

```

                                Z_PAYROLL_2_PDF_FORMS.txt
*-----*
*   Include           Z_PAYROLL_2_PDF_FORMS                      *
*-----*
*   Forms for REPORT: Z_PAYROLL_2_PDF                            *
*-----*

*
* F O R M S F O R M S F O R M S F O R M S F O R M S F O R M S *
*

*-----*
*           Form  INIT_PARAMETERS                                *
*-----*
FORM init_parameters.

    gf_molga = '05'.

ENDFORM.                  " INIT_PARAMETERS

*-----*
*   FORM A100_LOG_CREATE                                         *
*-----*
FORM a100_log_create.

    DATA:wa_log TYPE bal_s_log.

* Define some header data of this log
    wa_log-extnumber = 'Application Log'.
    wa_log-aluser    = sy-uname.
    wa_log-alprog    = sy-repid.

* Create a log
    CALL FUNCTION 'BAL_LOG_CREATE'
        EXPORTING
            i_s_log = wa_log
        EXCEPTIONS
            OTHERS  = 1.

    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.                  "A100_log_create

*-----*
*           Form  A999_LOG_DISPLAY                               *
*-----*
FORM a999_log_display.

    DATA: wa_display_profile TYPE bal_s_prof,
           wa_title          TYPE sytitle.

* To display profile
    CALL FUNCTION 'BAL_DSP_PROFILE_NO_TREE_GET'
        IMPORTING
            e_s_display_profile = wa_display_profile.

    wa_title = sy-title.
    SET TITLEBAR wa_title.

    MOVE wa_title TO wa_display_profile-title.
    wa_display_profile-use_grid = 'X'.

```

```

                                Z_PAYROLL_2_PDF_FORMS.txt
CALL FUNCTION 'BAL_DSP_LOG_DISPLAY'
  EXCEPTIONS
    OTHERS = 1.

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

ENDFORM.                                " A999_LOG_DISPLAY

*-----*
* FORM MSG_ADD_FREE_TEXT                                           *
*-----*
FORM msg_add_free_text USING      value(msgty)      TYPE symsgty
                                value(i_text)      TYPE c
                                value(i_probclass) TYPE balprobcl.

* add this message to log file
* (I_LOG_HANDLE is not specified, we want to add to the default log.
* If it does not exist we do not care =>EXCEPTIONS log_not_found = 0)
CALL FUNCTION 'BAL_LOG_MSG_ADD_FREE_TEXT'
  EXPORTING
    I_LOG_HANDLE =
    i_probclass  = i_probclass
    i_msgty      = msgty
    i_text       = i_text
  EXCEPTIONS
    log_not_found = 0
    OTHERS        = 1.

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.                                "msg_add_free_text

*-----*
* Form get_select_options                                           *
*-----*
FORM get_select_options
  USING uv_pernr TYPE persno.

  DATA: ls_selstab TYPE rsparams.

* Read the contents of the parameters and selection options of the current
program into an internal table selstab
CALL FUNCTION 'RS_REFRESH_FROM_SELECTOPTIONS'
  EXPORTING
    curr_report      = sy-repid
  TABLES
    selection_table  = gt_selstab
  EXCEPTIONS
    not_found        = 1
    no_report        = 2
    OTHERS            = 3.

* FM error
IF sy-subrc <> 0.
  CONCATENATE text-e05 text-e06 INTO gf_messtext SEPARATED BY space.
  PERFORM msg_add_free_text USING 'E' gf_messtext ''.
  PERFORM a999_log_display.
  STOP.
ELSE.
* Adjust gt_selstab.

```

```
* First clear all select-options
  DELETE gt_seltab WHERE selname = 'PNPPERNR'.
```

```
* Add a line with current (loop)-persno.
  ls_seltab-selname = 'PNPPERNR'.
  ls_seltab-kind    = 'S'.
  ls_seltab-sign    = 'I'.
  ls_seltab-option  = 'EQ'.
  ls_seltab-low     = uv_pernr.
  APPEND ls_seltab TO gt_seltab.
```

```
* Run report not necessary.
  CLEAR: ls_seltab.
  ls_seltab-selname = 'PRT_PROT'.
  ls_seltab-kind    = 'P'.
  ls_seltab-sign    = ' '.
  ls_seltab-option  = ' '.
  ls_seltab-low     = ' '.
  APPEND ls_seltab TO gt_seltab.
```

```
* Add Radiobuttons.
  ls_seltab-selname = 'PNPTIMR9'.
  ls_seltab-kind    = 'P'.
  ls_seltab-sign    = 'I'.
  ls_seltab-option  = 'EQ'.
  ls_seltab-low     = ' '.
  APPEND ls_seltab TO gt_seltab.
```

```
* Add Radiobuttons.
  ls_seltab-selname = 'PNPTIMRA'.
  ls_seltab-kind    = 'P'.
  ls_seltab-sign    = 'I'.
  ls_seltab-option  = 'EQ'.
  ls_seltab-low     = 'X'.
  APPEND ls_seltab TO gt_seltab.
ENDIF.
```

```
ENDFORM.                  " get_select_options
```

```
*-----*
*      Form  A000_LIST_2_SPOOL                      *
*-----*
```

```
FORM a000_list_2_spool
  USING   uv_pernr TYPE persno
  CHANGING cv_spool TYPE rspoid.
```

```
*- (1) Fetch Select-Options and Parameters
  PERFORM get_select_options USING uv_pernr.
```

```
*- (2) Get Print Parameters
  PERFORM get_print_parameters USING uv_pernr.
```

```
*- (3) Execute program and write list to spool
  PERFORM submit_prog_write_list_2_spool.
```

```
*- (4) Fetch spool number
  PERFORM fetch_spool_no USING uv_pernr
    CHANGING cv_spool.
```

```
ENDFORM.                  " A000_LIST_2_SPOOL
```

```
*-----*
*      Form  SUBMIT_PROG_WRITE_LIST_2_SPOOL          *
*-----*
```

```
FORM submit_prog_write_list_2_spool.
```

IF gf_valid <> space.

```
* Call transaction: PC00_M05_CEDT
  SUBMIT rpcedtn0
    WITH SELECTION-TABLE gt_selstab
    TO SAP-SPOOL
    SPOOL PARAMETERS gs_params
    WITHOUT SPOOL DYNPRO
    AND RETURN.
```

COMMIT WORK AND WAIT.

