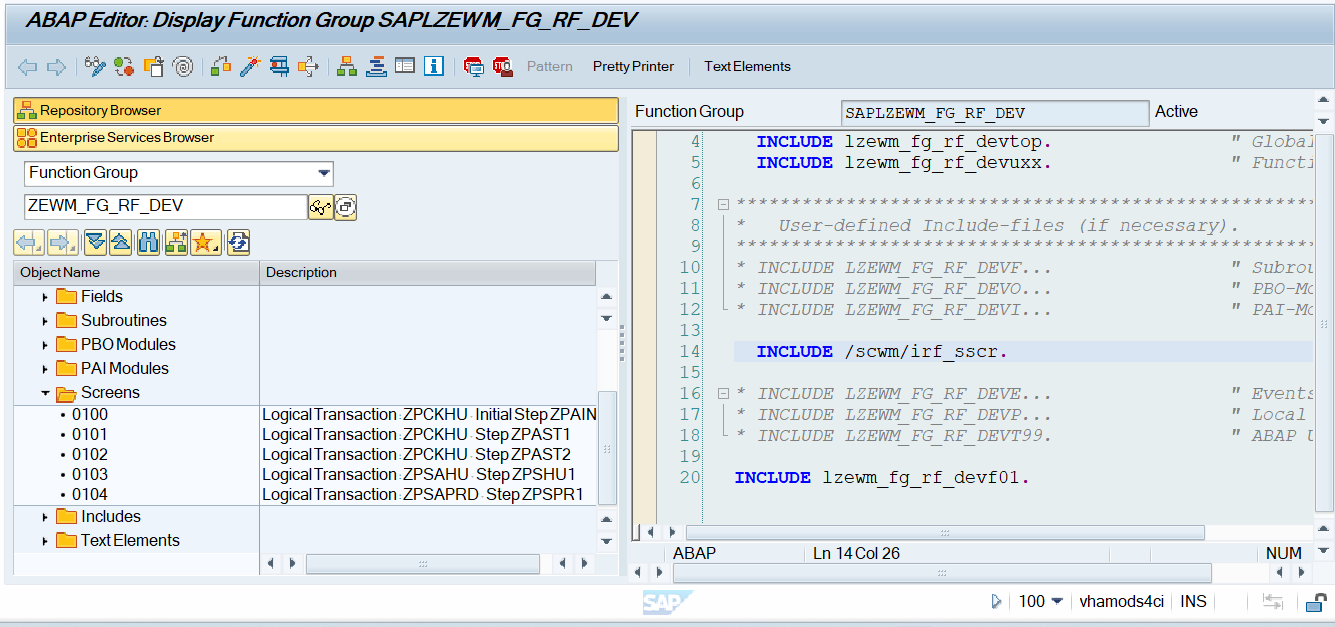
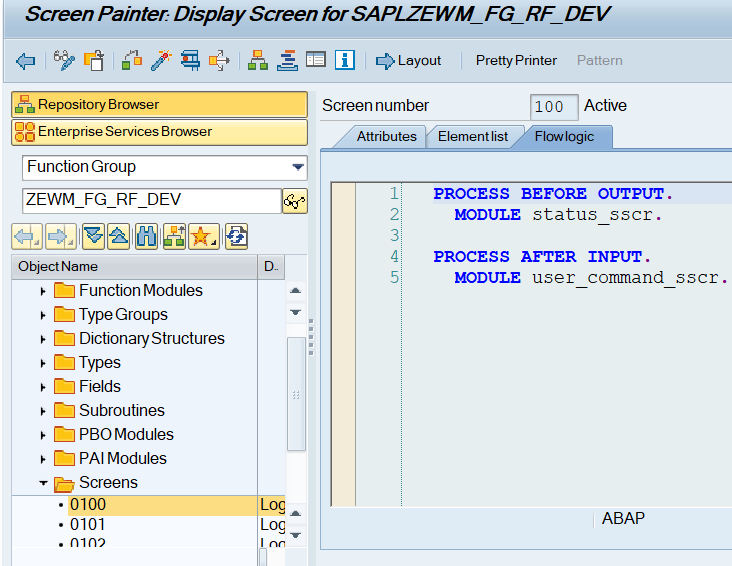
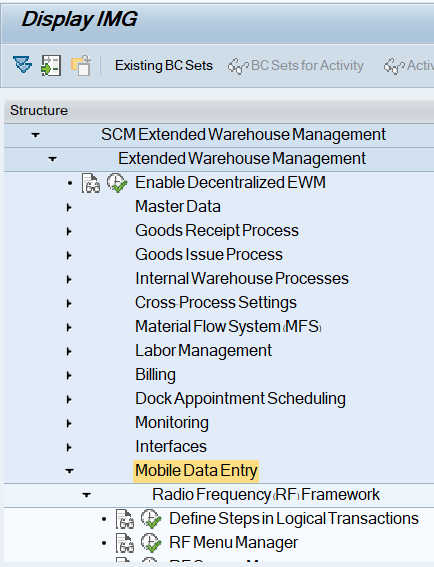
SAP EWM – RF Screen Framework

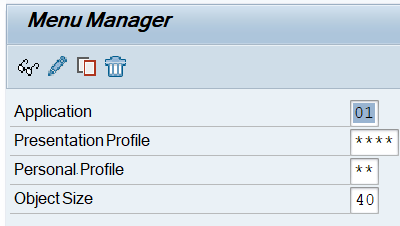
**Create custom screens for creating the Handling Units with the Supplier scanned handling unit number followed by its GR posting and Creation/Confirmation of Warehouse Task.**

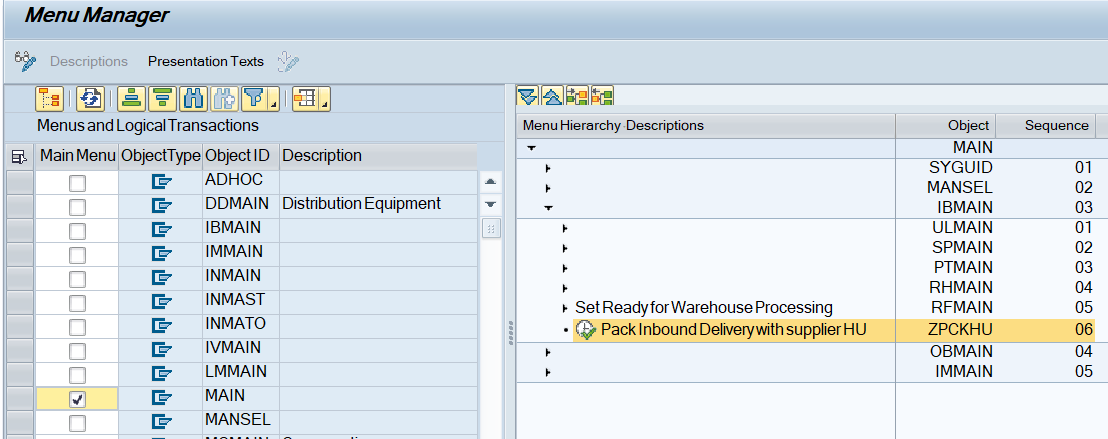
Create custom function group/required sub screens with layouts as shown below and add the below include program in it: INCLUDE /scwm/irf\_sscr 



Add below shown buttons as per requirement in RF Menu Manager in below SPRO path:

