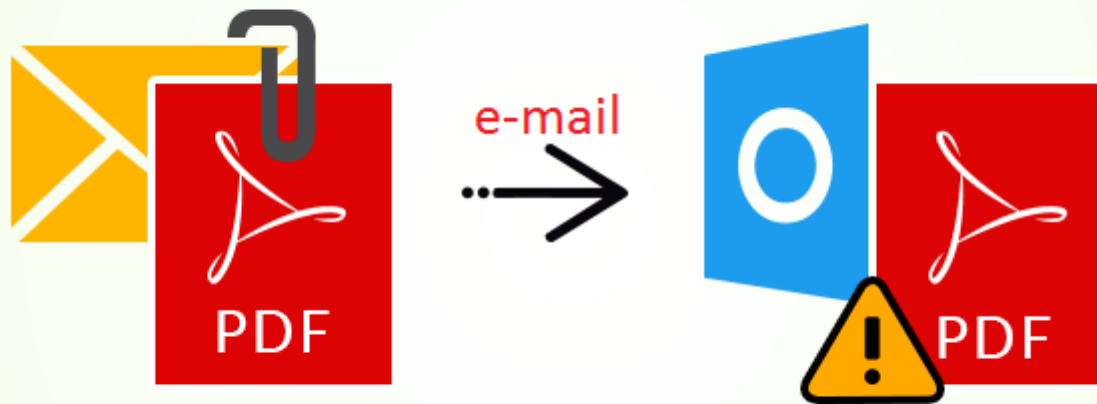


How to Email Smartform as PDF Attachment to Multiple Users?

By **AmitaJain** - July 25, 2017

Shares



Issue while sending Smartform as PDF Attachment?

Sending **Smartform output as PDF** in an e-mail is a topic which has been queried and answered multiple times in numerous forums. But, *whenever we get this requirement, we need to scramble for our previous notes or hit the search engine*. Recently, when I received this request from my business, it was nothing different to me as well. 😊

After delivering this development, I was looking for a repository where I can put my code so that if I have the same requirement in my future Projects, I know, exactly where to look. At this

point, which repository is better than SAPYard for SAP Technical stuff. (just kidding) 😊

Apart from the motive of saving my snippet in the web, the other better objective was to reveal to all freshers in SAP ABAP that, you just have to follow 4 steps for this requirement.

1. Get the Smartform output in OTF (Other Text Format).
2. Convert the OTF to Hexa decimal string format.
3. Convert the Hexa decimal string format to Binary format.
4. Build the email body and attach the Binary table as an attachment using CL_DOCUMENT_BCS-ADD_ATTACHMENT and send the email using CL_BCS class.

If you understand these four steps, you need not look the search engine in future for such developments. 😊 .

Let me be little more specific with the technicalities of the four steps.

1. Smartform Output in OTF

When you call your smartform, you would get the Smart Forms OTF Table in IMPORTING parameter JOB_OUTPUT_INFO component **OTFDATA** which is the structure for Smart Forms: Table OTF.

For the below snippet, the OTF data would be in w_return-otfdata.

```
* Call Smartform
CALL FUNCTION v_form_name
  EXPORTING
    control_parameters = w_ctrlop
```

Shares

```
output_options      = w_compop
user_settings       = abap_true
IMPORTING
  job_output_info    = w_return " This will have all output
EXCEPTIONS
  formatting_error    = 1
  internal_error      = 2
  send_error          = 3
  user_canceled       = 4
OTHERS               = 5.
```

```
* Get Output Text Format (OTF)
i_otf[] = w_return-otfdata[].
```

2. Convert the OTF to Hexa decimal string format.

The below snippet is to convert the OTF to XSTRING format.

```
* Import HexaString and filesize
CALL FUNCTION 'CONVERT_OTF'
EXPORTING
  format              = 'PDF'
  max_linewidth       = 132
IMPORTING
  bin_filesize        = v_len_in
  bin_file            = i_xstring " This is NOT Binary. This is Hexa
TABLES
  otf                 = i_otf
  lines               = i_tline
EXCEPTIONS
  err_max_linewidth   = 1
  err_format          = 2
  err_conv_not_possible = 3
OTHERS                = 4.
```