ENDIF.

ENDFORM. " EXEC_PROG_WRITE_LIST_2_SPOOL

```
*-----*
*      Form  GET_PRINT_PARAMETERS      *
*-----*
```

```
FORM get_print_parameters
  USING uv_pernr TYPE persno.
```

```
DATA: lf_list_name TYPE syplst,
      lf_list_text TYPE syprtxt.
```

```
CLEAR: gf_valid,
       gs_params.
```

```
CONCATENATE pnppabrj
            _
            pnppabrp
            _
            uv_pernr
            INTO lf_list_text.
```

```
MOVE: uv_pernr TO lf_list_name.
```

```
CALL FUNCTION 'GET_PRINT_PARAMETERS'
  EXPORTING
    destination      = 'LOCL'
    immediately      = ' '
    list_name         = lf_list_name
    list_text         = lf_list_text
    no_dialog         = 'X'
  IMPORTING
    out_parameters    = gs_params
    valid             = gf_valid
  EXCEPTIONS
    archive_info_not_found = 1
    invalid_print_params  = 2
    invalid_archive_params = 3
    OTHERS               = 4.
```

```
* FM error
IF sy-subrc <> 0 OR sy-msgty = 'E'.
  CONCATENATE text-e05 text-e07 INTO gf_messtext SEPARATED BY space.
  PERFORM msg_add_free_text USING 'E' gf_messtext ''.
  MESSAGE ID sy-msgid TYPE sy-msgty NUMBER sy-msgno INTO gf_messtext WITH
sy-msgv1 sy-msgv2 sy-msgv3 sy-msgv4.
  PERFORM msg_add_free_text USING 'E' gf_messtext ''.
  PERFORM a999_log_display.
  STOP.
ENDIF.
```

Z_PAYROLL_2_PDF_FORMS.txt

ENDFORM.

" GET_PRINT_PARAMETERS

```
*-----*
*      Form  FETCH_SPOOL_NO      *
*-----*
FORM fetch_spool_no
  USING   uv_pernr TYPE persno
  CHANGING cv_spool TYPE rspoid.

  DATA: ls_tsp01 TYPE tsp01.

* Get the highest spoolnumber
  SELECT rqident
    FROM tsp01 AS t1
   INTO cv_spool
  WHERE rqtitle = gs_params-prtxt
    AND  rqcretime = ( SELECT MAX( rqcretime ) FROM tsp01 WHERE rqtitle =
t1~rqtitle ).
  ENDSELECT.

* Or like this =>
  SELECT *
    FROM tsp01
   INTO ls_tsp01
 WHERE rqtitle = gs_params-prtxt ORDER BY rqcretime DESCENDING.
  cv_spool = ls_tsp01-rqident.
  EXIT.
  ENDSELECT.

  IF sy-subrc <> 0.
* Issue error-message: => No spoolnumber found!
  CONCATENATE text-e08 uv_pernr INTO gf_messtext SEPARATED BY space.
  PERFORM msg_add_free_text USING 'E' gf_messtext '2'.
  ENDIF.
```

ENDFORM.

" FETCH_SPOOL_NO

```
*-----*
*      Form  GET_PERSONELL_NUMBERS      *
*-----*
FORM get_personell_numbers .

* Create itab from select-options table
  SELECT pernr
    INTO CORRESPONDING FIELDS OF TABLE gt_pernr
   FROM pa0002
  WHERE pernr IN pnppernr
    AND begda LE sy-datum
    AND endda GE sy-datum.

  SORT gt_pernr.
```

ENDFORM.

" GET_PERSONELL_NUMBERS

```
*-----*
*      Form  B000_CREATE_PDF_FROM_SPOOL      *
*-----*
FORM b000_create_pdf_from_spool
  USING uv_spool TYPE rspoid.

* Only continue if a spoolnumber is present.
  CHECK uv_spool <> 0.
```

REFRESH: gt1_pdf.
 CLEAR: gf_pdf_bytecount.

FORMAT INTENSIFIED OFF.

* Convert Spool to PDF

```
CALL FUNCTION 'CONVERT_ABAPSPoolJOB_2_PDF'
  EXPORTING
    src_spoolid           = uv_spool
    no_dialog             = co_false
    dst_device            = co_device
    get_size_from_format  = 'X'
  IMPORTING
    pdf_bytecount         = gf_pdf_bytecount
  TABLES
    pdf                   = gt1_pdf
  EXCEPTIONS
    err_no_abap_spooljob  = 1
    err_no_spooljob       = 2
    err_no_permission     = 3
    err_conv_not_possible = 4
    err_bad_destdevice    = 5
    user_cancelled        = 6
    err_spoolerror        = 7
    err_temseerror        = 8
    err_btcjob_open_failed = 9
    err_btcjob_submit_failed = 10
    err_btcjob_close_failed = 11
    OTHERS                 = 12.
```

* Catch error.

IF sy-subrc <> 0.

