\*&----------------------------------------------------------------------------\*  
\*& Report  ZHR\_CREATE\_WBS  
\*&----------------------------------------------------------------------------\*  
\*& Modification Log  
\*& MOD#   Transport#   Date         Description  
\*& CH01   XO9K908572   11/01/2021   FBUS-142 - WBS Element - Company Code  
\*&                                  errors  
\*&----------------------------------------------------------------------------\*  
REPORT ZHR\_CREATE\_WBS.  
  
DATA: IT\_WBS\_STATUS TYPE TABLE OF BAPI\_WBS\_MNT\_SYSTEM\_STATUS,  
      WA\_WBS\_STATUS TYPE BAPI\_WBS\_MNT\_SYSTEM\_STATUS.  
  
TYPES: BEGIN OF TY\_PRPS,  
         PSPNR TYPE PS\_POSNR,  
         POSID TYPE PS\_POSID,  
         OBJNR TYPE J\_OBJNR,  
         PSPHI TYPE PS\_PSPHI,  
       END OF TY\_PRPS.  
  
TYPES: BEGIN OF TY\_PROJ,  
         PSPNR TYPE PS\_POSNR,  
         PSPID TYPE PS\_PSPID,  
         OBJNR TYPE J\_OBJNR,  
       END OF TY\_PROJ.  
  
TYPES: BEGIN OF TY\_PROJ\_ASSOC,  
         POSID TYPE PS\_POSID,  
         PSPHI TYPE PS\_PSPHI,  
         PSPID TYPE PS\_PSPID,  
       END OF TY\_PROJ\_ASSOC.  
  
  
DATA: IT\_PRPS       TYPE TABLE OF TY\_PRPS,  
      IT\_PROJ       TYPE TABLE OF TY\_PROJ,  
      IT\_JEST       TYPE TABLE OF JEST,  
      IT\_PROJ\_ASSOC TYPE TABLE OF TY\_PROJ\_ASSOC,  
      WA\_PROJ\_ASSOC TYPE TY\_PROJ\_ASSOC,  
      WA\_JEST       TYPE JEST,  
      WA\_PRPS       TYPE  TY\_PRPS,  
      WA\_PROJ       TYPE  TY\_PROJ,  
      LV\_POSID\_OLD  TYPE PS\_POSID,  
      LV\_POSID\_NEW  TYPE PS\_POSID,  
      LV\_PSPID\_OLD  TYPE PS\_PSPID,  
      LV\_PSPID\_NEW  TYPE PS\_PSPID,  
      LV\_LONG\_PROJ  TYPE PS\_PSPID.  
  
DATA: WA\_PROJDEF TYPE BAPI\_BUS2001\_NEW,  
      IT\_RETURN  TYPE TABLE OF BAPIRET2,  
\*      WA\_RETURN  TYPE BAPIRET2,  
      IT\_EXTIN   TYPE TABLE OF BAPIPAREX,  
      IT\_EXTOUT  TYPE TABLE OF BAPIPAREX,  
      IT\_RETURN1 TYPE TABLE OF BAPIRET2,  
      IT\_RETURN3 TYPE BAPIRETURN1,  
\*      wa\_RETURN3 TYPE BAPIRETURN1,  
      IT\_RESULT3 TYPE TABLE OF BAPI\_STATUS\_RESULT,  
      WA\_RESULT3 TYPE BAPI\_STATUS\_RESULT,  
      WA\_RETURN2 TYPE BAPIRET2,  
      WA\_RETURN3 TYPE BAPIRET2,  
      WA\_RETURN1 TYPE BAPIRET2.  
  
DATA: LV\_GO       TYPE C LENGTH 1.  
\*  
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\_WBS,  
         PSPID TYPE PS\_PSPID,  
         POSID TYPE PS\_POSID,  
         BUKRS TYPE BUKRS,  
         FLAG  TYPE CHAR1,  
       END      OF T\_WBS.  
  
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,  
      LTA\_LGART  TYPE TABLE OF TY\_LGART,  
      LV\_BTRTL   TYPE PA0001-BTRTL.  
  
DATA: LV\_VAR  TYPE BDC\_FVAL,  
      LV\_CHAR TYPE C LENGTH 2.  
DATA: LOGNUMBER TYPE BAL\_T\_LGNM.  
DATA: LT\_LOGH TYPE BAL\_T\_LOGH.  
  
DATA: IT\_WBS TYPE TABLE OF BAPI\_BUS2054\_NEW,  
      WA\_WBS TYPE BAPI\_BUS2054\_NEW.  
DATA: LV\_NUM\_I TYPE I.  
  
DATA:  
  LD\_E\_S\_MSG\_HANDLE       TYPE BALMSGHNDL,  
  LD\_E\_MSG\_WAS\_LOGGED     TYPE BOOLEAN,  
  LD\_E\_MSG\_WAS\_DISPLAYED TYPE BOOLEAN.  
  
  
\*      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\_WBS     TYPE T\_WBS,  
      LWA\_WBS\_DUP TYPE T\_WBS,  
      LTA\_WBS     TYPE TABLE OF T\_WBS,  
      LV\_STRING1  TYPE STRING,  
      LV\_STRING2  TYPE STRING,  
      LV\_STRING3  TYPE STRING,  
      LT\_HANDLE   TYPE BALLOGHNDL,  
      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,  
      LV\_LEN      TYPE I,  
      LV\_FLAG     TYPE C LENGTH 1,  
      LV\_GL       TYPE STRING.  
\*      LV\_KOSTL TYPE KOSTL.  
  
DATA: LV\_SEQNO(6) TYPE N VALUE '000001',  
      LC\_SEQNO(6) TYPE N VALUE '000000'.  
  
DATA : GS\_PROJECT\_DEFINITION     LIKE BAPI\_PROJECT\_DEFINITION,  
       GS\_PROJECT\_DEFINITION\_UPD LIKE BAPI\_PROJECT\_DEFINITION\_UP.  
  
DATA : GT\_METHOD\_PROJECT           LIKE BAPI\_METHOD\_PROJECT OCCURS 0 WITH HEADER LINE,  
       GT\_WBS\_ELEMENT\_TABLE\_UPDATE LIKE BAPI\_WBS\_ELEMENT\_UPDATE OCCURS 0 WITH HEADER LINE,  
       GT\_WBS\_ELEMENT\_TABLE        LIKE BAPI\_WBS\_ELEMENT OCCURS 0 WITH HEADER LINE,  
       GT\_WBS\_HIERARCHIE\_TABLE     LIKE BAPI\_WBS\_HIERARCHIE OCCURS 0 WITH HEADER LINE,  
       GT\_RETURN                   LIKE BAPIRETURN1 OCCURS 0 WITH HEADER LINE,  
       GT\_RETURN1                  LIKE BAPIRET2 OCCURS 0 WITH HEADER LINE,  
       GT\_MESSAGE\_TABLE            LIKE BAPI\_METH\_MESSAGE OCCURS 0 WITH HEADER LINE.  
  
