\*&---------------------------------------------------------------------\*  
\*& Report  ZHR\_2010\_DELETE  
\*&  
\*&---------------------------------------------------------------------\*  
\*&  
\*&  
\*&---------------------------------------------------------------------\*  
REPORT zhr\_2010\_delete\_main.  
  
  
TYPE-POOLS : slis.  
  
TYPES: BEGIN OF ty\_xml\_line,  
         data(256) TYPE x,  
       END OF ty\_xml\_line.  
  
TYPES: BEGIN OF t\_2010updt,  
         pernr     TYPE pernr\_d,  
         lgart     TYPE bdc\_fval,  
         date      TYPE bdc\_fval,  
         custfield TYPE bdc\_fval,  
         amount    TYPE bdc\_fval,  
         suba      TYPE bdc\_fval,  
         wbs       TYPE bdc\_fval,  
         extdoc    TYPE bdc\_fval,  
         bukrs     TYPE bdc\_fval,  
       END      OF t\_2010updt.  
  
TYPES: BEGIN OF ty\_2010\_final,  
         seqnr   TYPE seqnr,  
         subty   TYPE subty,  
         pernr   TYPE persno,  
         aedtm   TYPE aedat,  
         uname   TYPE aenam,  
         lgart   TYPE lgart,  
         endda   TYPE endda,  
         begda   TYPE begda,  
         betrg   TYPE pad\_amt7s,  
         anzhl   TYPE enanz,  
         prakn   TYPE prakn,  
         versl   TYPE vrsch,  
         exbel   TYPE exbel,  
         custfld TYPE hr\_usrfld,  
       END OF ty\_2010\_final.  
  
  
DATA: lv\_proj      TYPE string,  
      lv\_acco      TYPE string,  
      lv\_task      TYPE string,  
      lv\_grpname   TYPE apq\_grpn,  
      lv\_subrc     TYPE syst\_subrc,  
      t\_messtab    TYPE TABLE OF bdcmsgcoll,  
      wa\_messtab   TYPE bdcmsgcoll,  
      lv\_bukrs     TYPE bdc\_fval,  
      lv\_kostl     TYPE kostl,  
      lv\_curr      TYPE bdc\_fval,  
      lv\_aznhl     TYPE enanz,  
      lv\_log       TYPE  bal\_s\_log,  
      lv\_msghnd    TYPE balloghndl,  
      wa\_slg       TYPE bal\_s\_msg,  
      lv\_butxt     TYPE butxt,  
      lv\_num(10)   TYPE c,  
      fieldcatalog TYPE slis\_t\_fieldcat\_alv WITH HEADER LINE,  
      gd\_layout    TYPE slis\_layout\_alv,  
      gd\_repid     LIKE sy-repid,  
      g\_save       TYPE c VALUE 'X',  
      g\_variant    TYPE disvariant,  
      gx\_variant   TYPE disvariant,  
      g\_exit       TYPE c,  
      ispfli       TYPE TABLE OF spfli,  
      lta\_pa2010   TYPE TABLE OF ty\_2010\_final,  
      lwa\_pa2010   TYPE ty\_2010\_final,  
      it\_p2010     TYPE TABLE OF p2010,  
      wa\_p2010     TYPE  p2010.  
  
