\*&---------------------------------------------------------------------\*  
\*& Report  ZXML  
\*&  
\*&---------------------------------------------------------------------\*  
\*& Report to upload 2010 data  
\*&\*  
\*&---------------------------------------------------------------------\*  
REPORT ZHR\_2010\_UPDATE\_WITHWBS.  
  
TABLES PA0008.  
  
\*  
TYPES: BEGIN OF TY\_XML\_LINE,  
         DATA(256) TYPE X,  
       END OF TY\_XML\_LINE.  
  
TYPES: BEGIN OF TY\_LGART,  
         LGART TYPE LGART,  
       END OF TY\_LGART.  
  
TYPES: BEGIN OF T\_2010UPDT,  
         PERNR     TYPE BDC\_FVAL,  
         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,  
         RATE      TYPE BDC\_FVAL,  
         UNITS     TYPE BDC\_FVAL,  
         PRAKN     TYPE BDC\_FVAL,  
         PRAKZ     TYPE BDC\_FVAL,  
         REVER     TYPE BDC\_FVAL,  
         KOSTL     TYPE BDC\_FVAL,  
         MAJOR     TYPE BDC\_FVAL,  
         FLAG      TYPE BDC\_FVAL,  
         TRFGR     TYPE BDC\_FVAL,  
         TRFST     TYPE BDC\_FVAL,  
         FRINGE    TYPE BDC\_FVAL,  
         CALC      TYPE BDC\_FVAL,  
         FWAGERT   TYPE BDC\_FVAL,  
         FFRINGRT  TYPE BDC\_FVAL,  
         SHIFTPREM TYPE BDC\_FVAL,  
         TAXOVER   TYPE BDC\_FVAL,  
       END      OF T\_2010UPDT.  
  
DATA: LV\_EXBEL TYPE EXBEL,  
      LV\_STDAZ  TYPE ENSTD,  
      LV\_BWGRL  TYPE PTM\_VBAS7S,  
      LV\_PERNR1 TYPE PERNR\_D,  
      LV\_SUBTY TYPE SUBTY,  
      LV\_BEGDA TYPE BEGDA.  
  
  
  
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,  
      LWA\_LGART   TYPE TY\_LGART,  
      LWA\_LGART\_C TYPE TY\_LGART,  
      LTA\_LGART\_C TYPE TABLE OF TY\_LGART,  
      LTA\_LGART   TYPE TABLE OF TY\_LGART,  
      LV\_BTRTL    TYPE PA0001-BTRTL,  
      LV\_LENGTH   TYPE I,  
      LWA\_PA0008  TYPE PA0008,  
      LTA\_PA0008  TYPE TABLE OF PA0008,  
      LV\_TOTAL    TYPE PAD\_AMT7S,  
      LV\_T510RTE  TYPE PAD\_AMT7S,  
      LV\_PW\_RATE  TYPE CHAR20,  
      LV\_LD8      TYPE CHAR1,  
      LV\_LD9      TYPE CHAR1.  
  
DATA: LV\_VAR  TYPE BDC\_FVAL,  
      LV\_CHAR TYPE C LENGTH 2.  
  
  
DATA: LV\_NUM\_I TYPE I.  
  
\*      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,  
      LT\_HANDLE   TYPE BALLOGHNDL,  
      LV\_INDEX    TYPE I,  
      LV\_LEN1     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,  
      LV\_LEN      TYPE I,  
      LV\_WAGERATE TYPE CHAR1,  
      LV\_FLAG     TYPE C LENGTH 1,  
      LWA\_FORE    TYPE ZFOREMANFLG,  
      LTA\_FORE    TYPE TABLE OF ZFOREMANFLG.  
\*      LV\_KOSTL TYPE KOSTL.  
  
\*range for wageyptes  
TYPES:RNG\_LGART TYPE RANGE OF LGART.  
DATA: RNG\_TBL TYPE RNG\_LGART,  
      WA\_RNG  TYPE LINE OF RNG\_LGART.  
  
SELECTION-SCREEN BEGIN OF BLOCK B1 WITH FRAME TITLE TEXT-001.  
PARAMETERS : REPROCES RADIOBUTTON GROUP RB1.  
PARAMETERS : FULL RADIOBUTTON GROUP RB1.  
  
SELECTION-SCREEN END OF BLOCK B1.  
  
SELECTION-SCREEN BEGIN OF BLOCK B2 WITH FRAME TITLE TEXT-002.  
PARAMETERS : K1 RADIOBUTTON GROUP RB2.  
PARAMETERS : K2 RADIOBUTTON GROUP RB2.  
PARAMETERS : K3 RADIOBUTTON GROUP RB2.  
PARAMETERS : U1 RADIOBUTTON GROUP RB2.  
PARAMETERS : U2 RADIOBUTTON GROUP RB2.  
PARAMETERS : U3 RADIOBUTTON GROUP RB2.  
PARAMETERS : U4 RADIOBUTTON GROUP RB2.  
SELECTION-SCREEN END OF BLOCK B2.  
  
  
START-OF-SELECTION.  
  
  "Call the WBS automation program to create all Projects and WBS elements  
  SUBMIT ZHR\_CREATE\_WBS WITH K1 = K1  
                        WITH K2 = K2  
                        WITH K3 = K3  
                        WITH U1 = U1  
                        WITH U2 = U2  
                        WITH U3 = U3  
   AND RETURN.  
  
  
  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.  
  
\* 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.  
  
\*so for these wagetypes I should map the units instead of the amount,  
\*and then put the rate i the extra pay/valuation field  
  
        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, LWA\_PA0008, LTA\_PA0008, LV\_TOTAL,  LV\_PW\_RATE, LV\_T510RTE .  
  
\*\*\*\*\*\*\*\*\*\*\*\*  
\*FBus-159 acting foreman changes  
\* had to move location of forman logic as the date in the file was need to read the table rather than sy-datum  
\*\*\*\*\*\*\*\*\*\*\*\*  
  
\*select current it8 record as values are needed for logic  
          SELECT \* FROM PA0008  
            INTO  TABLE LTA\_PA0008  
            WHERE PERNR = LWA\_2010-PERNR AND  
                    BEGDA <= LWA\_2010-DATE AND  
          ENDDA >= LWA\_2010-DATE.  
  