3. Convert the Hexa decimal string format to Binary format.

```
* Convert Hexa String to Binary format
CALL FUNCTION 'SCMS_XSTRING_TO_BINARY'
  EXPORTING
    buffer      = i_xstring
  TABLES
    binary_tab = i_objbin[].
```

Shares

Just a thought. If you are able to find any function to convert OTF directly to Binary Format then you can skip the 2nd step of converting OTF to Hexa. 😊

4. Use CL_DOCUMENT_BCS class and method ADD_ATTACHMENT to attach the document and use CL_BCS class to email the document.

If you want to read more about CL_BCS class usage, then please check this useful post '[Send an e-mail with subject line greater than 50 characters](#)'.

If you have a deadline, I can understand, you do not want more Gyan (*Sanskrit work which roughly translates to Preaching*). 😊 You are more interested in getting a working code which you can fit into your system. Do not worry, I will not disappoint you. *Please find a working snippet which converts the Smartform to PDF and emails it as a PDF attachment.*

```
*-----*
* Program to convert smartform as PDF and email PDF to
* multiple recipients.
*-----*
TABLES adr6.
*-----*
*-----*
```

```

* Global Data Declaration for ease
* In actual project, please do not define all these in global
*-----*
DATA: i_otf      TYPE itcoo   OCCURS 0 WITH HEADER LINE,
      i_tline    LIKE tline   OCCURS 0 WITH HEADER LINE,
      i_record   LIKE solisti1 OCCURS 0 WITH HEADER LINE,
      i_xstring  TYPE xstring,
* Objects to send mail.
      i_objpack  LIKE sopcklsti1 OCCURS 0 WITH HEADER LINE,
      i_objtxt   LIKE solisti1 OCCURS 0 WITH HEADER LINE,
      i_objbin   LIKE solix     OCCURS 0 WITH HEADER LINE,
      i_reclist  LIKE somlreci1 OCCURS 0 WITH HEADER LINE,
* Work Area declarations
      wa_objhead TYPE soli_tab,
      w_ctrlop   TYPE ssfctrlop,
      w_compop   TYPE ssfcompop,
      w_return   TYPE ssfcrescl,
      wa_buffer  TYPE string,
* Variables declarations
      v_form_name TYPE rs38l_fnam,
      v_len_in    LIKE sood-objlen.

*-----*
* Selection Screen
*-----*
SELECTION-SCREEN BEGIN OF BLOCK b1 WITH FRAME TITLE text-t01.
PARAMETERS:      p_sform TYPE tdsfname OBLIGATORY.
SELECT-OPTIONS:  s_email FOR adr6-smtp_addr NO INTERVALS OBLIGATORY.
SELECTION-SCREEN END OF BLOCK b1.

*-----*
END-OF-SELECTION.
*-----*
* Call your Smart
  PERFORM smartform_driver_call.

* Convert Smartform data to Hexa
  PERFORM convert_otf_to_xstring.

```

Shares

```
* Convert Hexa to Binary
PERFORM str_to_binary.

* Send the email using cl_bcs class
PERFORM send_multiple_user.

*-----*
* Sub Routines
*-----*
FORM smartform_driver_call .

* Call you smartform
CALL FUNCTION 'SSF_FUNCTION_MODULE_NAME'
  EXPORTING
    formname          = p_sform
  IMPORTING
    fm_name           = v_form_name
  EXCEPTIONS
    no_form            = 1
    no_function_module = 2
    OTHERS             = 3.

IF sy-subrc <> 0.
  MESSAGE ID sy-msgid TYPE sy-msgty NUMBER sy-msgno
  WITH sy-msgv1 sy-msgv2 sy-msgv3 sy-msgv4.
ENDIF.

* Set the control parameter
w_ctrlop-getotf = abap_true.
w_ctrlop-no_dialog = abap_true.
w_compop-tdnoprev = abap_true.
w_ctrlop-preview = space.
w_compop-tddest = 'LOCL'.

* Call Smartform
CALL FUNCTION v_form_name
  EXPORTING
    control_parameters = w_ctrlop
    output_options      = w_compop
```

