

SDN Community Contribution

(This is not an official SAP document.)

Disclaimer & Liability Notice

This document may discuss sample coding or other information that does not include SAP official interfaces and therefore is not supported by SAP. 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 the information, code or methods suggested in this document, and anyone using these methods does so at his/her own risk.

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

Applies To:

BW 3.5

Summary

BW provides a mechanism for sending query results as email using [Information Broadcasting](#). But user requirements sometime get really innovative & difficult to handle with the standard delivered tools. If you need the ability to execute a BW query and send the results as a HTML email from an ABAP program, you can use the following code sample. This has been implemented as a Function Module interface.

By: [Duraij Athavan Raja](#)

Company: Atos Origin Middle East

Date: 14th March 2006

The CODE

1. As a first step , create a Function group (if you are not familiar with creating function modules and function groups check this link http://help.sap.com/saphelp_nw04/helpdata/en/26/64f623fa8911d386e70000e82011b8/frameset.htm)
2. Copy paste the following code in the top include (global data) of the function group

```
DATA: send_request      TYPE REF TO cl_bcs.
DATA: text              TYPE bcsy_text.
DATA: document         TYPE REF TO cl_document_bcs.
DATA: sender           TYPE REF TO if_sender_bcs.
DATA: recipient        TYPE REF TO if_recipient_bcs.
DATA: bcs_exception    TYPE REF TO cx_bcs.
DATA: sent_to_all      TYPE os_boolean.
DATA: conlength TYPE i ,
      conlengths TYPE so_obj_len ,
      result_content TYPE string .
DATA: e_r_page TYPE REF TO cl_rsr_www_page.
data: content_length      TYPE w3param-cont_len ,
      content_type TYPE w3param-cont_type,
      return_code TYPE w3param-ret_code .

data: html type standard table of w3html .
data: server type string ,
      port type string .
data: wa_rec type ad_smtpadr .
data: bcs_message type string .
```

3. Create the Function Module with parameters as shown in the following screen shots.

Import parameters:

Function module: Y_EMAIL_QRESULTS Active

Attributes Import Export Changing Tables Exceptions Source code

Parameter Name	Type...	Associated Type	Default value	Opt...	Pa...	Short text
SENDER_ID	TYPE	AD_SMTPADR		<input type="checkbox"/>	<input type="checkbox"/>	Sender Email address
SUBJECT	TYPE	SO_OBJ_DES		<input type="checkbox"/>	<input type="checkbox"/>	Short description of contents
				<input type="checkbox"/>	<input type="checkbox"/>	
				<input type="checkbox"/>	<input type="checkbox"/>	

Tables:

Function module: Y_EMAIL_QRESULTS Active

Attributes Import Export Changing Tables Exceptions Source code

Parameter Name	Type spec.	Associated Type	Optional	Short text	Long text
QUERY_PARAMS	TYPE	RRXW3TQUERY	<input type="checkbox"/>	Query Table	
RECEIPIENTS	TYPE	BCSY_SMTPA	<input type="checkbox"/>	BCS: Internal Table with SMTP Address..	
RETURN	TYPE	TABLE_OF_STRINGS	<input checked="" type="checkbox"/>	Return Parameter	
			<input type="checkbox"/>		