  
\*range ffor wageyptes  
TYPES:RNG\_LGART TYPE RANGE OF LGART.  
DATA: RNG\_TBL TYPE RNG\_LGART,  
      WA\_RNG  TYPE LINE OF RNG\_LGART.  
  
DATA: IT\_DEF\_PROJ\_MAP TYPE TABLE OF ZHR\_DEF\_PROJ\_MAP,  
      WA\_DEF\_PROJ\_MAP TYPE ZHR\_DEF\_PROJ\_MAP.  
  
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.  
  
  SELECT \* FROM ZHR\_DEF\_PROJ\_MAP INTO TABLE IT\_DEF\_PROJ\_MAP.  
  
  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+23(11)  = 'PROJECT\_NUM'.  
          SPLIT WA\_XML\_TAB AT '>' INTO LV\_STRING1 LV\_STRING2.  
          SPLIT LV\_STRING2 AT '<' INTO LV\_STRING2 LV\_STRING3.  
          LWA\_WBS-PSPID = LV\_STRING2.  
          CLEAR: LV\_STRING1, LV\_STRING2, LV\_STRING3.  
  
\*\*\* CH01 - FBUS-142 - WBS Element - Company Code \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
        ELSEIF WA\_XML\_TAB+23(7)  = 'COMPANY'.  
          SPLIT WA\_XML\_TAB AT '>' INTO LV\_STRING1 LV\_STRING2.  
          SPLIT LV\_STRING2 AT '<' INTO LV\_STRING2 LV\_STRING3.  
          LWA\_WBS-BUKRS = LV\_STRING2.  
          CLEAR: LV\_STRING1, LV\_STRING2, LV\_STRING3.  
\*\*\* CH01 - FBUS-142 - WBS Element - Company Code \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
  
  
\*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 ).  
  
          CLEAR: LV\_GL.  
          LV\_GL = LV\_STRING2+3(6).  
  
  
\*Logic to pull company code from first 3 char of gl string  
          LWA\_WBS-BUKRS = LV\_STRING2+0(3).  
          IF LWA\_WBS-BUKRS+0(1) = '5'.  
            CONCATENATE 'U' LWA\_WBS-BUKRS INTO LWA\_WBS-BUKRS.  
          ELSE.  
            CONCATENATE 'C' LWA\_WBS-BUKRS INTO LWA\_WBS-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\_WBS-FLAG = 'X'.  
            LWA\_WBS-POSID = LV\_STRING2.  
          ELSE.  
            IF LV\_STRING2  NE '0000000000000000000000000000000'.  
              CONCATENATE LV\_STRING2+0(21) LV\_STRING2+28(3) INTO LWA\_WBS-POSID.  
            ENDIF.  
          ENDIF.  
  
        ENDIF.  
  
\*\*\* FlynnCAN: WBS create Program is not creating WBS elements \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 23/10 \*\*\*  
  
\*        IF LWA\_WBS-PSPID IS NOT INITIAL AND  
\*           LWA\_WBS-POSID IS NOT INITIAL AND  
\*           ( LV\_GL IS NOT INITIAL AND  
\*             LV\_GL NE '000000' ).  
        IF LWA\_WBS-POSID IS NOT INITIAL.  
          IF LWA\_WBS-PSPID IS INITIAL.  
            CLEAR: WA\_DEF\_PROJ\_MAP.  
            READ TABLE IT\_DEF\_PROJ\_MAP INTO WA\_DEF\_PROJ\_MAP  
            WITH KEY BUKRS = LWA\_WBS-BUKRS+1(3).  
            IF SY-SUBRC = 0.  
              LWA\_WBS-PSPID = WA\_DEF\_PROJ\_MAP-PSPID.  
            ENDIF.  
          ELSE.  
\*\*\* CH01 - FBUS-142 - WBS Element - Company Code \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 11/01 \*\*\*  
            CONCATENATE LWA\_WBS-BUKRS+1(3) LWA\_WBS-PSPID INTO LWA\_WBS-PSPID.  
\*\*\* CH01 - FBUS-142 - WBS Element - Company Code \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 11/01 \*\*\*  
          ENDIF.  
  
          APPEND LWA\_WBS TO LTA\_WBS.  
          CLEAR: LWA\_WBS.  
        ENDIF.  
\*        ENDIF .  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
      ELSE.  
        EXIT.  
      ENDIF.  
  
    ENDDO.  
  ENDIF.  
  
\*CLOSE THE FILE AFTER READING THE DATA  
  CLOSE DATASET GV\_FILENAME.  
  
  
  IF LTA\_WBS IS NOT INITIAL.  
    SORT LTA\_WBS BY PSPID  
                    POSID ASCENDING.  
    DELETE ADJACENT DUPLICATES FROM LTA\_WBS COMPARING PSPID  
                                                      POSID.  
  
    "Get projects  
    REFRESH: IT\_PROJ.  
    SELECT PSPNR  
           PSPID  
           OBJNR  
      FROM PROJ  
      INTO TABLE IT\_PROJ  
      FOR ALL ENTRIES IN LTA\_WBS  
      WHERE PSPID = LTA\_WBS-PSPID.  
  
    "Get WBS elements  
    REFRESH: IT\_PRPS.  
    SELECT PSPNR  
           POSID  
           OBJNR  
           PSPHI  
      FROM PRPS  
      INTO TABLE IT\_PRPS  
      FOR ALL ENTRIES IN LTA\_WBS  
      WHERE POSID = LTA\_WBS-POSID.  
  
    CLEAR: WA\_PRPS .  
    LOOP AT IT\_PRPS INTO WA\_PRPS.  
  
