# E-mail ABAP Report List Output in HTML Format

SAP DEVELOPER NETWORK

# 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.

# E-mail ABAP Report List Output in HTML Format

SAP DEVELOPER NETWORK

## Applies To:

WAS 6.40

## Summary

Code sample for emailing ABAP report list output as HTML email. Note this doesn't support ALV GRID reports. This has been implemented as a Function Module interface.

**By**: Durairaj Athavan Raja

**Company**: Atos Origin Middle East

**Date**: 4th April 2006

## The CODE

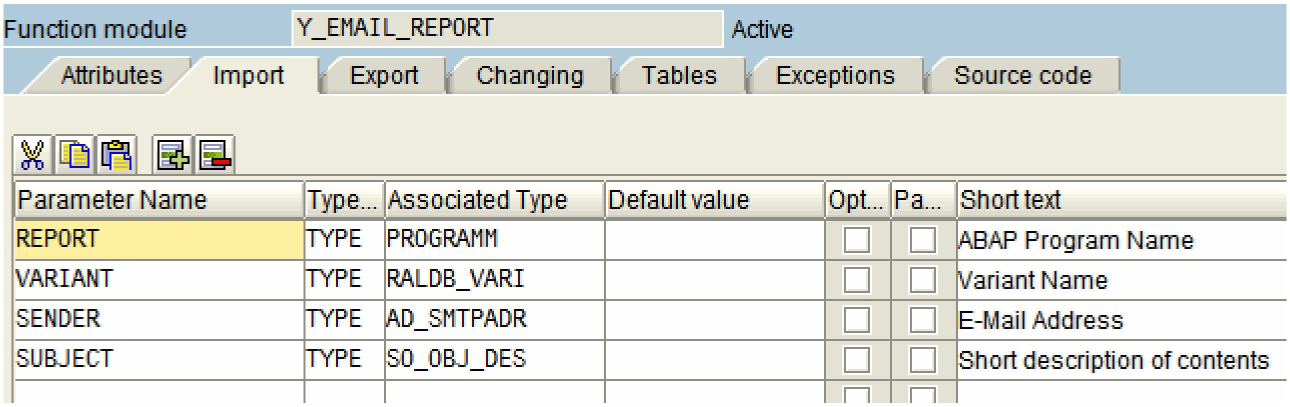
1. As a first step , create a Function group (if you are not familiar with creating function modules and function groups check this link http://help.sap.com/saphelp_nw04/helpdata/en/26/64f623fa8911d386e70000e82011b8/frameset.htm)

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_id             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: content_length     TYPE   w3param-cont_len ,
        content_type       TYPE   w3param-cont_type,
        return_code TYPE   w3param-ret_code .
  DATA: listobject TYPE TABLE OF abaplist.
  DATA: html_wa TYPE w3html.