          IF SY-SUBRC = 0.  
            READ TABLE LTA\_PA0008 INTO LWA\_PA0008  INDEX 1.  
  
  
  
  
\*add values to compare later (t510) value).  
            ADD LWA\_PA0008-BET01 TO LV\_TOTAL.  
            ADD LWA\_PA0008-BET02 TO LV\_TOTAL.  
          ENDIF.  
  
  
          IF LV\_LD8 = 'Y' OR LV\_LD9 = 'Y'.  
\*select employees pay scale lead level from the custom table  
            SELECT \* FROM   ZFOREMANFLG  
              INTO TABLE LTA\_FORE  
                  WHERE    TRFGR = LWA\_PA0008-TRFGR AND  
                 BEGDA <= LWA\_2010-DATE AND  
                 ENDDA >= LWA\_2010-DATE.  
  
            IF SY-SUBRC = 0.  
  
\*loop at foreman entries  
              LOOP AT LTA\_FORE INTO  LWA\_FORE.  
  
\*if the lead level is not inital.  
  
                LWA\_2010-TRFGR = LWA\_PA0008-TRFGR.  
  
                IF LV\_LD8 = 'Y' AND LWA\_FORE-FORLEVL IS NOT INITIAL.  
                  LWA\_2010-TRFST = LWA\_FORE-FORLEVL.  
                ELSEIF LV\_LD9 = 'Y' AND LWA\_FORE-LEADLVL IS NOT INITIAL.  
                  LWA\_2010-TRFST = LWA\_FORE-LEADLVL.  
                ENDIF.  
  
  
              ENDLOOP.  
  
  
  
              IF LWA\_2010-TRFST  IS INITIAL.  
                CLEAR LWA\_2010-TRFGR.  
              ENDIF.  
  
\*check rate against t510  
\*select rate from t510 based on IT8 values and foreman level  
              SELECT SINGLE BETRG FROM T510 INTO  LV\_T510RTE  
                WHERE MOLGA = '10' AND  
                      TRFAR = LWA\_PA0008-TRFAR AND  
                      TRFGB = LWA\_PA0008-TRFGB AND  
                      TRFGR = LWA\_PA0008-TRFGR AND  
                      TRFST = LWA\_2010-TRFST AND  
                      BEGDA <= LWA\_2010-DATE AND  
                      ENDDA >= LWA\_2010-DATE.  
  
\*if a value is found proceed with logic  
\*\*Compare rate from IT0008 total cumulated and Rate found in T510. If Rate from T510 is lower  
\*  than IT0008 total clear LWA\_2010-TRFGR.  
  
              IF  LV\_T510RTE IS NOT INITIAL.  
  
  
                IF LV\_TOTAL GT LV\_T510RTE.  
                  CLEAR LWA\_2010-TRFGR.  
                  CLEAR LWA\_2010-TRFST.  
                ENDIF.  
  
              ENDIF.  
  
            ENDIF.  
  
          ENDIF.  
  
  
\*if the wage type is one that is needed for prevailaing wage move updated rate to the based wage type evaulation field  
          IF LWA\_2010-FRINGE = 'Y'.  
            IF LWA\_2010-LGART = '1100' OR LWA\_2010-LGART = '1101' OR LWA\_2010-LGART = '1102' OR LWA\_2010-LGART = '1107' OR LWA\_2010-LGART = '1120' OR LWA\_2010-LGART = '1131'  
                OR LWA\_2010-LGART = '1132' OR LWA\_2010-LGART = '1133' OR LWA\_2010-LGART = '1134'.  
              LWA\_2010-RATE = LWA\_2010-FWAGERT.  
            ENDIF.  
          ENDIF.  
  
\*FBUS-207  
\*validation to clear the rate if less than the total  
\*select current it8 record as values are needed for logic  
\*only done for wage type 1100  
  
          IF LWA\_2010-LGART = '1100' OR LWA\_2010-LGART = '1101' OR LWA\_2010-LGART = '1102' OR LWA\_2010-LGART = '1107' OR LWA\_2010-LGART = '1120' OR LWA\_2010-LGART = '1131'  
                OR LWA\_2010-LGART = '1132' OR LWA\_2010-LGART = '1133' OR LWA\_2010-LGART = '1134'.  
  
\*if usd4 divide by 100  
  
            IF LWA\_PA0008-WAERS = 'USD4'.  
              LV\_TOTAL = LV\_TOTAL / 100.  
            ENDIF.  
  
            IF LWA\_PA0008-WAERS+0(2) = 'US'.  
              IF ( LWA\_2010-RATE LT LV\_TOTAL OR LV\_WAGERATE = '2') AND LWA\_2010-RATE NE '0'.  
                LV\_PW\_RATE = LWA\_2010-RATE.  
                CLEAR LWA\_2010-RATE.  
              ENDIF.  
            ENDIF.  
  
          ENDIF.  
  
          APPEND LWA\_2010 TO LTA\_2010.  
  
\*logic for prevailing project  
          IF LWA\_2010-FRINGE = 'Y'.  
  
            IF LWA\_2010-LGART = '1100' OR LWA\_2010-LGART = '1107' OR LWA\_2010-LGART = '1120'.  
  
\*added after load in xob 200 maybe data related where the fringe rates in the file were blank  
              IF LWA\_2010-FFRINGRT NE '0' AND LV\_WAGERATE = '1'.  
  
                IF LWA\_2010-CALC = '1'.  
                  LWA\_2010-LGART = '9FRR'.  
                  LWA\_2010-RATE = LWA\_2010-FFRINGRT.  
                  APPEND LWA\_2010 TO LTA\_2010.  
                ELSEIF  LWA\_2010-CALC = '2'.  
                  LWA\_2010-LGART = '9FRP'.  
                  LWA\_2010-RATE = LWA\_2010-FFRINGRT.  
                  APPEND LWA\_2010 TO LTA\_2010.  
                ENDIF.  
  