Shares

```
user_settings      = abap_true
IMPORTING
  job_output_info  = w_return " This will have all output
EXCEPTIONS
  formatting_error  = 1
  internal_error    = 2
  send_error        = 3
  user_canceled     = 4
  OTHERS            = 5.

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. " SMARTFORM_DRIVER_CALL

FORM str_to_binary .

* Convert Hexa String to Binary format
CALL FUNCTION 'SCMS_XSTRING_TO_BINARY'
EXPORTING
  buffer      = i_xstring
TABLES
  binary_tab = i_objbin[].
* Sy-subrc check not required.

ENDFORM. " STR_TO_BINARY

FORM send_mail USING in_mailid.

DATA: salutation TYPE string.
DATA: body TYPE string.
DATA: footer TYPE string.

DATA: lo_send_request TYPE REF TO cl_bcs,
      lo_document     TYPE REF TO cl_document_bcs,
      lo_sender       TYPE REF TO if_sender_bcs,
      lo_recipient     TYPE REF TO if_recipient_bcs VALUE IS INITIAL, lt_message_body TYPE bcs
```

Shares

```
lx_document_bcs TYPE REF TO cx_document_bcs,  
lv_sent_to_all TYPE os_boolean.  
  
"create send request  
lo_send_request = cl_bcs=>create_persistent( ).  
  
"create message body and subject  
salutation = 'Dear Sir/Madam,'.  
APPEND salutation TO lt_message_body.  
APPEND INITIAL LINE TO lt_message_body.  
  
body = 'Please find the attached the Smartform in PDF format.'.  
APPEND body TO lt_message_body.  
APPEND INITIAL LINE TO lt_message_body.  
  
footer = 'With Regards,'.  
APPEND footer TO lt_message_body.  
footer = 'www.sapyard.com.'.  
APPEND footer TO lt_message_body.  
"put your text into the document  
lo_document = cl_document_bcs=>create_document(  
i_type = 'RAW'  
i_text = lt_message_body  
i_subject = 'Smartform in PDF' ).  
  
*DATA: l_size TYPE sood-objlen. " Size of Attachment  
*l_size = l_lines * 255.  
TRY.  
    lo_document->add_attachment(  
        EXPORTING  
        i_attachment_type = 'PDF'  
        i_attachment_subject = 'Your Smartform'  
        i_att_content_hex = i_objbin[] ).  
    CATCH cx_document_bcs INTO lx_document_bcs.  
ENDTRY.  
  
* Add attachment  
* Pass the document to send request  
lo_send_request->set_document( lo_document ).
```