  DATA: html TYPE STANDARD TABLE OF w3html .
  DATA: wa_rec TYPE ad_smtpadr .
  DATA: bcs_message TYPE string .
  DATA: tmp_str TYPE string .
  DATA: prog TYPE programm ,
        var TYPE raldb_vari .
```

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

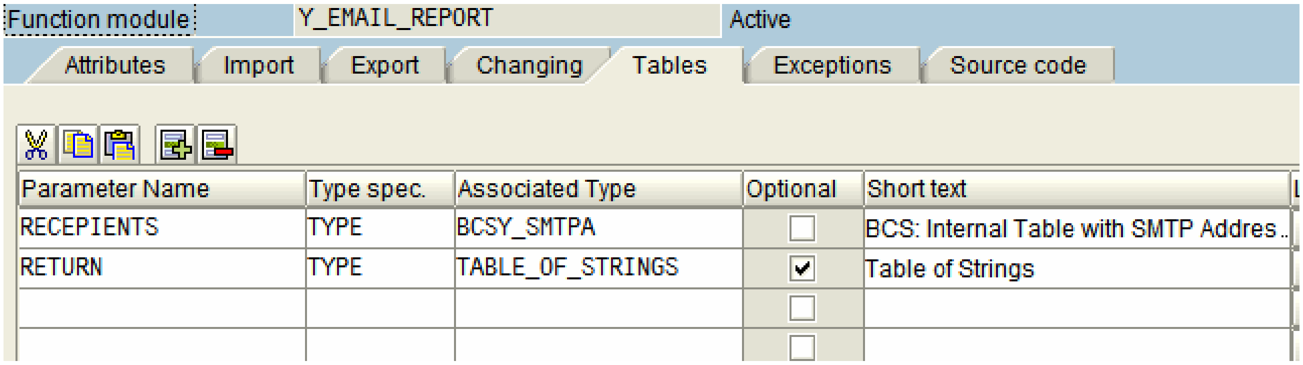
**Import parameters:**



Function module Y_EMAIL_REPORT Active

| Parameter Name | Type... | Associated Type | Default value | Opt... | Pa... | Short text |
|---|---|---|---|---|---|---|
| REPORT | TYPE | PROGRAMM | | □ | □ | ABAP Program Name |
| VARIANT | TYPE | RALDB_VARI | | □ | □ | Variant Name |
| SENDER | TYPE | AD_SMTPADR | | □ | □ | E-Mail Address |
| SUBJECT | TYPE | SO_OBJ_DES | | □ | □ | Short description of contents |
| | | | | □ | □ | |

Tables:



Function module Y_EMAIL_REPORT Active

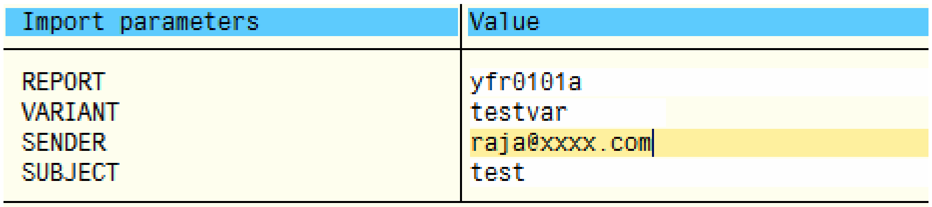
| Parameter Name | Type spec. | Associated Type | Optional | Short text |
|---|---|---|---|---|
| RECEPIENTS | TYPE | BCSY_SMTPA | □ | BCS: Internal Table with SMTP Addres... |
| RETURN | TYPE | TABLE_OF_STRINGS | ☑ | Table of Strings |
| | | | □ | |
| | | | □ | |

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

```
FUNCTION y_email_report.
*"----------------------------------------------------------------------
*"*"Local Interface:
*"  IMPORTING
*"     REFERENCE(REPORT) TYPE  PROGRAMM
*"     REFERENCE(VARIANT) TYPE  RALDB_VARI
*"     REFERENCE(SENDER) TYPE  AD_SMTPADR
*"     REFERENCE(SUBJECT) TYPE  SO_OBJ_DES
*"  TABLES
*"      RECEPIENTS TYPE  BCSY_SMTPA
*"      RETURN TYPE  TABLE_OF_STRINGS OPTIONAL
*"----------------------------------------------------------------------


  IF NOT recepients[] IS INITIAL .
    CLEAR: var, prog .
    MOVE: report TO prog ,
          variant TO var .
    TRANSLATE prog TO UPPER CASE .
    TRANSLATE var TO UPPER CASE .
    SUBMIT (prog) USING SELECTION-SET var  EXPORTING LIST TO MEMORY AND RETURN.
    CLEAR: listobject , html .
    REFRESH : listobject, html .

    CALL FUNCTION 'LIST_FROM_MEMORY'
      TABLES
        listobject = listobject.

    CALL FUNCTION 'WWW_HTML_FROM_LISTOBJECT'
      EXPORTING
        report_name = prog
      TABLES
        html        = html
        listobject  = listobject.
    CLEAR tmp_str .
    CLEAR : html_wa .
    LOOP AT html INTO html_wa .
      CONCATENATE tmp_str html_wa INTO tmp_str.

    ENDLOOP .
    CLEAR: conlength,conlengths .

    conlength = STRLEN( tmp_str ) .
    conlengths = conlength .

    TRY.
        CLEAR send_request .

        send_request = cl_bcs=>create_persistent( ).

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

*     add document to send request
        CALL METHOD send_request->set_document( document ).

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

        CLEAR wa_rec .

        LOOP AT recepients 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:

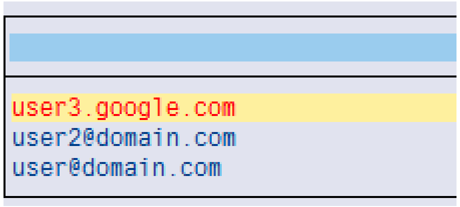
Import Parameters:



| Import parameters | Value |
|---|---|
| REPORT | yfr0101a |
| VARIANT | testvar |
| SENDER | raja@xxxx.com |
| SUBJECT | test |

Tables parameter:

Receipients table:



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

## 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.