  
              ENDIF.  
  
\*Fbus-214 PW rate logic. If there is a value in the place holder rate then create a wage type  
\* with that rate for 9PWG  
\*lv\_wagerate = '1'  
  
              IF LV\_PW\_RATE  IS NOT INITIAL AND LV\_WAGERATE NE '3' .  
                LWA\_2010-LGART = '9PWG'.  
                LWA\_2010-RATE = LV\_PW\_RATE.  
                APPEND LWA\_2010 TO LTA\_2010.  
              ENDIF.  
  
  
\*FBUS-277 Update 2010 load for overtime  
            ELSEIF LWA\_2010-LGART = '1101' OR LWA\_2010-LGART = '1102' OR LWA\_2010-LGART = '1131'  
                OR LWA\_2010-LGART = '1132' OR LWA\_2010-LGART = '1133' OR LWA\_2010-LGART = '1134'.  
  
\*added after load in xob 200 maybe data related where the fringe rates in the file were blank  
              IF LWA\_2010-FFRINGRT NE '0' AND LV\_WAGERATE = '1'.  
  
                IF LWA\_2010-CALC = '1'.  
                  LWA\_2010-LGART = '9FOR'.  
                  LWA\_2010-RATE = LWA\_2010-FFRINGRT.  
                  APPEND LWA\_2010 TO LTA\_2010.  
                ELSEIF  LWA\_2010-CALC = '2'.  
                  LWA\_2010-LGART = '9FOP'.  
                  LWA\_2010-RATE = LWA\_2010-FFRINGRT.  
                  APPEND LWA\_2010 TO LTA\_2010.  
                ENDIF.  
  
              ENDIF.  
  
\*Fbus-214 PW rate logic. If there is a value in the place holder rate then create a wage type  
\* with that rate for 9PWG  
\*lv\_wagerate = '1'  
  
              IF LV\_PW\_RATE  IS NOT INITIAL AND LV\_WAGERATE NE '3' .  
                LWA\_2010-LGART = '9PWG'.  
                LWA\_2010-RATE = LV\_PW\_RATE.  
                APPEND LWA\_2010 TO LTA\_2010.  
              ENDIF.  
  
  
            ENDIF.  
  
  
  
          ENDIF.  
  
\*If its a shift premium create the wage tyype 1210 entry  
          IF LWA\_2010-SHIFTPREM = 'Y'.  
  
            LWA\_2010-LGART = '1210'.  
            APPEND LWA\_2010 TO LTA\_2010.  
          ENDIF.  
  
          CLEAR: LWA\_2010, LV\_LD8,  LV\_LD9.  
  
        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(11)  = 'PROJECT\_LOC'.  
          SPLIT WA\_XML\_TAB AT '>' INTO LV\_STRING1 LV\_STRING2.  
          SPLIT LV\_STRING2 AT '<' INTO LV\_STRING2 LV\_STRING3.  
          LWA\_2010-TAXOVER = 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.  
  
\*fbus 223 wage rate type tag  
        ELSEIF WA\_XML\_TAB+24(4)  = 'RATE'.  
          SPLIT WA\_XML\_TAB AT '>' INTO LV\_STRING1 LV\_STRING2.  
          SPLIT LV\_STRING2 AT '<' INTO LV\_STRING2 LV\_STRING3.  
          LV\_WAGERATE = 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.  
          LV\_LENGTH = STRLEN( LV\_STRING2 ).  
          IF LV\_LENGTH = '10'.  
            LWA\_2010-EXTDOC = LV\_STRING2+2(8).  
          ELSEIF LV\_LENGTH = '11'.  
            LWA\_2010-EXTDOC = LV\_STRING2+3(8).  
          ELSEIF LV\_LENGTH = '12'.  
            LWA\_2010-EXTDOC = LV\_STRING2+4(8).  
          ENDIF..  
          CLEAR: LV\_STRING1, LV\_STRING2, LV\_STRING3, LV\_LENGTH.  
  
        ELSEIF WA\_XML\_TAB+19(4)  = 'RATE'.  
          SPLIT WA\_XML\_TAB AT '>' INTO LV\_STRING1 LV\_STRING2.  
          SPLIT LV\_STRING2 AT '<' INTO LV\_STRING2 LV\_STRING3.  
          LWA\_2010-RATE = LV\_STRING2.  
          CLEAR: LV\_STRING1, LV\_STRING2, LV\_STRING3.  
  
        ELSEIF WA\_XML\_TAB+19(4)  = 'UNIT'.  
          SPLIT WA\_XML\_TAB AT '>' INTO LV\_STRING1 LV\_STRING2.  
          SPLIT LV\_STRING2 AT '<' INTO LV\_STRING2 LV\_STRING3.  
          LWA\_2010-UNITS = LV\_STRING2.  
          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.  
  
          "&-------------------------------------------------------------------------------------------------------------&" ++Kriba "JIRA 490  
\* Jira-568  
\*Jira 1474 - had to switch back for night/day shift as they were loading wrong  
        ELSEIF WA\_XML\_TAB+19(3)  = 'LD5'.  
\*        ELSEIF wa\_xml\_tab+13(3)  = 'LD5'.  
\* Jira-568  
          SPLIT WA\_XML\_TAB AT '>' INTO LV\_STRING1 LV\_STRING2.  
          SPLIT LV\_STRING2 AT '<' INTO LV\_STRING2 LV\_STRING3.  
          LWA\_2010-PRAKN = 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.  
          "&-------------------------------------------------------------------------------------------------------------&"  
  
          "&-------------------------------------------------------------------------------------------------------------&" ++Kriba "JIRA 805  
        ELSEIF WA\_XML\_TAB(8)  = '<REVERSE'.  
          SPLIT WA\_XML\_TAB AT '>' INTO LV\_STRING1 LV\_STRING2.  
          SPLIT LV\_STRING2 AT '<' INTO LV\_STRING2 LV\_STRING3.  
          LWA\_2010-REVER = 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.  
          "&-------------------------------------------------------------------------------------------------------------&"  
  
          "&-------------------------------------------------------------------------------------------------------------&" ++Mike "JIRA 568  
        ELSEIF WA\_XML\_TAB+13(3)  = 'LD7'.  
          CLEAR: LV\_STRING2.  
          SPLIT WA\_XML\_TAB AT '>' INTO LV\_STRING1 LV\_STRING2.  
          SPLIT LV\_STRING2 AT '<' INTO LV\_STRING2 LV\_STRING3.  
\* Jira-568  
          IF LV\_STRING2 = 'Y'.  
            LWA\_2010-MAJOR = LV\_STRING2.  
\*           CONCATENATE lwa\_2010-major lwa\_2010-major INTO lwa\_2010-major.  
          ELSE.  
            CLEAR: LWA\_2010-MAJOR.  
          ENDIF.  
\* Jira-568  
\*          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.  
  