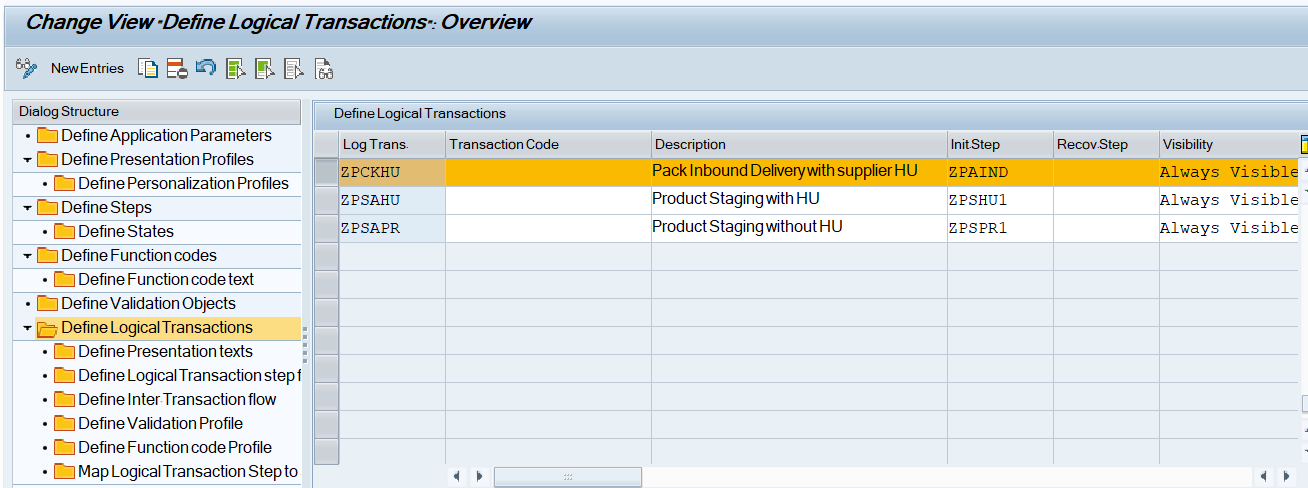


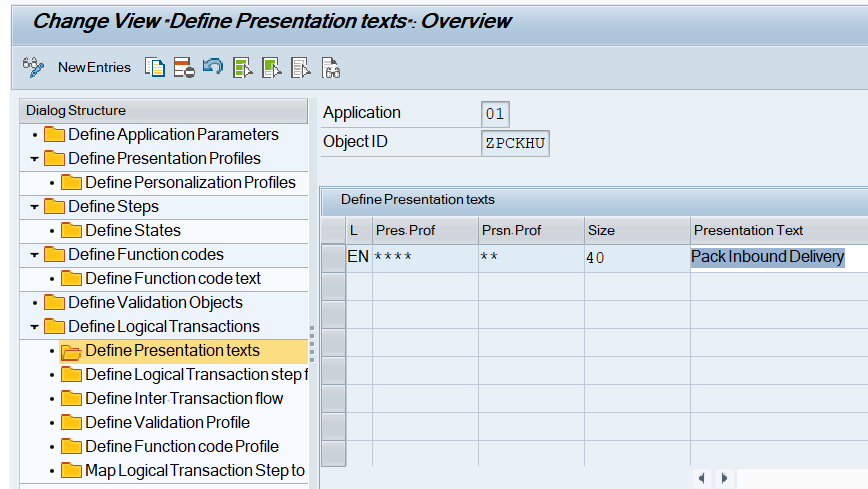




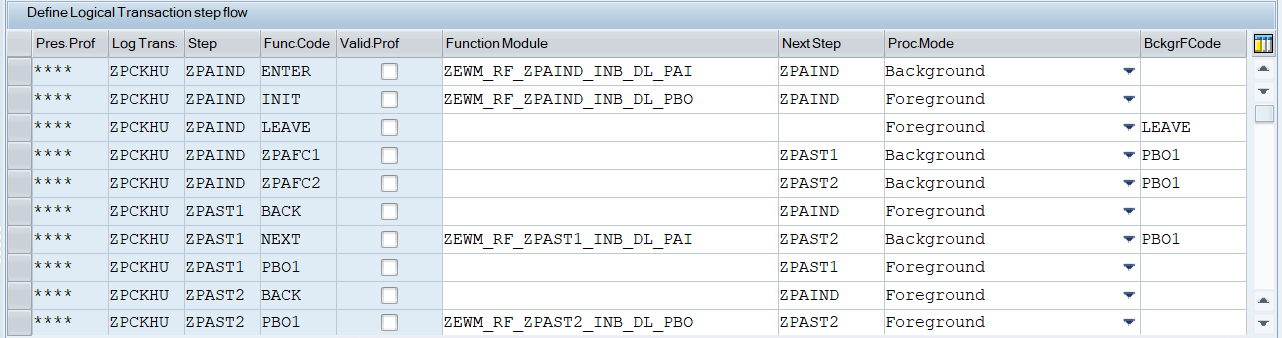
Configuration done in Define steps in Logical Transaction:

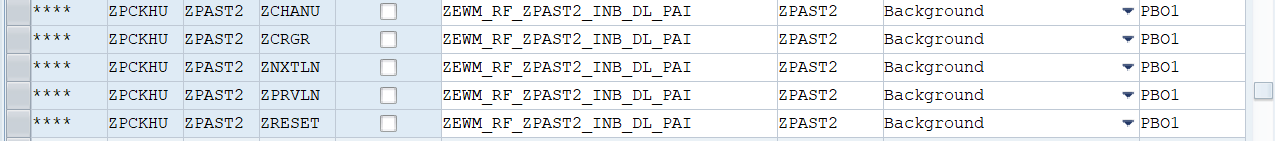
* Define Logical Transaction



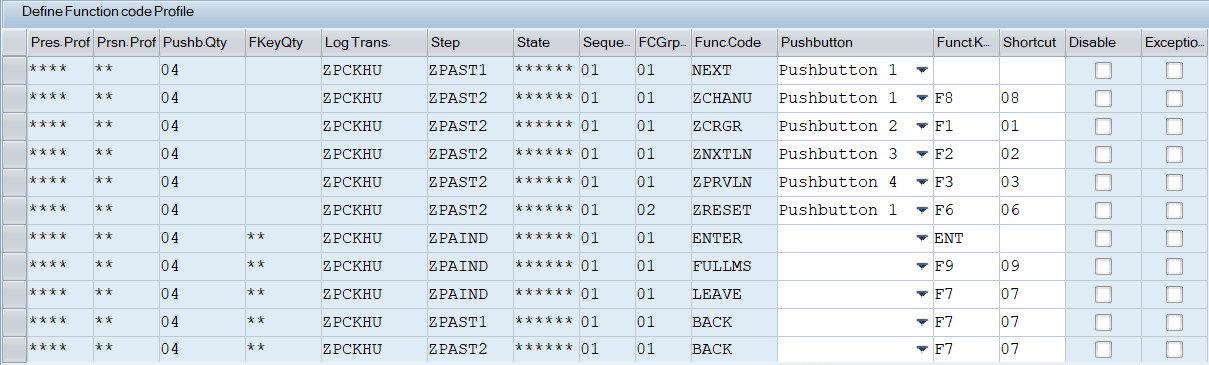


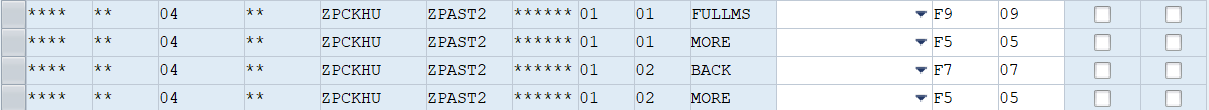
Logical Transaction step flow:



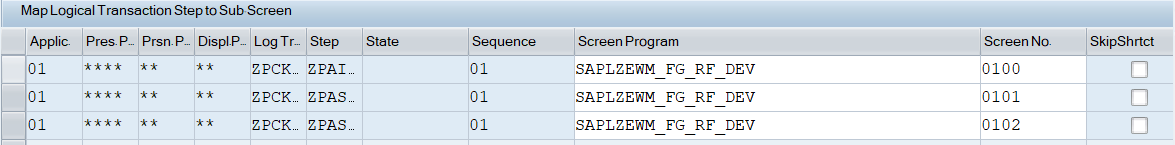


Define Function code Profile.

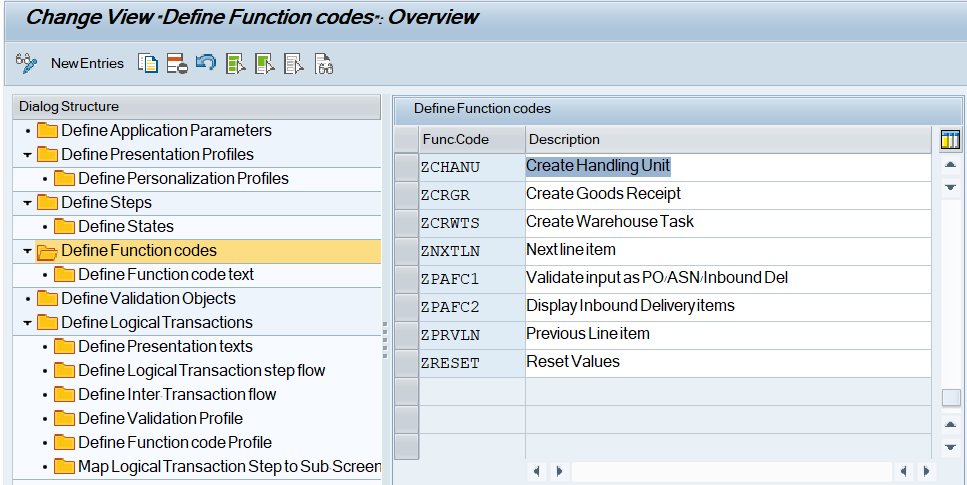




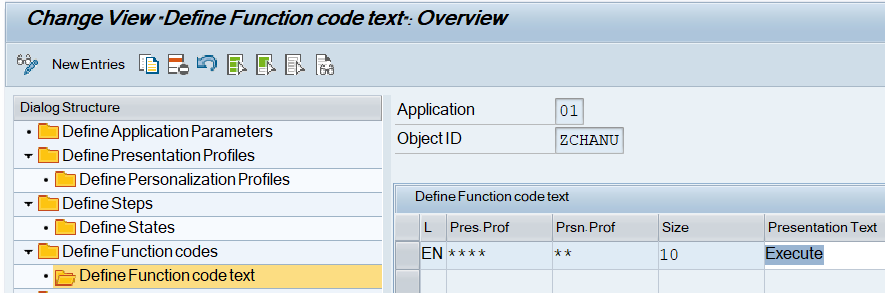
Map Logical Transaction Step to Sub-Screen



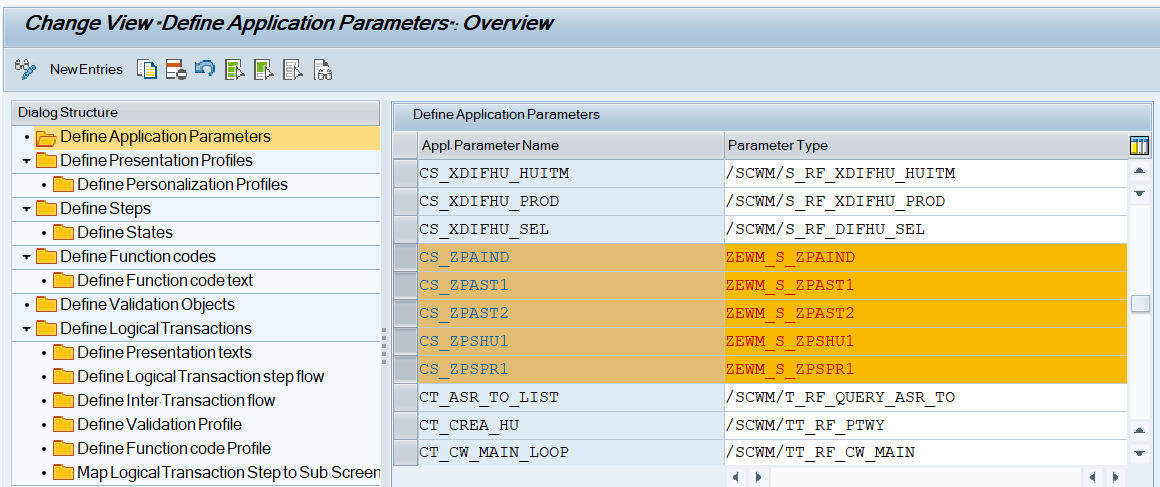
Define custom function codes.

