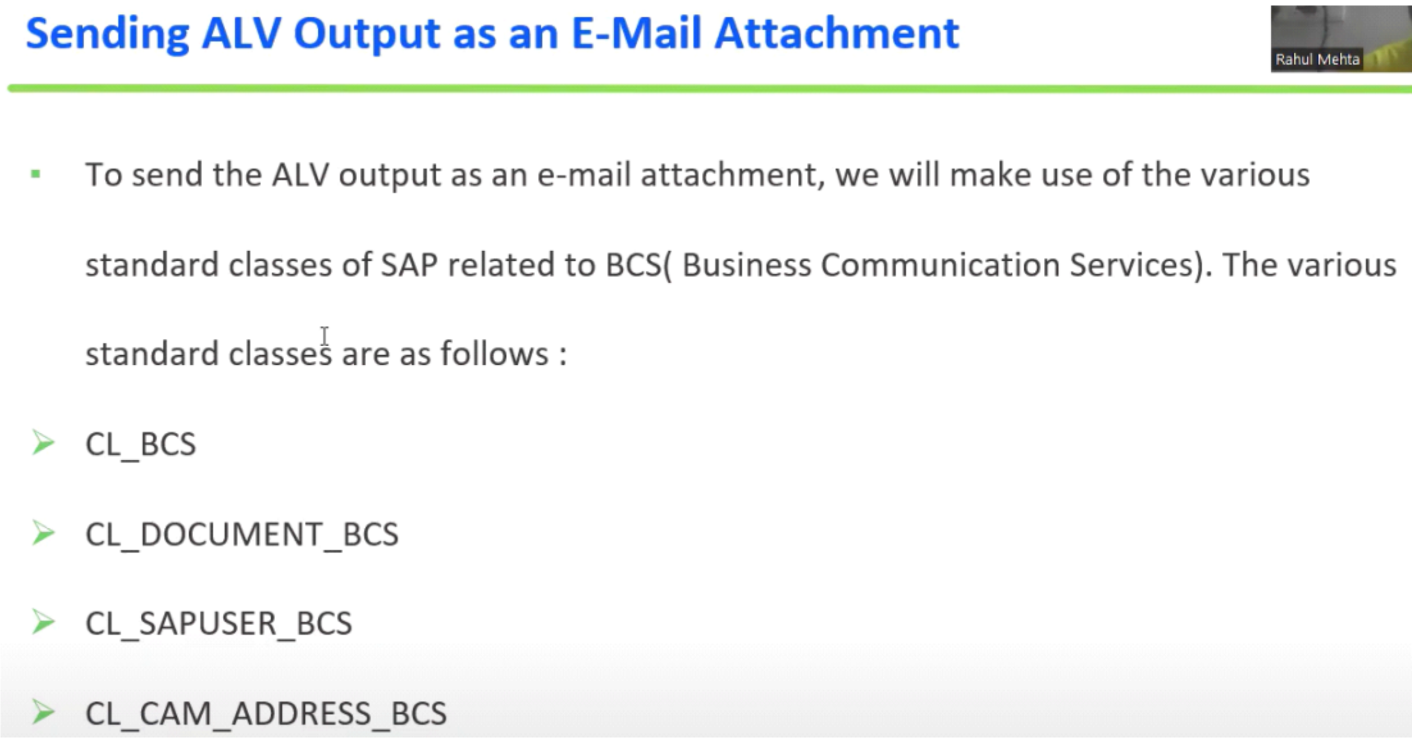
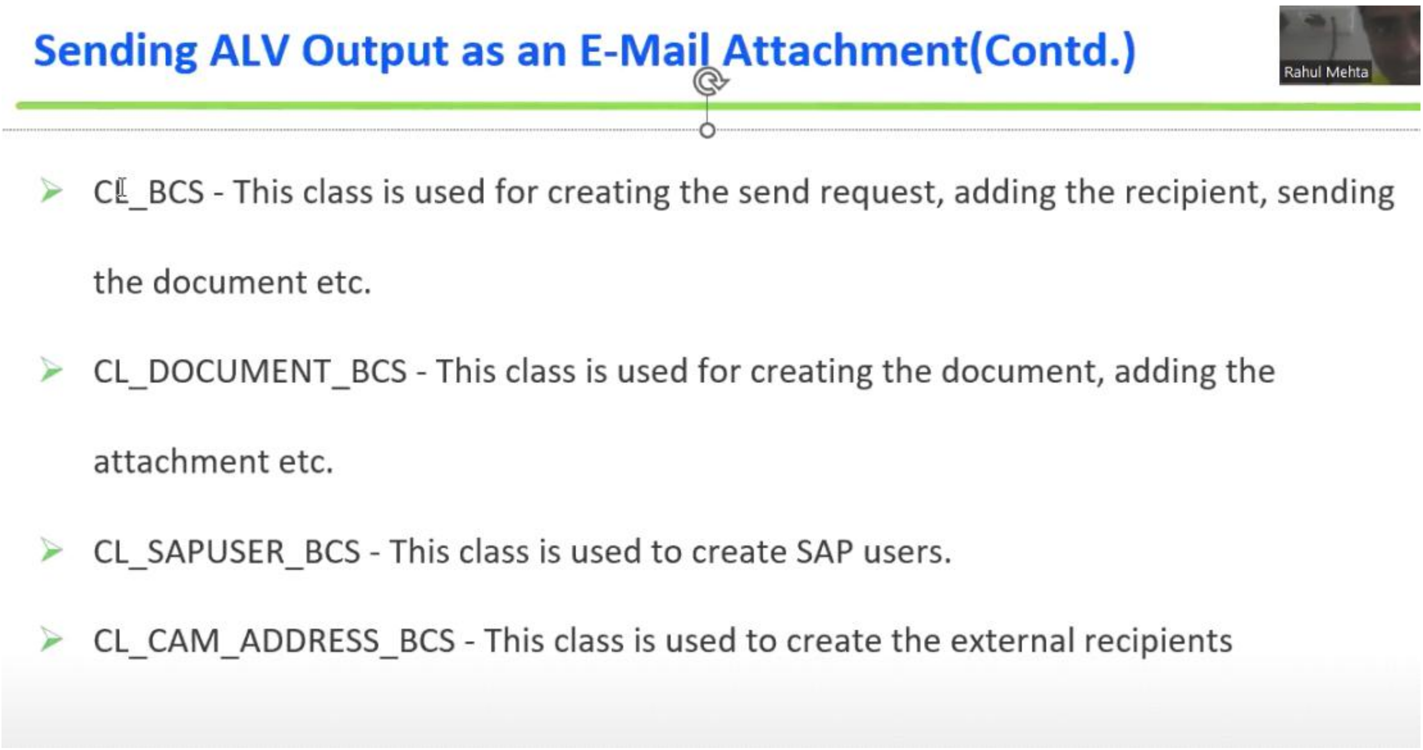
Send Mail