* Issue error-message: =>

```
MESSAGE ID sy-msgid TYPE sy-msgty NUMBER sy-msgno INTO gf_messtext WITH
sy-msgv1 sy-msgv2 sy-msgv3 sy-msgv4.
PERFORM msg_add_free_text USING 'E' gf_messtext ''.
ENDIF.
```

CALL FUNCTION 'SX_TABLE_LINE_WIDTH_CHANGE'

```
TABLES
  content_in           = gt1_pdf
  content_out          = SOLI_TAB
EXCEPTIONS
  OTHERS                = 4.
```

CALL FUNCTION 'SCMS_BINARY_TO_XSTRING'

```
EXPORTING
  input_length         = gf_pdf_bytecount
IMPORTING
  buffer               = e_xstring
TABLES
  binary_tab           = SOLI_TAB
EXCEPTIONS
  OTHERS                = 2.
```

CALL FUNCTION 'SCMS_XSTRING_TO_BINARY'

```
EXPORTING
  buffer               = e_xstring
TABLES
  binary_tab           = lt_solix.
```

gt_pdf[] = lt_solix[].

ENDFORM. " B000_CREATE_PDF_FROM_SPOOL

```

                                Z_PAYROLL_2_PDF_FORMS.txt
*-----*
FORM c000_download_pdf_2_server
  USING uv_pernr      TYPE persno
        uv_spool      TYPE rspoid.

* Only continue if a spoolnumber is present.
  CHECK uv_spool <> 0.

  DATA: lt_pdf      TYPE solix,
        lf_filename TYPE localfile.

* Check pdf-content.
  IF gf_pdf_bytecount LE '6059'.
    EXIT.
  ENDIF.

* Path & Filename
  PERFORM create_path_and_filename USING uv_pernr
                                       CHANGING lf_filename.

*Open file
  OPEN DATASET gf_path_U FOR OUTPUT IN BINARY MODE.

  IF sy-subrc <> 0.
* Issue error-message: => Opening of dataset went wrong
    CONCATENATE text-e01 gf_path_U text-e02 INTO gf_messtext SEPARATED BY
space.
    PERFORM msg_add_free_text USING 'E' gf_messtext ''.
    EXIT.

  ELSE.

    CONCATENATE gf_path text-e04 INTO gf_messtext SEPARATED BY space.
    PERFORM msg_add_free_text USING 'S' gf_messtext ''.

  ENDIF.

*Transfer content to file
  LOOP AT gt_pdf INTO lt_pdf.

    TRANSFER lt_pdf TO gf_path_U.

  ENDLOOP.

* Close file
  CLOSE DATASET gf_path_U.

ENDFORM.                                " C000_DOWNLOAD_PDF_2_SERVER

*-----*
*          Form  CREATE_PATH_AND_FILENAME          *
*-----*
FORM create_path_and_filename
  USING uv_pernr      TYPE persno
        CHANGING cf_filename TYPE localfile.

* Clear path and get the new one for next record
  CLEAR: gf_path.
  PERFORM get_path.

```

** If Other period is chosen @ sel-screen:

```

CONCATENATE pnppabrp      "Year (Other period)
            pnppabrp      "Period (Other Period)
            uv_pernr      "Personnal number
            co_ext
            INTO cf_filename.

```

```

REPLACE '?' WITH cf_filename INTO gf_path.
cf_filename = 'temp.pdf'.
REPLACE '?' WITH cf_filename INTO gf_path_U.

```

ENDFORM. " CREATE_PATH_AND_FILENAME

```

*-----*
*      Form  GET_PATH      *
*-----*

```

FORM get_path.

```

* Get pathname
CALL FUNCTION 'FILE_GET_NAME_USING_PATH'
  EXPORTING
    logical_path      = 'HCM_PAYROLL_SLIP_OUT'
    file_name         = '?'
  IMPORTING
    file_name_with_path = gf_path
  EXCEPTIONS
    path_not_found     = 1
    missing_parameter  = 2
    operating_system_not_found = 3
    file_system_not_found = 4
    OTHERS              = 5.

```

```

* If error, quit.
IF sy-subrc <> 0.
  MESSAGE ID sy-msgid
    TYPE 'E'
    NUMBER sy-msgno
    WITH sy-msgv1 sy-msgv2 sy-msgv3 sy-msgv4.
ENDIF.
gf_path_U = gf_path.
ENDFORM. " GET_PATH

```

```

*-----*
*      Form  CHECK_Payrollperiod      *
*-----*

```

FORM check_payrollperiod.

```

DATA: lf_period TYPE pnppabrp,
      lf_year   TYPE pnppabrpj,
      lf_state  TYPE vwsta,
      lf_date_1 TYPE sydatum,
      lf_date_2 TYPE sydatum,
      ls_t569v  TYPE t569v.

```

```

IF pnptimr9 = 'X'.
  lf_period = pnpdispp .
  lf_year   = pnpdispj .
ELSEIF pnptimra = 'X'.
  lf_period = pnppabrp .

```



```

    lf_year    = pnppabrpj.
ENDIF.

```

```

* Get the payroll area from T569V

```

```

SELECT SINGLE *
  INTO ls_t569v
  FROM t569v
  WHERE abkrs = pnpxabkr
        AND vwsaz = '1'.

```

```

* Construct date from period-year from T569V

```

```

CONCATENATE '01'
            ls_t569v-pabrp
            ls_t569v-pabrpj
  INTO lf_date_2.

```

```

* Construct date from period-year on selection-screen

```

```

CONCATENATE '01'
            lf_period
            lf_year
  INTO lf_date_1.

```

```

* IF period of the selection-screen is less than the period in T569V (or
transactie PA03) => go on

```

```

  IF lf_date_1 < lf_date_2.
  ENDIF.