4. Copy paste the following code in the Source code section of the Function Module.

```
FUNCTION y_email_qresults.
*-----
* "Local Interface:
* " IMPORTING
* "     REFERENCE(SENDER_ID) TYPE  AD_SMTPADR
* "     REFERENCE(SUBJECT) TYPE  SO_OBJ_DES
* " TABLES
* "     QUERY_PARAMS TYPE  RRXW3TQUERY
* "     RECEIPIENTS TYPE  BCSY_SMTPA
* "     RETURN TYPE  TABLE_OF_STRINGS OPTIONAL
*-----
DATA: tmp_str TYPE string .

IF NOT receipients[] IS INITIAL .
    MOVE: 'text/html' TO content_type .
    CLEAR: e_r_page,result_content .

    CALL FUNCTION 'RRW3_WEBRFC'
* EXPORTING
* I_R_PARAMETER      =
    IMPORTING
        e_r_page      = e_r_page
    TABLES
        query_string  = query_params
* html              = html
* MIME              =
    CHANGING
        return_code   = return_code
        content_type   = content_type
        content_length = content_length .

    result_content = e_r_page->n_sx_output-string.

    REPLACE ALL OCCURRENCES OF '/webrfc/bw30' IN result_content WITH '/sap/bw/Mime/BEx' .
    REPLACE ALL OCCURRENCES OF 'mime/BEx/StyleSheets/' IN result_content WITH
'/sap/bw/mime/BEx/StyleSheets/' .

    CLEAR tmp_str .
    CONCATENATE
        '<meta http-equiv="content-type" content="text/html; charset=iso-8859-1">'
        cl_abap_char_utilities=>cr_lf
```

```
'<script type="text/javascript"
src="/sap/bw/Mime/BEx/JavaScript/JSBW_C_Std.js"></script>'
cl_abap_char_utilities=>cr_lf

INTO tmp_str .

REPLACE ALL OCCURRENCES OF
'<meta http-equiv="content-type" content="text/html; charset=iso-8859-1">' IN
result_content WITH
tmp_str .
CLEAR tmp_str .
REPLACE ALL OCCURRENCES OF
'BWReports_ie5.css' IN result_content WITH
'BWReports_ie6.css' .

REPLACE ALL OCCURRENCES OF
'Mime/BEx/StyleSheets/' IN result_content WITH
'/sap/bw/Mime/BEx/StyleSheets/' .

REPLACE ALL OCCURRENCES OF 'Mime/BEx/Icons/' IN result_content WITH
'/sap/bw/Mime/BEx/Icons/' .

REPLACE ALL OCCURRENCES OF
'/sap/bw/Mime/BEx/icons/' IN result_content WITH
'/sap/bw/Mime/BEx/Icons/' .

CLEAR : server, port .
CALL FUNCTION 'TH_GET_VIRT_HOST_DATA'
EXPORTING
  protocol          = 1
  virt_idx          = 0
* LOCAL            = 1
IMPORTING
  hostname          = server
  port              = port
EXCEPTIONS
  not_found         = 1
  internal_error    = 2
  OTHERS            = 3 .
CLEAR tmp_str .
CONCATENATE 'http://' server ':' port '/sap/bw/' INTO tmp_str .

REPLACE ALL OCCURRENCES OF '/sap/bw/' IN result_content WITH tmp_str .
```

```
CLEAR tmp_str .
CLEAR: conlength,conlengths .

conlength = STRLEN( result_content ) .
conlengths = conlength .

REFRESH text .
CALL FUNCTION 'SCMS_STRING_TO_FTEXT'
  EXPORTING
    text      = result_content
  TABLES
    ftext_tab = text.

* CALL FUNCTION 'SWA_STRING_TO_TABLE'
*   EXPORTING
*     character_string      = result_content
*     append                = 'X'
*     line_size             = '255'
**  CHECK_TABLE_TYPE       = ' '
*   IMPORTING
*     character_table       = text
**  TOTAL_LENGTH           =
**  LINE_SIZE_USED         =
**  LINES_FILLED           =
**  LAST_LINE_LENGTH       =
*   EXCEPTIONS
*     no_flat_charlike_structure = 1
*     OTHERS                 = 2 .

TRY.
  CLEAR send_request .

  send_request = cl_bcs=>create_persistent( ).

  CLEAR document .
  document = cl_document_bcs=>create_document(
    i_type      = 'HTM'
    i_text      = text
    i_length    = conlengths
    i_subject   = subject ).

*   add document to send request
```

```
CALL METHOD send_request->set_document( document ).

CLEAR sender .
sender = cl_cam_address_bcs=>create_internet_address( sender_id ).
CALL METHOD send_request->set_sender
  EXPORTING
    i_sender = sender.

CLEAR wa_rec .

LOOP AT recipients INTO wa_rec .
  CLEAR recipient .

  recipient = cl_cam_address_bcs=>create_internet_address(
    wa_rec ).

* add recipient with its respective attributes to send request
  CALL METHOD send_request->add_recipient
    EXPORTING
      i_recipient = recipient
      i_express   = 'X'.
ENDLOOP .
CALL METHOD send_request->set_status_attributes
  EXPORTING
    i_requested_status = 'E'
    i_status_mail      = 'E'.

CALL METHOD send_request->set_send_immediately( 'X' ).
* ----- send document -----
CALL METHOD send_request->send(
  EXPORTING
    i_with_error_screen = 'X'
  RECEIVING
    result              = sent_to_all ).
IF sent_to_all = 'X'.
  APPEND 'Mail sent successfully ' TO return .
ENDIF.

COMMIT WORK.

CATCH cx_bcs INTO bcs_exception.
  bcs_message = bcs_exception->get_text( ).
  APPEND bcs_message TO return .
```



```
EXIT.  
  
ENDTRY.  
ELSE .  
  
    APPEND 'Specify email address for sending' TO return .  
ENDIF .
```

ENDFUNCTION.

5. Tips on passing parameters:

Import Parameters:

Import parameters	Value
SENDER_ID	raja@xxxxx.com
SUBJECT	Test mail from FM

Tables parameter:

Recipients table:

user3.google.com
user2@domain.com
user@domain.com

Query_params table

NAME	VALUE
VARIABLE_SCREE	
VAR_VALUE_EXT_3	[REDACTED]
VAR_NAME_3	0H_CCTR
VAR_VALUE_EXT_2	0012002
VAR_NAME_2	0P_FPER
VAR_NODE_I0BJNM_1	0HIER_NODE
VAR_VALUE_EXT_1	[REDACTED]
VAR_NAME_1	[REDACTED]
QUERY	YP [REDACTED] ABST_1
INFOCUBE	ZC [REDACTED]
CMD	LDOC

Author Bio



Durairaj Athavan Raja works as Business System Analyst with Atos Origin Middle East and has been involved in SAP development for over 8 years. He is a big fan of SDN.