25 September 2025 15:49



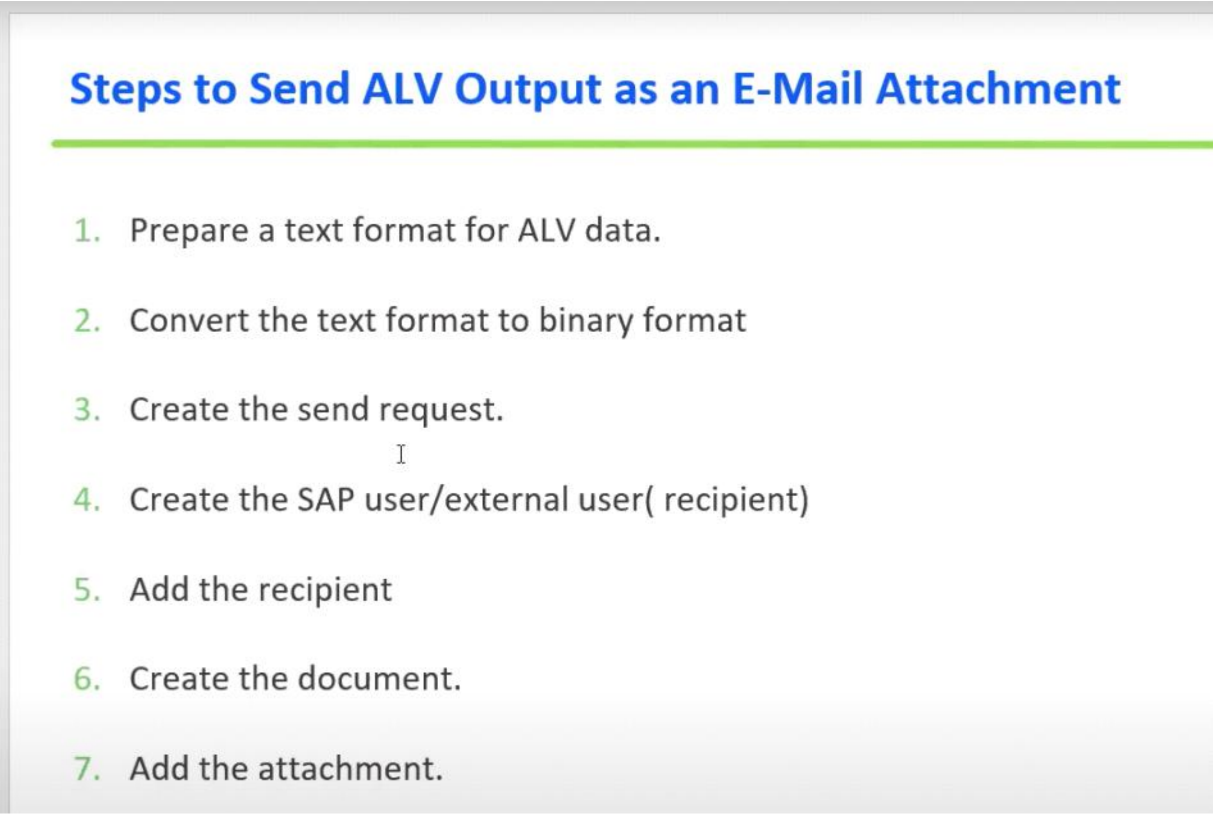
Sending ALV Output as an E-Mail Attachment