\*      Declaring the structure for the XML internal table  
TYPES: BEGIN OF ty\_xml,  
         raw(2000) TYPE c,  
       END OF ty\_xml.  
\*      Declaring the XML internal table  
DATA:  g\_t\_xml\_tab TYPE TABLE OF ty\_xml INITIAL SIZE 0.  
\*      Declaring the work area for the XML internal table  
DATA:  wa\_xml\_tab TYPE ty\_xml.  
\*      Declaring the string to contain the data for the XML internal table  
DATA:  g\_str TYPE string.  
\*      Declaring the string to contain x string  
DATA:  g\_xmldata TYPE xstring.  
\*      Declaring the table to contain the parsed data  
DATA:  g\_t\_xml\_info TYPE TABLE OF smum\_xmltb INITIAL SIZE 0.  
\*      Declaring the work area for the internal table containing the parsed data.  
DATA:  g\_s\_xml\_info LIKE LINE OF g\_t\_xml\_info.  
\*      Declaring the table to contain the returned messages from the parsing FM  
DATA:  g\_t\_return TYPE STANDARD TABLE OF bapiret2.  
\*      Declaring the work area for the return table  
DATA:  wa\_return LIKE LINE OF g\_t\_return.  
\*      Declaring the structure for the table containing fields in the XML file  
TYPES: BEGIN OF struc\_people,  
         homepernr(8),  
         uname(4)     TYPE c,  
         userid(32),  
       END OF struc\_people.  
\*      Declaring the internal table containing the fields in the XML file  
DATA:  g\_t\_employeerequest TYPE TABLE OF struc\_people.  
\*      Declaring the work area for the internal table containing the fields in the      \*      XML file  
DATA:  g\_s\_employeerequest LIKE LINE OF g\_t\_employeerequest.  
DATA: gv\_filename TYPE string.  
DATA:gt\_xml\_table TYPE STANDARD TABLE OF ty\_xml\_line.  
DATA: gv\_xml\_table\_size TYPE i .  
DATA: lwa\_2010   TYPE t\_2010updt,  
      lta\_2010   TYPE TABLE OF t\_2010updt,  
      lv\_string1 TYPE string,  
      lv\_string2 TYPE string,  
      lv\_string3 TYPE string,  
      lv\_handle  TYPE balloghndl,  
      lt\_handle  TYPE bal\_t\_logh,  
      lv\_index   TYPE i,  
      lv\_pernr   TYPE char8,  
      lv\_date    TYPE char8,  
      lv\_posid   TYPE ps\_posid,  
      lta2\_2010  TYPE TABLE OF pa2010,  
      lwa2\_2010  TYPE pa2010.  
RANGES: ra\_pernr FOR pernr-pernr OCCURS 0.  
DATA: wa\_retlist TYPE bapireturn1.  
DATA: wa\_bapireturn1 TYPE bapireturn1.  
  
INFOTYPES:2010.  
TABLES: pernr.  
  
  
  
SELECTION-SCREEN BEGIN OF BLOCK b3 WITH FRAME TITLE text-003.  
PARAMETERS : t1 RADIOBUTTON GROUP rb3.  
PARAMETERS : t2 RADIOBUTTON GROUP rb3.  
SELECTION-SCREEN END OF BLOCK b3.  
  
SELECTION-SCREEN BEGIN OF BLOCK b1 WITH FRAME TITLE text-001.  
SELECT-OPTIONS : p\_pernr FOR p2010-pernr.  
SELECT-OPTIONS : p\_extdoc FOR p2010-exbel.  
SELECT-OPTIONS : p\_cust FOR p2010-custfld.  
SELECT-OPTIONS : p\_lgart FOR p2010-lgart.  
SELECT-OPTIONS : p\_aedtm FOR p2010-aedtm.  
SELECT-OPTIONS : p\_user FOR sy-uname.  
SELECT-OPTIONS : p\_begda FOR p2010-begda.  
SELECTION-SCREEN END OF BLOCK b1.  
  
  
  
  
START-OF-SELECTION.  
  
\*  IF k1 = 'X'.  
\*    CONCATENATE '/usr/sap/' sy-sysid '/DVEBMGS00/work/EC\_PAYROLL\_K1.xml' INTO gv\_filename.  
\*  ELSEIF k2 = 'X'.  
\*    CONCATENATE '/usr/sap/' sy-sysid '/DVEBMGS00/work/EC\_PAYROLL\_K2.xml' INTO gv\_filename.  
\*  ELSEIF k3 = 'X'.  
\*    CONCATENATE '/usr/sap/' sy-sysid '/DVEBMGS00/work/EC\_PAYROLL\_K3.xml' INTO gv\_filename.  
\*  ELSEIF u1 = 'X'.  
\*    CONCATENATE '/usr/sap/' sy-sysid '/DVEBMGS00/work/EC\_PAYROLL\_U1.xml' INTO gv\_filename.  
\*  ELSEIF u2 = 'X'.  
\*    CONCATENATE '/usr/sap/' sy-sysid '/DVEBMGS00/work/EC\_PAYROLL\_U2.xml' INTO gv\_filename.  
\*  ELSEIF u3 = 'X'.  
\*    CONCATENATE '/usr/sap/' sy-sysid '/DVEBMGS00/work/EC\_PAYROLL\_U3.xml' INTO gv\_filename.  
\*  ELSEIF u4 = 'X'.  
\*    CONCATENATE '/usr/sap/' sy-sysid '/DVEBMGS00/work/EC\_PAYROLL\_U4.xml' INTO gv\_filename.  
\*  ENDIF.  
  
  
  
\*  GV\_FILENAME = '/usr/sap/XOB/DVEBMGS00/work/EC\_Payroll.xml'.  
\* open the xml file for reading data  
  OPEN DATASET gv\_filename FOR INPUT IN TEXT MODE ENCODING DEFAULT.  
  IF sy-subrc NE 0.  
\*    MESSAGE ‘Error opening the XML file’ TYPE ‘E’.  
  ELSE.  
  