          "&-------------------------------------------------------------------------------------------------------------&" ++Mike Jira FBUS-187  
        ELSEIF WA\_XML\_TAB+19(4)  = 'WS21'.  
          CLEAR: LV\_STRING2.  
          SPLIT WA\_XML\_TAB AT '>' INTO LV\_STRING1 LV\_STRING2.  
          SPLIT LV\_STRING2 AT '<' INTO LV\_STRING2 LV\_STRING3.  
  
  
          IF LV\_STRING2 = 'N'.  
            LWA\_2010-MAJOR = ' '.  
\*           CONCATENATE lwa\_2010-major lwa\_2010-major INTO lwa\_2010-major.  
          ELSEIF  LV\_STRING2 = 'W'.  
            LWA\_2010-MAJOR = 'W'.  
          ELSEIF  LV\_STRING2 = 'L'.  
            LWA\_2010-MAJOR = 'L'.  
          ELSEIF  LV\_STRING2 = 'A'.  
            LWA\_2010-MAJOR = 'A'.  
          ENDIF.  
  
          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.  
  
          CLEAR: LV\_LEN.  
          LV\_LEN = STRLEN( LV\_STRING2 ).  
  
          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.  
  
\* if the cust field is 24 instead of 31 - meaning without the Cost center - set the whole string as WBS elements  
\* and later in the code set the Cos Center to the default cost center of the employee  
          IF LV\_LEN < 31.  
            LWA\_2010-FLAG = 'X'.  
            LWA\_2010-WBS = LV\_STRING2.  
          ELSE.  
            IF LV\_STRING2  NE '0000000000000000000000000000000'.  
              CONCATENATE LV\_STRING2+0(21) LV\_STRING2+28(3) INTO LWA\_2010-WBS.  
            ENDIF.  
          ENDIF.  
  
\* if cust field has Subaccount as 0000000  
          IF  LV\_STRING2+21(7) EQ '0000000'.  
            LWA\_2010-FLAG = 'X'.  
          ENDIF.  
  
\*Shift Premium  
        ELSEIF WA\_XML\_TAB+13(4)   = 'LD10'.  
          SPLIT WA\_XML\_TAB AT '>' INTO LV\_STRING1 LV\_STRING2.  
          SPLIT LV\_STRING2 AT '<' INTO LV\_STRING2 LV\_STRING3.  
          IF LV\_STRING2 = 'Y'.  
            LWA\_2010-SHIFTPREM = LV\_STRING2.  
          ENDIF.  
  
\*Store value for foreman flags depending on if its ld8 set flag to Y for processing later  
        ELSEIF WA\_XML\_TAB+13(3)   = 'LD8' .  
  
          SPLIT WA\_XML\_TAB AT '>' INTO LV\_STRING1 LV\_STRING2.  
          SPLIT LV\_STRING2 AT '<' INTO LV\_STRING2 LV\_STRING3.  
          IF LV\_STRING2 = 'Y'.  
            LV\_LD8 = 'Y'.  
          ENDIF.  
  
\*Store value for foreman flags depending on if its ld8 set flag to Y for processing later  
        ELSEIF WA\_XML\_TAB+13(3)   = 'LD9'.  
  
          SPLIT WA\_XML\_TAB AT '>' INTO LV\_STRING1 LV\_STRING2.  
          SPLIT LV\_STRING2 AT '<' INTO LV\_STRING2 LV\_STRING3.  
          IF LV\_STRING2 = 'Y'.  
            LV\_LD9 = 'Y'.  
          ENDIF.  
  
  
\*USA changes. If prevaling wage project set flag value  
        ELSEIF WA\_XML\_TAB+31(15) = 'NG\_WAGE\_PROJECT'.  
          IF WA\_XML\_TAB+47(1) = 'Y'.  
            LWA\_2010-FRINGE = 'Y'.  
          ENDIF.  
  
\*Prevaling Wage reate  
        ELSEIF  WA\_XML\_TAB+25(15) = 'REVAILING\_WAGE>'.  
          SPLIT WA\_XML\_TAB AT '>' INTO LV\_STRING1 LV\_STRING2.  
          SPLIT LV\_STRING2 AT '<' INTO LV\_STRING2 LV\_STRING3.  
          LWA\_2010-FWAGERT = LV\_STRING2.  
          CLEAR: LV\_STRING1, LV\_STRING2, LV\_STRING3.  
  
\*Prevaling fringe rate  
        ELSEIF WA\_XML\_TAB+25(15) = 'REVAILING\_FRING'.  
          SPLIT WA\_XML\_TAB AT '>' INTO LV\_STRING1 LV\_STRING2.  
          SPLIT LV\_STRING2 AT '<' INTO LV\_STRING2 LV\_STRING3.  
          LWA\_2010-FFRINGRT = LV\_STRING2.  
          CLEAR: LV\_STRING1, LV\_STRING2, LV\_STRING3.  
  
\*Calculation Method  
        ELSEIF WA\_XML\_TAB+26(15) = 'CALCULATION\_MET'.  
          SPLIT WA\_XML\_TAB AT '>' INTO LV\_STRING1 LV\_STRING2.  
          SPLIT LV\_STRING2 AT '<' INTO LV\_STRING2 LV\_STRING3.  
          LWA\_2010-CALC = LV\_STRING2.  
          CLEAR: LV\_STRING1, LV\_STRING2, LV\_STRING3.  
  