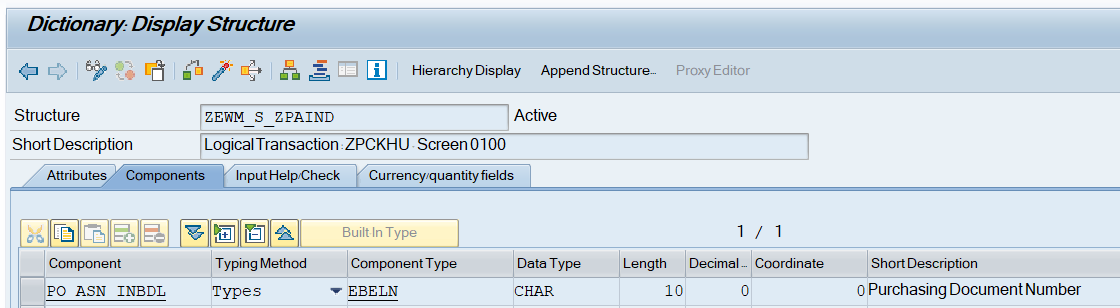


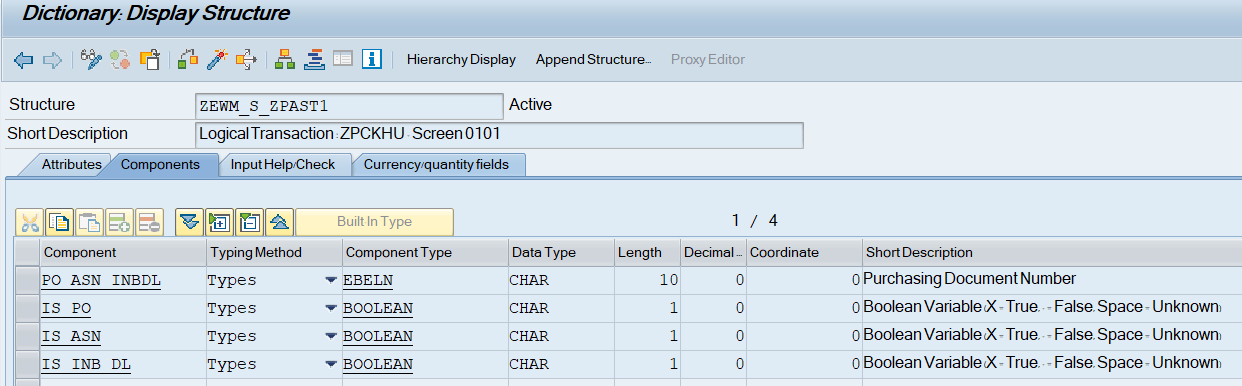
Define function code text for each custom function code as shown below:

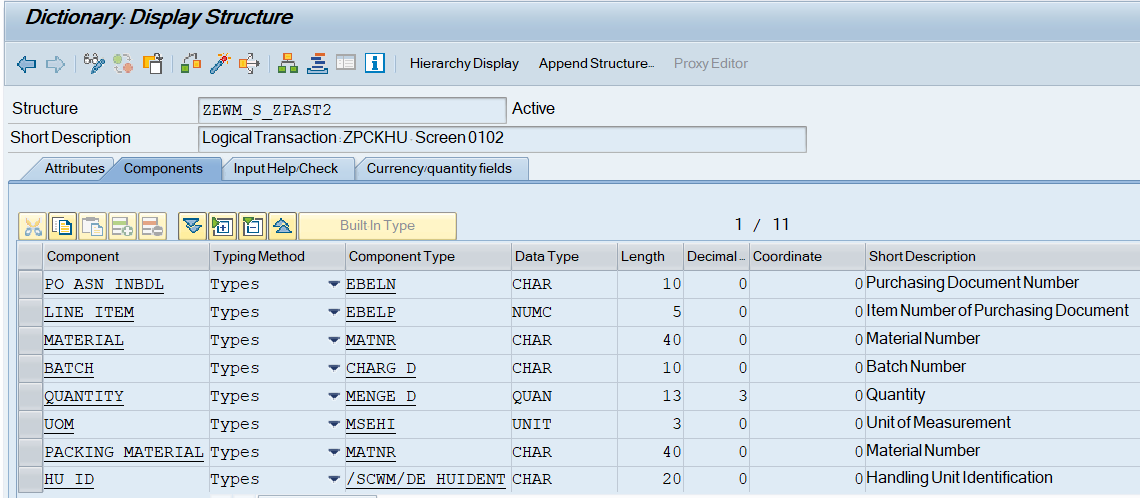


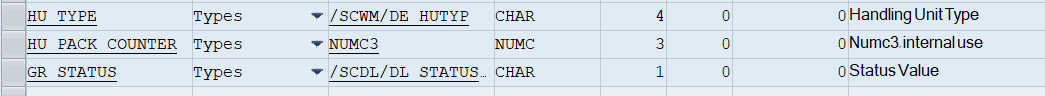
Define screen parameters structures.











Technical configuration details of complete development:

Logical Transaction: ZPCKHU  Pack Inbound Delivery with supplier HU

Steps:

ZPAIND      Initial Step                   Screen: 0100     Screen Structure: ZEWM\_S\_ZPAIND

ZPAST1      Validate input as PO/ASN/Inb DL Screen: 0101     Screen Structure: ZEWM\_S\_ZPAST1

ZPAST2      Display Inbound Delivery items  Screen: 0102     Screen Structure: ZEWM\_S\_ZPAST2

FM:

ZEWM\_RF\_ZPAIND\_INB\_DL\_PAI

ZEWM\_RF\_ZPAIND\_INB\_DL\_PBO

ZEWM\_RF\_ZPAST1\_INB\_DL\_PAI

ZEWM\_RF\_ZPAST2\_INB\_DL\_PAI

-------------------------------------------------------------------------------------------------------------------------------------------

FUNCTION zewm\_rf\_zpaind\_inb\_dl\_pbo.  
\*"----------------------------------------------------------------------  
\*"\*"Local Interface:  
\*"  CHANGING  
\*"     REFERENCE(CS\_ZPAIND) TYPE  ZEWM\_S\_ZPAIND  
\*"     REFERENCE(CS\_ZPAST1) TYPE  ZEWM\_S\_ZPAST1  
\*"     REFERENCE(CS\_ZPAST2) TYPE  ZEWM\_S\_ZPAST2  
\*"----------------------------------------------------------------------  
  
  CLEAR:cs\_zpaind,  
        cs\_zpast1.  
  
\*Set screen parameters  
  CALL METHOD /scwm/cl\_rf\_bll\_srvc=>init\_screen\_param.  
  
  CALL METHOD /scwm/cl\_rf\_bll\_srvc=>set\_screen\_param  
    EXPORTING  
      iv\_param\_name = gc\_param\_cs\_zpaind.  
  
  CALL METHOD /scwm/cl\_rf\_bll\_srvc=>set\_screen\_param  
    EXPORTING  
      iv\_param\_name = gc\_param\_cs\_zpast1.  
  
  CALL METHOD /scwm/cl\_rf\_bll\_srvc=>set\_screen\_param  
    EXPORTING  
      iv\_param\_name = gc\_param\_cs\_zpast2.  
  
ENDFUNCTION.

 -------------------------------------------------------------------------------------------------------------------------------------------

 -------------------------------------------------------------------------------------------------------------------------------------------

FUNCTION zewm\_rf\_zpaind\_inb\_dl\_pai.  
\*"----------------------------------------------------------------------  
\*"\*"Local Interface:  
\*"  CHANGING  
\*"     REFERENCE(CS\_ZPAIND) TYPE  ZEWM\_S\_ZPAIND  
\*"     REFERENCE(CS\_ZPAST1) TYPE  ZEWM\_S\_ZPAST1  
\*"     REFERENCE(CS\_ZPAST2) TYPE  ZEWM\_S\_ZPAST2  
\*"----------------------------------------------------------------------  
  
  DATA:lv\_inb\_dl  TYPE /scdl/dl\_docno\_int,  
       lv\_ibd\_no  TYPE /scdl/dl\_docno\_int,  
       lv\_counter TYPE i.  
  
  DATA:ls\_huhdr TYPE /scwm/s\_huhdr\_int.  
  
  DATA:lt\_huitm TYPE /scwm/tt\_huitm\_int,  
       lt\_huref TYPE /scwm/tt\_huref\_int.  
  
  DATA:lo\_packing TYPE REF TO /scwm/cl\_wm\_packing.  
  
  CLEAR gt\_process\_data.  
  
  IF cs\_zpaind-po\_asn\_inbdl IS INITIAL.  
    MESSAGE e006(zewm01).  
  ENDIF.  
  
\*Set values in the screen to be navigated  
  cs\_zpast1-po\_asn\_inbdl = cs\_zpaind-po\_asn\_inbdl.  
  cs\_zpast2-po\_asn\_inbdl = cs\_zpaind-po\_asn\_inbdl.  
  
\*Check for input value as PO/ASN/Inbound Delivery  
  
  SELECT SINGLE @abap\_true  
  FROM ekko  
  INTO @DATA(lv\_exists)  
  WHERE ebeln EQ @cs\_zpaind-po\_asn\_inbdl.  
  IF sy-subrc EQ 0.  
    DATA(lv\_po) = abap\_true.  
    lv\_counter = 1.  
  ENDIF.  
  
  CLEAR lv\_exists.  
  SELECT SINGLE @abap\_true  
  FROM /scdl/db\_proch\_i  
  INTO @lv\_exists  
  WHERE refdocno\_extasn EQ @cs\_zpaind-po\_asn\_inbdl.  
  IF sy-subrc EQ 0.  
    DATA(lv\_asn) = abap\_true.  
  
