  METHOD po\_invoice\_posting\_new.  
\*\*\*\*--------------------------------------------------------------------\*\*\*\*  
\*-------------Populate BAPI BAPI\_INCOMINGINVOICE\_CREATE1-------------------\*  
\*\*\*\*--------------------------------------------------------------------\*\*\*\*  
  
\*&---------------------------------------------------------------------\*  
\*\*--------------------------Data Declaration--------------------------\*\*  
\*&---------------------------------------------------------------------\*  
    DATA: lt\_item        TYPE STANDARD TABLE OF bapi\_incinv\_create\_item,  
          lt\_glacnt      TYPE STANDARD TABLE OF bapi\_incinv\_create\_gl\_account,  
          lt\_taxdata     TYPE STANDARD TABLE OF bapi\_incinv\_create\_tax,  
          lt\_return      TYPE STANDARD TABLE OF bapiret2,  
          lt\_exception   TYPE STANDARD TABLE OF zcl\_z\_google\_extract\_t\_mpc\_ext=>ts\_exceptions1,  
  
          ls\_processed   TYPE zcl\_z\_google\_extract\_t\_mpc\_ext=>ts\_processed1,  
          ls\_return      TYPE bapiret2,  
          ls\_ereturn     TYPE bapiret2,  
          ls\_header      TYPE bapi\_incinv\_create\_header,  
          ls\_item        TYPE bapi\_incinv\_create\_item,  
          ls\_glacnt      TYPE bapi\_incinv\_create\_gl\_account,  
          ls\_taxdata     TYPE bapi\_incinv\_create\_tax,  
          ls\_exception   TYPE zcl\_z\_google\_extract\_t\_mpc\_ext=>ts\_exceptions1,  
  
          ex\_doc\_output  TYPE  sydatum,  
          ex\_base\_output TYPE sy-datum,  
          lv\_year(4)     TYPE c,  
          lv\_month(2)    TYPE c,  
          lv\_day(2)      TYPE c,  
          lv\_len\_month   TYPE i,  
          lv\_len\_day     TYPE i,  
          lv\_lines(2)    TYPE n,  
          lv\_status      TYPE bapi\_incinv\_create\_status-rbstat,  
          lv\_invno       TYPE bapi\_incinv\_fld-inv\_doc\_no,  
          lv\_fyear       TYPE bapi\_incinv\_fld-fisc\_year,  
          lv\_sap\_intuom  TYPE meins.  
\*&---------------------------------------------------------------------\*  
\*\*--------------------------     Constant   --------------------------\*\*  
\*&---------------------------------------------------------------------\*  
    CONSTANTS:c\_doctyp TYPE blart VALUE 'RE',  
              c\_curusd TYPE waers VALUE 'USD',  
              c\_true   TYPE char4 VALUE 'true',  
              c\_zpom   TYPE blart VALUE 'ZPOM',  
              c\_scan   TYPE char6 VALUE 'Scanned',  
              c\_bapi   TYPE char4 VALUE 'BAPI'.  
  
    DESCRIBE TABLE it\_processed LINES lv\_lines.  
    READ TABLE it\_processed INTO ls\_processed INDEX 1.  
  
\*----Fetch PO Item Data from EKPO----\*  
    SELECT ebeln,ebelp,matnr,txz01,menge,meins,netpr,mwskz,txjcd,webre  
      FROM ekpo  
      INTO TABLE @DATA(lt\_ekpo)  
      WHERE ebeln = @ls\_processed-zpono.  
      IF sy-subrc EQ 0.  
        sort lt\_ekpo by ebeln ebelp webre.  
      ENDIF.  
\*----Fetch PO History from EKBE----\*  
    SELECT lfbnr,ebeln,ebelp,gjahr,belnr,menge,buzei,lfpos,shkzg,bewtp " SOSWAIN  
      INTO TABLE @DATA(lt\_ekbe)  
      FROM ekbe  
      WHERE ebeln = @ls\_processed-zpono  
      AND (  bewtp = 'E' OR bewtp = 'Q' ) . " added by SOSWAIN  
  
\*--Formatting the date and month for field zdocdate  
    SPLIT ls\_processed-zdocdate AT '-' INTO lv\_year lv\_month lv\_day.  
    lv\_len\_month = cl\_abap\_list\_utilities=>dynamic\_output\_length( lv\_month ).  
    lv\_len\_day   = cl\_abap\_list\_utilities=>dynamic\_output\_length( lv\_day ).  
  