- To send the ALV output as an e-mail attachment, we will make use of the various standard classes of SAP related to BCS(Business Communication Services). The various standard classes are as follows :
- CL_BCS
- CL_DOCUMENT_BCS
- CL_SAPUSER_BCS
- CL_CAM_ADDRESS_BCS



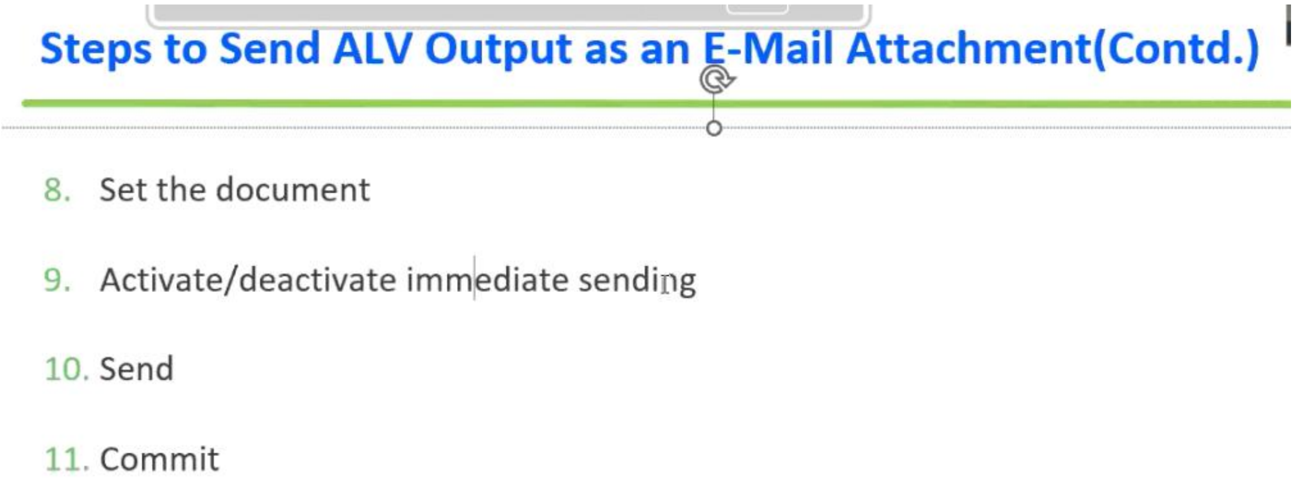
Sending ALV Output as an E-Mail Attachment(Contd.)

- CL_BCS - This class is used for creating the send request, adding the recipient, sending the document etc.
- CL_DOCUMENT_BCS - This class is used for creating the document, adding the attachment etc.
- CL_SAPUSER_BCS - This class is used to create SAP users.
- CL_CAM_ADDRESS_BCS - This class is used to create the external recipients



Steps to Send ALV Output as an E-Mail Attachment

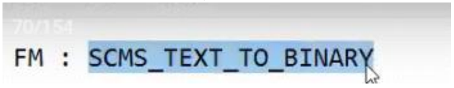
1. Prepare a text format for ALV data.
2. Convert the text format to binary format
3. Create the send request.
4. Create the SAP user/external user(recipient)
5. Add the recipient
6. Create the document.
7. Add the attachment.



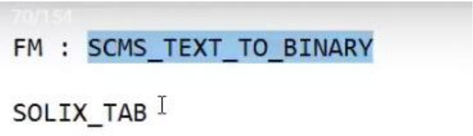
Steps to Send ALV Output as an E-Mail Attachment(Contd.)