  
        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'.          "Object name  
  LV\_LOG-SUBOBJECT = 'Z2010LOAD'.  
  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            = LT\_HANDLE  
    EXCEPTIONS  
      LOG\_HEADER\_INCONSISTENT = 1  
      OTHERS                  = 2.  
  IF SY-SUBRC = 0.  
  ENDIF.  
  
\*LLoop at the 2010 values  
  LOOP AT LTA\_2010 INTO LWA\_2010.  
\*    IF LWA\_2010-WBS IS NOT INITIAL.  
    CLEAR: LV\_KOSTL.  
  
    WA\_SLG-MSGTY = 'S'.  
    WA\_SLG-MSGID = 'ZSLG'.  
    WA\_SLG-MSGNO = '000'.  
    CONCATENATE ' ' LWA\_2010-EXTDOC INTO WA\_SLG-MSGV1.  
  
    CALL FUNCTION 'BAL\_LOG\_MSG\_ADD'  
      EXPORTING  
        I\_LOG\_HANDLE     = LV\_MSGHND  
        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.  
  
    WA\_SLG-MSGTY = 'S'.  
    WA\_SLG-MSGID = 'ZSLG'.  
    WA\_SLG-MSGNO = '001'.  
    CONCATENATE ' ' LWA\_2010-PERNR INTO WA\_SLG-MSGV1.  
  
    CALL FUNCTION 'BAL\_LOG\_MSG\_ADD'  
      EXPORTING  
        I\_LOG\_HANDLE     = LV\_MSGHND  
        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.  
  
  
    WA\_SLG-MSGTY = 'S'.  
    WA\_SLG-MSGID = 'ZSLG'.  
    WA\_SLG-MSGNO = '002'.  
    CONCATENATE ' ' LWA\_2010-LGART INTO WA\_SLG-MSGV1.  
  
    CALL FUNCTION 'BAL\_LOG\_MSG\_ADD'  
      EXPORTING  
        I\_LOG\_HANDLE     = LV\_MSGHND  
        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.  
  
  
    WA\_SLG-MSGTY = 'S'.  
    WA\_SLG-MSGID = 'ZSLG'.  
    WA\_SLG-MSGNO = '003'.  
    CONCATENATE ' ' LWA\_2010-AMOUNT INTO WA\_SLG-MSGV1.  
  
    CALL FUNCTION 'BAL\_LOG\_MSG\_ADD'  
      EXPORTING  
        I\_LOG\_HANDLE     = LV\_MSGHND  
        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 wbs element variable  
    CLEAR: LV\_POSID.  
  
    IF LWA\_2010-WBS+3  NE '000000000000000000000'.  
      IF LWA\_2010-WBS IS NOT INITIAL.  
        SELECT SINGLE POSID FROM PRPS  
          INTO  LV\_POSID  
    WHERE POSID = LWA\_2010-WBS.  
        IF SY-SUBRC NE 0.  
  
          WA\_SLG-MSGTY = 'E'.  
          WA\_SLG-MSGID = 'ZSLG'.  
          WA\_SLG-MSGNO = '004'.  
          CONCATENATE ' ' LWA\_2010-WBS INTO WA\_SLG-MSGV1.  
  
          CALL FUNCTION 'BAL\_LOG\_MSG\_ADD'  
            EXPORTING  
              I\_LOG\_HANDLE     = LV\_MSGHND  
              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.  
  
          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.  
  
  
\*process next record  
          CONTINUE.  
  
        ELSE.  
          WA\_SLG-MSGTY = 'S'.  
          WA\_SLG-MSGID = 'ZSLG'.  
          WA\_SLG-MSGNO = '006'.  
          CONCATENATE ' ' LWA\_2010-WBS INTO WA\_SLG-MSGV1.  
  
          CALL FUNCTION 'BAL\_LOG\_MSG\_ADD'  
            EXPORTING  
              I\_LOG\_HANDLE     = LV\_MSGHND  
              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.  
  
  
        ENDIF.  
      ENDIF.  
    ENDIF.  
  
  
\*add1 to counter and pass to the variable with proper formatting  
    LV\_INDEX = LV\_INDEX + 1.  
    LV\_NUM = LV\_INDEX.  
  
\* if the length of cust\_field is 24, or the Subaccount in cust field is 0000000  
\* then replace the cost center with the standard cost center for that prnr  
    IF LWA\_2010-FLAG = 'X' OR LWA\_2010-SUBA IS INITIAL.  
      CLEAR: LV\_KOSTL.  
      SELECT SINGLE KOSTL FROM PA0001  
      INTO LV\_KOSTL WHERE PERNR = LWA\_2010-PERNR AND  
                          BEGDA <= LWA\_2010-DATE AND  
                          ENDDA > LWA\_2010-DATE .  
      IF LV\_KOSTL IS NOT INITIAL.  
        LWA\_2010-SUBA = LV\_KOSTL.  
        LWA\_2010-KOSTL = LWA\_2010-SUBA.  
      ENDIF.  
    ENDIF.  
  
    "JIRA 701 - set subaccount to cst center field  
\*  CONCATENATE lwa\_2010-suba(3) '-' lwa\_2010-suba+3(2) '-' lwa\_2010-suba+5(2) into lwa\_2010-suba.  
  
\*company code use from file, if does not xituse IT1  
    SELECT SINGLE BUTXT FROM T001  
      INTO LV\_BUTXT  
      WHERE BUKRS = LWA\_2010-BUKRS.  
    IF SY-SUBRC = 0.  
      LV\_BUKRS = LWA\_2010-BUKRS.  
    ELSE.  
  
\*select the company code  
      SELECT SINGLE BUKRS FROM PA0001  
      INTO LV\_BUKRS WHERE PERNR = LWA\_2010-PERNR AND  
      BEGDA <= LWA\_2010-DATE AND  
  ENDDA >= LWA\_2010-DATE.  
    ENDIF.  
  