    IF lv\_len\_day = 1.  
      CONCATENATE '0' lv\_day  INTO lv\_day.  
    ENDIF.  
    IF lv\_len\_month = 1.  
      CONCATENATE '0' lv\_month  INTO lv\_month.  
    ENDIF.  
  
    CONCATENATE lv\_year lv\_month lv\_day  INTO ex\_doc\_output.  
    CLEAR:lv\_month,lv\_day,lv\_year,lv\_len\_month,lv\_len\_month.  
  
\*--Formatting the date and month for field zbaselinedate  
    SPLIT ls\_processed-zbaselinedate AT '-' INTO lv\_year lv\_month lv\_day.  
    lv\_len\_month = cl\_abap\_list\_utilities=>dynamic\_output\_length( lv\_month ).  
    lv\_len\_day   = cl\_abap\_list\_utilities=>dynamic\_output\_length( lv\_day ).  
  
    IF lv\_len\_day = 1.  
      CONCATENATE '0' lv\_day  INTO lv\_day.  
    ENDIF.  
    IF lv\_len\_month = 1.  
      CONCATENATE '0' lv\_month  INTO lv\_month.  
    ENDIF.  
  
    CONCATENATE lv\_year lv\_month lv\_day  INTO ex\_base\_output.  
    CLEAR:lv\_month,lv\_day,lv\_year,lv\_len\_month,lv\_len\_month.  
  
    ls\_header-invoice\_ind  = abap\_true.  
    ls\_header-doc\_type     = c\_doctyp.  
    ls\_header-doc\_date     = ex\_doc\_output."ls\_processed-zdocdate."sy-datum.changed on 5/20 Parvathy  
    ls\_header-pstng\_date   = sy-datum.  
    ls\_header-ref\_doc\_no   = ls\_processed-zinvno.  
    ls\_header-comp\_code    = ls\_processed-zcompanycode.  
    ls\_header-currency     = c\_curusd."ls\_processed-zcurrency.  
    ls\_header-gross\_amount = ls\_processed-zgrossamount.  
    ls\_header-pmnttrms     = ls\_processed-zbilltoemail." ls\_processed-zpaymentterm .changed on 5/20 Parvathy, zbilltoemail is the payment term key and zpayment term is the payment term dscription  
    ls\_header-bline\_date   = ex\_base\_output."ls\_processed-zbaselinedate.  
    ls\_header-diff\_inv     = ls\_processed-zsuppliername2.  
    ls\_header-del\_costs    = ls\_processed-zinvfreightamt.  
    ls\_header-header\_txt   = ls\_processed-zheadertext.  
    ls\_header-pymt\_meth    = ls\_processed-zpaymentmethod.  
    ls\_header-pmnt\_block   = ls\_processed-zpaymentblock.  
    ls\_header-iv\_category  = ls\_processed-zinvtype.  
    ls\_header-partner\_bk   = ls\_processed-zbankpartnertype.  
    ls\_header-housebankid  = ls\_processed-zhousebankaccid.  
    ls\_header-pmtmthsupl   = ls\_processed-zpaymentmethodsupp.  
  
    IF ls\_processed-zautocaltax = c\_true.  
      ls\_header-calc\_tax\_ind = abap\_true.  
    ELSE.  
      ls\_header-calc\_tax\_ind = ''.  
    ENDIF.  
    ls\_header-exch\_rate    = ls\_processed-zexchangerate.  
    ls\_header-payee\_payer  = ls\_processed-zalternatepayee.  
  
\*----GL Details----\*  
    ls\_glacnt-gl\_account       = ls\_processed-zdirectpostgl.  
    ls\_glacnt-invoice\_doc\_item = ls\_processed-zdocitmininvdoc.  
    ls\_glacnt-item\_amount      = ls\_processed-zunitprice.  
    ls\_glacnt-item\_text        = ls\_processed-zitemdesc.  
  