    DO.  
\* Transfer the contents from the file to the work area of the internal table  
      READ DATASET gv\_filename INTO wa\_xml\_tab.  
      IF sy-subrc EQ 0.  
        CONDENSE wa\_xml\_tab.  
\*       Append the contents of the work area to the internal table  
        APPEND wa\_xml\_tab TO g\_t\_xml\_tab.  
  
        IF wa\_xml\_tab+1(4) = 'EMPL'.  
          SPLIT wa\_xml\_tab AT '>' INTO lv\_string1 lv\_string2.  
          SPLIT lv\_string2 AT '<' INTO lv\_string2 lv\_string3.  
          lwa\_2010-pernr = lv\_string2.  
          CLEAR: lv\_string1, lv\_string2, lv\_string3.  
  
        ELSEIF wa\_xml\_tab+1(4) = 'VALI'.  
          SPLIT wa\_xml\_tab AT '>' INTO lv\_string1 lv\_string2.  
          SPLIT lv\_string2 AT '<' INTO lv\_string2 lv\_string3.  
          lwa\_2010-date  = lv\_string2.  
          CLEAR: lv\_string1, lv\_string2, lv\_string3.  
  
        ELSEIF wa\_xml\_tab+1(4) = 'WAGE'.  
          SPLIT wa\_xml\_tab AT '>' INTO lv\_string1 lv\_string2.  
          SPLIT lv\_string2 AT '<' INTO lv\_string2 lv\_string3.  
          lwa\_2010-lgart = lv\_string2.  
          CLEAR: lv\_string1, lv\_string2, lv\_string3.  
  
        ELSEIF wa\_xml\_tab+1(4) = 'NUMB'.  
          SPLIT wa\_xml\_tab AT '>' INTO lv\_string1 lv\_string2.  
          SPLIT lv\_string2 AT '<' INTO lv\_string2 lv\_string3.  
          lv\_aznhl = lv\_string2.  
          lv\_string2 = lv\_aznhl.  
          lwa\_2010-amount = lv\_string2+0(5).  
          CLEAR: lv\_string1, lv\_string2, lv\_string3.  
  
          CALL FUNCTION 'CONVERSION\_EXIT\_ALPHA\_INPUT'  
            EXPORTING  
              input  = lwa\_2010-pernr  
            IMPORTING  
              output = lwa\_2010-pernr.  
  
          APPEND lwa\_2010 TO lta\_2010.  
  
          ra\_pernr-sign   = 'I'.  
          ra\_pernr-option = 'EQ'.  
          ra\_pernr-low    = lwa\_2010-pernr."lowest EE number or exact EE number  
\*          ra\_pernr-high  = "highest EE number or empty (if only one EE)  
          APPEND ra\_pernr.  
  
          CLEAR lwa\_2010.  
  
        ELSEIF wa\_xml\_tab+23(11)  = 'PROJECT\_NUM'.  
          SPLIT wa\_xml\_tab AT '>' INTO lv\_string1 lv\_string2.  
          SPLIT lv\_string2 AT '<' INTO lv\_string2 lv\_string3.  
          lv\_proj = lv\_string2.  
          CLEAR: lv\_string1, lv\_string2, lv\_string3.  
  
        ELSEIF wa\_xml\_tab+23(4)  = 'ACCO'.  
          SPLIT wa\_xml\_tab AT '>' INTO lv\_string1 lv\_string2.  
          SPLIT lv\_string2 AT '<' INTO lv\_string2 lv\_string3.  
          lv\_acco = lv\_string2.  
          CLEAR: lv\_string1, lv\_string2, lv\_string3.  
  
        ELSEIF wa\_xml\_tab+23(4)  = 'TASK'.  
          SPLIT wa\_xml\_tab AT '>' INTO lv\_string1 lv\_string2.  
          SPLIT lv\_string2 AT '<' INTO lv\_string2 lv\_string3.  
          lv\_task = lv\_string2.  
          CLEAR: lv\_string1, lv\_string2, lv\_string3.  
  
        ELSEIF wa\_xml\_tab+1(4)  = 'EXTD'.  
          SPLIT wa\_xml\_tab AT '>' INTO lv\_string1 lv\_string2.  
          SPLIT lv\_string2 AT '<' INTO lv\_string2 lv\_string3.  
          lwa\_2010-extdoc = lv\_string2+2(8).  
          CLEAR: lv\_string1, lv\_string2, lv\_string3.  
  