8. Set the document
9. Activate/deactivate immediate sending
10. Send
11. Commit

To convert to binary format

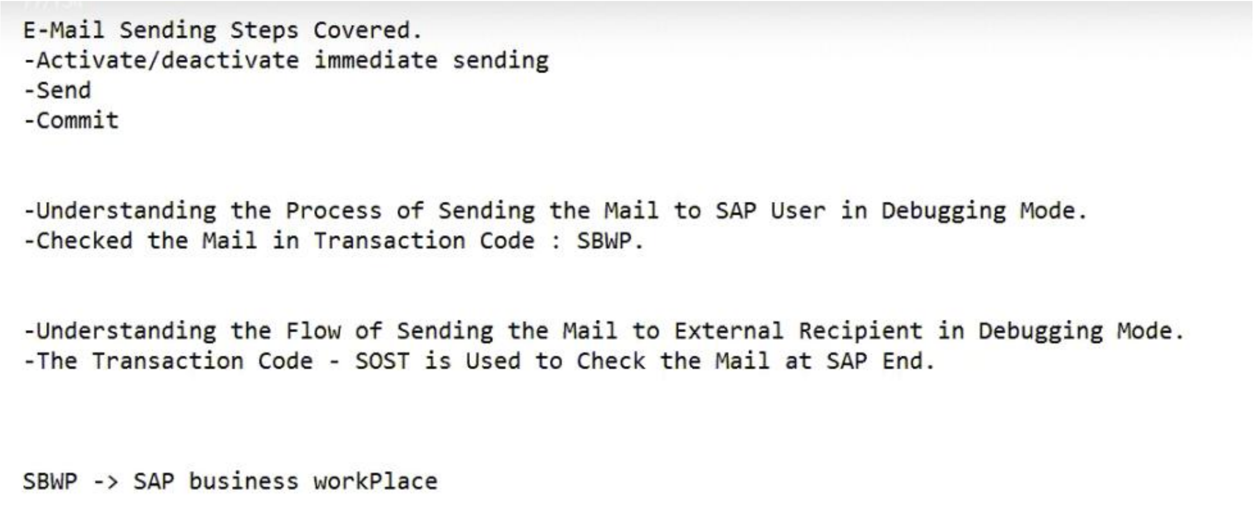


```
FM : SCMS_TEXT_TO_BINARY
```



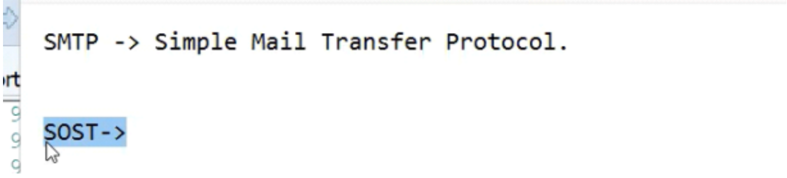
```
FM : SCMS_TEXT_TO_BINARY

SOLIX_TAB
```

ZALV_SEND_EMAIL1_640



```
E-Mail Sending Steps Covered.
-Activate/deactivate immediate sending
-Send
-Commit


-Understanding the Process of Sending the Mail to SAP User in Debugging Mode.
-Checked the Mail in Transaction Code : SBWP.


-Understanding the Flow of Sending the Mail to External Recipient in Debugging Mode.
-The Transaction Code - SOST is Used to Check the Mail at SAP End.



SBWP -> SAP business workPlace
```