```

```

* IF period of the selection-screen is equal on the period in T569V => Check de
status

```

```

  IF lf_date_1 = lf_date_2.

```

```

* Get status of the payroll area

```

```

SELECT SINGLE state
  INTO lf_state
  FROM t569v
  WHERE abkrs = pnpxabkr
        AND vwsaz = '1'
        AND pabrpj = lf_year
        AND pabrp = lf_period.

```

```

  IF lf_state NE '3'.

```

```

    CALL FUNCTION 'POPUP_TO_DISPLAY_TEXT'

```

```

      EXPORTING
        titel           = text-002
        textline1       = text-003
*      TEXTLINE2       = ' '
        start_column    = 25
        start_row       = 6.

```

```

    STOP.

```

```

  ENDIF.

```

```

ENDIF.

```

```

* IF period of the selection-screen is greater than the period in T569V =>
error, issue message.

```

```

  IF lf_date_1 > lf_date_2.

```

```

    CALL FUNCTION 'POPUP_TO_DISPLAY_TEXT'

```

```

      EXPORTING
        titel           = text-002
        textline1       = text-004
*      TEXTLINE2       = ' '
        start_column    = 25
        start_row       = 6.

```

STOP.

ENDIF.

ENDFORM. " CHECK_PAYROLLPERIODE

```
*-----*
*      Form  EMAIL_THE_FILES      *
*-----*
```

FORM email_the_files.

```
DATA: ls_dir_list    TYPE eps2fili,
      lf_usrid_long  TYPE comm_id_long,
      lf_mailaddress TYPE comm_id_long,
      lf_error       TYPE boole_d.
```

CLEAR: lf_error.

```
* Get mail-address for Persnr
  PERFORM get_mailaddress_for_pernr USING ls_pernr
                                     CHANGING lf_mailaddress
                                             lf_error.
```

```
* If no mail address is found send it to the general email address
if lf_mailaddress is initial.
```

```
  CONCATENATE text-S08
               ls_pernr
               INTO gf_messtext SEPARATED BY space.
  PERFORM msg_add_free_text USING 'E' gf_messtext '2'.
```

```
PERFORM Get_general_address using lf_mailaddress..
endif.
if sy-subrc = 0.
```

```
* Send the mail with PDF-attachment.
  PERFORM send_mail_with_attachment USING ls_pernr
                                         lf_mailaddress
                                         CHANGING lf_error.
```

endif.

ENDFORM. " EMAIL_THE_FILES

```
*-----*
*      Form  SEND_MAIL_WITH_ATTACHMENT      *
*-----*
```

```
FORM send_mail_with_attachment
  USING m_pernr TYPE ty_pernr
        uv_mailaddress TYPE comm_id_long
  CHANGING cf_error TYPE boole_d.
```

```
DATA: lt_contents TYPE STANDARD TABLE OF solist1.
```

```
* Frame the Body of the Email
  PERFORM frame_mail_body USING m_pernr
                               cf_error
                               CHANGING lt_contents.
```

```
* Send Mail
```

```

                                Z_PAYROLL_2_PDF_FORMS.txt
PERFORM send_mail USING      uv_mailaddress
                             lt_contents
                             m_pernr
CHANGING cf_error.

ENDFORM.                        " SEND_MAIL_WITH_ATTACHMENT

*-----*
*          Form  GET_MAILADDRESS_FOR_PERNR          *
*-----*
FORM get_mailaddress_for_pernr
  USING      m_pernr          TYPE ty_pernr
  CHANGING   cf_mailaddress TYPE comm_id_long
            cf_error          TYPE boole_d.

DATA: lt_p0105          TYPE p0105ar_tab,
      ls_p0105          TYPE p0105,
      lf_pernr          TYPE pernr_d.

* Get pernr from filename
  lf_pernr = ls_pernr.

* Read Infotype 0105 (Communication)
  CALL FUNCTION 'HR_READ_INFOTYPE'
    EXPORTING
      tclas          = 'A'
      pernr          = lf_pernr    "
      infty          = '0105'
      begda          = sy-datum
      endda          = sy-datum    " '99991231'
    * BYPASS_BUFFER  = ' '
    * LEGACY_MODE    = ' '
    * IMPORTING
    * SUBRC          =
    TABLES
      infty_tab      = lt_p0105
    EXCEPTIONS
      infty_not_found = 1
      OTHERS         = 2.

* catch errors.
  IF sy-subrc <> 0.

* Issue error-message: =>
  MESSAGE ID sy-msgid TYPE sy-msgty NUMBER sy-msgno INTO gf_messtext WITH
sy-msgv1 sy-msgv2 sy-msgv3 sy-msgv4.
  PERFORM msg_add_free_text USING 'E' gf_messtext ''.
  cf_error = 'X'.
  ELSE.

* Check if an email address is found.
*   READ TABLE lt_p0105 INTO ls_p0105 INDEX 1.

  IF sy-subrc = 0.
    Loop at lt_p0105 INTO ls_p0105
      where subty = '0010'.

      MOVE: ls_p0105-usrid_long TO cf_mailaddress.
    endloop.
  ENDIF.
  if cf_mailaddress is initial.

* Issue error-message: =>
  MESSAGE ID sy-msgid TYPE sy-msgty NUMBER sy-msgno INTO gf_messtext WITH
sy-msgv1 sy-msgv2 sy-msgv3 sy-msgv4.
  PERFORM msg_add_free_text USING 'E' gf_messtext ''.
  cf_error = 'X'.

```

endif.

ENDIF.