        ELSEIF wa\_xml\_tab+23(4)  = 'SUBA'.  
          SPLIT wa\_xml\_tab AT '>' INTO lv\_string1 lv\_string2.  
          SPLIT lv\_string2 AT '<' INTO lv\_string2 lv\_string3.  
          lwa\_2010-suba = lv\_string2.  
          CONCATENATE  lwa\_2010-suba+0(3) '-' lwa\_2010-suba+3(2) '-' lwa\_2010-suba+5(2) INTO lwa\_2010-suba.  
          CLEAR: lv\_string1, lv\_string2, lv\_string3.  
  
\*WBS element & cust field both dervied from GL string  
  
        ELSEIF wa\_xml\_tab+19(4)  = 'WS17'.  
          SPLIT wa\_xml\_tab AT '>' INTO lv\_string1 lv\_string2.  
          SPLIT lv\_string2 AT '<' INTO lv\_string2 lv\_string3.  
  
          IF lv\_string2  NE '0000000000000000000000000000000'.  
            lwa\_2010-custfield = lv\_string2.  
  
\*Logic to pull company code from first 3 char of gl string  
            lwa\_2010-bukrs = lv\_string2+0(3).  
  
            IF     lwa\_2010-bukrs+0(1) = '5'.  
              CONCATENATE 'U' lwa\_2010-bukrs INTO lwa\_2010-bukrs.  
            ELSE.  
              CONCATENATE 'C' lwa\_2010-bukrs INTO lwa\_2010-bukrs.  
            ENDIF.  
  
            CONCATENATE lv\_string2+0(21) lv\_string2+28(3) INTO lwa\_2010-wbs.  
  
          ENDIF.  
  
        ENDIF.  
      ELSE.  
        EXIT.  
      ENDIF.  
    ENDDO.  
  ENDIF.  
  
\* Close the file after reading the data  
  CLOSE DATASET gv\_filename.  
  
\*Transfer the contents from the internal table to a string  
  IF NOT g\_t\_xml\_tab IS INITIAL.  
    CONCATENATE LINES OF g\_t\_xml\_tab INTO g\_str SEPARATED BY space.  
  ENDIF.  
  
  
\*Object values for the SLG log  
  lv\_log-object = 'Z2010\_DEL'.          "Object name  
  lv\_log-subobject = 'Z2010\_DEL'.  
  lv\_log-aluser = sy-uname.        "Username  
  lv\_log-alprog = sy-repid.          "Report name  
  
  
\*Function to Create the SLG log  
  CALL FUNCTION 'BAL\_LOG\_CREATE'  
    EXPORTING  
      i\_s\_log                 = lv\_log  
    IMPORTING  
      e\_log\_handle            = lv\_handle  
    EXCEPTIONS  
      log\_header\_inconsistent = 1  
      OTHERS                  = 2.  
  IF sy-subrc = 0.  
  ENDIF.  
  
  
  
\*\*\* program will only run for the pernrs from the file  
\*  IF pernr-pernr NOT IN ra\_pernr.  
\*    REJECT.  
\*  ENDIF.  
  
\*\*\* program will only run for the pernrs from the file  
  
\*  IF p\_extdoc IS NOT INITIAL.  
\*START-OF-SELECTION.  
  
\*\*\* will populate all infotyPes declared using "INFOTYPE"  keyword for the PERNRs in the selection screen  
\*GET PERNR.  
\*\*\* will fetch all the 2010 infotype records between the dates given in the selection screen and the external document number provided  
\*  PROVIDE \* FROM P2010 BETWEEN PN-BEGDA AND PN-ENDDA. "WHERE P2010-EXBEL = P\_EXTDOC.  
  
  SELECT  seqnr  
          subty  
          pernr  
          aedtm  
          uname  
          lgart  
          endda  
          begda  
          betrg  
          anzhl  
          prakn  
          versl  
          exbel  
          custfld  
    FROM pa2010 INTO TABLE  
    lta\_pa2010  
    WHERE pernr   IN p\_pernr AND  
          aedtm   IN p\_aedtm AND  
          custfld IN p\_cust AND  
          lgart   IN p\_lgart AND  
          exbel   IN p\_extdoc AND  
          uname   IN p\_user AND  
          begda   IN p\_begda.  
  IF sy-subrc = 0.  
  
    CLEAR: lwa\_pa2010.  
    LOOP AT lta\_pa2010 INTO lwa\_pa2010.  
      MOVE-CORRESPONDING lwa\_pa2010 TO wa\_p2010.  
      APPEND wa\_p2010 TO it\_p2010.  
      CLEAR: wa\_p2010.  
    ENDLOOP.  
  