    IF lv\_counter IS INITIAL.  
      lv\_counter = 1.  
    ELSE.  
      lv\_counter = lv\_counter + 1.  
    ENDIF.  
  ENDIF.  
  
  lv\_inb\_dl = cs\_zpaind-po\_asn\_inbdl.  
  CALL FUNCTION 'CONVERSION\_EXIT\_ALPHA\_INPUT'  
    EXPORTING  
      input  = lv\_inb\_dl  
    IMPORTING  
      output = lv\_inb\_dl.  
  
  CLEAR lv\_exists.  
  SELECT SINGLE @abap\_true  
  FROM /scdl/db\_proch\_i  
  INTO @lv\_exists  
  WHERE docno EQ @lv\_inb\_dl.  
  IF sy-subrc EQ 0.  
    DATA(lv\_inb\_delv) = abap\_true.  
  
    IF lv\_counter IS INITIAL.  
      lv\_counter = 1.  
    ELSE.  
      lv\_counter = lv\_counter + 1.  
    ENDIF.  
  ENDIF.  
  
  IF lv\_counter IS INITIAL.  
    MESSAGE e006(zewm01).  
  ELSE.  
    IF lv\_counter GT 1.  
      /scwm/cl\_rf\_bll\_srvc=>set\_prmod( EXPORTING iv\_prmod = '1' ).  
      /scwm/cl\_rf\_bll\_srvc=>set\_fcode( EXPORTING iv\_fcode = gc\_scr\_fcode\_zpafc1 ).  
    ELSE.  
  
\*-------------------------------------------------------------------------------------  
\*Get data for processing against a PO  
\*-------------------------------------------------------------------------------------  
      IF lv\_po EQ abap\_true.  
  
\*Validate if multiple inbound deliveries exists for a PO  
        SELECT DISTINCT COUNT( docno )  
        FROM /scdl/db\_proci\_i  
        WHERE refdocno\_po EQ @cs\_zpaind-po\_asn\_inbdl  
        INTO @DATA(lv\_total\_inb\_delv).  
        IF sy-subrc EQ 0 AND  
           lv\_total\_inb\_delv GT 1.  
          MESSAGE e008(zewm01).  
        ENDIF.  
  
        SELECT docid,  
               docno,  
               itemid,  
               itemno,  
               productno,  
               productid,  
               batchno,  
               qty,  
               uom,  
               refdocno\_po,  
               /scwm/whno  
        FROM /scdl/db\_proci\_i  
        WHERE refdocno\_po EQ @cs\_zpaind-po\_asn\_inbdl  
        INTO CORRESPONDING FIELDS OF TABLE @gt\_process\_data.  
        IF sy-subrc NE 0.  
          MESSAGE e012(zewm01) WITH cs\_zpaind-po\_asn\_inbdl.  
        ENDIF.  
      ENDIF.  
\*-------------------------------------------------------------------------------------  
  
\*-------------------------------------------------------------------------------------  
\*Get data for processing against a ASN  
\*-------------------------------------------------------------------------------------  
      IF lv\_asn EQ abap\_true.  
  
\*Validate if multiple inbound deliveries exists for a ASN  
        SELECT DISTINCT COUNT( docno )  
        FROM /scdl/db\_proch\_i  
        WHERE refdocno\_extasn EQ @cs\_zpaind-po\_asn\_inbdl  
        INTO @lv\_total\_inb\_delv.  
        IF sy-subrc EQ 0 AND  
           lv\_total\_inb\_delv GT 1.  
          MESSAGE e009(zewm01).  
        ENDIF.  
  
        SELECT a~docid,  
               a~docno,  
               b~itemid,  
               b~itemno,  
               b~productno,  
               b~productid,  
               b~batchno,  
               b~qty,  
               b~uom,  
               a~refdocno\_extasn,  
               b~/scwm/whno  
         FROM /scdl/db\_proch\_i AS a INNER JOIN /scdl/db\_proci\_i AS b ON b~docid EQ a~docid AND  
                                                                        b~docno EQ a~docno  
         WHERE a~refdocno\_extasn EQ @cs\_zpaind-po\_asn\_inbdl  
         INTO CORRESPONDING FIELDS OF TABLE @gt\_process\_data.  
        IF sy-subrc NE 0.  
          MESSAGE e012(zewm01) WITH cs\_zpaind-po\_asn\_inbdl.  
        ENDIF.  
      ENDIF.  
\*-------------------------------------------------------------------------------------  
  
\*-------------------------------------------------------------------------------------  
\*Get data for processing against a Inbound Delivery  
\*-------------------------------------------------------------------------------------  
      IF lv\_inb\_delv EQ abap\_true.  
  
        SELECT docid,  
               docno,  
               itemid,  
               itemno,  
               productno,  
               productid,  
               batchno,  
               qty,  
               uom,  
               /scwm/whno  
        FROM /scdl/db\_proci\_i  
        WHERE docno EQ @lv\_inb\_dl  
        INTO CORRESPONDING FIELDS OF TABLE @gt\_process\_data.  
        IF sy-subrc NE 0.  
          MESSAGE e012(zewm01) WITH cs\_zpaind-po\_asn\_inbdl.  
        ENDIF.  
      ENDIF.  
\*-------------------------------------------------------------------------------------  
  
\*-------------------------------------------------------------------------------------  
\*Exclude the line items for which the GR is already done  
\*-------------------------------------------------------------------------------------  
      IF gt\_process\_data IS NOT INITIAL.  
        SELECT docid,  
               itemid,  
               status\_type,  
               status\_value  
        FROM /scdl/db\_status  
        FOR ALL ENTRIES IN @gt\_process\_data  
        WHERE docid        EQ @gt\_process\_data-docid  AND  
              itemid       EQ @gt\_process\_data-itemid AND  
              status\_type  EQ 'DGR'                  AND  
              status\_value EQ '9'  
        INTO TABLE @DATA(lt\_status).  
        IF sy-subrc EQ 0.  
          LOOP AT lt\_status ASSIGNING FIELD-SYMBOL(<lfs\_status>).  
            DELETE gt\_process\_data WHERE docid  EQ <lfs\_status>-docid AND  
                                         itemid EQ <lfs\_status>-itemid.  
          ENDLOOP.  
        ENDIF.  
      ENDIF.  
  
\*-------------------------------------------------------------------------------------  
\*Reduce the quantity already packed from quantity to be packed.  
\*-------------------------------------------------------------------------------------  
      IF lo\_packing IS NOT BOUND.  
        lo\_packing = NEW #( ).  
      ENDIF.  
  
      IF gt\_process\_data IS NOT INITIAL.  
        SELECT a~docid,  
               a~itemid,  
               b~guid\_parent,  
               b~qdocid,  
               b~qitmid,  
               c~guid\_hu,  
               c~huident  
        FROM /scdl/db\_proci\_i AS a INNER JOIN /scwm/quan AS b ON b~qdocid  EQ a~docid  AND  
                                                                 b~qitmid  EQ a~itemid AND  
                                                                 b~qdoccat EQ 'PDI'  
                                   INNER JOIN /scwm/huhdr AS c ON c~guid\_hu EQ b~guid\_parent  
        FOR ALL ENTRIES IN @gt\_process\_data  
        WHERE a~docno  EQ @gt\_process\_data-docno  AND  
              a~itemno EQ @gt\_process\_data-itemno  
        INTO TABLE @DATA(lt\_hu\_data).  
        IF sy-subrc EQ 0.  
          SORT lt\_hu\_data BY docid itemid guid\_hu huident.  
  
          LOOP AT lt\_hu\_data ASSIGNING FIELD-SYMBOL(<lfs\_hu\_data>).  
  
            CLEAR:ls\_huhdr,  
                  lt\_huitm,  
                  lt\_huref.  
  
            CALL METHOD lo\_packing->get\_hu  
              EXPORTING  
                iv\_huident = <lfs\_hu\_data>-huident  
              IMPORTING  
                es\_huhdr   = ls\_huhdr  
                et\_huitm   = lt\_huitm  
                et\_huref   = lt\_huref  
              EXCEPTIONS  
                not\_found  = 1  
                OTHERS     = 2.  
            IF sy-subrc <> 0.  
              MESSAGE ID sy-msgid TYPE sy-msgty NUMBER sy-msgno WITH sy-msgv1 sy-msgv2 sy-msgv3 sy-msgv4.  
            ENDIF.  
  
            LOOP AT lt\_huitm ASSIGNING FIELD-SYMBOL(<lfs\_huitm>).  
              READ TABLE gt\_process\_data ASSIGNING FIELD-SYMBOL(<lfs\_process\_data>)  
                  WITH KEY docid  = <lfs\_hu\_data>-docid  
                           itemid = <lfs\_hu\_data>-itemid.  
              IF sy-subrc EQ 0.  
                <lfs\_process\_data>-qty = <lfs\_process\_data>-qty - <lfs\_huitm>-quan.  
              ENDIF.  
            ENDLOOP.  
  
          ENDLOOP.  
        ENDIF.  
      ENDIF.  
  
\*-------------------------------------------------------------------------------------  
\*Map data to the screen structure  
\*-------------------------------------------------------------------------------------  
      SORT gt\_process\_data BY itemno.  
      DATA(ls\_process\_data) = gt\_process\_data[ 1 ].  
  
      CLEAR lv\_ibd\_no.  
      lv\_ibd\_no = ls\_process\_data-docno.  
      CALL FUNCTION 'CONVERSION\_EXIT\_ALPHA\_OUTPUT'  
        EXPORTING  
          input  = lv\_ibd\_no  
        IMPORTING  
          output = lv\_ibd\_no.  
  
      cs\_zpast2-po\_asn\_inbdl = lv\_ibd\_no.  
      cs\_zpast2-line\_item    = ls\_process\_data-itemno.  
      cs\_zpast2-material     = ls\_process\_data-productno.  
      cs\_zpast2-batch        = ls\_process\_data-batchno.  
      cs\_zpast2-quantity     = ls\_process\_data-qty.  
      cs\_zpast2-uom          = ls\_process\_data-uom.  
  