Shares

```
"Create sender
lo_sender = cl_sapuser_bcs=>create( sy-uname ).

"Set sender
lo_send_request->set_sender( lo_sender ).

"Create recipient
lo_recipient = cl_cam_address_bcs=>create_internet_address( in_mailid ).

*Set recipient
lo_send_request->add_recipient(
    EXPORTING
    i_recipient = lo_recipient
    i_express = abap_true
    ).

lo_send_request->add_recipient( lo_recipient ).

* Send email
lo_send_request->send(
    EXPORTING
    i_with_error_screen = abap_true
    RECEIVING
    result = lv_sent_to_all ).

CONCATENATE 'Email sent to' in_mailid INTO data(lv_msg) SEPARATED BY space.
WRITE:/ lv_msg COLOR COL_POSITIVE.
SKIP.

* Commit Work to send the email
COMMIT WORK.

ENDFORM.

FORM send_multiple_user.

    DATA: in_mailid TYPE ad_smtpadr.

* Begin of sending email to multiple users
```

```

* If business want email to be sent to all users at one time, it can be done

* For now we do not want to send 1 email to multiple users
* Mail has to be sent one email at a time
  LOOP AT s_email.
    CLEAR in_mailid.
    in_mailid = s_email-low.
    PERFORM send_mail USING in_mailid .
  ENDLOOP.

ENDFORM. " SEND_MULTIPLE_USER


FORM convert_otf_to_xstring .

* Get Output Text Format (OTF)
  i_otf[] = w_return-otfdata[].

* Import Binary file and filesize
  CALL FUNCTION 'CONVERT_OTF'
    EXPORTING
      format                = 'PDF'
      max_linewidth         = 132
    IMPORTING
      bin_filesize          = v_len_in
      bin_file              = i_xstring " This is NOT Binary. This is Hexa
    TABLES
      otf                   = i_otf
      lines                 = i_tline
    EXCEPTIONS
      err_max_linewidth     = 1
      err_format            = 2
      err_conv_not_possible = 3
      OTHERS                = 4.

* Sy-subrc check not checked

ENDFORM.

```

Are you new to SAPUI5? [Check the issues faced by fresher in SAPUI5.](#)

For our test, we created a simple smartform with just a logo and one text line.




Hit the test icon to **test the smartform stand alone.**

You would see the function module linked to this smartform. Hit the test icon again. Now you see the parameters of the function, hit test icon again.


Shares

Function Builder: Initial Screen

 Reassign...

Function Module

Test Function Module: Initial Screen

 Debugging

Test for function group
Function module
Uppercase/Lowercase ☐

Import parameters	Value
ARCHIVE_INDEX	00/00/0000
ARCHIVE_INDEX_TAB	0 Entries
ARCHIVE_PARAMETERS	
CONTROL_PARAMETERS	
MAIL_APPL_OBJ	
MAIL_RECIPIENT	
MAIL_SENDER	
OUTPUT_OPTIONS	<Initial>
USER_SETTINGS	X

Print:

OutputDevice

Frontend Printer

Page selection

Spool Request

Name

Title

Authorization

Spool Control

☐ Print Now

☐ Delete After Output

☐ New Spool Request

☐ Close Spool Request

Spool Retention Day(s)

Storage Mode

Number of Copies

Number

☐ Group (1-1-1,2-2-2,3-3-3,...)

Cover Page Settings

SAP cover page

Recipient

Department

Shares



This is our smartform output.

Are you HANA ABAP ready? [Check how you can write your first Program in HANA.](#)

Now, our program should **attach this smartform output as PDF and email it to multiple users.**

Let's execute our program. Give the smartform name and the email ids. For the test, we are sending just to two users. You might enter as many users you want.

Shares

Send Smartform as PDF attachment to mulitple

Input Parameters

Smartform name	z
Recipients Email	mail@sapyard.com

Multiple Selection for Recipients Email

Select Single Values (1) | Select Ranges | Exclude Single V

O. Single value

<input type="checkbox"/>	mail@sapyard.com	<input type="checkbox"/>
<input checked="" type="checkbox"/>	mailsapyard@gmail.com	<input checked="" type="checkbox"/>

Execute and you would get this message (if you are testing our program).

Send Smartform as PDF attachment to mulitple

Send Smartform as PDF attachment to mulitple

Email sent to mail@sapyard.com

Email sent to mailsapyard@gmail.com

Go to t-code **SOST** to see if email really went out. Yes!! it did.

SAPconnect: Send Requests from 07/24/2017 to 07/25/2017

Refresh

Period | **Send Status** | Sender | Options | Attribs.

Send date: to:
 Sent at: to:
 Maximum No. of Hits:

Refresh









All Send Requests

Status	Send Met...	Doc. Title	Sender	Recipient	Send Date	Send Ti...	M...
	By E-Mail	Smartform in PDF		mailsapyard@gmail.com	07/25/2017	10:24:10	<u>6..</u>
	By E-Mail	Smartform in PDF		mail@sapyard.com	07/25/2017	10:24:10	<u>6..</u>

Let us check what this email has for us.