  
\*depending on the first value of the company code pass the proper currency  
    IF LV\_BUKRS+0(1) = 'C'.  
  
  
\*select wage types from custom table that are allowances for canada as they are treated differently  
      IF LTA\_LGART IS INITIAL.  
        SELECT \* FROM ZTALLLOWANCES  
          INTO CORRESPONDING FIELDS OF TABLE LTA\_LGART  
          WHERE COUNTRY = 'CAN'.  
        IF SY-SUBRC = 0.  
  
          LOOP AT LTA\_LGART INTO LWA\_LGART.  
            WA\_RNG-SIGN = 'I'.  
            WA\_RNG-OPTION = 'EQ'.  
            WA\_RNG-LOW = LWA\_LGART-LGART.  
            APPEND WA\_RNG TO RNG\_TBL.  
          ENDLOOP .  
  
        ENDIF.  
      ENDIF.  
  
    ELSE.  
  
  
  
\*select wage types from custom table that are allowances for usa as they are treated differently  
      IF LTA\_LGART IS INITIAL.  
        SELECT \* FROM ZTALLLOWANCES  
          INTO CORRESPONDING FIELDS OF TABLE LTA\_LGART  
          WHERE COUNTRY = 'USA'.  
  
        IF SY-SUBRC = 0.  
  
          LOOP AT LTA\_LGART INTO LWA\_LGART.  
            WA\_RNG-SIGN = 'I'.  
            WA\_RNG-OPTION = 'EQ'.  
            WA\_RNG-LOW = LWA\_LGART-LGART.  
            APPEND WA\_RNG TO RNG\_TBL.  
          ENDLOOP .  
  
        ENDIF.  
      ENDIF.  
    ENDIF.  
  
\*select the currency  
    SELECT SINGLE WAERS FROM PA0008  
    INTO LV\_CURR WHERE PERNR = LWA\_2010-PERNR AND  
    BEGDA <= LWA\_2010-DATE AND  
ENDDA >= LWA\_2010-DATE.  
  
  
  
  
\*group name used for bdc creation  
    CONCATENATE 'WF' SY-DATUM+4(2) SY-DATUM+6(2)  
                SY-UZEIT INTO LV\_GRPNAME.  
  
\* format for selection  
    LV\_DATE = LWA\_2010-DATE.  
  
\*Format date for recording  
  
\*    IF sy-sysid = 'XOB' AND sy-mandt = '100'.  
\*   CONCATENATE  LWA\_2010-DATE+0(4) '.'  LWA\_2010-DATE+4(2)  '.' LWA\_2010-DATE+6(2) INTO  LWA\_2010-DATE .  
\*    ELSE.  
  
  
  
    CONCATENATE LWA\_2010-DATE+0(4) '-' LWA\_2010-DATE+4(2) '-'  
   LWA\_2010-DATE+6(2) INTO LWA\_2010-DATE .  
\*    ENDIF.  
  
    "&-------------------------------------------------------------------------------------------------------------&" ++Kriba "JIRA 805  
    IF LWA\_2010-REVER = 'X'.  
  
\*      CONCATENATE '-' LWA\_2010-AMOUNT INTO LWA\_2010-AMOUNT.  
  
      CLEAR: LV\_LEN1.  
      LV\_LEN1 = STRLEN( LWA\_2010-AMOUNT ).  
      IF LWA\_2010-AMOUNT(1) = '-'.  
        LWA\_2010-AMOUNT = LWA\_2010-AMOUNT+1(LV\_LEN1).  
      ELSE.  
        CONCATENATE '-' LWA\_2010-AMOUNT INTO LWA\_2010-AMOUNT.  
      ENDIF.  
  
\*jira fb-1578  
\* always take negative value  
      IF LWA\_2010-AMOUNT GT '0'.  
        LWA\_2010-AMOUNT = LWA\_2010-AMOUNT \* -1.  
      ENDIF.  
  
  
    ENDIF.  
    "&-------------------------------------------------------------------------------------------------------------&"  
  
\*if reprocess mode is selected only process records that haven't been created all ready  
    IF REPROCES = 'X'.  
  
      IF LTA\_LGART\_C IS INITIAL.  
        SELECT \* FROM ZTALLLOWANCES  
          INTO CORRESPONDING FIELDS OF TABLE LTA\_LGART\_C  
          WHERE COUNTRY = 'CAN'.  
      ENDIF.  
  
      CLEAR: LTA2\_2010, LWA2\_2010.  
      REFRESH LTA2\_2010.  
  
      LV\_PERNR = LWA\_2010-PERNR.  
  
      CALL FUNCTION 'CONVERSION\_EXIT\_ALPHA\_INPUT'  
        EXPORTING  
          INPUT  = LV\_PERNR  
        IMPORTING  
          OUTPUT = LV\_PERNR.  
  
      "&-------------------------------------------------------------------------------------------------------------&" ++Kriba "JIRA 1511  
  
      CLEAR: LWA\_LGART\_C.  
      READ TABLE LTA\_LGART INTO LWA\_LGART\_C WITH KEY LGART = LWA\_2010-LGART.  
      IF SY-SUBRC = 0.  
  
        "&-------------------------------------------------------------------------------------------------------------&" ++Kriba "FBUS 334  
  
        CLEAR:  LV\_EXBEL,  
                LV\_STDAZ,  
                LV\_BWGRL,  
                LV\_PERNR1,  
                LV\_SUBTY ,  
                LV\_BEGDA .  
  
        LV\_EXBEL = LWA\_2010-EXTDOC.  
        LV\_STDAZ = LWA\_2010-UNITS.  
        LV\_BWGRL = LWA\_2010-RATE.  
        LV\_PERNR1 = LV\_PERNR.  
        LV\_SUBTY = LWA\_2010-LGART.  
        LV\_BEGDA = LV\_DATE.  
  