  
\*    "DELETE P2010 WHERE EXBEL NE P\_EXTDOC.  
\*    IF P\_AEDTM IS NOT INITIAL.  
\*      DELETE LTA\_PA2010 WHERE AEDTM NE P\_AEDTM.  
\*    ENDIF.  
\*  
\*    "DELETE P2010 WHERE cust NE P\_EXTDOC.  
\*    IF P\_CUST IS NOT INITIAL.  
\*      DELETE LTA\_PA2010 WHERE CUSTFLD NOT IN P\_CUST.  
\*    ENDIF.  
\*  
\*    "DELETE P2010 WHERE lgart NE P\_lgart.  
\*    IF P\_LGART IS NOT INITIAL.  
\*      DELETE LTA\_PA2010 WHERE LGART NOT IN P\_LGART.  
\*    ENDIF.  
\*  
\*  
\*    "DELETE P2010 WHERE lgart NE P\_EXTDOC.  
\*    IF P\_EXTDOC IS NOT INITIAL.  
\*      DELETE  LTA\_PA2010  WHERE EXBEL NOT IN P\_EXTDOC.  
\*    ENDIF .  
\*  
\*  
\*    "DELETE P2010 WHERE lgart NE P\_EXTDOC.  
\*    IF P\_USER IS NOT INITIAL.  
\*      DELETE  LTA\_PA2010  WHERE UNAME NE P\_USER.  
\*    ENDIF.  
\*  
\*    "DELETE P2010 WHERE lgart NE P\_EXTDOC.  
\*    IF P\_BEGDA IS NOT INITIAL.  
\*      DELETE  LTA\_PA2010  WHERE BEGDA NE P\_BEGDA.  
\*    ENDIF.  
    CLEAR: wa\_p2010.  
    LOOP AT it\_p2010 INTO wa\_p2010.  
\*\*\*loop into the file for the PERNR being currently processed - to get the External document number  
      "READ TABLE LTA\_2010 INTO LWA\_2010 WITH KEY PERNR = PERNR-PERNR LGART = P\_LGART.  
      "IF SY-SUBRC = 0.  
  
\*    LOOP AT P2010.  
\*    MOVE-CORRESPONDING P2010 TO LWA\_PA2010.  
\*      LWA\_PA2010-PERNR = P2010-PERNR.  
\*      APPEND LWA\_PA2010 TO LTA\_PA2010.  
  
      IF  t1 NE 'X'.  
  
  
  
    WA\_SLG-MSGTY = 'S'.  
    WA\_SLG-MSGID = 'ZSLG'.  
    WA\_SLG-MSGNO = '007'.  
    CONCATENATE wa\_p2010-PERNR  wa\_p2010-lgart wa\_p2010-begda wa\_p2010-exbel INTO WA\_SLG-MSGV1 SEPARATED BY '/'.  
  
       CALL FUNCTION 'BAL\_LOG\_MSG\_ADD'  
      EXPORTING  
        I\_LOG\_HANDLE     =  lv\_handle  
        I\_S\_MSG          = WA\_SLG  
      EXCEPTIONS  
        LOG\_NOT\_FOUND    = 1  
        MSG\_INCONSISTENT = 2  
        LOG\_IS\_FULL      = 3  
        OTHERS           = 4.  
    IF SY-SUBRC <> 0.  
\* Implement suitable error handling here  
    ENDIF.  
  
  
        CLEAR  wa\_slg.  
  
\*  
\*          CALL FUNCTION 'BAL\_DB\_SAVE'  
\*            EXPORTING  
\*              I\_CLIENT         = SY-MANDT  
\*              I\_SAVE\_ALL       = 'X'  
\*            EXCEPTIONS  
\*              LOG\_NOT\_FOUND    = 1  
\*              SAVE\_NOT\_ALLOWED = 2  
\*              NUMBERING\_ERROR  = 3  
\*              OTHERS           = 4.  
\*          IF SY-SUBRC <> 0.  
\*\* Implement suitable error handling here  
\*          ENDIF.  
  
  
        CALL FUNCTION 'BAPI\_EMPLOYEE\_ENQUEUE'  
          EXPORTING  
            number = wa\_p2010-pernr  
          IMPORTING  
            return = wa\_bapireturn1.  
  
\*\*\* delete the infotype record  
          CALL FUNCTION 'HR\_INFOTYPE\_OPERATION'  
            EXPORTING  
              infty         = '2010'  
              number        = wa\_p2010-pernr  
              subtype       = wa\_p2010-subty  
\*             OBJECTID      = RECORD\_4002-OBJPS  
\*             LOCKINDICATOR = RECORD\_4002-SPRPS  
              validityend   = wa\_p2010-endda  
              validitybegin = wa\_p2010-begda  
              recordnumber  = wa\_p2010-seqnr  
              record        = wa\_p2010  
              operation     = 'DEL'  
              tclas         = 'A'  
              dialog\_mode   = '1'  
            IMPORTING  
              return        = wa\_retlist.  
  