ENDFORM. " GET_MAILADDRESS_FOR_PERNR

```
*-----*
*      Form  FRAME_MAIL_BODY      *
*-----*
```

```
FORM frame_mail_body
  USING      m_pernr      TYPE ty_pernr
            cf_error      TYPE boole_d
  CHANGING ct_contents TYPE ccrctt_text_tab.
```

CHECK: cf_error = ''.

```
*-- Local data declaration to hold the textpool
DATA: tl_textpool TYPE STANDARD TABLE OF textpool,
      wl_textpool TYPE textpool,
      wa_contents TYPE solist1,
      lf_yearperiod TYPE char7.
```

```
*get the month description
m_monthnr = pnpabr.
m_langu = 'EN'.
CALL FUNCTION 'ISP_GET_MONTH_NAME'
  EXPORTING
    *   DATE           = T_DATE
        LANGUAGE       = m_langu
        MONTH_NUMBER   = m_monthnr
  IMPORTING
    *   LANGU_BACK     =
        LONGTEXT       = m_month.
```

```
*-- Read the Entire Textpool into an Internal table
READ TEXTPOOL sy-repid INTO tl_textpool LANGUAGE sy-langu.
IF sy-subrc IS INITIAL.
  SORT tl_textpool BY id key.
ENDIF.
```

```
*-- Font start
CLEAR wa_contents.
wa_contents-line = c_style_start.
APPEND wa_contents TO ct_contents.
```

```
perform New_line using 1
                        ct_contents.
```

*-- Body text

```
PERFORM get_body_text USING m_pernr
                        CHANGING wa_contents
                                ct_contents.
```

```
perform New_line using 10
                        ct_contents.
```

```
*-- Subject of the Mail
  CONCATENATE text-005 m_month pnpabrj INTO w_mail_subj SEPARATED BY space.
```

```
ENDFORM.          " FRAME_MAIL_BODY
```

```
*-----*
*      Form  SEND_MAIL      *
*-----*
```

```
FORM send_mail
```

```
  USING      uv_mailaddress TYPE comm_id_long
             ut_contents   TYPE ccrctt_text_tab
             m_pernr       TYPE ty_pernr
  CHANGING   cf_error      TYPE boole_d.
```

```
  CHECK: cf_error = ''.
```

```
*-- Local data declaration for sending mail
```

```
DATA: l_send_request TYPE REF TO cl_bcs,
      l_document      TYPE REF TO cl_document_bcs,
      l_sender        TYPE REF TO cl_sapuser_bcs,
      l_sub           TYPE char50,
      l_recipient      TYPE REF TO if_recipient_bcs,
      t1_contents     TYPE STANDARD TABLE OF soli,
      l_doc_len       TYPE so_obj_len,
      l_cnt           TYPE sy-tabix,
      l_rcv_email     TYPE adr6-smtp_addr,
      l_result        TYPE sy-binpt,
      l_bcs_exception TYPE REF TO cx_bcs,
      l_subj          TYPE string,
      wa_contents     TYPE solist1,
      lf_path         TYPE eseftappl.
```

```
DATA: str1 TYPE string,
      str2 TYPE string,
      str3 TYPE string,
      str4 TYPE string,
      str5 TYPE string.
```

```
  MOVE: gf_path TO lf_path.
```

```
  TRY.
```

```
*-- Create persistent send request
      l_send_request = cl_bcs=>create_persistent( ).
      t1_contents[] = ut_contents[].
```

```
*-- Get the length of the Document
      DESCRIBE TABLE t1_contents LINES l_cnt.
      READ TABLE t1_contents INTO wa_contents INDEX l_cnt.
      l_doc_len = ( l_cnt - 1 ) * 255 + STRLEN( wa_contents ).
```

```
*-- Subject of the mail
      l_sub = w_mail_subj.
```

```
*-- Create Document
      l_document = cl_document_bcs=>create_document(
        i_type       = c_htm
        i_text       = t1_contents
        i_length     = l_doc_len
        i_subject    = l_sub
        i_language   = sy-langu
        i_importance = '1' ).
```

```
*-- Subject of the mail
      MOVE w_mail_subj TO l_subj.
      w_document = l_document.
```

```
  TRY.
```

```

                                Z_PAYROLL_2_PDF_FORMS.txt
*-- Set the Message Subject
    CALL METHOD l_send_request->set_message_subject
    EXPORTING
        ip_subject = l_subj.
    CATCH cx_sy_dyn_call_illegal_method.
    ENDTRY.

*-- Add document to send request
    CALL METHOD l_send_request->set_document( l_document ).

*-- Do send delivery info for successfull mails
    CALL METHOD l_send_request->set_status_attributes
    EXPORTING
        i_requested_status = 'E'
        i_status_mail      = 'A'.

**-- Set sender -----
    DATA: l_d_addr      TYPE adr6-smtp_addr,
           lf_sender_mail TYPE ad_smtpadr,
           internet_sender TYPE REF TO if_sender_bcs.

    MOVE: text-mai TO lf_sender_mail. "xxxx@xxx.xx".
    l_d_addr = lf_sender_mail. "gs_finaa_last-mail_send_addr.

    internet_sender = cl_cam_address_bcs=>create_internet_address(
    i_address_string = l_d_addr ).

    CALL METHOD l_send_request->set_sender
    EXPORTING
        i_sender = internet_sender.

*-----

*-- To frame the attachments for the mail
    PERFORM frame_attachments USING m_pernr.

*-- Add the recipients to the Send mail

SPLIT uv_mailaddress AT ';' INTO: str1 str2 str3 str4 str5.
do 5 times.
    case sy-index.
        when 1.
            l_rcv_email = str1.
        when 2.
            l_rcv_email = str2.
        when 3.
            l_rcv_email = str3.
        when 4.
            l_rcv_email = str4.
        when OTHERS.
            l_rcv_email = str5.
    endcase.

    CHECK NOT l_rcv_email IS INITIAL.
    l_recipient = cl_cam_address_bcs=>create_internet_address(
                                                    l_rcv_email ).
    CALL METHOD l_send_request->add_recipient
    EXPORTING
        i_recipient = l_recipient
        i_express   = 'X'.

enddo.

*-- Send Email
    CALL METHOD l_send_request->send(
    EXPORTING

```

```

                                Z_PAYROLL_2_PDF_FORMS.txt
i_with_error_screen = 'X'
RECEIVING
result                = l_result ).