        "&-------------------------------------------------------------------------------------------------------------&" ++Kriba "FBUS 334  
  
        "this is a 1500 series wagetype - check the units, date and external doc number  
        SELECT SINGLE \* FROM PA2010 INTO  LWA2\_2010  
        WHERE PERNR = LV\_PERNR1 AND  
              SUBTY = LV\_SUBTY AND  
              BEGDA = LV\_BEGDA AND  
              EXBEL = LV\_EXBEL AND  
              STDAZ = LV\_STDAZ AND  
              BWGRL = LV\_BWGRL.  
  
      ELSE.  
  
        "this is a non 1500 series wagetype - check the amount field to the STDAZ field  
        SELECT \* FROM PA2010 INTO TABLE LTA2\_2010  
        WHERE PERNR = LV\_PERNR AND  
              SUBTY = LWA\_2010-LGART AND  
              BEGDA = LV\_DATE AND  
              EXBEL = LWA\_2010-EXTDOC AND  
              STDAZ = LWA\_2010-AMOUNT.  
  
      ENDIF.  
  
      "&-------------------------------------------------------------------------------------------------------------&"  
  
      IF SY-SUBRC = 0.  
\*select from 2010 table to check if a record already exists based on employee ID, wagetype  
\*amount, date, custifeld, docnumber  
  
        SORT LTA2\_2010 ASCENDING BY LGART EXBEL.  
        READ TABLE  LTA2\_2010 INTO LWA2\_2010 WITH KEY LGART = LWA\_2010-LGART  
        EXBEL = LWA\_2010-EXTDOC .  
\*If record exsits skip to next record  
        IF SY-SUBRC = 0.  
          CLEAR LTA2\_2010.  
  
          WA\_SLG-MSGTY = 'E'.  
          WA\_SLG-MSGID = 'ZSLG'.  
          WA\_SLG-MSGNO = '005'.  
          CONCATENATE ' ' LWA\_2010-EXTDOC INTO WA\_SLG-MSGV1.  
  
          CALL FUNCTION 'BAL\_LOG\_MSG\_ADD'  
            EXPORTING  
              I\_LOG\_HANDLE     = LV\_MSGHND  
              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.  
  
          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.  
  
  
          CONTINUE.  
        ENDIF.  
  
      ENDIF.  
  
    ENDIF.  
  
  
\*wagetype logic  
    IF LWA\_2010-LGART IN RNG\_TBL.  
      LWA\_2010-AMOUNT =  LWA\_2010-UNITS.  
  
      "&-------------------------------------------------------------------------------------------------------------&" ++Kriba "JIRA 805  
      IF LWA\_2010-REVER = 'X'.  
  
\*      CONCATENATE '-' LWA\_2010-AMOUNT INTO LWA\_2010-AMOUNT.  
  
        CLEAR: LV\_LEN1.  
        LV\_LEN1 = STRLEN( LWA\_2010-AMOUNT ).  
        IF LWA\_2010-AMOUNT(1) = '-'.  
          LWA\_2010-AMOUNT = LWA\_2010-AMOUNT+1(LV\_LEN1).  
        ELSE.  
          CONCATENATE '-' LWA\_2010-AMOUNT INTO LWA\_2010-AMOUNT.  
        ENDIF.  
  
      ENDIF.  
      "&-------------------------------------------------------------------------------------------------------------&"  
    ENDIF.  
    .  
  
  
  
    "&-------------------------------------------------------------------------------------------------------------&" ++Kriba  "JIRA 701  
\* Flag will be set to 'X' when length < 24 and if the subaccount is 0000000  
    IF LWA\_2010-FLAG IS INITIAL.  
  
      LV\_KOSTL = LWA\_2010-CUSTFIELD+21(7).  
  
      CONCATENATE LV\_KOSTL(3) '-' LV\_KOSTL+3(2) '-' LV\_KOSTL+5(2) INTO LV\_KOSTL.  
  
\*    CALL FUNCTION 'CONVERSION\_EXIT\_ALPHA\_INPUT'  
\*      EXPORTING  
\*        INPUT  = LV\_KOSTL  
\*      IMPORTING  
\*        OUTPUT = LV\_KOSTL.  
  
      LWA\_2010-KOSTL = LV\_KOSTL.  
  
    ENDIF.  
    "&-------------------------------------------------------------------------------------------------------------&"  
  
    "&-------------------------------------------------------------------------------------------------------------&" ++Kriba  "JIRA 490  
\*    M, D, E, or N.  
\*    If we receive an M, E or N  
\*    and the employee is in PA0001-BTRTL = 6451, 6452, or 6453 create on the PA2010-PRAKN field 04.  
\*select the Personnel Subarea  
\*    CLEAR: lv\_btrtl.  
\*    SELECT SINGLE btrtl FROM pa0001  
\*    INTO lv\_btrtl  WHERE pernr = lwa\_2010-pernr AND  
\*      begda <= lwa\_2010-date AND  
\*  endda >= lwa\_2010-date.  
\*    IF lv\_btrtl IS NOT INITIAL.  
\*      IF ( lwa\_2010-prakn = 'M' OR lwa\_2010-prakn = 'E' OR lwa\_2010-prakn = 'N' ) AND  
\*         ( lv\_btrtl = '6451' OR lv\_btrtl = '6452' OR lv\_btrtl = '6453' ) .  
\*        lwa\_2010-prakn = '04'.  
\*      ENDIF.  
  
    IF  LWA\_2010-PRAKN = 'M' OR LWA\_2010-PRAKN = 'E' OR LWA\_2010-PRAKN = 'N'  OR LWA\_2010-PRAKN = 'D'.  
      CLEAR: LV\_VAR.  
      CONCATENATE LWA\_2010-PRAKN LWA\_2010-PRAKN INTO LWA\_2010-PRAKN .  
    ENDIF.  
  
\*      SELECT SINGLE prakz FROM t510p  
\*        INTO lwa\_2010-prakz  
\*        WHERE prakn = lwa\_2010-prakn  AND modif = '70'    .  
  