\*-------------------------------------------------------------------------------------  
\*Get packing material for the warehouse  
\*-------------------------------------------------------------------------------------  
      IF gt\_process\_data IS NOT INITIAL.  
        DATA(lv\_warehouseno) = gt\_process\_data[ 1 ]-/scwm/whno.  
      ENDIF.  
  
      IF lv\_warehouseno IS NOT INITIAL.  
        SELECT SINGLE packing\_material  
        FROM zewm\_t\_mat\_conf  
        WHERE warehouse\_no EQ @lv\_warehouseno  
        INTO @DATA(lv\_packing\_material).  
        IF sy-subrc EQ 0.  
          cs\_zpast2-packing\_material = lv\_packing\_material.  
        ENDIF.  
      ENDIF.  
\*-------------------------------------------------------------------------------------  
  
      /scwm/cl\_rf\_bll\_srvc=>set\_lgnum( iv\_lgnum = lv\_warehouseno ).  
      /scwm/cl\_rf\_bll\_srvc=>set\_prmod( EXPORTING iv\_prmod = '1' ).  
      /scwm/cl\_rf\_bll\_srvc=>set\_fcode( EXPORTING iv\_fcode = gc\_scr\_fcode\_zpafc2 ).  
  
    ENDIF.  
  ENDIF.  
  
ENDFUNCTION.

-------------------------------------------------------------------------------------------------------------------------------------------

-------------------------------------------------------------------------------------------------------------------------------------------

FUNCTION zewm\_rf\_zpast1\_inb\_dl\_pai.  
\*"----------------------------------------------------------------------  
\*"\*"Local Interface:  
\*"  CHANGING  
\*"     REFERENCE(CS\_ZPAIND) TYPE  ZEWM\_S\_ZPAIND  
\*"     REFERENCE(CS\_ZPAST1) TYPE  ZEWM\_S\_ZPAST1  
\*"     REFERENCE(CS\_ZPAST2) TYPE  ZEWM\_S\_ZPAST2  
\*"----------------------------------------------------------------------  
  
  DATA:lo\_packing TYPE REF TO /scwm/cl\_wm\_packing.  
  
  DATA:lv\_ibd\_no TYPE /scdl/dl\_docno\_int.  
  
  DATA:ls\_huhdr TYPE /scwm/s\_huhdr\_int.  
  
  DATA:lt\_huitm TYPE /scwm/tt\_huitm\_int,  
       lt\_huref TYPE /scwm/tt\_huref\_int.  
  
  DATA(lv\_fcode) = /scwm/cl\_rf\_bll\_srvc=>get\_fcode( ).  
  
  IF lv\_fcode EQ 'NEXT'.  
  
\*-------------------------------------------------------------------------------------  
\*Get data for processing against a PO  
\*-------------------------------------------------------------------------------------  
    IF cs\_zpast1-is\_po EQ abap\_true.  
  
\*Validate if multiple inbound deliveries exists for a PO  
      SELECT DISTINCT COUNT( docno )  
      FROM /scdl/db\_proci\_i  
      WHERE refdocno\_po EQ @cs\_zpast1-po\_asn\_inbdl  
      INTO @DATA(lv\_total\_inb\_delv).  
      IF sy-subrc EQ 0 AND  
         lv\_total\_inb\_delv GT 1.  
        MESSAGE e008(zewm01).  
      ENDIF.  
  
      SELECT docid,  
             docno,  
             itemid,  
             itemno,  
             productno,  
             productid,  
             batchno,  
             qty,  
             uom,  
             refdocno\_po,  
             /scwm/whno  
      FROM /scdl/db\_proci\_i  
      WHERE refdocno\_po EQ @cs\_zpast1-po\_asn\_inbdl  
      INTO CORRESPONDING FIELDS OF TABLE @gt\_process\_data.  
      IF sy-subrc NE 0.  
        MESSAGE e012(zewm01) WITH cs\_zpast1-po\_asn\_inbdl.  
      ENDIF.  
    ENDIF.  
\*-------------------------------------------------------------------------------------  
  
\*-------------------------------------------------------------------------------------  
\*Get data for processing against a ASN  
\*-------------------------------------------------------------------------------------  
    IF cs\_zpast1-is\_asn EQ abap\_true.  
  
\*Validate if multiple inbound deliveries exists for a ASN  
      SELECT DISTINCT COUNT( docno )  
      FROM /scdl/db\_proch\_i  
      WHERE refdocno\_extasn EQ @cs\_zpast1-po\_asn\_inbdl  
      INTO @lv\_total\_inb\_delv.  
      IF sy-subrc EQ 0 AND  
         lv\_total\_inb\_delv GT 1.  
        MESSAGE e009(zewm01).  
      ENDIF.  
  
      SELECT a~docid,  
             a~docno,  
             b~itemid,  
             b~itemno,  
             b~productno,  
             b~productid,  
             b~batchno,  
             b~qty,  
             b~uom,  
             a~refdocno\_extasn,  
             b~/scwm/whno  
       FROM /scdl/db\_proch\_i AS a INNER JOIN /scdl/db\_proci\_i AS b ON b~docid EQ a~docid AND  
                                                                      b~docno EQ a~docno  
       WHERE a~refdocno\_extasn EQ @cs\_zpast1-po\_asn\_inbdl  
       INTO CORRESPONDING FIELDS OF TABLE @gt\_process\_data.  
      IF sy-subrc NE 0.  
        MESSAGE e012(zewm01) WITH cs\_zpast1-po\_asn\_inbdl.  
      ENDIF.  
  
    ENDIF.  
\*-------------------------------------------------------------------------------------  
  
\*-------------------------------------------------------------------------------------  
\*Get data for processing against a Inbound Delivery  
\*-------------------------------------------------------------------------------------  
    IF cs\_zpast1-is\_inb\_dl EQ abap\_true.  
  
      SELECT docid,  
             docno,  
             itemid,  
             itemno,  
             productno,  
             productid,  
             batchno,  
             qty,  
             uom,  
             /scwm/whno  
      FROM /scdl/db\_proci\_i  
      WHERE docno EQ @cs\_zpast1-po\_asn\_inbdl  
      INTO CORRESPONDING FIELDS OF TABLE @gt\_process\_data.  
      IF sy-subrc NE 0.  
        MESSAGE e011(zewm01) WITH cs\_zpast1-po\_asn\_inbdl.  
      ENDIF.  
  
    ENDIF.  
\*-------------------------------------------------------------------------------------  
  
\*-------------------------------------------------------------------------------------  
\*Exclude the line items for which the GR is already done  
\*-------------------------------------------------------------------------------------  
    IF gt\_process\_data IS NOT INITIAL.  
      SELECT docid,  
             itemid,  
             status\_type,  
             status\_value  
      FROM /scdl/db\_status  
      FOR ALL ENTRIES IN @gt\_process\_data  
      WHERE docid        EQ @gt\_process\_data-docid  AND  
            itemid       EQ @gt\_process\_data-itemid AND  
            status\_type  EQ 'DGR'                  AND  
            status\_value EQ '9'  
      INTO TABLE @DATA(lt\_status).  
      IF sy-subrc EQ 0.  
        LOOP AT lt\_status ASSIGNING FIELD-SYMBOL(<lfs\_status>).  
          DELETE gt\_process\_data WHERE docid  EQ <lfs\_status>-docid AND  
                                       itemid EQ <lfs\_status>-itemid.  
        ENDLOOP.  
      ENDIF.  
    ENDIF.  
  
\*-------------------------------------------------------------------------------------  
\*Reduce the quantity already packed from quantity to be packed.  
\*-------------------------------------------------------------------------------------  
    IF lo\_packing IS NOT BOUND.  
      lo\_packing = NEW #( ).  
    ENDIF.  
  
    IF gt\_process\_data IS NOT INITIAL.  
      SELECT a~docid,  
             a~itemid,  
             b~guid\_parent,  
             b~qdocid,  
             b~qitmid,  
             c~guid\_hu,  
             c~huident  
      FROM /scdl/db\_proci\_i AS a INNER JOIN /scwm/quan AS b ON b~qdocid  EQ a~docid  AND  
                                                               b~qitmid  EQ a~itemid AND  
                                                               b~qdoccat EQ 'PDI'  
                                 INNER JOIN /scwm/huhdr AS c ON c~guid\_hu EQ b~guid\_parent  
      FOR ALL ENTRIES IN @gt\_process\_data  
      WHERE a~docno  EQ @gt\_process\_data-docno  AND  
            a~itemno EQ @gt\_process\_data-itemno  
      INTO TABLE @DATA(lt\_hu\_data).  
      IF sy-subrc EQ 0.  
        SORT lt\_hu\_data BY docid itemid guid\_hu huident.  
  
        LOOP AT lt\_hu\_data ASSIGNING FIELD-SYMBOL(<lfs\_hu\_data>).  
  
          CLEAR:ls\_huhdr,  
                lt\_huitm,  
                lt\_huref.  
  
          CALL METHOD lo\_packing->get\_hu  
            EXPORTING  
              iv\_huident = <lfs\_hu\_data>-huident  
            IMPORTING  
              es\_huhdr   = ls\_huhdr  
              et\_huitm   = lt\_huitm  
              et\_huref   = lt\_huref  
            EXCEPTIONS  
              not\_found  = 1  
              OTHERS     = 2.  
          IF sy-subrc <> 0.  
            MESSAGE ID sy-msgid TYPE sy-msgty NUMBER sy-msgno WITH sy-msgv1 sy-msgv2 sy-msgv3 sy-msgv4.  
          ENDIF.  
  
          LOOP AT lt\_huitm ASSIGNING FIELD-SYMBOL(<lfs\_huitm>).  
            READ TABLE gt\_process\_data ASSIGNING FIELD-SYMBOL(<lfs\_process\_data>)  
                WITH KEY docid  = <lfs\_hu\_data>-docid  
                         itemid = <lfs\_hu\_data>-itemid.  
            IF sy-subrc EQ 0.  
              <lfs\_process\_data>-qty = <lfs\_process\_data>-qty - <lfs\_huitm>-quan.  
            ENDIF.  
          ENDLOOP.  
  