Shares

Display Document: Smartform in PDF

 Reply... Reply w/Reference... New message



Doc. contents

Attributes

Recipient list

Attachments



Smartform in PDF

Created  

Dear Sir/Madam,

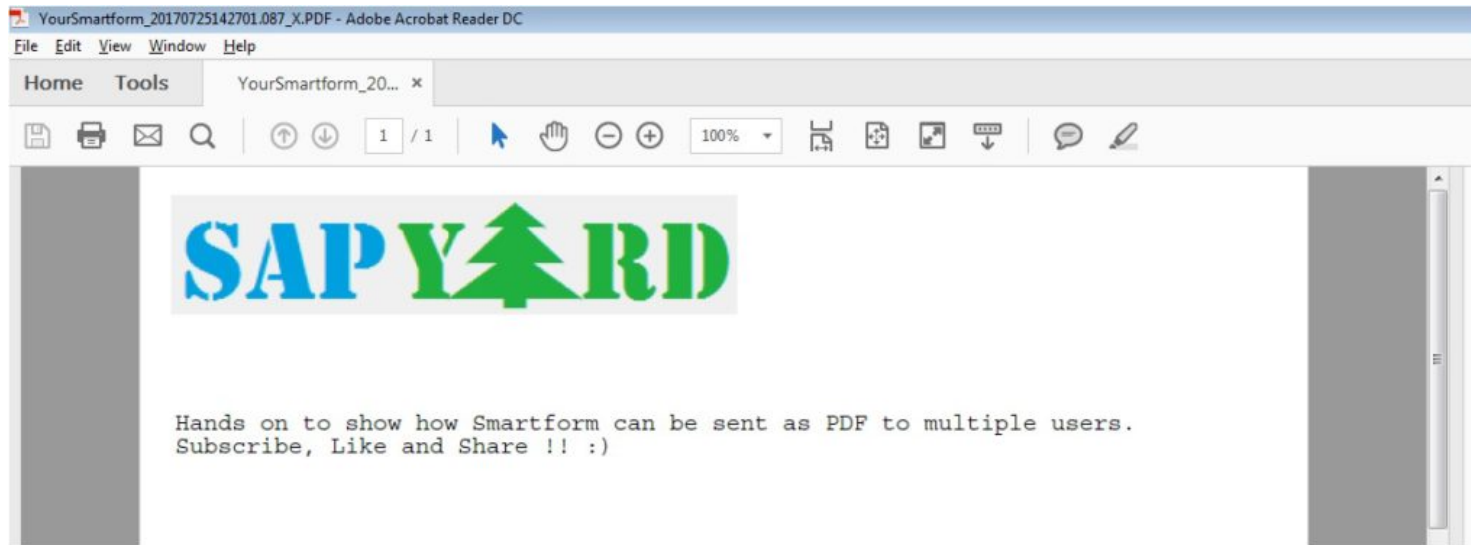
Please find the Smartform in PDF format.

With Regards,
www.sapyard.com.

 
Your Smartform

Cool, we have the email body and an attachment.

Let us check if the attachment is the same smartform we checked above in stand alone mode.



Shares

Bingo!! It is in deed the smartform in the PDF format.

*I am planning to refer to this page for my future use case where I have a similar requirement.
What do you think?*

I want to hear now from you now. Do you have a better approach?

If you have or not, in either case, **please leave your comments** and let us start a healthy debate.

Amitajain

Amita is a young and dynamic SAP Consultant. Apart from SAP ABAP, she is skilled in Microsoft Excel, SQL, PHP, C++, and Android. She has Master of Computer Applications (MCA) degree and currently focussed on SAPUI5 and Fiori Developments. Find more about Amita on [LinkedIn](#).

Shares

1000000000

1000000000

1000000000

1000000000

1000000000

1000000000

1000000000