      CLEAR: WA\_PROJ\_ASSOC.  
      CALL FUNCTION 'CONVERSION\_EXIT\_ABPRJ\_OUTPUT'  
        EXPORTING  
          INPUT  = WA\_PRPS-PSPHI  
        IMPORTING  
          OUTPUT = WA\_PROJ\_ASSOC-PSPID.             "LONG PROJ  
  
      WA\_PROJ\_ASSOC-PSPHI = WA\_PRPS-PSPHI.          "SHORT PROJ  
      WA\_PROJ\_ASSOC-POSID = WA\_PRPS-POSID.          "WBS  
  
      "check if this project does not already exist in the internal table  
      READ TABLE IT\_PROJ WITH KEY PSPID = WA\_PROJ\_ASSOC-PSPID TRANSPORTING NO FIELDS.  
      IF SY-SUBRC <> 0.  
        APPEND WA\_PROJ\_ASSOC TO IT\_PROJ\_ASSOC.  
      ENDIF.  
  
    ENDLOOP.  
  
    "get the projects for the additional projects associated with WBS elements (special case)  
    IF IT\_PROJ\_ASSOC IS NOT INITIAL.  
      SORT IT\_PROJ\_ASSOC BY POSID  
                            PSPHI  
                            PSPID.  
      DELETE ADJACENT DUPLICATES FROM IT\_PROJ\_ASSOC COMPARING POSID  
                                                              PSPHI  
                                                              PSPID.  
  
      SELECT PSPNR  
             PSPID  
             OBJNR  
        FROM PROJ  
        APPENDING TABLE IT\_PROJ  
        FOR ALL ENTRIES IN IT\_PROJ\_ASSOC  
        WHERE PSPID = IT\_PROJ\_ASSOC-PSPID.  
    ENDIF.  
  
    "Get the status for all Projects and WBS elements  
    IF IT\_PROJ IS NOT INITIAL.  
      REFRESH: IT\_JEST.  
      SELECT \* FROM JEST  
        APPENDING TABLE IT\_JEST  
        FOR ALL ENTRIES IN IT\_PROJ  
        WHERE OBJNR = IT\_PROJ-OBJNR.  
    ENDIF.  
  
    IF IT\_PROJ IS NOT INITIAL.  
      SELECT \* FROM JEST  
        APPENDING TABLE IT\_JEST  
        FOR ALL ENTRIES IN IT\_PRPS  
        WHERE OBJNR = IT\_PRPS-OBJNR.  
    ENDIF.  
  
  
  
\*\*Object values for the SLG log  
\*    LV\_LOG-OBJECT = 'ZHR\_WBS\_CR'.          "Object name  
\*    LV\_LOG-SUBOBJECT = 'ZHR\_WBS\_CR'.  
\*    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\_MSGHND  
\*      EXCEPTIONS  
\*        LOG\_HEADER\_INCONSISTENT = 1  
\*        OTHERS                  = 2.  
\*    IF SY-SUBRC = 0.  
\*    ENDIF.  
  
\*Object values for the SLG log  
    LV\_LOG-OBJECT = 'Z2010'.          "Object name  
    LV\_LOG-SUBOBJECT = 'Z2010\_WBS'.  
    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\_MSGHND  
      EXCEPTIONS  
        LOG\_HEADER\_INCONSISTENT = 1  
        OTHERS                  = 2.  
    IF SY-SUBRC = 0.  
      REFRESH: LT\_LOGH.  
      APPEND LV\_MSGHND TO LT\_LOGH.  
    ENDIF.  
  
  
  
    CLEAR: LWA\_WBS.  
    LOOP AT LTA\_WBS INTO LWA\_WBS.  
  
      CLEAR: WA\_SLG.  
      REFRESH: IT\_WBS.  
  
\*      CLEAR: LV\_POSID\_NEW,  
\*             LV\_GO.  
\*      LV\_PSPID\_NEW = LWA\_WBS-PSPID.  
\*  
\*      IF LV\_PSPID\_OLD IS INITIAL.  
\*        LV\_PSPID\_OLD = LWA\_WBS-PSPID.  
\*        LV\_PSPID\_NEW = LWA\_WBS-PSPID.  
\*        LV\_GO = 'X'.  
\*      ELSE.  
\*        IF LV\_PSPID\_OLD NE LV\_PSPID\_NEW.  
\*          LV\_GO = 'X'.  
\*          LV\_PSPID\_OLD = LV\_PSPID\_NEW.  
\*        ENDIF.  
\*      ENDIF.  
  
\*      IF LV\_GO = 'X'.  
      "Condition 1: Check if the project exists  
      CLEAR: WA\_PROJ.  
      READ TABLE IT\_PROJ INTO WA\_PROJ WITH KEY PSPID = LWA\_WBS-PSPID.  
      IF SY-SUBRC = 0.  
  
        "Condition 1.1: If project exists, check if the project is released  
        CLEAR: WA\_JEST.  
        READ TABLE IT\_JEST INTO WA\_JEST WITH KEY OBJNR = WA\_PROJ-OBJNR  
                                                 STAT = 'I0002'.  
        IF SY-SUBRC = 0.  
          "If the project is released, then create the WBS element below the project  
  
          "Condition 1.1.1: If project exists and is released, then check is WBS exists  
          CLEAR: WA\_JEST,  
                 WA\_PRPS.  
          READ TABLE IT\_PRPS INTO WA\_PRPS WITH KEY POSID = LWA\_WBS-POSID.  
          IF SY-SUBRC = 0.  
  
            "Condition 1.1.1.1: If project exists and is released, and WBS element exists, then check if its released  
            CLEAR: WA\_JEST.  
            READ TABLE IT\_JEST INTO WA\_JEST WITH KEY OBJNR = WA\_PRPS-OBJNR  
                                                     STAT = 'I0002'.  
            IF SY-SUBRC = 0.  
              "if project exists and is released, and WBS element exists and is released, then dont do anything  
  
              WA\_SLG-MSGTY = 'W'.  
              WA\_SLG-MSGID = 'ZHRWBS'.  
              WA\_SLG-MSGNO = '000'.  
              CONCATENATE  'Project ' LWA\_WBS-PSPID ' exists and released '  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.  
  
              CLEAR: WA\_SLG-MSGV1.  
              CONCATENATE  'WBS ' LWA\_WBS-POSID ' exists and released' 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.  
  
  
            ELSE.  
              "if project exists and is released, and WBS element exists and is not released, then release WBS element  
              PERFORM RELEASE\_WBS\_ELEMENT.  
  
            ENDIF.  
  
          ELSE.  
            "if project exists and is released, and WBS element does not exists  
            PERFORM CREATE\_WBS\_ELEMENT.  
  
          ENDIF.  
  
        ELSE.  
          "If the project is not released, release the project and then create the WBS element beneath it.  
\*          PERFORM RELEASE\_PROJECT.  
  
          PERFORM CREATE\_WBS\_ELEMENT.  
  
        ENDIF.  
  