For External Email check



```
SMTP -> Simple Mail Transfer Protocol.


SOST->
```

```
*&---------------------------------------------------------------------*
*& Report ZALV_SEND_EMAIL1_640
*&---------------------------------------------------------------------*
*&
*&---------------------------------------------------------------------*
REPORT zalv_send_email1_640.

PARAMETERS : p_erdat TYPE  erdat DEFAULT sy-datum.          "'20250207'
PARAMETERS : p_email AS CHECKBOX DEFAULT 'X'.

TYPES : BEGIN OF lty_vbak,
          vbeln TYPE vbeln_va,
          ernam TYPE ernam,
          netwr TYPE netwr_ak,
        END OF lty_vbak.
TYPES : BEGIN OF lty_data,
          data(100) TYPE c,
        END OF lty_data.

DATA : lt_vbak TYPE TABLE OF lty_vbak.

DATA : lt_data TYPE TABLE OF lty_data,
       ls_data TYPE lty_data.

DATA : lt_binary_data TYPE solix_tab.
DATA : lo_bcs TYPE REF TO cl_bcs.
DATA : lo_sapuser TYPE REF TO cl_sapuser_bcs.
DATA : lo_external_user TYPE REF TO cl_cam_address_bcs.
DATA : lo_external_user1 TYPE REF TO cl_cam_address_bcs.
DATA : lv_subject TYPE so_obj_des.
DATA : lt_text TYPE soli_tab.
DATA : lo_document           TYPE REF TO cl_document_bcs,
       lv_attachment_subject TYPE so_obj_des,
       lv_result             TYPE boolean.

DATA : lt_fieldcat TYPE slis_t_fieldcat_alv.

SELECT vbeln ernam netwr
  FROM vbak
  INTO TABLE lt_vbak
  WHERE erdat = p_erdat.

IF p_email = 'X'.
```

```
* 1) Prepare a text format for  ALV Data.

  CONCATENATE 'Sales Document Number' 'User Name ' 'Amount' INTO ls_data-
data SEPARATED BY ' ! '.
  APPEND ls_data TO lt_data.
  CLEAR ls_data.

  LOOP AT lt_vbak INTO DATA(ls_vbak).
    ls_data-data = |{ ls_vbak-vbeln } ! { ls_vbak-ernam } ! { ls_vbak-
netwr }|.
    APPEND ls_data TO lt_data.
    CLEAR ls_data.
  ENDLOOP.

* 2) convert text format to binary format

  CALL FUNCTION 'SCMS_TEXT_TO_BINARY'
*   EXPORTING
*     FIRST_LINE            = 0
*     LAST_LINE             = 0
*     APPEND_TO_TABLE       = ' '
*     MIMETYPE              = ' '
*     ENCODING              =
*   IMPORTING
*     OUTPUT_LENGTH         =
    TABLES
      text_tab   = lt_data
      binary_tab = lt_binary_data
    EXCEPTIONS
      failed     = 1
      OTHERS     = 2.

* 3) create a send request

  TRY.
      CALL METHOD cl_bcs=>create_persistent
        RECEIVING
          result = lo_bcs.
    CATCH cx_send_req_bcs.
  ENDTRY.

* 4) create a sap user / external user (recipient).

  TRY.
      CALL METHOD cl_sapuser_bcs=>create
        EXPORTING
          i_user = 'TECH_54'
        RECEIVING
          result = lo_sapuser.
    CATCH cx_address_bcs.
  ENDTRY.

* 5) ADD THE RECIPIENT.

  TRY.
      CALL METHOD lo_bcs->add_recipient
        EXPORTING
          i_recipient  = lo_sapuser
*         i_express    =
*         i_copy       =
*         i_blind_copy =
*         i_no_forward =
        .
    CATCH cx_send_req_bcs.
  ENDTRY.

  TRY.
      CALL METHOD cl_cam_address_bcs=>create_internet_address
        EXPORTING
          i_address_string = 'vinodreddyk@gmail.com'
*         i_address_name   =
*         i_incl_sapuser   =
        RECEIVING
          result           = lo_external_user.
    CATCH cx_address_bcs.
  ENDTRY.

  TRY.
      CALL METHOD lo_bcs->add_recipient
        EXPORTING
          i_recipient = lo_external_user
*         i_express   =
*         i_copy      =
*         i_blind_copy =
*         i_no_forward =
        .
    CATCH cx_send_req_bcs.
  ENDTRY.

*  for adding more mails in cc and bcc.

  TRY.
      CALL METHOD cl_cam_address_bcs=>create_internet_address
        EXPORTING
          i_address_string = 'vinod@gmail.com'
*         i_address_name   =
*         i_incl_sapuser   =
        RECEIVING
          result           = lo_external_user1.
    CATCH cx_address_bcs.
  ENDTRY.

  TRY.
      CALL METHOD lo_bcs->add_recipient
        EXPORTING
          i_recipient  = lo_external_user1
*         i_express    =
          i_copy       = 'X'
          i_blind_copy = 'X'
*         i_no_forward =
        .
    CATCH cx_send_req_bcs.
  ENDTRY.

* 6) create the document

  DESCRIBE TABLE lt_vbak LINES DATA(lv_lines).
  lv_subject = |{ p_erdat+6(2) }:{ p_erdat+4(2) }:{ p_erdat+
0(4) } Number Of Sales Order : { lv_lines }|.

  lt_text = VALUE #( ( line = 'Dear Sir, ' )
                     ( line = 'Below is the Attached Document ')
                     ( line = 'Thanks & Regards' )
                     ( line = 'Vinod' ) ).

  TRY.
      CALL METHOD cl_document_bcs=>create_document
        EXPORTING
          i_type    = 'RAW'
          i_subject = lv_subject
*         i_length  =
*         i_language      = SPACE
*         i_importance    =
*         i_sensitivity   =
          i_text    = lt_text
*         i_hex     =
*         i_header  =
*         i_sender  =
*         iv_vsi_profile  =
*         iv_vsi_scan_off =
        RECEIVING
          result    = lo_document.
    CATCH cx_document_bcs.
  ENDTRY.

* 7) Add the attachment.

  CONCATENATE 'Sales Orders ' '_ ' p_erdat+6(2) ':' p_erdat+
4(2) ':' p_erdat+0(4) INTO lv_attachment_subject.

  TRY.
```

```
      CALL METHOD lo_document->add_attachment
        EXPORTING
          i_attachment_type    = 'XLS'
          i_attachment_subject = lv_attachment_subject
*         i_attachment_size    =
*         i_attachment_language = SPACE
*         i_att_content_text   =
          i_att_content_hex    = lt_binary_data
*         i_attachment_header  =
*         iv_vsi_profile       =
*         iv_vsi_scan_off      =
*         i_attachment_filename =
        .
    CATCH cx_document_bcs.
  ENDTRY.

* 8) set the document.

  TRY.
      CALL METHOD lo_bcs->set_document
        EXPORTING
          i_document = lo_document.
    CATCH cx_send_req_bcs.
  ENDTRY.

* 9) Activate or Deactivate immediate sending

  TRY.
      CALL METHOD lo_bcs->set_send_immediately
        EXPORTING
          i_send_immediately = 'X'.
    CATCH cx_send_req_bcs.
  ENDTRY.

* 10) send

  TRY.
      CALL METHOD lo_bcs->send
*  EXPORTING
*    i_with_error_screen = SPACE
        RECEIVING
          result = lv_result.
    CATCH cx_send_req_bcs.
  ENDTRY.

* 11) Commit only for external email not for sap users.( within sap do not
require commit work ).

  IF lv_result IS NOT INITIAL.
    COMMIT WORK.
  ENDIF.


ENDIF.


lt_fieldcat = VALUE #( ( col_pos = 1 fieldname = 'VBELN' seltext_l = ' Sale
s Document Number' )
                        ( col_pos = 2 fieldname = 'ERNAM' seltext_l = ' Use
r Name' )
                        ( col_pos = 3 fieldname = 'NETWR' seltext_l = ' Amo
unt' ) ).

CALL FUNCTION 'REUSE_ALV_GRID_DISPLAY'
  EXPORTING
    it_fieldcat    = lt_fieldcat
  TABLES
    t_outtab       = lt_vbak
  EXCEPTIONS
    program_error  = 1
    OTHERS         = 2.
```