```

```

COMMIT WORK.
wait up to 1 seconds.

```

```

* Sending of the mail went succesfully
IF l_result = 'X'.

```

```

    CONCATENATE text-S05
                text-S11
                m_pernr
                text-S06
                uv_mailaddress
    INTO gf_messtext SEPARATED BY space.

```

```

    PERFORM msg_add_free_text USING 'S' gf_messtext '2'.

```

```

ELSE.

```

```

    CONCATENATE text-S05
                text-S11
                m_pernr
                text-S07
                uv_mailaddress
    INTO gf_messtext SEPARATED BY space.

```

```

    PERFORM msg_add_free_text USING 'E' gf_messtext '3'.

```

```

    cf_error = 'X'.

```

```

ENDIF.

```

```

* Sending of the mail went wrong
CATCH cx_bcs INTO l_bcs_exception.
IF l_result NE 'X'.

```

```

    CONCATENATE text-S05
                text-S07
                uv_mailaddress
    INTO gf_messtext SEPARATED BY space.

```

```

    PERFORM msg_add_free_text USING 'E' gf_messtext '3'.
    cf_error = 'X'.

```

```

ENDIF.

```

```

ENDTRY.

```

```

COMMIT WORK.                                "Commit work

```

```

ENDFORM.                                " SEND_MAIL

```

```

*-----*
* Form Name      : frame_attachments *
*-----*
* Description    : To frame the attachments for the mail to be sent *
*-----*

```

```

FORM frame_attachments
    USING m_pernr          TYPE ty_pernr.

```

```

*-- Local Data declaration
DATA: l_subject   TYPE so_obj_des,
      l_att_type  TYPE soodk-objtp,
      w_file      TYPE dvasdocid,

```

```

                                Z_PAYROLL_2_PDF_FORMS.txt
w_extn(5)    TYPE c,
wa_gt1_pdf LIKE LINE OF gt1_pdf,
wa_gt_pdf LIKE LINE OF gt_pdf.

*-- Subject of the Attachment
CONCATENATE pnppabrp
            _
            pnppabrp
            _
            m_pernr
            INTO l_subject.

*-- Format of the Attachment
l_att_type = 'pdf'.

IF gt_pdf[] IS NOT INITIAL.
  TRY.
*-- Add Attachment to the Document
    CALL METHOD w_document->add_attachment
      EXPORTING
        i_attachment_type    = l_att_type
        i_attachment_subject = l_subject
        i_att_content_hex    = gt_pdf.

    CATCH cx_document_bcs.
  ENDTRY.
ENDIF.

ENDFORM.                    " frame_attachments

*-----*
* Form GET_BODY_TEXT *
*-----*
FORM get_body_text
  USING m_pernr TYPE ty_pernr
  CHANGING c_wa_contents TYPE solisti1
           ct_contents TYPE ccrctt_text_tab.

* Get pernr from filename
lf_pernr = m_pernr.

* Get Communication language
SELECT SINGLE sprsl rufnm vorna
  INTO (lf_sprsl, lf_rufnm, lf_vorna)
  FROM pa0002
  WHERE pernr = lf_pernr
  AND begda LE sy-datum
  AND endda GE sy-datum.

IF sy-subrc <> 0.
  lf_sprsl = 'E'.
ENDIF.

* Read (S010)-text
CALL FUNCTION 'READ_TEXT'
  EXPORTING
    id = 'ST'
    language = 'E'
    name = 'ZHR_MAIL_BODY'
    object = 'TEXT'
  TABLES
    lines = ta_tlinetab
  EXCEPTIONS
    id = 1
    language = 2

```



```

                                Z_PAYROLL_2_PDF_FORMS.txt
name                           = 3
not_found                      = 4
object                         = 5
reference_check                = 6
wrong_access_to_archive        = 7
OTHERS                         = 8.

* Catch errors.
IF sy-subrc <> 0.
* Issue error-message: =>
  MESSAGE ID sy-msgid TYPE sy-msgty NUMBER sy-msgno INTO gf_messtext WITH
sy-msgv1 sy-msgv2 sy-msgv3 sy-msgv4.
  PERFORM msg_add_free_text USING 'E' gf_messtext ''.
ENDIF.

* Build MM-YYYYY
  CONCATENATE m_month
              pnppabrj
              INTO lf_yearperiod SEPARATED BY space.

* (line 1)
PERFORM get_text_rule using 1
                          c_wa_contents-line.

*Take the vorna if rufnm is empty
if lf_rufnm is initial.
  CONCATENATE c_wa_contents-line
              lf_vorna
              INTO c_wa_contents SEPARATED BY space.
else.
  CONCATENATE c_wa_contents-line
              lf_rufnm
              INTO c_wa_contents SEPARATED BY space.
endif.

  CONCATENATE c_wa_contents
              ,
              INTO c_wa_contents.
  APPEND c_wa_contents TO ct_contents.

perform New_line using 2
                    ct_contents.

* (line 2)
PERFORM get_text_rule using 2
                          c_wa_contents-line.

  CONCATENATE c_wa_contents
              lf_yearperiod
              INTO c_wa_contents SEPARATED BY space.
  APPEND c_wa_contents TO ct_contents.

perform New_line using 2
                    ct_contents.

* (line 3)
PERFORM get_text_rule using 3
                          c_wa_contents-line.

  APPEND c_wa_contents TO ct_contents.

perform New_line using 2
                    ct_contents.

* (line 4)
PERFORM get_text_rule using 4
                          c_wa_contents-line.

  APPEND c_wa_contents TO ct_contents.