        ENDLOOP.  
      ENDIF.  
    ENDIF.  
  
\*-------------------------------------------------------------------------------------  
\*Map data to the screen structure  
\*-------------------------------------------------------------------------------------  
    SORT gt\_process\_data BY itemno.  
    DATA(ls\_process\_data) = gt\_process\_data[ 1 ].  
  
    CLEAR lv\_ibd\_no.  
    lv\_ibd\_no = ls\_process\_data-docno.  
    CALL FUNCTION 'CONVERSION\_EXIT\_ALPHA\_OUTPUT'  
      EXPORTING  
        input  = lv\_ibd\_no  
      IMPORTING  
        output = lv\_ibd\_no.  
  
    cs\_zpast2-po\_asn\_inbdl = lv\_ibd\_no.  
    cs\_zpast2-line\_item    = ls\_process\_data-itemno.  
    cs\_zpast2-material     = ls\_process\_data-productno.  
    cs\_zpast2-batch        = ls\_process\_data-batchno.  
    cs\_zpast2-quantity     = ls\_process\_data-qty.  
    cs\_zpast2-uom          = ls\_process\_data-uom.  
  
\*-------------------------------------------------------------------------------------  
\*Get packing material for the warehouse  
\*-------------------------------------------------------------------------------------  
  
    DATA(lv\_warehouseno) = /scwm/cl\_rf\_bll\_srvc=>get\_lgnum( ).  
  
    IF lv\_warehouseno IS NOT INITIAL.  
      SELECT SINGLE packing\_material  
      FROM zewm\_t\_mat\_conf  
      WHERE warehouse\_no EQ @lv\_warehouseno  
      INTO @DATA(lv\_packing\_material).  
      IF sy-subrc EQ 0.  
        cs\_zpast2-packing\_material = lv\_packing\_material.  
      ENDIF.  
    ENDIF.  
\*-------------------------------------------------------------------------------------  
  
  ENDIF.  
  
ENDFUNCTION.

-------------------------------------------------------------------------------------------------------------------------------------------

-------------------------------------------------------------------------------------------------------------------------------------------

FUNCTION zewm\_rf\_zpast2\_inb\_dl\_pai.  
\*"----------------------------------------------------------------------  
\*"\*"Local Interface:  
\*"  CHANGING  
\*"     REFERENCE(CS\_ZPAIND) TYPE  ZEWM\_S\_ZPAIND  
\*"     REFERENCE(CS\_ZPAST1) TYPE  ZEWM\_S\_ZPAST1  
\*"     REFERENCE(CS\_ZPAST2) TYPE  ZEWM\_S\_ZPAST2  
\*"----------------------------------------------------------------------  
  
  DATA(lv\_fcode) = /scwm/cl\_rf\_bll\_srvc=>get\_fcode( ).  
  
  CLEAR cs\_zpast2-gr\_status.  
  
  CASE lv\_fcode.  
    WHEN 'ZCHANU'.  
  
      IF cs\_zpast2-quantity IS INITIAL.  
        MESSAGE e010(zewm01).  
      ENDIF.  
  
      CALL FUNCTION 'ZEWM\_RF\_CREATE\_AND\_PACK\_HU'  
        CHANGING  
          cs\_zpast2 = cs\_zpast2.  
  
    WHEN 'ZCRGR'.  
  
      CALL FUNCTION 'ZEWM\_RF\_POST\_GOODS\_RECEIPT'  
        CHANGING  
          cs\_zpast2 = cs\_zpast2.  
  
      CALL FUNCTION 'ZEWM\_RF\_CRT\_AND\_CONF\_TASK'  
        CHANGING  
          cs\_zpast2 = cs\_zpast2.  
  
      cs\_zpast2-gr\_status = abap\_true.  
  
    WHEN 'ZNXTLN'.  
      IF gt\_process\_data IS NOT INITIAL.  
        DATA(lv\_index) = line\_index( gt\_process\_data[ itemno = cs\_zpast2-line\_item ] ).  
  
        IF line\_exists( gt\_process\_data[ lv\_index + 1 ] ).  
          DATA(ls\_process\_data) = gt\_process\_data[ lv\_index + 1 ].  
          CLEAR cs\_zpast2-hu\_pack\_counter.  
        ELSE.  
          ls\_process\_data = gt\_process\_data[ lv\_index ].  
        ENDIF.  
  
        cs\_zpast2-line\_item = ls\_process\_data-itemno.  
        cs\_zpast2-material  = ls\_process\_data-productno.  
        cs\_zpast2-batch     = ls\_process\_data-batchno.  
        cs\_zpast2-quantity  = ls\_process\_data-qty - ls\_process\_data-qty\_packed.  
        cs\_zpast2-uom       = ls\_process\_data-uom.  
  
      ENDIF.  
    WHEN 'ZPRVLN'.  
      IF gt\_process\_data IS NOT INITIAL.  
        lv\_index = line\_index( gt\_process\_data[ itemno = cs\_zpast2-line\_item ] ).  
  
        IF line\_exists( gt\_process\_data[ lv\_index - 1 ] ).  
          ls\_process\_data = gt\_process\_data[ lv\_index - 1 ].  
        ELSE.  
          ls\_process\_data = gt\_process\_data[ lv\_index ].  
        ENDIF.  
  
        cs\_zpast2-line\_item = ls\_process\_data-itemno.  
        cs\_zpast2-material  = ls\_process\_data-productno.  
        cs\_zpast2-batch     = ls\_process\_data-batchno.  
        cs\_zpast2-quantity  = ls\_process\_data-qty - ls\_process\_data-qty\_packed.  
        cs\_zpast2-uom       = ls\_process\_data-uom.  
      ENDIF.  
  
    WHEN 'ZRESET'.  
      CLEAR cs\_zpast2-hu\_id.  
  
    WHEN OTHERS.  
  ENDCASE.  
  
ENDFUNCTION.

-------------------------------------------------------------------------------------------------------------------------------------------

-------------------------------------------------------------------------------------------------------------------------------------------

CALL FUNCTION 'ZEWM\_RF\_CREATE\_AND\_PACK\_HU'  
        CHANGING  
          cs\_zpast2 = cs\_zpast2.

FUNCTION zewm\_rf\_create\_and\_pack\_hu.  
\*"----------------------------------------------------------------------  
\*"\*"Local Interface:  
\*"  CHANGING  
\*"     REFERENCE(CS\_ZPAST2) TYPE  ZEWM\_S\_ZPAST2  
\*"----------------------------------------------------------------------  
  
  DATA:lv\_inb\_dl     TYPE /scdl/dl\_docno\_int,  
       lv\_rejected   TYPE boole\_d,  
       lv\_severity   TYPE bapi\_mtype,  
       lv\_matid\_guid TYPE /scwm/de\_matid.  
  
  DATA:lo\_pack     TYPE REF TO /scwm/cl\_dlv\_pack\_ibdl,  
       lo\_message  TYPE REF TO /scdl/cl\_dm\_message,  
       lo\_dlv\_save TYPE REF TO /scwm/cl\_dlv\_management\_prd,  
       lo\_packing  TYPE REF TO /scwm/cl\_wm\_packing.  
  
  DATA:ls\_levels         TYPE /scwm/s\_ps\_level\_int,  
       ls\_hu\_create      TYPE /scwm/s\_huhdr\_create\_ext,  
       ls\_huhdr          TYPE /scwm/s\_huhdr\_int,  
       ls\_psp\_hdr        TYPE /scwm/s\_ps\_header\_int,  
       ls\_packspec\_level TYPE /scwm/s\_ps\_level\_int,  
       ls\_psp\_content    TYPE /scwm/s\_packspec\_nested,  
       ls\_material       TYPE /scwm/s\_pack\_stock,  
       ls\_quantity       TYPE /scwm/s\_quan,  
       ls\_hu\_ident       TYPE /scwm/hu\_ident.  
  
  DATA:lt\_hu           TYPE /scwm/t\_gm\_hu,  
       lt\_docid        TYPE /scwm/tt\_docid,  
       lt\_psp\_content  TYPE /scwm/tt\_packspec\_nested,  
       lt\_elementgroup TYPE /scwm/tt\_ps\_elementgroup,  
       lt\_message      TYPE /scdl/dm\_message\_tab,  
       lt\_crea\_hu      TYPE /scwm/tt\_to\_prep\_hu\_int.  
  
  DATA(ls\_process) = gt\_process\_data[ itemno = cs\_zpast2-line\_item ].  
  
\*------------------------------------------------------------------------------------------------------  
\*Fill packaging specification relevant info in case it is present for material.  
\*------------------------------------------------------------------------------------------------------  
  IF ls\_process-productid IS NOT INITIAL.  
    CALL FUNCTION '/SCWM/PS\_PACKSPEC\_GET'  
      EXPORTING  
        iv\_guid\_ps          = ls\_process-productid  
        iv\_read\_elements    = 'X'  
      IMPORTING  
        es\_packspec\_header  = ls\_psp\_hdr  
        et\_packspec\_content = lt\_psp\_content  
        et\_elementgroup     = lt\_elementgroup  
        es\_packspec\_level   = ls\_packspec\_level  
      EXCEPTIONS  
        error               = 1  
        OTHERS              = 2.  
    IF sy-subrc <> 0.  
    ENDIF.  
  
    IF lt\_psp\_content IS NOT INITIAL.  
      ls\_psp\_content = lt\_psp\_content[ 1 ].  
  
      IF line\_exists( ls\_psp\_content-levels[ hu\_create = abap\_true ] ).  
        ls\_levels = ls\_psp\_content-levels[ hu\_create = abap\_true ].  
  
        ls\_hu\_create-tare\_var       = ls\_levels-tare\_var.  
        ls\_hu\_create-closed\_package = ls\_levels-closed\_package.  
  