\* Unlock Pernr  
          CALL FUNCTION 'BAPI\_EMPLOYEE\_DEQUEUE'  
            EXPORTING  
              number = lwa\_2010-pernr.  
  
  
  
  
\*\*\*Loop at the return message tab  
      IF wa\_retlist IS NOT INITIAL.  
        wa\_slg-msgty = wa\_retlist-type.  
        wa\_slg-msgid = wa\_retlist-id.  
        wa\_slg-msgno = wa\_retlist-number.  
        wa\_slg-msgv1 = wa\_retlist-message(50).  
        wa\_slg-msgv2 = wa\_retlist-message+49(50).  
        wa\_slg-msgv3 = wa\_retlist-message+99(50).  
        wa\_slg-msgv4 = wa\_retlist-message+149(50).  
      ELSE.  
        wa\_slg-msgty = 'S'.  
        wa\_slg-msgid = 000.  
        wa\_slg-msgno = 000.  
        CONCATENATE '2010 Record for Emp' wa\_p2010-pernr 'deleted successfully' INTO wa\_slg-msgv1 SEPARATED BY space.  
      ENDIF.  
  
  
\*add the values ot the slg log  
      CALL FUNCTION 'BAL\_LOG\_MSG\_ADD'  
        EXPORTING  
          i\_log\_handle     = lv\_handle  
          i\_s\_msg          = wa\_slg  
\* IMPORTING  
\*         E\_S\_MSG\_HANDLE   =  
\*         E\_MSG\_WAS\_LOGGED =  
\*         E\_MSG\_WAS\_DISPLAYED       =  
        EXCEPTIONS  
          log\_not\_found    = 1  
          msg\_inconsistent = 2  
          log\_is\_full      = 3  
          OTHERS           = 4.  
      IF sy-subrc <> 0.  
\* Implement suitable error handling here  
      ENDIF.  
  
  
\*          CALL FUNCTION 'BAL\_DB\_SAVE'  
\*            EXPORTING  
\*              I\_CLIENT         = SY-MANDT  
\*              I\_SAVE\_ALL       = 'X'  
\*            EXCEPTIONS  
\*              LOG\_NOT\_FOUND    = 1  
\*              SAVE\_NOT\_ALLOWED = 2  
\*              NUMBERING\_ERROR  = 3  
\*              OTHERS           = 4.  
\*          IF SY-SUBRC <> 0.  
\*\* Implement suitable error handling here  
\*          ENDIF.  
  
      CLEAR wa\_slg.  
  
  
  
      ENDIF.  
\*    ENDLOOP.  
  
  
  
    ENDLOOP.  
  ENDIF.  
  
\*  ENDPROVIDE.  
  
END-OF-SELECTION.  
  
  PERFORM build\_fieldcatalog.  
  PERFORM display\_alv\_report.  
\*  ENDIF.  
  
  
  