      ELSE.  
        "Condition 2: If Project does not exist, then create the Project along with the WBS element.  
  
        PERFORM CREATE\_PROJECT.  
  
        PERFORM CREATE\_WBS\_ELEMENT.  
  
      ENDIF.  
\*      ENDIF.  
  
      IF WA\_SLG IS NOT INITIAL.  
  
        CALL FUNCTION 'BAL\_DB\_SAVE'  
          EXPORTING  
            I\_T\_LOG\_HANDLE   = LT\_LOGH  
          IMPORTING  
            E\_NEW\_LOGNUMBERS = LOGNUMBER  
          EXCEPTIONS  
            LOG\_NOT\_FOUND    = 1  
            SAVE\_NOT\_ALLOWED = 2  
            NUMBERING\_ERROR  = 3  
            OTHERS           = 4.  
  
      ENDIF.  
  
  
    ENDLOOP.  
  
  
    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.  
  
  
    IF NOT LV\_MSGHND IS INITIAL.  
      CALL FUNCTION 'BAL\_LOG\_DELETE'  
        EXPORTING  
          I\_LOG\_HANDLE  = LV\_MSGHND  
        EXCEPTIONS  
          LOG\_NOT\_FOUND = 0  
          OTHERS        = 0.  
      IF SY-SUBRC <> 0.  
      ENDIF.  
    ENDIF.  
  
  
  ENDIF.  
  
  
  
  
\*&---------------------------------------------------------------------\*  
\*&      Form  CREATE\_PROJECT  
\*&---------------------------------------------------------------------\*  
\*       text  
\*----------------------------------------------------------------------\*  
\*  -->  p1        text  
\*  <--  p2        text  
\*----------------------------------------------------------------------\*  
FORM CREATE\_PROJECT .  
  
  WA\_PROJDEF-PROJECT\_DEFINITION = LWA\_WBS-PSPID.  
\*  CLEAR: WA\_PROJDEF-PROJECT\_DEFINITION.  
\*  CONCATENATE LWA\_WBS-BUKRS LWA\_WBS-PSPID INTO WA\_PROJDEF-PROJECT\_DEFINITION.  
  WA\_PROJDEF-DESCRIPTION = LWA\_WBS-PSPID.  
  WA\_PROJDEF-PROJECT\_PROFILE = '0000003'.  
  
  WA\_PROJDEF-COMPANY\_CODE     = LWA\_WBS-BUKRS.  
  WA\_PROJDEF-BUSINESS\_AREA    = ''.  
  WA\_PROJDEF-CONTROLLING\_AREA = LWA\_WBS-BUKRS.  
  WA\_PROJDEF-PLANT            = LWA\_WBS-BUKRS.  
  WA\_PROJDEF-PROJECT\_CURRENCY = 'CA'.  
  
  CALL FUNCTION 'BAPI\_PS\_INITIALIZATION'.  
  
  CALL FUNCTION 'BAPI\_BUS2001\_CREATE'  
    EXPORTING  
      I\_PROJECT\_DEFINITION = WA\_PROJDEF  
    TABLES  
      ET\_RETURN            = IT\_RETURN  
      EXTENSIONIN          = IT\_EXTIN  
      EXTENSIONOUT         = IT\_EXTOUT.  
  
  CLEAR: WA\_RETURN2.  
  READ TABLE IT\_RETURN INTO WA\_RETURN2 WITH KEY TYPE = 'S'.  
  IF SY-SUBRC = 0.  
  
    CALL FUNCTION 'BAPI\_BUS2001\_SET\_STATUS'  
      EXPORTING  
        PROJECT\_DEFINITION = LWA\_WBS-PSPID  
\*       UNDO\_SYSTEM\_STATUS =  
\*       UNDO\_USER\_STATUS   =  
        SET\_SYSTEM\_STATUS  = 'REL'  
\*       SET\_USER\_STATUS    =  
      IMPORTING  
        RETURN             = IT\_RETURN3  
      TABLES  
        E\_RESULT           = IT\_RESULT3.  
  
    CALL FUNCTION 'BAPI\_PS\_PRECOMMIT'  
      TABLES  
        ET\_RETURN = IT\_RETURN1.  
    READ TABLE IT\_RETURN1 INTO WA\_RETURN1 WITH KEY TYPE = 'E'.  
    IF SY-SUBRC EQ 0.  
      CALL FUNCTION 'BAPI\_TRANSACTION\_ROLLBACK'.  
    ELSE.  
      CALL FUNCTION 'BAPI\_TRANSACTION\_COMMIT'  
        EXPORTING  
          WAIT = ABAP\_TRUE.  
    ENDIF.  
  
  ELSE.  
  
    CLEAR: WA\_RETURN2,  
           WA\_SLG.  
    LOOP AT IT\_RETURN INTO WA\_RETURN2.  
  
\*      WA\_SLG-MSGTY = WA\_RETURN-TYPE.  
\*      WA\_SLG-MSGID = 'ZHRWBS'.  
\*      WA\_SLG-MSGNO = '000'.  
\*      WA\_SLG-MSGV1 = WA\_RETURN-MESSAGE(50).  
\*      WA\_SLG-MSGV2 = WA\_RETURN-MESSAGE+50(50).  
  
      CALL FUNCTION 'BAL\_LOG\_MSG\_ADD\_FREE\_TEXT'  
        EXPORTING  
          I\_LOG\_HANDLE        = LV\_MSGHND  
          I\_MSGTY             = WA\_RETURN2-TYPE  
\*         I\_PROBCLASS         = '4'  
          I\_TEXT              = WA\_RETURN2-MESSAGE  
\*         I\_S\_CONTEXT         =  
\*         I\_S\_PARAMS          =  
\*         I\_DETLEVEL          = '1'  
        IMPORTING  
          E\_S\_MSG\_HANDLE      = LD\_E\_S\_MSG\_HANDLE  
          E\_MSG\_WAS\_LOGGED    = LD\_E\_MSG\_WAS\_LOGGED  
          E\_MSG\_WAS\_DISPLAYED = LD\_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.  
  
\*\*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.  
\*  
\*      ENDIF.  
    ENDLOOP.  
  
  
  ENDIF.  
  
ENDFORM.  
  
\*&---------------------------------------------------------------------\*  
\*&      Form  CREATE\_WBS\_ELEMENT  
\*&---------------------------------------------------------------------\*  
\*       text  
\*----------------------------------------------------------------------\*  
\*  -->  p1        text  
\*  <--  p2        text  
\*----------------------------------------------------------------------\*  
FORM CREATE\_WBS\_ELEMENT.  
  