Send mail 2

26 September 2025 15:40

```
BADI in me21n


METHOD if_ex_me_process_po_cust~process_item.

    IF sy-uname = 'TECH_54'.

      DATA : lt_text1  TYPE soli_tab.

      CALL METHOD im_item->get_previous_data
        IMPORTING
          ex_data = DATA(ls_old)
        EXCEPTIONS
          no_data = 1
          OTHERS  = 2.
      IF sy-subrc <> 0.
*       Implement suitable error handling here
      ENDIF.

      CALL METHOD im_item->get_data
        RECEIVING
          re_data = DATA(ls_new).

* 1) Prepare a text format for  ALV Data.
*      lt_data = VALUE #( ( data = |Material Number ! Plant ! Purchasing Document Number !
 Old Quantity ! New Quantity ! User Name ! UTC Time Stamp | )
*            ( data = |{ ls_new-matnr } ! { ls_new-werks } ! { ls_new-ebeln } ! { ls_old-
menge } ! { ls_new-menge } ! { sy-uname } ! { sy-dATUM }{ sy-uzeit }| ) ).
* 2) convert text format to binary format
      " Build HTML body with table

      CLEAR lt_text1.
      APPEND '<html><body>' TO lt_text1.
      APPEND '<p>Dear Sir,</p>' TO lt_text1.
      APPEND '<p>Below is the PO change details :</p>' TO lt_text1.

      APPEND '<table border="1" cellspacing="0" cellpadding="5">' TO lt_text1.
      APPEND '<tr> <th>Purchasing Document</th><th>Material Number</th><th>Plant</th><th>
Old Quantity</th><th>New Quantity</th><th>Unit</th><th>User Name</th><th>Change Date</th>
</tr>' TO lt_text1.

      APPEND |<tr><td>{ ls_new-ebeln }</td><td>{ ls_new-matnr }</td><td>{ ls_new-werks }
</td><td>{ ls_old-menge }</td><td>{ ls_new-menge }</td><td>{ ls_new-meins }</td><td>{ sy-
uname }</td><td>{ sy-datum+6(2) }:{ sy-datum+4(2) }:{ sy-datum+0(4) } {
                sy-uzeit+0(2) }:{ sy-uzeit+2(2) }:{ sy-uzeit+4(2) }</td>
</tr>| TO  lt_text1.

      APPEND '</table>' TO lt_text1.
      APPEND '<br/><p>Thanks & Regards,<br/>Vinod</p>' TO lt_text1.
      APPEND '</body></html>' TO lt_text1.


* 3) create a send request
      TRY.
          CALL METHOD cl_bcs=>create_persistent
            RECEIVING
              result = DATA(lo_bcs).
        CATCH cx_send_req_bcs.
      ENDTRY.

*   4) create a sap user / external user (recipient).
```

```
      TRY.
          CALL METHOD cl_sapuser_bcs=>create
            EXPORTING
              i_user = 'TECH_54'
            RECEIVING
              result = DATA(lo_sapuser).
        CATCH cx_address_bcs.
      ENDTRY.

*   5) ADD THE RECIPIENT.
      TRY.
          CALL METHOD lo_bcs->add_recipient
            EXPORTING
              i_recipient = lo_sapuser.
        CATCH cx_send_req_bcs.
      ENDTRY.

*       for Exter Mail connect

      TRY.
          CALL METHOD cl_cam_address_bcs=>create_internet_address
            EXPORTING
              i_address_string = 'vinod.k@ladera.com'
              i_address_name   = 'K Vinod'
*             i_incl_sapuser   = 'X'
            RECEIVING
              result           = DATA(lo_external_user).
        CATCH cx_address_bcs.
      ENDTRY.

      TRY.
          CALL METHOD lo_bcs->add_recipient
            EXPORTING
              i_recipient = lo_external_user.
        CATCH cx_send_req_bcs.
      ENDTRY.

* 6) create the document
      TRY.
          CALL METHOD cl_document_bcs=>create_document
            EXPORTING
              i_type    = 'HTM'
              i_subject = ' Changed PO Details'
              i_text    = lt_text1
            RECEIVING
              result    = DATA(lo_document).
        CATCH cx_document_bcs.
      ENDTRY.

* 7) Add the attachment.
*       TRY.
*           CALL METHOD lo_document->add_attachment
*             EXPORTING
*               i_attachment_type    = 'XLS'
*               i_attachment_subject = |created po on { sy-datum }|
**              i_attachment_size    =
**              i_attachment_language = SPACE
**              i_att_content_text   =
*               i_att_content_hex    = lt_binary
**              i_attachment_header  =
**              iv_vsi_profile       =
**              iv_vsi_scan_off      =
**              i_attachment_filename =
*             .
*         CATCH cx_document_bcs.
*       ENDTRY.
* 8) set the document.
      TRY.
          CALL METHOD lo_bcs->set_document
```

```
            EXPORTING
              i_document = lo_document.
        CATCH cx_send_req_bcs.
      ENDTRY.

* 9) Activate or Deactivate immediate sending
      TRY.
          CALL METHOD lo_bcs->set_send_immediately
            EXPORTING
              i_send_immediately = 'X'.
        CATCH cx_send_req_bcs.
      ENDTRY.

* 10) send
      TRY.
          CALL METHOD lo_bcs->send
*  EXPORTING
*    i_with_error_screen = SPACE
            RECEIVING
              result = DATA(lv_result).
        CATCH cx_send_req_bcs.
      ENDTRY.

* 11) Commit only for external email not for sap users.( within sap do not require commit
work ).
      IF lv_result IS NOT INITIAL.
        COMMIT WORK.
      ENDIF.


    ENDIF.

  ENDMETHOD.
```