```

```

perform New_line using 2
                        ct_contents.

* (line 5)
PERFORM get_text_rule using 5
                        c_wa_contents-line.
  APPEND c_wa_contents TO ct_contents.

perform New_line using 2
                        ct_contents.

* (line 6)
PERFORM get_text_rule using 6
                        c_wa_contents-line.
  APPEND c_wa_contents TO ct_contents.

perform New_line using 2
                        ct_contents.

* (line 7)
PERFORM get_text_rule using 7
                        c_wa_contents-line.
  APPEND c_wa_contents TO ct_contents.

ENDFORM.                " GET_BODY_TEXT
*&-----*
*&      Form  GET_GENERAL_ADDRESS
*&-----*
*      text
*-----*
*  -->  p1      text
*  <--  p2      text
*-----*
FORM GET_GENERAL_ADDRESS using my_mailaddress.
* Read (S010)-text
  CALL FUNCTION 'READ_TEXT'
    EXPORTING
      id          = 'ST'
      language    = 'E'
      name        = 'ZHR_MAIL_GENERAL'
      object      = 'TEXT'
    TABLES
      lines       = ta_tlinetab
    EXCEPTIONS
      id          = 1
      language    = 2
      name        = 3
      not_found   = 4
      object      = 5
      reference_check = 6
      wrong_access_to_archive = 7
      OTHERS      = 8.

* Catch errors.
  IF sy-subrc <> 0.
* Issue error-message: =>
  MESSAGE ID sy-msgid TYPE sy-msgty NUMBER sy-msgno INTO gf_messtext WITH
sy-msgv1 sy-msgv2 sy-msgv3 sy-msgv4.
  PERFORM msg_add_free_text USING 'E' gf_messtext ''.
  ENDIF.

* (line 1)
  READ TABLE ta_tlinetab INTO ls_tlinetab INDEX 1.
  if ls_tlinetab-tdline is initial.
* Issue error-message: =>
  CONCATENATE text-S09
                'ZHR_MAIL_GENERAL'

```

```

                                Z_PAYROLL_2_PDF_FORMS.txt
                                text-S05
                                text-S10
                                ls_pernr
                                INTO gf_messtext SEPARATED BY space.
                                PERFORM msg_add_free_text USING 'E' gf_messtext '1'.
                                sy-subrc = 1.
                                exit.
                                endif.

                                my_mailaddress = ls_tlinetab-TDLINE.

* (line 2)
clear: ls_tlinetab.
READ TABLE ta_tlinetab INTO ls_tlinetab INDEX 2.
if not ls_tlinetab is initial. "5 rules
    CONCATENATE my_mailaddress
                ls_tlinetab-TDLINE
                INTO my_mailaddress SEPARATED BY ';'.
    clear: ls_tlinetab.
    READ TABLE ta_tlinetab INTO ls_tlinetab INDEX 3.
    if not ls_tlinetab is initial.
        CONCATENATE my_mailaddress
                    ls_tlinetab-TDLINE
                    INTO my_mailaddress SEPARATED BY ';'.
    clear: ls_tlinetab.
    READ TABLE ta_tlinetab INTO ls_tlinetab INDEX 4.
    if not ls_tlinetab is initial.
        CONCATENATE my_mailaddress
                    ls_tlinetab-TDLINE
                    INTO my_mailaddress SEPARATED BY ';'.
    clear: ls_tlinetab.
    READ TABLE ta_tlinetab INTO ls_tlinetab INDEX 5.
    if not ls_tlinetab is initial.
        CONCATENATE my_mailaddress
                    ls_tlinetab-TDLINE
                    INTO my_mailaddress SEPARATED BY ';'.
    endif.
endif.
endif.
endif.
if not my_mailaddress is initial.
    sy-subrc = 0.
endif.

ENDFORM.                                " GET_GENERAL_ADDRESS
*&-----*
*&      Form  GET_TEXT_RULE
*&-----*
*      text
*-----*
*      -->P_1      text
*-----*
FORM GET_TEXT_RULE  USING my_index
                      my_line.

READ TABLE ta_tlinetab INTO ls_tlinetab INDEX my_index.

my_line = ls_tlinetab-tdline.

ENDFORM.                                " GET_TEXT_RULE
*&-----*
*&      Form  NEW_LINE
*&-----*
*      text
*-----*
*      -->P_2      text
*-----*
FORM NEW_LINE  USING Aanta1
                  my_contents TYPE ccrctt_text_tab.