  "At this point - the WBS element  
  "a) is not created  
  "b) is created under the same project  
  "c) is created under a different project  
  "- Check if that project is released - if not release it  
  
  CLEAR: WA\_JEST,  
         WA\_PRPS.  
  READ TABLE IT\_PRPS INTO WA\_PRPS WITH KEY POSID = LWA\_WBS-POSID.  
  IF SY-SUBRC = 0.  
    "WBS element exists  
  
\*\*\* Project related checks \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
    CLEAR: LV\_LONG\_PROJ .  
    CALL FUNCTION 'CONVERSION\_EXIT\_ABPRJ\_OUTPUT'  
      EXPORTING  
        INPUT  = WA\_PRPS-PSPHI  
      IMPORTING  
        OUTPUT = LV\_LONG\_PROJ.  
  
  
    "Condition 1: Check if the project associated with the WBS element exists  
    CLEAR: WA\_PROJ.  
    READ TABLE IT\_PROJ INTO WA\_PROJ WITH KEY PSPID = LV\_LONG\_PROJ.  
    IF SY-SUBRC = 0.  
  
\*      "Condition 1.1: If project exists, check if the project is released  
\*      CLEAR: WA\_JEST.  
\*      READ TABLE IT\_JEST INTO WA\_JEST WITH KEY OBJNR = WA\_PROJ-OBJNR  
\*                                               STAT = 'I0002'.  
\*      IF SY-SUBRC <> 0.  
  
\*        "if the project is not released  
\*        CALL FUNCTION 'BAPI\_PS\_INITIALIZATION'.  
\*  
\*        CLEAR: IT\_RETURN3.  
\*        REFRESH: IT\_RESULT3.  
\*        CALL FUNCTION 'BAPI\_BUS2001\_SET\_STATUS'  
\*          EXPORTING  
\*            PROJECT\_DEFINITION = LV\_LONG\_PROJ  
\*\*           UNDO\_SYSTEM\_STATUS =  
\*\*           UNDO\_USER\_STATUS   =  
\*            SET\_SYSTEM\_STATUS  = 'REL'  
\*\*           SET\_USER\_STATUS    =  
\*          IMPORTING  
\*            RETURN             = IT\_RETURN3  
\*          TABLES  
\*            E\_RESULT           = IT\_RESULT3.  
\*  
\*        CLEAR: WA\_RESULT3.  
\*        READ TABLE IT\_RESULT3 INTO WA\_RESULT3 WITH KEY STATUS\_ACTION = 'S'.  
\*        IF SY-SUBRC = 0.  
\*          CALL FUNCTION 'BAPI\_PS\_PRECOMMIT'  
\*            TABLES  
\*              ET\_RETURN = IT\_RETURN1.  
\*          READ TABLE IT\_RETURN1 INTO WA\_RETURN1 WITH KEY TYPE = 'E'.  
\*          IF SY-SUBRC EQ 0.  
\*            CALL FUNCTION 'BAPI\_TRANSACTION\_ROLLBACK'.  
\*          ELSE.  
\*            CALL FUNCTION 'BAPI\_TRANSACTION\_COMMIT'  
\*              EXPORTING  
\*                WAIT = ABAP\_TRUE.  
\*          ENDIF.  
\*  
\*        ELSE.  
\*  
\*          CLEAR: WA\_RESULT3,  
\*                 WA\_SLG,  
\*                 LD\_E\_S\_MSG\_HANDLE,  
\*                 LD\_E\_MSG\_WAS\_LOGGED,  
\*                 LD\_E\_MSG\_WAS\_DISPLAYED.  
\*          LOOP AT IT\_RESULT3 INTO WA\_RESULT3.  
\*\*      WA\_SLG-MSGTY = WA\_RETURN-TYPE.  
\*\*      WA\_SLG-MSGID = 'ZHRWBS'.  
\*\*      WA\_SLG-MSGNO = '000'.  
\*\*      WA\_SLG-MSGV1 = WA\_RESULT3-MESSAGE\_TEXT(50).  
\*\*      WA\_SLG-MSGV2 = WA\_RESULT3-MESSAGE\_TEXT+50(21).  
\*  
\*            CALL FUNCTION 'BAL\_LOG\_MSG\_ADD\_FREE\_TEXT'  
\*              EXPORTING  
\*                I\_LOG\_HANDLE        = LV\_MSGHND  
\*                I\_MSGTY             = WA\_RESULT3-MESSAGE\_TYPE  
\*\*               I\_PROBCLASS         = '4'  
\*                I\_TEXT              = WA\_RESULT3-MESSAGE\_TEXT  
\*\*               I\_S\_CONTEXT         =  
\*\*               I\_S\_PARAMS          =  
\*\*               I\_DETLEVEL          = '1'  
\*              IMPORTING  
\*                E\_S\_MSG\_HANDLE      = LD\_E\_S\_MSG\_HANDLE  
\*                E\_MSG\_WAS\_LOGGED    = LD\_E\_MSG\_WAS\_LOGGED  
\*                E\_MSG\_WAS\_DISPLAYED = LD\_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.  
\*  
\*\*\*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.  
\*\*  
\*\*      ENDIF.  
\*          ENDLOOP.  
\*  
\*        ENDIF.  
\*  
\*      ENDIF.  
\*  
    ELSE.  
  
      "If the project does not exist - create the project  
      WA\_PROJDEF-PROJECT\_DEFINITION = LV\_LONG\_PROJ.  
      WA\_PROJDEF-DESCRIPTION = LV\_LONG\_PROJ.  
      WA\_PROJDEF-PROJECT\_PROFILE = '0000003'.  
  
      WA\_PROJDEF-COMPANY\_CODE     = LWA\_WBS-BUKRS.  
      WA\_PROJDEF-BUSINESS\_AREA    = ''.  
      WA\_PROJDEF-CONTROLLING\_AREA = LWA\_WBS-BUKRS.  
      WA\_PROJDEF-PLANT            = LWA\_WBS-BUKRS.  
      WA\_PROJDEF-PROJECT\_CURRENCY = 'CA'.  
  
      CALL FUNCTION 'BAPI\_PS\_INITIALIZATION'.  
  
      CALL FUNCTION 'BAPI\_BUS2001\_CREATE'  
        EXPORTING  
          I\_PROJECT\_DEFINITION = WA\_PROJDEF  
        TABLES  
          ET\_RETURN            = IT\_RETURN  
          EXTENSIONIN          = IT\_EXTIN  
          EXTENSIONOUT         = IT\_EXTOUT.  
  