Send mail 3

26 September 2025 15:41

```
  METHOD if_ex_me_process_po_cust~process_item.

    IF sy-uname = 'TECH_54'.

      DATA : lt_text1  TYPE soli_tab.

      CALL METHOD im_item->get_previous_data
        IMPORTING
          ex_data = DATA(ls_old)
        EXCEPTIONS
          no_data = 1
          OTHERS  = 2.
      IF sy-subrc <> 0.
*       Implement suitable error handling here
      ENDIF.

      CALL METHOD im_item->get_data
        RECEIVING
          re_data = DATA(ls_new).

* 1) Prepare a text format for  ALV Data.

* 2) convert text format to binary format

      " Build HTML body with table

      CLEAR lt_text1.
      APPEND '<html><body>' TO lt_text1.
      APPEND '<p>Dear Sir,</p>' TO lt_text1.
      APPEND '<p>Below is the PO change details :</p>' TO lt_text1.

      APPEND '<table border="1" cellspacing="0" cellpadding="5">' TO lt_tex
t1.
      APPEND '<tr> <th>Purchasing Document</th><th>Material Number</th><th>
Plant</th><th>Old Quantity</th><th>New Quantity</th><th>Unit</th><th>
User Name</th><th>Change Date</th></tr>' TO lt_text1.

      APPEND |<tr><td>{ ls_new-ebeln }</td><td>{ ls_new-matnr }</td><td>
{ ls_new-werks }</td><td>{ ls_old-menge }</td><td>{ ls_new-menge }</td><td>
{ ls_new-meins }</td><td>{ sy-uname }</td><td>{ sy-datum+6(2) }:{ sy-datum+
4(2) }:{ sy-datum+0(4) } {
                sy-uzeit+0(2) }:{ sy-uzeit+2(2) }:{ sy-uzeit+4(2) }</td>
</tr>| TO  lt_text1.

      APPEND '</table>' TO lt_text1.
      APPEND '<br/><p>Thanks & Regards,<br/>Vinod</p>' TO lt_text1.
      APPEND '</body></html>' TO lt_text1.


* 3) create a send request
      TRY.
          CALL METHOD cl_bcs=>create_persistent
            RECEIVING
              result = DATA(lo_bcs).
        CATCH cx_send_req_bcs.
      ENDTRY.

*  4) create a sap user / external user (recipient).
*  5) ADD THE RECIPIENT.
*        for Exter Mail connect
      TRY.
          CALL METHOD cl_cam_address_bcs=>create_internet_address
            EXPORTING
              i_address_string = 'vinod.k@ladera.com'
              i_address_name   = 'K Vinod'
              i_incl_sapuser   = 'X'
            RECEIVING
              result           = DATA(lo_external_user).
        CATCH cx_address_bcs.
      ENDTRY.

      TRY.
          CALL METHOD lo_bcs->add_recipient
            EXPORTING
              i_recipient = lo_external_user.
        CATCH cx_send_req_bcs.
      ENDTRY.

* 6) create the document
      TRY.
          CALL METHOD cl_document_bcs=>create_document
            EXPORTING
              i_type    = 'HTM'
              i_subject = ' Changed PO Details'
              i_text    = lt_text1
            RECEIVING
              result    = DATA(lo_document).
        CATCH cx_document_bcs.
      ENDTRY.

* 7) Add the attachment.

* 8) set the document.
      TRY.
          CALL METHOD lo_bcs->set_document
            EXPORTING
              i_document = lo_document.
        CATCH cx_send_req_bcs.
      ENDTRY.

* 9) Activate or Deactivate immediate sending
      TRY.
          CALL METHOD lo_bcs->set_send_immediately
            EXPORTING
              i_send_immediately = 'X'.
        CATCH cx_send_req_bcs.
      ENDTRY.

* 10) send
      TRY.
          CALL METHOD lo_bcs->send
*  EXPORTING
*    i_with_error_screen = SPACE
            RECEIVING
              result = DATA(lv_result).
        CATCH cx_send_req_bcs.
      ENDTRY.

* 11) Commit only for external email not for sap users.( within sap do not
require commit work ).
      IF lv_result IS NOT INITIAL.
        COMMIT WORK.
      ENDIF.
```

```
  ENDIF.

ENDMETHOD.
```