    IF ls\_header-calc\_tax\_ind = ''.  
      READ TABLE it\_processed INTO ls\_processed INDEX lv\_lines.  
      ls\_taxdata-tax\_amount    = ls\_processed-ztotaltaxamt.  
      APPEND ls\_taxdata TO lt\_taxdata.  
    ENDIF.  
  
    LOOP AT it\_processed INTO ls\_processed.  
  
      ls\_item-invoice\_doc\_item = ls\_processed-zitemno.  
      ls\_item-po\_number        = ls\_processed-zpono.  
      ls\_item-po\_item          = ls\_processed-zitemno.  
  
      CALL FUNCTION 'CONVERSION\_EXIT\_CUNIT\_INPUT'  
        EXPORTING  
          input    = ls\_processed-zitemunit  
          language = sy-langu  
        IMPORTING  
          output   = lv\_sap\_intuom.  
  
      ls\_item-po\_unit          = lv\_sap\_intuom."ls\_processed-zitemunit.  
      ls\_item-quantity         = ls\_processed-zitemqty.  
      ls\_item-item\_amount      = ls\_processed-zitemqty \* ls\_processed-zunitprice."ls\_processed-ztotalamount.  
      ls\_item-item\_text        = ls\_processed-zitemdesc.  
\*\*\*\*\*moving value that is fetched to tax code and jurisdiction.  
      ls\_item-tax\_code         = ls\_processed-zitemtax.  
      ls\_item-taxjurcode       = ls\_processed-ztaxjursidiction.  
  
\*----GR Based PO----\*  
      READ TABLE lt\_ekpo INTO DATA(ls\_ekpo) WITH KEY ebeln = ls\_processed-zpono  
                                                     ebelp = ls\_processed-zitemno  
                                                     webre = 'X'  
                                                     BINARY SEARCH.  
      IF sy-subrc EQ 0.  
        DATA :lt\_ekbe\_1    LIKE lt\_ekbe,  
              lt\_ekbe\_2    LIKE lt\_ekbe,  
              lv\_total\_qty TYPE menge\_d.  
\*-- Create the 1st part list in internal table lt\_ekbe\_1  
        LOOP AT lt\_ekbe INTO DATA(ls\_ekbe) WHERE ebeln = ls\_processed-zpono AND  
                                                 ebelp = ls\_processed-zitemno AND  
                                                 bewtp = 'E'.  
          IF ls\_ekbe-belnr EQ ls\_ekbe-lfbnr.  
            APPEND ls\_ekbe TO lt\_ekbe\_1.  
          ELSE.  
            READ TABLE lt\_ekbe\_1 TRANSPORTING NO FIELDS WITH KEY belnr = ls\_ekbe-lfbnr  
                                                                 shkzg = 'S'.  
            IF sy-subrc EQ 0.  
              DELETE lt\_ekbe\_1 WHERE belnr = ls\_ekbe-lfbnr.  
            ENDIF.  
          ENDIF.  
        ENDLOOP.  
        CLEAR ls\_ekbe.  
        SORT lt\_ekbe BY ebeln ebelp bewtp shkzg.  
        DATA(lt\_ekbe1) = lt\_ekbe.  
        DELETE lt\_ekbe1 WHERE ebelp NE ls\_processed-zitemno.  
        DELETE lt\_ekbe1 WHERE bewtp NE 'Q'.  
        SORT lt\_ekbe1 BY lfbnr.  
\*-- Create the 2nd part list in internal table lt\_ekbe\_2  
        sort lt\_ekbe\_1 by belnr.  
        LOOP AT lt\_ekbe1 INTO DATA(ls\_ekbe1).  
          ls\_ekbe = ls\_ekbe1.  
          READ TABLE lt\_ekbe\_1 INTO DATA(ls\_ekbe\_1) WITH KEY belnr = ls\_ekbe-lfbnr BINARY SEARCH.  
          IF sy-subrc EQ 0.  
            IF ls\_ekbe-shkzg = 'S'.  
              lv\_total\_qty = lv\_total\_qty + ls\_ekbe-menge.  
            ELSEIF ls\_ekbe-shkzg = 'H'.  
              lv\_total\_qty = lv\_total\_qty - ls\_ekbe-menge.  
            ENDIF.  
          ENDIF.  
          AT END OF lfbnr.  
            ls\_ekbe-menge = lv\_total\_qty.  
            CLEAR lv\_total\_qty.  
            APPEND ls\_ekbe TO lt\_ekbe\_2.  
          ENDAT.  
\*          ELSE.  
\*            ls\_item-quantity             = ls\_processed-zitemqty. "ls\_ekbe-menge.  
\*            ls\_item-ref\_doc              = ls\_ekbe-belnr.  
\*            ls\_item-ref\_doc\_it           = ls\_ekbe-buzei.  
\*            ls\_item-ref\_doc\_year         = ls\_ekbe-gjahr.  
\*            APPEND ls\_item TO lt\_item.  
\*          ENDIF.  
        ENDLOOP.  
  