\*  IF p\_extdoc IS INITIAL.  
\*  
\*\*\*\* will fetch all the 2010 infotype records between the dates given in the selection screen and the external document number provided  
\*    PROVIDE \* FROM p2010 BETWEEN pn-begda AND pn-endda.  
\*  
\*      DELETE p2010 WHERE aedtm NE p\_aedtm.  
\*  
\*\*\*\*loop into the file for the PERNR being currently processed - to get the External document number  
\*      "READ TABLE LTA\_2010 INTO LWA\_2010 WITH KEY PERNR = PERNR-PERNR LGART = P\_LGART.  
\*      "IF SY-SUBRC = 0.  
\*      LOOP AT p2010.  
\*        CALL FUNCTION 'BAPI\_EMPLOYEE\_ENQUEUE'  
\*          EXPORTING  
\*            number = p2010-pernr  
\*          IMPORTING  
\*            return = wa\_bapireturn1.  
\*  
\*        IF wa\_bapireturn1-type NE 'E'.  
\*\*\*\* delete the infotype record  
\*          CALL FUNCTION 'HR\_INFOTYPE\_OPERATION'  
\*            EXPORTING  
\*              infty         = '2010'  
\*              number        = p2010-pernr  
\*              subtype       = p2010-subty  
\*\*             OBJECTID      = RECORD\_4002-OBJPS  
\*\*             LOCKINDICATOR = RECORD\_4002-SPRPS  
\*              validityend   = p2010-endda  
\*              validitybegin = p2010-begda  
\*              recordnumber  = p2010-seqnr  
\*              record        = p2010  
\*              operation     = 'DEL'  
\*              tclas         = 'A'  
\*              dialog\_mode   = '1'  
\*            IMPORTING  
\*              return        = wa\_retlist.  
\*  
\*\* Unlock Pernr  
\*          CALL FUNCTION 'BAPI\_EMPLOYEE\_DEQUEUE'  
\*            EXPORTING  
\*              number = lwa\_2010-pernr.  
\*        ENDIF.  
\*      ENDLOOP.  
\*  
\*\*\*\*Loop at the return message tab  
\*      IF wa\_retlist IS NOT INITIAL.  
\*        wa\_slg-msgty = wa\_retlist-type.  
\*        wa\_slg-msgid = wa\_retlist-id.  
\*        wa\_slg-msgno = wa\_retlist-number.  
\*        wa\_slg-msgv1 = wa\_retlist-message(50).  
\*        wa\_slg-msgv2 = wa\_retlist-message+49(50).  
\*        wa\_slg-msgv3 = wa\_retlist-message+99(50).  
\*        wa\_slg-msgv4 = wa\_retlist-message+149(50).  
\*      ELSE.  
\*        wa\_slg-msgty = 'S'.  
\*        wa\_slg-msgid = 000.  
\*        wa\_slg-msgno = 000.  
\*        CONCATENATE '2010 Record for Emp' p2010-pernr 'deleted successfully' INTO wa\_slg-msgv1 SEPARATED BY space.  
\*      ENDIF.  
\*  
\*\*add the values ot the slg log  
\*      CALL FUNCTION 'BAL\_LOG\_MSG\_ADD'  
\*        EXPORTING  
\*          i\_log\_handle     = lv\_handle  
\*          i\_s\_msg          = wa\_slg  
\*\* IMPORTING  
\*\*         E\_S\_MSG\_HANDLE   =  
\*\*         E\_MSG\_WAS\_LOGGED =  
\*\*         E\_MSG\_WAS\_DISPLAYED       =  
\*        EXCEPTIONS  
\*          log\_not\_found    = 1  
\*          msg\_inconsistent = 2  
\*          log\_is\_full      = 3  
\*          OTHERS           = 4.  
\*      IF sy-subrc <> 0.  
\*\* Implement suitable error handling here  
\*      ENDIF.  
\*  
\*      CLEAR wa\_slg.  
\*  
\*      "ENDIF.  
\*  
\*    ENDPROVIDE.  
\*  
\*  ENDIF.  
  
  IF lv\_handle IS NOT INITIAL.  
    INSERT lv\_handle INTO TABLE lt\_handle.  
    CALL FUNCTION 'BAL\_DB\_SAVE'  
      EXPORTING  
        i\_client         = sy-mandt  
\*       I\_IN\_UPDATE\_TASK = ' '  
        i\_save\_all       = 'X'  
        i\_t\_log\_handle   = lt\_handle  
\*       I\_2TH\_CONNECTION = ' '  
\*       I\_2TH\_CONNECT\_COMMIT       = ' '  
\*       I\_LINK2JOB       = 'X'  
\* IMPORTING  
\*       E\_NEW\_LOGNUMBERS =  
\*       E\_SECOND\_CONNECTION        =  
      EXCEPTIONS  
        log\_not\_found    = 1  
        save\_not\_allowed = 2  
        numbering\_error  = 3  
        OTHERS           = 4.  
    IF sy-subrc <> 0.  
\* Implement suitable error handling here  
    ENDIF.  
  ENDIF.  
  
  
  
\*&---------------------------------------------------------------------\*  
\*&      Form  BUILD\_FIELDCATALOG  
\*&---------------------------------------------------------------------\*  
\*       text  
\*----------------------------------------------------------------------\*  
\*  -->  p1        text  
\*  <--  p2        text  
\*----------------------------------------------------------------------\*  
FORM build\_fieldcatalog .  
  fieldcatalog-fieldname   = 'PERNR'.  
  fieldcatalog-seltext\_m   = 'Employee ID'.  
  fieldcatalog-col\_pos     = 0.  
  APPEND fieldcatalog TO fieldcatalog.  
  CLEAR  fieldcatalog.  
  
  fieldcatalog-fieldname   = 'AEDTM'.  
  fieldcatalog-seltext\_m   = 'Changed On'.  
  fieldcatalog-col\_pos     = 1.  
  APPEND fieldcatalog TO fieldcatalog.  
  CLEAR  fieldcatalog.  
  
  fieldcatalog-fieldname   = 'UNAME'.  
  fieldcatalog-seltext\_m   = 'Name of Person Who Changed Object'.  
  fieldcatalog-col\_pos     = 2.  
  APPEND fieldcatalog TO fieldcatalog.  
  CLEAR  fieldcatalog.  
  