      CLEAR: WA\_RETURN2.  
      READ TABLE IT\_RETURN INTO WA\_RETURN2 WITH KEY TYPE = 'S'.  
      IF SY-SUBRC = 0.  
  
        CALL FUNCTION 'BAPI\_BUS2001\_SET\_STATUS'  
          EXPORTING  
            PROJECT\_DEFINITION = LWA\_WBS-PSPID  
\*           UNDO\_SYSTEM\_STATUS =  
\*           UNDO\_USER\_STATUS   =  
            SET\_SYSTEM\_STATUS  = 'REL'  
\*           SET\_USER\_STATUS    =  
          IMPORTING  
            RETURN             = IT\_RETURN3  
          TABLES  
            E\_RESULT           = IT\_RESULT3.  
  
        CALL FUNCTION 'BAPI\_PS\_PRECOMMIT'  
          TABLES  
            ET\_RETURN = IT\_RETURN1.  
        READ TABLE IT\_RETURN1 INTO WA\_RETURN1 WITH KEY TYPE = 'E'.  
        IF SY-SUBRC EQ 0.  
          CALL FUNCTION 'BAPI\_TRANSACTION\_ROLLBACK'.  
        ELSE.  
          CALL FUNCTION 'BAPI\_TRANSACTION\_COMMIT'  
            EXPORTING  
              WAIT = ABAP\_TRUE.  
        ENDIF.  
  
      ELSE.  
  
        CLEAR: WA\_RETURN2,  
                WA\_SLG.  
        LOOP AT IT\_RETURN INTO WA\_RETURN2.  
  
\*      WA\_SLG-MSGTY = WA\_RETURN-TYPE.  
\*      WA\_SLG-MSGID = 'ZHRWBS'.  
\*      WA\_SLG-MSGNO = '000'.  
\*      WA\_SLG-MSGV1 = WA\_RETURN-MESSAGE(50).  
\*      WA\_SLG-MSGV2 = WA\_RETURN-MESSAGE+50(50).  
  
          CALL FUNCTION 'BAL\_LOG\_MSG\_ADD\_FREE\_TEXT'  
            EXPORTING  
              I\_LOG\_HANDLE        = LV\_MSGHND  
              I\_MSGTY             = WA\_RETURN2-TYPE  
\*             I\_PROBCLASS         = '4'  
              I\_TEXT              = WA\_RETURN2-MESSAGE  
\*             I\_S\_CONTEXT         =  
\*             I\_S\_PARAMS          =  
\*             I\_DETLEVEL          = '1'  
            IMPORTING  
              E\_S\_MSG\_HANDLE      = LD\_E\_S\_MSG\_HANDLE  
              E\_MSG\_WAS\_LOGGED    = LD\_E\_MSG\_WAS\_LOGGED  
              E\_MSG\_WAS\_DISPLAYED = LD\_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.  
  
        ENDLOOP.  
  
      ENDIF.  
  
    ENDIF.  
\*\*\* Project related checks \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
  
\*\*\* WBS elemnt related checks \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
    "Check if WBS element has released status  
    CLEAR: WA\_JEST.  
    READ TABLE IT\_JEST INTO WA\_JEST WITH KEY OBJNR = WA\_PRPS-OBJNR  
                                             STAT = 'I0002'.  
    IF SY-SUBRC <> 0.  
  
      PERFORM RELEASE\_WBS\_ELEMENT.  
  
    ENDIF.  
\*\*\* WBS elemnt related checks \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
  
  ELSE.  
    "WBS element does not exist  
  
  
    CLEAR: LV\_LONG\_PROJ.  
    CALL FUNCTION 'CONVERSION\_EXIT\_ABPRJ\_OUTPUT'  
      EXPORTING  
        INPUT  = LWA\_WBS-PSPID  
      IMPORTING  
        OUTPUT = LV\_LONG\_PROJ.  
    IF LV\_LONG\_PROJ IS INITIAL.  
      LV\_LONG\_PROJ = LWA\_WBS-PSPID.  
    ENDIF.  
  
    "Condition 1: Check if the project associated with the WBS element exists  
    CLEAR: WA\_PROJ.  
    READ TABLE IT\_PROJ INTO WA\_PROJ WITH KEY PSPID = LV\_LONG\_PROJ.  
    IF SY-SUBRC = 0.  
  
    ENDIF.  
  
\*  CLEAR: LWA\_WBS\_DUP,  
\*          WA\_WBS.  
\*  REFRESH: IT\_WBS.  
\*  LOOP AT LTA\_WBS INTO LWA\_WBS\_DUP WHERE PSPID = LWA\_WBS-PSPID.  
    REFRESH: IT\_WBS.  
    CLEAR: WA\_WBS.  
    WA\_WBS-WBS\_ELEMENT = LWA\_WBS-POSID.  
    WA\_WBS-DESCRIPTION = LWA\_WBS-POSID.  
    WA\_WBS-COMPANY\_CODE =       LWA\_WBS-BUKRS.  
    WA\_WBS-BUSINESS\_AREA =      ''.  
    WA\_WBS-CONTROLLING\_AREA =   LWA\_WBS-BUKRS.  
    WA\_WBS-PLANT =              LWA\_WBS-BUKRS.  
    WA\_WBS-CURRENCY =           'CA'.  
    APPEND WA\_WBS TO IT\_WBS.  
  
\*  ENDLOOP.  
  
    CALL FUNCTION 'BAPI\_PS\_INITIALIZATION'.  
  
    CALL FUNCTION 'BAPI\_BUS2054\_CREATE\_MULTI'  
      EXPORTING  
        I\_PROJECT\_DEFINITION = LWA\_WBS-PSPID  
      TABLES  
        IT\_WBS\_ELEMENT       = IT\_WBS  
        ET\_RETURN            = IT\_RETURN  
        EXTENSIONIN          = IT\_EXTIN  
        EXTENSIONOUT         = IT\_EXTOUT.  
  
    CLEAR: WA\_RETURN.  
    READ TABLE IT\_RETURN INTO WA\_RETURN WITH KEY TYPE = 'S'.  
    IF SY-SUBRC = 0.  
  
      CLEAR: WA\_WBS\_STATUS.  
      REFRESH: IT\_WBS\_STATUS.  
      LOOP AT IT\_WBS INTO WA\_WBS.  
        WA\_WBS\_STATUS-WBS\_ELEMENT = WA\_WBS-WBS\_ELEMENT.  
        WA\_WBS\_STATUS-SET\_SYSTEM\_STATUS = 'REL'.  
        APPEND WA\_WBS\_STATUS TO IT\_WBS\_STATUS.  
      ENDLOOP.  
  