```

```

                                Z_PAYROLL_2_PDF_FORMS.txt
data: d_wa_contents TYPE solisti1.
do aantal times.
*-- New line
    CLEAR d_wa_contents.
    d_wa_contents-line = c_new_line.
    APPEND d_wa_contents TO my_contents.
ENDDO.

ENDFORM.                                " NEW_LINE
*&-----*
*&      Form  ENCRYPT_PDF
*&-----*
*      text
*-----*
*      -->P_GF_PATH  text
*      -->P_LF_BSNR  text
*-----*
FORM ENCRYPT_PDF  USING      P_GF_PATH
                           P_GF_PATH_U
                           P_LF_BSNR.

data: PARAM LIKE SXPGCOLIST-PARAMETERS,
      RESULTADO LIKE BTCXPM occurs 1,
      STATUS LIKE EXTCMDSEX-STATUS.

CONCATENATE P_GF_PATH_U
            'output'
            P_GF_PATH
            'user_pw'
            P_LF_BSNR
            INTO param SEPARATED BY space.

CALL FUNCTION 'SXPG_COMMAND_EXECUTE'
EXPORTING
COMMANDNAME = 'Z_ENCRYPT_PDF'
OPERATINGSYSTEM = 'Linux'

ADDITIONAL_PARAMETERS = PARAM
IMPORTING
STATUS = STATUS
TABLES
EXEC_PROTOCOL = RESULTADO
EXCEPTIONS
NO_PERMISSION = 1
COMMAND_NOT_FOUND = 2
PARAMETERS_TOO_LONG = 3
SECURITY_RISK = 4
WRONG_CHECK_CALL_INTERFACE = 5
PROGRAM_START_ERROR = 6
PROGRAM_TERMINATION_ERROR = 7
X_ERROR = 8
PARAMETER_EXPECTED = 9
TOO_MANY_PARAMETERS = 10
ILLEGAL_COMMAND = 11
WRONG_ASYNCHRONOUS_PARAMETERS = 12
CANT_ENQ_TBTCO_ENTRY = 13
JOBCOUNT_GENERATION_ERROR = 14
OTHERS = 15.

IF sy-subrc <> 0.
* Issue error-message: => encrypting the pdf file
    CONCATENATE text-e10 gf_path INTO gf_messtext SEPARATED BY space.
    PERFORM msg_add_free_text USING 'E' gf_messtext ''.
    EXIT.

```

```

                                Z_PAYROLL_2_PDF_FORMS.txt
ELSE.

    CONCATENATE gf_path text-e09 INTO gf_messtext SEPARATED BY space.
    PERFORM msg_add_free_text USING 'S' gf_messtext ''.

ENDIF.

ENDFORM.                                " ENCRYPT_PDF

*-----*
*      Form  UPLOAD_DATA                *
*-----*
FORM upload_data
    CHANGING cf_error                TYPE boole_d.

    DATA:    ls_pdf                TYPE solix,
              lf_path_and_filename TYPE eseftapl.

    CHECK: cf_error = ''.

    CLEAR: gt_pdf, ls_pdf.
    REFRESH: gt_pdf.

* Complete padnaam + filenaam
MOVE: gf_path TO lf_path_and_filename.

* Dataset openen.
OPEN DATASET lf_path_and_filename FOR INPUT IN BINARY MODE.

IF sy-subrc <> 0.
* Issue error-message: => Opening of dataset went wrong
CONCATENATE text-e11 lf_path_and_filename INTO gf_messtext SEPARATED BY
space.
    PERFORM msg_add_free_text USING 'E' gf_messtext ''.
    cf_error = 'X'.
    EXIT.

ELSE.

* Dataset lezen en transferren naar itab
DO.

    READ DATASET lf_path_and_filename INTO ls_pdf.

    IF sy-subrc = 0.

        APPEND ls_pdf TO gt_pdf.

    ELSE.

        APPEND ls_pdf TO gt_pdf.

    EXIT.
    ENDIF.

ENDDO.

ENDIF.
* Dataset sluiten
CLOSE DATASET lf_path_and_filename.

ENDFORM.                                " UPLOAD_DATA
*&-----*
*&      Form  DELETE_PDF                *
*&-----*
*      text
*-----*

```

```

                                Z_PAYROLL_2_PDF_FORMS.txt
*      -->P_GF_PATH  text
*-----*
FORM DELETE_PDF  USING      P_GF_PATH.

data: PARAM LIKE SXPGCOLIST-PARAMETERS,
      RESULTADO LIKE BTCXPM occurs 1,
      STATUS LIKE EXTCMDSEXEX-STATUS.

PARAM = P_GF_PATH.

CALL FUNCTION 'SXPG_COMMAND_EXECUTE'
EXPORTING
  COMMANDNAME = 'Z_DEL'
  OPERATINGSYSTEM = 'Linux'

ADDITIONAL_PARAMETERS = PARAM
IMPORTING
  STATUS = STATUS
TABLES
  EXEC_PROTOCOL = RESULTADO
EXCEPTIONS
  NO_PERMISSION = 1
  COMMAND_NOT_FOUND = 2
  PARAMETERS_TOO_LONG = 3
  SECURITY_RISK = 4
  WRONG_CHECK_CALL_INTERFACE = 5
  PROGRAM_START_ERROR = 6
  PROGRAM_TERMINATION_ERROR = 7
  X_ERROR = 8
  PARAMETER_EXPECTED = 9
  TOO_MANY_PARAMETERS = 10
  ILLEGAL_COMMAND = 11
  WRONG_ASYNCHRONOUS_PARAMETERS = 12
  CANT_ENQ_TBTCO_ENTRY = 13
  JOBCOUNT_GENERATION_ERROR = 14
  OTHERS = 15.

IF sy-subrc <> 0.
* Issue error-message: => cannot delete the pdf file
  CONCATENATE text-e12 gf_path INTO gf_messtext SEPARATED BY space.
  PERFORM msg_add_free_text USING 'E' gf_messtext ''.
  EXIT.
ENDIF.

ENDFORM.                  " DELETE_PDF
*&-----*
*&      Form  DELETE_SPOOLFILE
*&-----*
*      text
*-----*
*      -->P_LV_SPOOL  text
*-----*
FORM DELETE_SPOOLFILE USING      P_LV_SPOOL.
* Get the highest spoolnumber
  delete from tsp01
  WHERE RQIDENT = p_lv_spool.

ENDFORM.                  " DELETE_SPOOLFILE

```