha TYPE upc_ys_cha,
lt_cha TYPE STANDARD TABLE OF upc_ys_cha,
ls_return type bapiret2,
lt_return type standard table of bapiret2.

data: l_area type upc_y_area.

l_area = request->get_form_field( 'area' ).
do.
  clear l_variable.
  add 1 to l_cntr.
  concatenate 'var' l_cntr into l_name.
  l_variable = request->get_form_field( l_name ).
  if l_variable is initial.
    exit.
  endif.
  CLEAR: l_subrc, l_type, ls_hiese1, lt_cha, ls_cha.

  CALL FUNCTION 'API_SEMBPS_VARIABLE_GETDETAIL'
    EXPORTING
      i_area      = l_area
      i_variable  = l_variable
    IMPORTING
      e_subrc     = l_subrc
      es_return  = ls_return
      e_type      = l_type
      es_hiese1  = ls_hiese1
    TABLES
      etk_cha    = lt_cha.
  if l_subrc <> 0.
    l_error = 'X'.
    append ls_return to lt_return.
    exit.
  endif.

  CASE l_type.
    WHEN 'HIERARCHY'.
      read table lt_cha index 1 into ls_cha.
      concatenate 'value' l_cntr '_1' into l_name.
      l_value = request->get_form_field( l_name ).
      ls_hiese1-seqno = '1'.
      ls_hiese1-chanm = ls_cha-chanm.
```

```

ls_hiesel-nodenameext = l_value.

CALL FUNCTION 'API_SEMBPS_VARIABLE_SET'
  EXPORTING
    i_area      = l_area
    i_variable  = l_variable
    is_hiesel   = ls_hiesel
  IMPORTING
    e_subrc    = l_subrc
    es_return   = ls_return.
if l_subrc <> 0.
  l_error = 'X'.
  append ls_return to lt_return.
  exit.
endif.

WHEN 'CHAR'.
* fill a normal selection table
sort lt_cha by chanm.
clear l_cntr2.
CLEAR lt_varsel.
loop at lt_cha into ls_cha.
  add 1 to l_cntr2.
  concatenate 'value' l_cntr '_' l_cntr2 into l_name.
  l_value = request->get_form_field( l_name ).
  ls_varsel-seqno = 1.
  ls_varsel-chanm = ls_cha-chanm.
  ls_varsel-sign = 'I'.
  ls_varsel-option = 'EQ'.
  ls_varsel-low = l_value.
  APPEND ls_varsel TO lt_varsel.
endloop.
CALL FUNCTION 'API_SEMBPS_VARIABLE_SET'
  EXPORTING
    i_area      = l_area
    i_variable  = l_variable
  IMPORTING
    e_subrc    = l_subrc
    es_return   = ls_return
  TABLES
    itk_varsel = lt_varsel.
if l_subrc <> 0.
  l_error = 'X'.
  append ls_return to lt_return.
  exit.
endif.

when 'ATTRIBUTE'.
* variables type attribute cannot be set!

when 'NUMBER'.
concatenate 'value' l_cntr '_1' into l_name.
l_num_value = request->get_form_field( l_name ).
CALL FUNCTION 'API_SEMBPS_VARIABLE_SET'
  EXPORTING
    i_area      = l_area
    i_variable  = l_variable
    I_NUMVALUE  = l_num_value
  IMPORTING
    e_subrc    = l_subrc
    es_return   = ls_return.
if l_subrc <> 0.
  l_error = 'X'.
  append ls_return to lt_return.
  exit.
endif.

```

```

        WHEN OTHERS.
        ENDCASE.
    enddo.

CALL FUNCTION 'API_SEMBPS_POST'
    IMPORTING
        e_subrc    = l_subrc
        es_return  = ls_return.
if l_subrc <> 0.
    l_error = 'X'.
    append ls_return to lt_return.
endif.

* build up the url
* get the name of the BPS-application
l_bps_appl = request->get_form_field( 'bps-appl' ).
l_page = request->get_form_field( 'bps-page' ).

if L_bps_appl is initial.
* produce an error
    l_error = 'X'.
else.
    concatenate  '/sap/bc/bsp/sap/'
        l_bps_appl
        '/'
        l_bps_appl
        '.htm'
    into l_url.
    if not l_page is initial.
        concatenate l_url  '?bps-page=' l_page into l_url.
    endif.
endif.

*we only display the web interface if everything went okay.
if l_error is initial.
%>
<frameset framespacing=0 frameborder=0 border=0>
<frame src="<%= l_url%>" name="x" framespacing=0 frameborder=0 border=0
marginwidth=0 marginheight=0>
</frameset>
<%
    else. "l_error is initial.
%>
<body>
<table CELLSPACING='10' class="">
<TR>
<TD>
<h2>Serious problem...</h2>
</TD>
</TR>
<%
    loop at lt_return into ls_return.
%>
<TR>
<TD>
<h3><%= ls_return-message %></h3>
</TD>
</TR>
<%
    endloop.
%>
</table>
</body>
<%
    endif. "l_error is initial.
%>
</html>

```

www.sdn.sap.com/irj/sdn/howtoguides