\*    ENDIF.  
    "&-------------------------------------------------------------------------------------------------------------&"  
  
  
  
    IF LWA\_2010-WBS+3 EQ '000000000000000000000'.  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
\*New call with cost center and exception  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
      CALL FUNCTION 'ZHR\_2010UPLOADUSA'  
        EXPORTING  
          CTU          = 'X'  
          MODE         = 'N'   "A  
          UPDATE       = 'L'  
          GROUP        = LV\_GRPNAME  
          USER         = SY-UNAME  
          KEEP         = 'X'  
\*         HOLDDATE     =  
          NODATA       = '/'  
          PERNR\_001    = LWA\_2010-PERNR  
          TIMR6\_002    = 'X'  
          BEGDA\_003    = LWA\_2010-DATE  
          ENDDA\_004    = LWA\_2010-DATE  
          CHOIC\_005    = '2010'  
          BEGDA\_006    = LWA\_2010-DATE  
          LGART\_007    = LWA\_2010-LGART  
          STDAZ\_008    = LWA\_2010-AMOUNT  
          WAERS\_009    = LV\_CURR  
          VERSL\_010    = LWA\_2010-MAJOR  
          PRAKN\_011    = LWA\_2010-PRAKN  
          EXBEL\_012    = LWA\_2010-EXTDOC  
          CUSTFLD\_013  = LWA\_2010-CUSTFIELD  
          KOSTL\_014    = LWA\_2010-KOSTL  
          PS\_POSID\_015 = LWA\_2010-WBS  
          BUKRS\_016    = LWA\_2010-BUKRS  
          BEGDA\_017    = LWA\_2010-DATE  
          LGART\_018    = LWA\_2010-LGART  
          STDAZ\_019    = LWA\_2010-AMOUNT  
          WAERS\_020    = LV\_CURR  
          VERSL\_021    = LWA\_2010-MAJOR  
          PRAKN\_022    = LWA\_2010-PRAKN  
          EXBEL\_023    = LWA\_2010-EXTDOC  
          CUSTFLD\_024  = LWA\_2010-CUSTFIELD  
          BWGRL\_25     = LWA\_2010-RATE  
          TRFGR\_26     = LWA\_2010-TRFGR  
          TRFST\_27     = LWA\_2010-TRFST  
\*         WTART\_28     = LWA\_2010-TAXOVER \*jira 1816  
        IMPORTING  
          SUBRC        = LV\_SUBRC  
        TABLES  
          MESSTAB      = T\_MESSTAB.  
  
    ELSE.  
  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
\*New call with cost center and exception  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
      CALL FUNCTION 'ZHR\_2010UPLOADUSA'  
        EXPORTING  
          CTU          = 'X'  
          MODE         = 'N'   "A  
          UPDATE       = 'L'  
          GROUP        = LV\_GRPNAME  
          USER         = SY-UNAME  
          KEEP         = 'X'  
\*         HOLDDATE     =  
          NODATA       = '/'  
          PERNR\_001    = LWA\_2010-PERNR  
          TIMR6\_002    = 'X'  
          BEGDA\_003    = LWA\_2010-DATE  
          ENDDA\_004    = LWA\_2010-DATE  
          CHOIC\_005    = '2010'  
          BEGDA\_006    = LWA\_2010-DATE  
          LGART\_007    = LWA\_2010-LGART  
          STDAZ\_008    = LWA\_2010-AMOUNT  
          WAERS\_009    = LV\_CURR  
          VERSL\_010    = LWA\_2010-MAJOR  
          PRAKN\_011    = LWA\_2010-PRAKN  
          EXBEL\_012    = LWA\_2010-EXTDOC  
          CUSTFLD\_013  = LWA\_2010-CUSTFIELD  
          KOSTL\_014    = LWA\_2010-KOSTL  
          PS\_POSID\_015 = LWA\_2010-WBS  
          BUKRS\_016    = LWA\_2010-BUKRS  
          BEGDA\_017    = LWA\_2010-DATE  
          LGART\_018    = LWA\_2010-LGART  
          STDAZ\_019    = LWA\_2010-AMOUNT  
          WAERS\_020    = LV\_CURR  
          VERSL\_021    = LWA\_2010-MAJOR  
          PRAKN\_022    = LWA\_2010-PRAKN  
          EXBEL\_023    = LWA\_2010-EXTDOC  
          CUSTFLD\_024  = LWA\_2010-CUSTFIELD  
          BWGRL\_25     = LWA\_2010-RATE  
          TRFGR\_26     = LWA\_2010-TRFGR  
          TRFST\_27     = LWA\_2010-TRFST  
\*          WTART\_28     = LWA\_2010-TAXOVER \*jira 1816  
        IMPORTING  
          SUBRC        = LV\_SUBRC  
        TABLES  
          MESSTAB      = T\_MESSTAB.  
    ENDIF.  
  
\*Loop at the return message tab  
    LOOP AT T\_MESSTAB INTO WA\_MESSTAB.  
  
      WA\_SLG-MSGTY = WA\_MESSTAB-MSGTYP.  
      WA\_SLG-MSGID = WA\_MESSTAB-MSGID.  
      WA\_SLG-MSGNO = WA\_MESSTAB-MSGNR.  
      WA\_SLG-MSGV1 = WA\_MESSTAB-MSGV1.  
      WA\_SLG-MSGV2 = WA\_MESSTAB-MSGV2.  
  
  
\*add the values ot the slg log  
      CALL FUNCTION 'BAL\_LOG\_MSG\_ADD'  
        EXPORTING  
          I\_LOG\_HANDLE     = LV\_MSGHND  
          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.  
  
  
  
  
      CALL FUNCTION 'BAL\_DB\_SAVE'  
        EXPORTING  
          I\_CLIENT         = SY-MANDT  
\*         I\_IN\_UPDATE\_TASK = ' '  
          I\_SAVE\_ALL       = 'X'  
\*         I\_T\_LOG\_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.  
  
  
  
    ENDLOOP.  
  
    CLEAR T\_MESSTAB.  
    REFRESH T\_MESSTAB.  
  
    CLEAR LWA\_2010.  
\*    ENDIF.  
  ENDLOOP.  
  
  CALL FUNCTION 'BAL\_DB\_SAVE'.  
  IF SY-SUBRC <> 0.  
\* Implement suitable error handling here  
  ENDIF.