        IF ls\_levels-flag\_weight EQ abap\_true.  
          ls\_hu\_create-unit\_gw  = ls\_levels-unit\_gw.  
          ls\_hu\_create-t\_weight = ls\_levels-t\_weight.  
          ls\_hu\_create-unit\_tw  = ls\_levels-unit\_tw.  
        ENDIF.  
  
        IF ls\_levels-flag\_vol EQ abap\_true.  
          ls\_hu\_create-unit\_gv  = ls\_levels-unit\_gv.  
          ls\_hu\_create-t\_volume = ls\_levels-t\_volume.  
          ls\_hu\_create-unit\_tv  = ls\_levels-unit\_tv.  
        ENDIF.  
  
        IF ls\_levels-flag\_dim EQ abap\_true.  
          ls\_hu\_create-length       = ls\_levels-length.  
          ls\_hu\_create-width        = ls\_levels-width.  
          ls\_hu\_create-height       = ls\_levels-height.  
          ls\_hu\_create-unit\_lwh     = ls\_levels-unit\_lwh.  
          ls\_hu\_create-max\_length   = ls\_levels-max\_length.  
          ls\_hu\_create-max\_width    = ls\_levels-max\_width.  
          ls\_hu\_create-max\_height   = ls\_levels-max\_height.  
          ls\_hu\_create-unit\_max\_lwh = ls\_levels-unit\_max\_lwh.  
        ENDIF.  
  
        IF ls\_levels-flag\_capa EQ abap\_true.  
          ls\_hu\_create-t\_capa   = ls\_levels-t\_capa.  
          ls\_hu\_create-max\_capa = ls\_levels-max\_capa.  
          ls\_hu\_create-tolc     = ls\_levels-tolc.  
        ENDIF.  
  
      ENDIF.  
    ENDIF.  
  
  ENDIF.  
\*------------------------------------------------------------------------------------------------------  
  
\*------------------------------------------------------------------------------------------------------  
\*Create Handling Unit  
\*------------------------------------------------------------------------------------------------------  
  IF lo\_pack IS NOT BOUND.  
    lo\_pack = NEW #( ).  
  ENDIF.  
  
\*Get packing material guid  
  IF cs\_zpast2-packing\_material IS NOT INITIAL.  
  
    CLEAR lv\_matid\_guid.  
    CALL FUNCTION '/SCWM/WC\_GET\_MATID'  
      EXPORTING  
        iv\_id    = cs\_zpast2-packing\_material  
      IMPORTING  
        ev\_matid = lv\_matid\_guid  
      EXCEPTIONS  
        error    = 1  
        OTHERS   = 2.  
    IF sy-subrc <> 0.  
      MESSAGE ID sy-msgid TYPE sy-msgty NUMBER sy-msgno WITH sy-msgv1 sy-msgv2 sy-msgv3 sy-msgv4.  
    ENDIF.  
  
\*Set warehouse number  
    lo\_pack->gv\_lgnum = /scwm/cl\_rf\_bll\_srvc=>get\_lgnum( ).  
  
\*Initialize By Delivery  
    lt\_docid = VALUE #( ( docid = ls\_process-docid ) ).  
  
    lo\_pack->init(  
      EXPORTING  
        iv\_lgnum = /scwm/cl\_rf\_bll\_srvc=>get\_lgnum( )  
        it\_docid = lt\_docid ).  
  
    ls\_hu\_create-hutyp = 'E1'.  
  
    CALL METHOD lo\_pack->/scwm/if\_pack\_bas~create\_hu  
      EXPORTING  
        iv\_pmat      = lv\_matid\_guid  
        is\_hu\_create = ls\_hu\_create  
      RECEIVING  
        es\_huhdr     = ls\_huhdr  
      EXCEPTIONS  
        error        = 1  
        OTHERS       = 2.  
    IF sy-subrc <> 0.  
      MESSAGE ID sy-msgid TYPE sy-msgty NUMBER sy-msgno WITH sy-msgv1 sy-msgv2 sy-msgv3 sy-msgv4.  
    ENDIF.  
  
\*Update supplier provided HU  
    ls\_hu\_ident = VALUE #( mandt   = sy-mandt  
                           guid\_hu = ls\_huhdr-guid\_hu  
                           idart   = 'E'  
                           huident = cs\_zpast2-hu\_id  ).  
  
    INSERT INTO /scwm/hu\_ident VALUES ls\_hu\_ident.  
  
\*Read created handling unit to populate its GUID in buffer  
    CALL FUNCTION '/SCWM/HUHEADER\_READ'  
      EXPORTING  
        iv\_guid\_hu   = ls\_huhdr-guid\_hu  
        iv\_huident   = ls\_huhdr-huident  
        iv\_db\_select = 'A'  
        iv\_lock      = 'X'  
      EXCEPTIONS  
        not\_found    = 1  
        input        = 2  
        error        = 3  
        deleted      = 4  
        OTHERS       = 5.  
    IF sy-subrc <> 0.  
      MESSAGE ID sy-msgid TYPE sy-msgty NUMBER sy-msgno WITH sy-msgv1 sy-msgv2 sy-msgv3 sy-msgv4.  
    ENDIF.  
  
\*------------------------------------------------------------------------------------------------------  
  
\*------------------------------------------------------------------------------------------------------  
\*Pack the stock to the empty HU  
\*------------------------------------------------------------------------------------------------------  
    ls\_material-qdoccat = 'PDI'.  
    ls\_material-qdocid  = ls\_process-docid.  
    ls\_material-qitmid  = ls\_process-itemid.  
    ls\_quantity-quan    = cs\_zpast2-quantity.  
    ls\_quantity-unit    = cs\_zpast2-uom.  
  
    CALL METHOD lo\_pack->/scwm/if\_pack\_bas~pack\_stock  
      EXPORTING  
        iv\_dest\_hu  = ls\_huhdr-guid\_hu  
        is\_material = ls\_material  
        is\_quantity = ls\_quantity  
      EXCEPTIONS  
        error       = 1  
        OTHERS      = 2.  
    IF sy-subrc <> 0.  
      MESSAGE ID sy-msgid TYPE sy-msgty NUMBER sy-msgno WITH sy-msgv1 sy-msgv2 sy-msgv3 sy-msgv4.  
    ENDIF.  
  
\*------------------------------------------------------------------------------------------------------  
\*Reduce the packed quantity from the quantity to be packed  
\*------------------------------------------------------------------------------------------------------  
    READ TABLE gt\_process\_data ASSIGNING FIELD-SYMBOL(<lfs\_process\_data>)  
    WITH KEY itemno = cs\_zpast2-line\_item.  
    IF <lfs\_process\_data> IS ASSIGNED.  
      <lfs\_process\_data>-qty\_packed = <lfs\_process\_data>-qty\_packed + cs\_zpast2-quantity.  
      cs\_zpast2-quantity = <lfs\_process\_data>-qty - <lfs\_process\_data>-qty\_packed.  
    ENDIF.  
  
\*Update HU packing counter  
    cs\_zpast2-hu\_pack\_counter = cs\_zpast2-hu\_pack\_counter + 1.  
  
    CLEAR cs\_zpast2-hu\_id.  
  
    lo\_pack->save(  
      EXPORTING  
        iv\_commit = 'X'  
        iv\_wait   = 'X'  
      EXCEPTIONS  
        error     = 1  
        OTHERS    = 2  
    ).  
    IF sy-subrc <> 0.  
      MESSAGE ID sy-msgid TYPE sy-msgty NUMBER sy-msgno WITH sy-msgv1 sy-msgv2 sy-msgv3 sy-msgv4.  
    ENDIF.  
  
  ENDIF.  
\*------------------------------------------------------------------------------------------------------  
  
ENDFUNCTION.

-------------------------------------------------------------------------------------------------------------------------------------------

-------------------------------------------------------------------------------------------------------------------------------------------

CALL FUNCTION 'ZEWM\_RF\_POST\_GOODS\_RECEIPT'  
        CHANGING  
          cs\_zpast2 = cs\_zpast2.

FUNCTION zewm\_rf\_post\_goods\_receipt.  
\*"----------------------------------------------------------------------  
\*"\*"Local Interface:  
\*"  CHANGING  
\*"     REFERENCE(CS\_ZPAST2) TYPE  ZEWM\_S\_ZPAST2  
\*"----------------------------------------------------------------------  
  
  DATA:lv\_inb\_dl   TYPE /scdl/dl\_docno\_int,  
       lv\_rejected TYPE boole\_d.  
  
  DATA: ls\_huhdr TYPE /scwm/s\_huhdr\_int.  
  
  DATA:lt\_hu      TYPE /scwm/t\_gm\_hu,  
       lt\_message TYPE /scdl/dm\_message\_tab,  
       lt\_huitm   TYPE /scwm/tt\_huitm\_int,  
       lt\_huref   TYPE /scwm/tt\_huref\_int,  
       lt\_k\_item  TYPE  /scdl/t\_sp\_k\_item,  
       lt\_docid   TYPE /scwm/tt\_docid,  
       lt\_rcode   TYPE  /scdl/t\_sp\_return\_code.  
  
  DATA:lo\_message  TYPE REF TO /scdl/cl\_dm\_message,  
       lo\_pack     TYPE REF TO /scwm/cl\_dlv\_pack\_ibdl,  
       lo\_packing  TYPE REF TO /scwm/cl\_wm\_packing,  
       lo\_gm       TYPE REF TO /scwm/cl\_goods\_movement,  
       lo\_dlv      TYPE REF TO /scdl/cl\_sp\_prd\_inb,  
       lo\_dlv\_save TYPE REF TO /scwm/cl\_dlv\_management\_prd.  
  
\*------------------------------------------------------------------------------------------------------  
\*Post goods receipt  
\*------------------------------------------------------------------------------------------------------  
  
  IF lo\_packing IS NOT BOUND.  
    lo\_packing = NEW #( ).  
  ENDIF.  
  
  IF lo\_gm IS NOT BOUND.  
    lo\_gm = NEW #( ).  
  ENDIF.  
  