  fieldcatalog-fieldname   = 'LGART'.  
  fieldcatalog-seltext\_m   = 'Wage Type'.  
  fieldcatalog-col\_pos     = 3.  
  fieldcatalog-outputlen   = 20.  
  APPEND fieldcatalog TO fieldcatalog.  
  CLEAR  fieldcatalog.  
  
  fieldcatalog-fieldname   = 'ENDDA'.  
  fieldcatalog-seltext\_m   = 'End Date'.  
  fieldcatalog-col\_pos     = 3.  
  fieldcatalog-outputlen   = 20.  
  APPEND fieldcatalog TO fieldcatalog.  
  CLEAR  fieldcatalog.  
  
  fieldcatalog-fieldname   = 'BEGDA'.  
  fieldcatalog-seltext\_m   = 'Start Date'.  
  fieldcatalog-col\_pos     = 3.  
  fieldcatalog-outputlen   = 20.  
  APPEND fieldcatalog TO fieldcatalog.  
  CLEAR  fieldcatalog.  
  
  fieldcatalog-fieldname   = 'BETRG'.  
  fieldcatalog-seltext\_m   = 'Wage Type Amount for Payments'.  
  fieldcatalog-col\_pos     = 3.  
  fieldcatalog-outputlen   = 20.  
  APPEND fieldcatalog TO fieldcatalog.  
  CLEAR  fieldcatalog.  
  
  fieldcatalog-fieldname   = 'ANZHL'.  
  fieldcatalog-seltext\_m   = 'Number per Time Unit for EE Remuneration Info'.  
  fieldcatalog-col\_pos     = 3.  
  fieldcatalog-outputlen   = 20.  
  APPEND fieldcatalog TO fieldcatalog.  
  CLEAR  fieldcatalog.  
  
  fieldcatalog-fieldname   = 'PRAKN'.  
  fieldcatalog-seltext\_m   = 'Premium Number'.  
  fieldcatalog-col\_pos     = 3.  
  fieldcatalog-outputlen   = 20.  
  APPEND fieldcatalog TO fieldcatalog.  
  CLEAR  fieldcatalog.  
  
  fieldcatalog-fieldname   = 'VERSL'.  
  fieldcatalog-seltext\_m   = 'Overtime Compensation Type'.  
  fieldcatalog-col\_pos     = 3.  
  fieldcatalog-outputlen   = 20.  
  APPEND fieldcatalog TO fieldcatalog.  
  CLEAR  fieldcatalog.  
  
  fieldcatalog-fieldname   = 'EXBEL'.  
  fieldcatalog-seltext\_m   = 'External Document Number'.  
  fieldcatalog-col\_pos     = 3.  
  fieldcatalog-outputlen   = 20.  
  APPEND fieldcatalog TO fieldcatalog.  
  CLEAR  fieldcatalog.  
  
  fieldcatalog-fieldname   = 'CUSTFLD'.  
  fieldcatalog-seltext\_m   = 'Customer-Specific Field'.  
  fieldcatalog-col\_pos     = 3.  
  fieldcatalog-outputlen   = 20.  
  APPEND fieldcatalog TO fieldcatalog.  
  CLEAR  fieldcatalog.  
  
ENDFORM.                    " BUILD\_FIELDCATALOG  
  
  
\*&---------------------------------------------------------------------\*  
\*&      Form  DISPLAY\_ALV\_REPORT  
\*&---------------------------------------------------------------------\*  
\*       text  
\*----------------------------------------------------------------------\*  
\*  -->  p1        text  
\*  <--  p2        text  
\*----------------------------------------------------------------------\*  
FORM display\_alv\_report .  
  gd\_repid = sy-repid.  
  
  SORT lta\_pa2010 BY pernr aedtm.  
  
  CALL FUNCTION 'REUSE\_ALV\_GRID\_DISPLAY'  
    EXPORTING  
      i\_callback\_program      = gd\_repid  
\*     I\_CALLBACK\_TOP\_OF\_PAGE  = 'TOP-OF-PAGE'  "see FORM  
      i\_callback\_user\_command = 'USER\_COMMAND'  
      it\_fieldcat             = fieldcatalog[]  
      i\_save                  = 'X'  
\*     IS\_VARIANT              = G\_VARIANT  
    TABLES  
      t\_outtab                = lta\_pa2010  
    EXCEPTIONS  
      program\_error           = 1  
      OTHERS                  = 2.  
  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.                    "DISPLAY\_ALV\_REPORT  
" DISPLAY\_ALV\_REPORTzhr\*