\*-- If no invoice posted  
        IF lt\_ekbe1 IS INITIAL.  
          CLEAR ls\_ekbe.  
          READ TABLE lt\_ekbe\_1 INTO ls\_ekbe INDEX 1.  
          IF sy-subrc EQ 0.  
            ls\_item-quantity             = ls\_processed-zitemqty. "ls\_ekbe-menge.  
            ls\_item-ref\_doc              = ls\_ekbe-belnr.  
            ls\_item-ref\_doc\_it           = ls\_ekbe-buzei.  
            ls\_item-ref\_doc\_year         = ls\_ekbe-gjahr.  
          ENDIF.  
        ENDIF.  
  
\*-- Create the final list  
        CLEAR lt\_ekbe1.  
        SORT lt\_ekbe\_2 by lfbnr.  
        LOOP AT lt\_ekbe\_1 INTO ls\_ekbe\_1.  
          READ TABLE lt\_ekbe\_2 INTO DATA(ls\_ekbe\_2) WITH KEY lfbnr = ls\_ekbe\_1-belnr BINARY SEARCH.  
          IF sy-subrc EQ 0.  
            IF ls\_ekbe\_1-menge >= ls\_ekbe\_2-menge.  
              ls\_ekbe\_2-menge = ls\_ekbe\_1-menge - ls\_ekbe\_2-menge.  
              APPEND ls\_ekbe\_2 TO lt\_ekbe1.  
            ENDIF.  
          ENDIF.  
        ENDLOOP.  
  
\*-- Filling the ITEM internal table for posting  
        DATA(lv\_inv\_qty) = ls\_processed-zitemqty.  
        LOOP AT lt\_ekbe1 INTO ls\_ekbe1.  
          IF lv\_inv\_qty GE ls\_ekbe1-menge.  
            lv\_inv\_qty = lv\_inv\_qty - ls\_ekbe1-menge.  
            ls\_item-quantity             = ls\_ekbe1-menge. "ls\_ekbe-menge.  
            ls\_item-ref\_doc              = ls\_ekbe1-lfbnr.  
            ls\_item-ref\_doc\_it           = ls\_ekbe1-buzei.  
            ls\_item-ref\_doc\_year         = ls\_ekbe1-gjahr.  
            APPEND ls\_item TO lt\_item.  
            CLEAR ls\_item.  
          ELSE.  
            IF lv\_inv\_qty GT 0.  
              ls\_item-quantity             = lv\_inv\_qty. "ls\_ekbe-menge.  
              ls\_item-ref\_doc              = ls\_ekbe1-lfbnr.  
              ls\_item-ref\_doc\_it           = ls\_ekbe1-buzei.  
              ls\_item-ref\_doc\_year         = ls\_ekbe1-gjahr.  
              APPEND ls\_item TO lt\_item.  
              CLEAR ls\_item.  
              lv\_inv\_qty = 0.  
              EXIT.  
            ENDIF.  
          ENDIF.  
        ENDLOOP.  
\*        DATA(lv\_inv\_qty) = ls\_processed-zitemqty.  
\*        LOOP AT lt\_ekbe\_2 INTO ls\_ekbe.  
\*          lv\_inv\_qty = lv\_inv\_qty - ls\_ekbe-menge.  
\*          IF lv\_inv\_qty GE 0.  
\*            ls\_item-quantity             = lv\_inv\_qty. "ls\_ekbe-menge.  
\*            ls\_item-ref\_doc              = ls\_ekbe-belnr.  
\*            ls\_item-ref\_doc\_it           = ls\_ekbe-buzei.  
\*            ls\_item-ref\_doc\_year         = ls\_ekbe-gjahr.  
\*            APPEND ls\_item TO lt\_item.  
\*          ELSE.  
\*          ENDIF.  
\*        ENDLOOP.  
      ENDIF.  
      IF ls\_item IS NOT INITIAL.  
        APPEND ls\_item TO lt\_item.  
      ENDIF.  
  