      CLEAR: IT\_RETURN3.  
      REFRESH: IT\_RESULT3.  
      CALL FUNCTION 'BAPI\_BUS2054\_SET\_STATUS'  
        IMPORTING  
          RETURN              = IT\_RETURN3  
        TABLES  
          I\_WBS\_SYSTEM\_STATUS = IT\_WBS\_STATUS  
\*         I\_WBS\_USER\_STATUS   =  
          E\_RESULT            = IT\_RESULT3.  
  
      IF IT\_RETURN3-TYPE = 'S' OR  
         IT\_RETURN3-TYPE IS INITIAL.  
        CALL FUNCTION 'BAPI\_PS\_PRECOMMIT'  
          TABLES  
            ET\_RETURN = IT\_RETURN1.  
        READ TABLE IT\_RETURN1 INTO WA\_RETURN1 WITH KEY TYPE = 'E'.  
        IF SY-SUBRC EQ 0.  
          CALL FUNCTION 'BAPI\_TRANSACTION\_ROLLBACK'.  
        ELSE.  
          CALL FUNCTION 'BAPI\_TRANSACTION\_COMMIT'  
            EXPORTING  
              WAIT = ABAP\_TRUE.  
        ENDIF.  
  
      ENDIF.  
  
    ELSE.  
  
      CLEAR: WA\_RETURN,  
             WA\_SLG,  
             LD\_E\_S\_MSG\_HANDLE,  
             LD\_E\_MSG\_WAS\_LOGGED,  
             LD\_E\_MSG\_WAS\_DISPLAYED.  
      LOOP AT IT\_RETURN INTO WA\_RETURN.  
\*      WA\_SLG-MSGTY = WA\_RETURN-TYPE.  
\*      WA\_SLG-MSGID = 'ZHRWBS'.  
\*      WA\_SLG-MSGNO = '000'.  
\*      WA\_SLG-MSGV1 = WA\_RETURN-MESSAGE(50).  
\*      WA\_SLG-MSGV2 = WA\_RETURN-MESSAGE+50(50).  
  
        CALL FUNCTION 'BAL\_LOG\_MSG\_ADD\_FREE\_TEXT'  
          EXPORTING  
            I\_LOG\_HANDLE        = LV\_MSGHND  
            I\_MSGTY             = WA\_RETURN-TYPE  
\*           I\_PROBCLASS         = '4'  
            I\_TEXT              = WA\_RETURN-MESSAGE  
\*           I\_S\_CONTEXT         =  
\*           I\_S\_PARAMS          =  
\*           I\_DETLEVEL          = '1'  
          IMPORTING  
            E\_S\_MSG\_HANDLE      = LD\_E\_S\_MSG\_HANDLE  
            E\_MSG\_WAS\_LOGGED    = LD\_E\_MSG\_WAS\_LOGGED  
            E\_MSG\_WAS\_DISPLAYED = LD\_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.  
  
      ENDLOOP.  
  
      CONCATENATE 'Project : '  LV\_LONG\_PROJ INTO WA\_RETURN-MESSAGE SEPARATED BY SPACE.  
  
      CALL FUNCTION 'BAL\_LOG\_MSG\_ADD\_FREE\_TEXT'  
        EXPORTING  
          I\_LOG\_HANDLE        = LV\_MSGHND  
          I\_MSGTY             = 'E'  
\*         I\_PROBCLASS         = '4'  
          I\_TEXT              = WA\_RETURN-MESSAGE  
\*         I\_S\_CONTEXT         =  
\*         I\_S\_PARAMS          =  
\*         I\_DETLEVEL          = '1'  
        IMPORTING  
          E\_S\_MSG\_HANDLE      = LD\_E\_S\_MSG\_HANDLE  
          E\_MSG\_WAS\_LOGGED    = LD\_E\_MSG\_WAS\_LOGGED  
          E\_MSG\_WAS\_DISPLAYED = LD\_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.  
  
    ENDIF.  
  
  ENDIF.  
  
\*  ENDIF.  
  
  
ENDFORM.  
  
\*&---------------------------------------------------------------------\*  
\*&      Form  RELEASE\_PROJECT  
\*&---------------------------------------------------------------------\*  
\*       text  
\*----------------------------------------------------------------------\*  
\*  -->  p1        text  
\*  <--  p2        text  
\*----------------------------------------------------------------------\*  
FORM RELEASE\_PROJECT .  
  
  CALL FUNCTION 'BAPI\_PS\_INITIALIZATION'.  
  
  CLEAR: IT\_RETURN3.  
  REFRESH: IT\_RESULT3.  
  CALL FUNCTION 'BAPI\_BUS2001\_SET\_STATUS'  
    EXPORTING  
      PROJECT\_DEFINITION = LWA\_WBS-PSPID  
\*     UNDO\_SYSTEM\_STATUS =  
\*     UNDO\_USER\_STATUS   =  
      SET\_SYSTEM\_STATUS  = 'REL'  
\*     SET\_USER\_STATUS    =  
    IMPORTING  
      RETURN             = IT\_RETURN3  
    TABLES  
      E\_RESULT           = IT\_RESULT3.  
  
  CLEAR: WA\_RESULT3.  
  READ TABLE IT\_RESULT3 INTO WA\_RESULT3 WITH KEY STATUS\_ACTION = 'S'.  
  IF SY-SUBRC = 0.  
    CALL FUNCTION 'BAPI\_PS\_PRECOMMIT'  
      TABLES  
        ET\_RETURN = IT\_RETURN1.  
    READ TABLE IT\_RETURN1 INTO WA\_RETURN1 WITH KEY TYPE = 'E'.  
    IF SY-SUBRC EQ 0.  
      CALL FUNCTION 'BAPI\_TRANSACTION\_ROLLBACK'.  
    ELSE.  
      CALL FUNCTION 'BAPI\_TRANSACTION\_COMMIT'  
        EXPORTING  
          WAIT = ABAP\_TRUE.  
    ENDIF.  
  
  ELSE.  
  
