

ABAP Code Sample for Download of a BSP Application

Applies To:

Web Application Server, ABAP Business Server Pages

Summary

This code will save a whole BSP application to a local drive. This code was provided by OSS as a workaround for the lack of search functionality within a whole BSP application. All page attributes, the layout and the event handlers of a BSP page are saved in a file/page.

When the BSP application is downloaded, one can search on the code via local tools like the standard File Explorer within Windows XP.

By: Eddy De Clercq

Company: Katholieke Universiteit Leuven

Date: 24 February 2005

The code

Create a report e.g. Z_DOWNLOAD_BSP and paste the following code in it:

```
*****
*&-----*
*& Report Z_DOWNLOAD_BSP
*&
*&-----*
*& The report asks for the name of the BSP application.
*& Via a dialog a local directory must be specified as target of the
*& download operation.
*& Each page of the application is downloaded to a separate file in
*& this directory: directory\pagename
*&-----*

REPORT Z_DOWNLOAD_BSP.
type-pools: SO2.
types: t_page_list type standard table of o2pagkey.

parameters: g_appl type O2APPLNAME.

constants: c_separator_line type string value
'*****'.
```

ABAP Code Sample for Download of a BSP Application

```
perform main.

*****
*****

form main.
data: l_pages type t_page_list,
l_ref_page type ref to cl_o2_api_pages,
l_table type o2pageline_table,
l_folder type string,
l_lower type string,
l_filename type string.
field-symbols: <f_page> type o2pagkey.

* get local directory path
call method CL_GUI_FRONTEND_SERVICES=>DIRECTORY_BROWSE
* EXPORTING
* WINDOW_TITLE =
* INITIAL_FOLDER =
CHANGING
SELECTED_FOLDER = l_folder
EXCEPTIONS
CNTL_ERROR = 1
ERROR_NO_GUI = 2
NOT_SUPPORTED_BY_GUI = 3
others = 4.
IF SY-SUBRC <> 0.
MESSAGE ID SY-MSGID TYPE SY-MSGTY NUMBER SY-MSGNO
WITH SY-MSGV1 SY-MSGV2 SY-MSGV3 SY-MSGV4.
return.
ENDIF.
if l_folder is initial. return. endif.

* get all pages of selected BSP application
perform get_all_pages changing l_pages.
```

ABAP Code Sample for Download of a BSP Application

```
* perform each page
loop at l_pages assigning <f_page>.

* load page data
perform load_page using <f_page>
changing l_ref_page.

if l_ref_page is bound.
* collect the content of the page in an internal table
perform print using l_ref_page
changing l_table.

if not l_table is initial.
l_lower = <f_page>-pagekey.
translate l_lower to lower case.
* determine file path for the page
concatenate l_folder '\\' l_lower into l_filename.
* download the content of the page to file
perform download using l_filename l_table.
endif.
endif.
endloop.

endform.

*****
form download using p_filename type string
p_table type standard table.
* download the content of an internal table to a local file

call method CL_GUI_FRONTEND_SERVICES=>GUI_DOWNLOAD
EXPORTING
FILENAME = p_filename
CHANGING
```

ABAP Code Sample for Download of a BSP Application

```
DATA_TAB = p_table
EXCEPTIONS
FILE_WRITE_ERROR = 1
NO_BATCH = 2
GUI_REFUSE_FILETRANSFER = 3
INVALID_TYPE = 4
NO_AUTHORITY = 5
UNKNOWN_ERROR = 6
HEADER_NOT_ALLOWED = 7
SEPARATOR_NOT_ALLOWED = 8
FILESIZE_NOT_ALLOWED = 9
HEADER_TOO_LONG = 10
DP_ERROR_CREATE = 11
DP_ERROR_SEND = 12
DP_ERROR_WRITE = 13
UNKNOWN_DP_ERROR = 14
ACCESS_DENIED = 15
DP_OUT_OF_MEMORY = 16
DISK_FULL = 17
DP_TIMEOUT = 18
FILE_NOT_FOUND = 19
DATAPROVIDER_EXCEPTION = 20
CONTROL_FLUSH_ERROR = 21
NOT_SUPPORTED_BY_GUI = 22
ERROR_NO_GUI = 23
others = 24.
IF SY-SUBRC <> 0.
MESSAGE ID SY-MSGID TYPE SY-MSGTY NUMBER SY-MSGNO
WITH SY-MSGV1 SY-MSGV2 SY-MSGV3 SY-MSGV4.
ENDIF.

endform.

*****
form get_all_pages changing p_pages type t_page_list.
```