Send Mail 4

08 October 2025 10:24

```
*&---------------------------------------------------------------------*
*& Report ZALV_SEND_EMAIL1_640
*&---------------------------------------------------------------------*
*&
*&---------------------------------------------------------------------*
REPORT zalv_send_email2_640.

PARAMETERS : p_r1 RADIOBUTTON GROUP r1,
             p_r2 RADIOBUTTON GROUP r1.

DATA : lt_text1 TYPE soli_tab,
       lt_text2 TYPE soli_tab.
DATA : lo_document TYPE REF TO cl_document_bcs.

SELECT * FROM zord_h_12 INTO TABLE @DATA(lt_header).
SELECT * FROM zord_i_12 INTO TABLE @DATA(lt_item).

SORT lt_header BY ono.

* 1) Prepare a text format for  ALV Data.

* 2) convert text format to binary format

" Build HTML body Header table

CLEAR lt_text1.
APPEND '<html><body>' TO lt_text1.
APPEND '<p>Dear Sir,</p>' TO lt_text1.
APPEND '<p>Below are the Header details :</p>' TO lt_text1.

APPEND '<table border="1" cellspacing="0" cellpadding="5">' TO lt_text1.
APPEND '<tr> <th>Order Number</th><th>Order Date</th><th>Payment Mode</th>
<th>Amount</th><th>Currency</th></tr>' TO lt_text1.
LOOP AT lt_header INTO DATA(ls_header).
  APPEND |<tr><td>{ ls_header-ono }</td><td>{ ls_header-odate }</td><td>
{ ls_header-pm }</td><td>{ ls_header-ta }</td><td>{ ls_header-curr }</td>
</tr>| TO  lt_text1.
ENDLOOP.
APPEND '</table>' TO lt_text1.
APPEND '<br/><p>Thanks & Regards,<br/> Vinod</p>' TO lt_text1.
APPEND '</body></html>' TO lt_text1.


" Build HTML body Item table
CLEAR lt_text2.
APPEND '<html><body>' TO lt_text2.
APPEND '<p>Dear Sir,</p>' TO lt_text2.
APPEND '<p>Below are the Item details :</p>' TO lt_text2.

APPEND '<table border="1" cellspacing="0" cellpadding="5">' TO lt_text2.
APPEND '<tr> <th>Order Number</th><th>Order Item Number</th><th>
Order Description</th><th>Item Cost</th></tr>' TO lt_text2.
LOOP AT lt_item INTO DATA(ls_item).
  APPEND |<tr><td>{ ls_item-ono }</td><td>{ ls_item-oin }</td><td>
{ ls_item-odesc }</td><td>{ ls_item-icost }</td></tr>| TO  lt_text2.
ENDLOOP.
APPEND '</table>' TO lt_text2.
APPEND '<br/><p>Thanks & Regards,<br/> Vinod</p>' TO lt_text2.
APPEND '</body></html>' TO lt_text2.
```

```
* 3) create a send request
TRY.
    CALL METHOD cl_bcs=>create_persistent
      RECEIVING
        result = DATA(lo_bcs).
  CATCH cx_send_req_bcs.
ENDTRY.

*  4) create a sap user / external user (recipient).

*  5) ADD THE RECIPIENT.
*        for Exter Mail connect
TRY.
    CALL METHOD cl_cam_address_bcs=>create_internet_address
      EXPORTING
        i_address_string = 'vinod.k@ladera.com'
        i_address_name   = 'K Vinod'
        i_incl_sapuser   = 'X'
      RECEIVING
        result           = DATA(lo_external_user).
  CATCH cx_address_bcs.
ENDTRY.

TRY.
    CALL METHOD lo_bcs->add_recipient
      EXPORTING
        i_recipient = lo_external_user.
  CATCH cx_send_req_bcs.
ENDTRY.

* 6) create the document

IF p_r1 = 'X'.
  TRY.
      CALL METHOD cl_document_bcs=>create_document
        EXPORTING
          i_type    = 'HTM'
          i_subject = 'Header Details'
          i_text    = lt_text1
        RECEIVING
          result    = lo_document.
    CATCH cx_document_bcs.
  ENDTRY.
  MESSAGE 'Header Sent Mail' TYPE 'I'.
ELSE.
  TRY.
      CALL METHOD cl_document_bcs=>create_document
        EXPORTING
          i_type    = 'HTM'
          i_subject = 'Item Details'
          i_text    = lt_text2
        RECEIVING
          result    = lo_document.
    CATCH cx_document_bcs.
  ENDTRY.
  MESSAGE 'Items Sent Mail' TYPE 'I'.

ENDIF.

* 7) Add the attachment.

* 8) set the document.
TRY.
    CALL METHOD lo_bcs->set_document
```

```
      EXPORTING
        i_document = lo_document.
  CATCH cx_send_req_bcs.
ENDTRY.

* 9) Activate or Deactivate immediate sending
TRY.
    CALL METHOD lo_bcs->set_send_immediately
      EXPORTING
        i_send_immediately = 'X'.
  CATCH cx_send_req_bcs.
ENDTRY.

* 10) send
TRY.
    CALL METHOD lo_bcs->send
*  EXPORTING
*    i_with_error_screen = SPACE
      RECEIVING
        result = DATA(lv_result).
  CATCH cx_send_req_bcs.
ENDTRY.

* 11) Commit only for external email not for sap users.( within sap do not
require commit work ).
IF lv_result IS NOT INITIAL.
  COMMIT WORK.
ENDIF.
```