    CLEAR: WA\_RESULT3,  
           WA\_SLG,  
           LD\_E\_S\_MSG\_HANDLE,  
           LD\_E\_MSG\_WAS\_LOGGED,  
           LD\_E\_MSG\_WAS\_DISPLAYED.  
    LOOP AT IT\_RESULT3 INTO WA\_RESULT3.  
\*      WA\_SLG-MSGTY = WA\_RETURN-TYPE.  
\*      WA\_SLG-MSGID = 'ZHRWBS'.  
\*      WA\_SLG-MSGNO = '000'.  
\*      WA\_SLG-MSGV1 = WA\_RESULT3-MESSAGE\_TEXT(50).  
\*      WA\_SLG-MSGV2 = WA\_RESULT3-MESSAGE\_TEXT+50(21).  
  
      CALL FUNCTION 'BAL\_LOG\_MSG\_ADD\_FREE\_TEXT'  
        EXPORTING  
          I\_LOG\_HANDLE        = LV\_MSGHND  
          I\_MSGTY             = WA\_RESULT3-MESSAGE\_TYPE  
\*         I\_PROBCLASS         = '4'  
          I\_TEXT              = WA\_RESULT3-MESSAGE\_TEXT  
\*         I\_S\_CONTEXT         =  
\*         I\_S\_PARAMS          =  
\*         I\_DETLEVEL          = '1'  
        IMPORTING  
          E\_S\_MSG\_HANDLE      = LD\_E\_S\_MSG\_HANDLE  
          E\_MSG\_WAS\_LOGGED    = LD\_E\_MSG\_WAS\_LOGGED  
          E\_MSG\_WAS\_DISPLAYED = LD\_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.  
  
\*\*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.  
\*  
\*      ENDIF.  
    ENDLOOP.  
  
  ENDIF.  
  
ENDFORM.  
  
\*&---------------------------------------------------------------------\*  
\*&      Form  RELEASE\_WBS\_ELEMENT  
\*&---------------------------------------------------------------------\*  
\*       text  
\*----------------------------------------------------------------------\*  
\*  -->  p1        text  
\*  <--  p2        text  
\*----------------------------------------------------------------------\*  
FORM RELEASE\_WBS\_ELEMENT .  
  
\*          BAPI\_BUS2054\_SET\_STATUS  Set/Reset Status for BUS2054 (Work Breakdown Structure)  
\*  CLEAR: LWA\_WBS\_DUP,  
\*          WA\_WBS.  
\*  REFRESH: IT\_WBS.  
\*  LOOP AT LTA\_WBS INTO LWA\_WBS\_DUP WHERE PSPID = LWA\_WBS-PSPID.  
  
\*  CLEAR: WA\_WBS.  
\*  WA\_WBS-WBS\_ELEMENT = LWA\_WBS-POSID.  
\*  WA\_WBS-DESCRIPTION = LWA\_WBS-POSID.  
\*  WA\_WBS-COMPANY\_CODE =       LWA\_WBS-BUKRS.  
\*  WA\_WBS-BUSINESS\_AREA =      ''.  
\*  WA\_WBS-CONTROLLING\_AREA =   LWA\_WBS-BUKRS.  
\*  WA\_WBS-PLANT =              LWA\_WBS-BUKRS.  
\*  WA\_WBS-CURRENCY =           'CA'.  
\*  APPEND WA\_WBS TO IT\_WBS.  
  
\*  ENDLOOP.  
  
  CALL FUNCTION 'BAPI\_PS\_INITIALIZATION'.  
\*  
\*  CLEAR: WA\_WBS\_STATUS.  
\*  REFRESH: IT\_WBS\_STATUS.  
\*  LOOP AT IT\_WBS INTO WA\_WBS.  
  WA\_WBS\_STATUS-WBS\_ELEMENT = LWA\_WBS-POSID.  
  WA\_WBS\_STATUS-SET\_SYSTEM\_STATUS = 'REL'.  
  APPEND WA\_WBS\_STATUS TO IT\_WBS\_STATUS.  
\*  ENDLOOP.  
  
  CLEAR: IT\_RETURN3.  
  REFRESH: IT\_RESULT3.  
  CALL FUNCTION 'BAPI\_BUS2054\_SET\_STATUS'  
    IMPORTING  
      RETURN              = IT\_RETURN3  
    TABLES  
      I\_WBS\_SYSTEM\_STATUS = IT\_WBS\_STATUS  
\*     I\_WBS\_USER\_STATUS   =  
      E\_RESULT            = IT\_RESULT3.  
  
  IF IT\_RETURN3-TYPE = 'S' OR  
     IT\_RETURN3-TYPE IS INITIAL.  
  
    CALL FUNCTION 'BAPI\_PS\_PRECOMMIT'  
      TABLES  
        ET\_RETURN = IT\_RETURN1.  
    READ TABLE IT\_RETURN1 INTO WA\_RETURN1 WITH KEY TYPE = 'E'.  
    IF SY-SUBRC EQ 0.  
      CALL FUNCTION 'BAPI\_TRANSACTION\_ROLLBACK'.  
    ELSE.  
      CALL FUNCTION 'BAPI\_TRANSACTION\_COMMIT'  
        EXPORTING  
          WAIT = ABAP\_TRUE.  
    ENDIF.  
  
  ELSE.  
  
    CLEAR: WA\_RESULT3,  
           WA\_SLG.  
    LOOP AT IT\_RESULT3 INTO WA\_RESULT3.  
\*      WA\_SLG-MSGTY = WA\_RETURN-TYPE.  
\*      WA\_SLG-MSGID = 'ZHRWBS'.  
\*      WA\_SLG-MSGNO = '000'.  
\*      WA\_SLG-MSGV1 = WA\_RESULT3-MESSAGE\_TEXT(50).  
\*      WA\_SLG-MSGV2 = WA\_RESULT3-MESSAGE\_TEXT+50(21).  
\*  
  
      CALL FUNCTION 'BAL\_LOG\_MSG\_ADD\_FREE\_TEXT'  
        EXPORTING  
          I\_LOG\_HANDLE        = LV\_MSGHND  
          I\_MSGTY             = WA\_RESULT3-MESSAGE\_TYPE  
\*         I\_PROBCLASS         = '4'  
          I\_TEXT              = WA\_RESULT3-MESSAGE\_TEXT  
\*         I\_S\_CONTEXT         =  
\*         I\_S\_PARAMS          =  
\*         I\_DETLEVEL          = '1'  
        IMPORTING  
          E\_S\_MSG\_HANDLE      = LD\_E\_S\_MSG\_HANDLE  
          E\_MSG\_WAS\_LOGGED    = LD\_E\_MSG\_WAS\_LOGGED  
          E\_MSG\_WAS\_DISPLAYED = LD\_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.  
  
  
\*\*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.  
\*  
\*      ENDIF.  
    ENDLOOP.  
  
  ENDIF.  
  
ENDFORM.