      CLEAR:lt\_ekbe\_2, lt\_ekbe\_1.  
      CLEAR: ls\_item, ls\_processed, ls\_ekbe, ls\_ekpo.  
    ENDLOOP.  
  
    SORT lt\_item BY invoice\_doc\_item po\_number po\_item ref\_doc ref\_doc\_it ref\_doc\_year quantity.  
    DELETE ADJACENT DUPLICATES FROM lt\_item COMPARING invoice\_doc\_item po\_number po\_item ref\_doc ref\_doc\_it ref\_doc\_year quantity.  
  
    CALL FUNCTION 'BAPI\_INCOMINGINVOICE\_CREATE1'  
      EXPORTING  
        headerdata       = ls\_header  
        invoicestatus    = '5'  
      IMPORTING  
        invoicedocnumber = lv\_invno  
        fiscalyear       = lv\_fyear  
      TABLES  
        itemdata         = lt\_item  
        taxdata          = lt\_taxdata  
        return           = lt\_return.  
  
    IF lv\_invno IS NOT INITIAL.  
      ex\_invoice\_no = lv\_invno.  
      CALL FUNCTION 'BAPI\_TRANSACTION\_COMMIT'.  
    ELSEIF lt\_return IS NOT INITIAL.  
  
      READ TABLE it\_processed INTO DATA(ls\_prc) INDEX 1.  
  
      DELETE lt\_return WHERE type = 'S'.  
      LOOP AT lt\_return INTO ls\_return.  
        ls\_ereturn-field      =  ls\_return-field.  
        ls\_ereturn-id         =  ls\_return-id.  
        ls\_ereturn-log\_msg\_no =  ls\_ereturn-log\_msg\_no.  
        ls\_ereturn-message    =  ls\_return-message.  
        ls\_ereturn-message\_v1 =  ls\_return-message\_v1.  
        ls\_ereturn-message\_v2 =  ls\_return-message\_v2.  
        ls\_ereturn-message\_v3 =  ls\_return-message\_v3.  
        ls\_ereturn-message\_v4 =  ls\_return-message\_v4.  
        ls\_ereturn-number     =  ls\_return-number.  
        ls\_ereturn-type       =  ls\_return-type.  
  
        APPEND ls\_ereturn TO et\_return.  
  
\*-------------------------\*  
\*----Update BAPI Messages into ZOCR\_EXCEPTION Table----\*  
\*-------------------------\*  
        ls\_exception-zimagefingerprint      = ls\_prc-zimagefingerprint.  
        ls\_exception-zinvno                 = ls\_prc-zinvno.  
        ls\_exception-zdoctype               = c\_zpom.  
        ls\_exception-zexceptionno           = ls\_return-message.  
        ls\_exception-zexceptionstatus       = c\_scan.  
        ls\_exception-zmessagerasiedduring   = c\_bapi.  
        ls\_exception-zlastchecked           = sy-uzeit.  
        APPEND ls\_exception TO lt\_exception.  
  
        CLEAR: ls\_ereturn, ls\_return, ls\_exception.  
      ENDLOOP.  
      CLEAR: ls\_prc.  
  
      et\_exceptions = lt\_exception.  
    ENDIF.  
  ENDMETHOD.