ABAP Code Sample for Download of a BSP Application

```
* get all pages of the application
select applname pagekey from o2pagdir into table p_pages
where applname = g_appl and
pagetype <> so2_controller.
endform.

*****

form load_page using p_key type o2pagkey
changing p_data type ref to cl_o2_api_pages.
* load the data of one page

clear p_data.

call method cl_o2_api_pages=>load_with_access_permission
exporting
p_mode = 'SHOW'
p_pagekey = p_key
p_version = 'A' "'I' for inactive version
importing
p_page = p_data
exceptions
others = 1.
endform.

*****

form print using p_data type ref to cl_o2_api_pages
changing p_table type o2pageline_table.
* collect the content of a page into an internal table

data: l_attrs type o2pagattr,
l_params type o2pagpar_tabletype,
l_source type rsourceset,
l_layout type o2pageline_table,
l_ev_handlers type so2_ev_handler_list,
l_ttyp(11) type c,
```

ABAP Code Sample for Download of a BSP Application

```
l_line type string.

field-symbols: <f_param> type o2pagpars,
<f_evh_name> type so2_ev_handler_name.

refresh p_table.

* get page properties
CALL METHOD P_DATA->GET_ATTRS
IMPORTING
P_ATTRS = l_attrs
EXCEPTIONS
others = 1.
check l_attrs-pagetype <> so2_controller.

append c_separator_line to p_table.
concatenate '*' l_attrs-applname '/' l_attrs-pagekey into l_line.
append l_line to p_table.
append c_separator_line to p_table.
append space to p_table.

* local types
if l_attrs-pagetype = so2_full_page.
CALL METHOD P_DATA->GET_TYPE_SOURCE
IMPORTING
P_SOURCE = l_source
EXCEPTIONS
others = 1.

if not l_source is initial.
append c_separator_line to p_table.
append '* local types' to p_table.
append c_separator_line to p_table.
append lines of l_source to p_table.
do 3 times.
```

ABAP Code Sample for Download of a BSP Application

```
append space to p_table.
enddo.
endif.
endif.

* page attributes
if l_attr-pagetype = so2_full_page or
l_attr-pagetype = so2_view_page.
CALL METHOD P_DATA->GET_PARAMETERS
IMPORTING
P_PARAMETERS = l_params
EXCEPTIONS
others = 1.

if not l_params is initial.
append c_separator_line to p_table.
append '* page attributes' to p_table.
append c_separator_line to p_table.

loop at l_params assigning <f_param>.
if <f_param>-typtyp = so2_paramtyp_type.
l_tytyp = 'TYPE'.
else.
l_tytyp = 'TYPE REF TO'.
endif.
concatenate <f_param>-aliasname l_tytyp
<f_param>-type into l_line
separated by space.
append l_line to p_table.
endloop.

do 3 times.
append space to p_table.
enddo.
endif.
```

ABAP Code Sample for Download of a BSP Application

```
endif.

* layout
append c_separator_line to p_table.
append '* layout' to p_table.
append c_separator_line to p_table.
CALL METHOD P_DATA->GET_PAGE
IMPORTING
P_CONTENT = l_layout
EXCEPTIONS
others = 1.
append lines of l_layout to p_table.
do 3 times.
append space to p_table.
enddo.

* event handler
if l_attr-pagetype = so2_full_page.
CALL METHOD P_DATA->GET_EVENT_HANDLER_LIST
EXPORTING
P_WITH_ALL_PREDEFINED = ' '
IMPORTING
P_EV_HANDLER_L = l_ev_handlers
EXCEPTIONS
others = 1.

loop at l_ev_handlers assigning <f_evh_name>.
append c_separator_line to p_table.
concatenate '* eventhandler' <f_evh_name>-name into l_line
separated by space.
append l_line to p_table.
append c_separator_line to p_table.
clear l_source.
CALL METHOD P_DATA->GET_EVENT_HANDLER
EXPORTING
```

ABAP Code Sample for Download of a BSP Application

```
P_NAME = <f_evh_name>-key
IMPORTING
P_SOURCE = l_source
* P_ATTRS =
EXCEPTIONS
others = 1.
append lines of l_source to p_table.
do 3 times.
append space to p_table.
enddo.
endloop.
endif.

endform.
```

Disclaimer & Liability Notice

This document may discuss sample coding, which does not include official interfaces and therefore is not supported. 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 of the code and methods suggested here, and anyone using these methods, is doing it under his/her own responsibility.

SAP offers no guarantees and assumes no responsibility or liability of any type with respect to the content of the technical article, including any liability resulting from incompatibility between the content of the technical article and the materials and services offered by SAP. You agree that you will not hold SAP responsible or liable with respect to the content of the Technical Article or seek to do so.

Author Bio



Eddy De Clercq has 20 years experience in computing. He currently works at the Katholieke Universiteit Leuven, the oldest university of the Low Countries and the largest Flemish university. Eddy makes part of the E-university team that creates mostly self services (web) applications.

Copyright © 2005 SAP AG, Inc. All Rights Reserved. SAP, mySAP, mySAP.com, xApps, xApp, 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, service names, trademarks and registered trademarks mentioned are the trademarks of their respective owners.