  IF lo\_message IS NOT BOUND.  
    lo\_message = NEW #( ).  
  ENDIF.  
  
  IF lo\_dlv IS NOT BOUND.  
    lo\_dlv = NEW #( ).  
  ENDIF.  
  
  IF lo\_pack IS NOT BOUND.  
    lo\_pack = NEW #( ).  
  ENDIF.  
  
  DATA(ls\_process) = gt\_process\_data[ itemno = cs\_zpast2-line\_item ].  
  
\*Set warehouse number  
  lo\_pack->gv\_lgnum = /scwm/cl\_rf\_bll\_srvc=>get\_lgnum( ).  
  
\*Initialize By Delivery  
  lt\_docid = VALUE #( ( docid = ls\_process-docid ) ).  
  
  lo\_pack->init(  
    EXPORTING  
      iv\_lgnum = /scwm/cl\_rf\_bll\_srvc=>get\_lgnum( )  
      it\_docid = lt\_docid ).  
  
\*Get Handling unit for the current inbound delivery line item  
  lv\_inb\_dl = cs\_zpast2-po\_asn\_inbdl.  
  CALL FUNCTION 'CONVERSION\_EXIT\_ALPHA\_INPUT'  
    EXPORTING  
      input  = lv\_inb\_dl  
    IMPORTING  
      output = lv\_inb\_dl.  
  
  SELECT a~docid,  
         a~itemid,  
         b~guid\_parent,  
         b~qdocid,  
         b~qitmid,  
         c~guid\_hu,  
         c~huident  
  FROM /scdl/db\_proci\_i AS a INNER JOIN /scwm/quan AS b ON b~qdocid  EQ a~docid  AND  
                                                           b~qitmid  EQ a~itemid AND  
                                                           b~qdoccat EQ 'PDI'  
                             INNER JOIN /scwm/huhdr AS c ON c~guid\_hu EQ b~guid\_parent  
  WHERE a~docno  EQ @lv\_inb\_dl AND  
        a~itemno EQ @cs\_zpast2-line\_item  
  INTO TABLE @DATA(lt\_hu\_data).  
  IF sy-subrc EQ 0.  
  
    LOOP AT lt\_hu\_data ASSIGNING FIELD-SYMBOL(<lfs\_hu\_data>).  
  
      CLEAR:ls\_huhdr,  
            lt\_huitm,  
            lt\_huref.  
  
      CALL METHOD lo\_packing->get\_hu  
        EXPORTING  
          iv\_huident = <lfs\_hu\_data>-huident  
        IMPORTING  
          es\_huhdr   = ls\_huhdr  
          et\_huitm   = lt\_huitm  
          et\_huref   = lt\_huref  
        EXCEPTIONS  
          not\_found  = 1  
          OTHERS     = 2.  
      IF sy-subrc <> 0.  
        MESSAGE ID sy-msgid TYPE sy-msgty NUMBER sy-msgno WITH sy-msgv1 sy-msgv2 sy-msgv3 sy-msgv4.  
      ENDIF.  
  
      LOOP AT lt\_huitm ASSIGNING FIELD-SYMBOL(<lfs\_itm>).  
        APPEND INITIAL LINE TO lt\_k\_item ASSIGNING FIELD-SYMBOL(<lfs\_k\_item>).  
        IF <lfs\_k\_item> IS ASSIGNED.  
          <lfs\_k\_item>-docid  = <lfs\_hu\_data>-docid.  
          <lfs\_k\_item>-itemid = <lfs\_itm>-qitmid.  
        ENDIF.  
      ENDLOOP.  
  
      APPEND INITIAL LINE TO lt\_hu ASSIGNING FIELD-SYMBOL(<lfs\_hu>).  
      IF <lfs\_hu> IS ASSIGNED.  
        <lfs\_hu>-lgnum   = /scwm/cl\_rf\_bll\_srvc=>get\_lgnum( ).  
        <lfs\_hu>-guid\_hu = <lfs\_hu\_data>-guid\_hu.  
        <lfs\_hu>-huident = <lfs\_hu\_data>-huident.  
      ENDIF.  
  
    ENDLOOP.  
  
\*Lock inbound delivery for GR posting  
    lo\_dlv->lock(  
      EXPORTING  
        inkeys       = lt\_k\_item  
        lockmode     = /scdl/if\_sp1\_locking=>sc\_exclusive\_lock  
        aspect       = /scdl/if\_sp\_c=>sc\_asp\_item  
      IMPORTING  
        rejected     = lv\_rejected  
        return\_codes = lt\_rcode ).  
  
\*Post GR  
    CALL METHOD /scwm/cl\_goods\_movement=>post\_hu  
      EXPORTING  
        it\_hu      = lt\_hu  
        iv\_gmcat   = /scwm/if\_docflow\_c=>sc\_gr  
      IMPORTING  
        eo\_message = lo\_message.  
  
    IF lo\_message IS BOUND.  
      lt\_message = lo\_message->get\_messages( iv\_msgty = 'E' ).  
  
      IF line\_exists( lt\_message[ msgty = 'E' ] ).  
        DATA(ls\_message) = lt\_message[ msgty = 'E' ].  
        MESSAGE ID ls\_message-msgid TYPE ls\_message-msgty NUMBER ls\_message-msgno  
        WITH ls\_message-msgv1 ls\_message-msgv2 ls\_message-msgv3 ls\_message-msgv4.  
      ENDIF.  
  
    ENDIF.  
  
\*------------------------------------------------------------------------------------------------------  
\*Save inbound delivery  
\*------------------------------------------------------------------------------------------------------  
    lo\_dlv\_save = /scwm/cl\_dlv\_management\_prd=>get\_instance( ).  
    lo\_dlv\_save->save(  
      IMPORTING  
        ev\_rejected = lv\_rejected  
        et\_message  = lt\_message ).  
  
    IF line\_exists( lt\_message[ msgty = 'E' ] ).  
      ls\_message = lt\_message[ msgty = 'E' ].  
      MESSAGE ID ls\_message-msgid TYPE ls\_message-msgty NUMBER ls\_message-msgno  
      WITH ls\_message-msgv1 ls\_message-msgv2 ls\_message-msgv3 ls\_message-msgv4.  
    ENDIF.  
  
    COMMIT WORK AND WAIT.  
  ENDIF.  
  
ENDFUNCTION.

-------------------------------------------------------------------------------------------------------------------------------------------

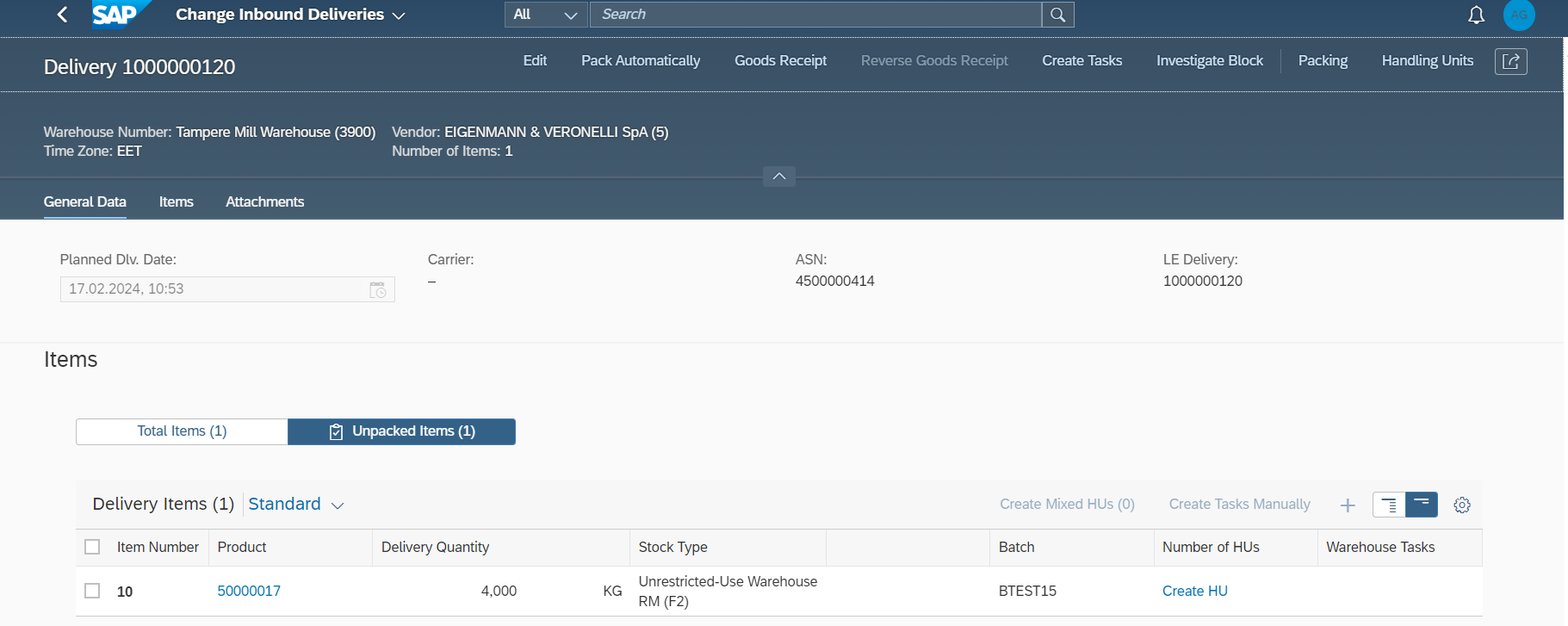
-------------------------------------------------------------------------------------------------------------------------------------------

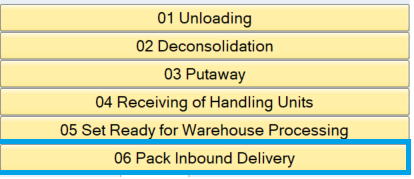
CALL FUNCTION 'ZEWM\_RF\_CRT\_AND\_CONF\_TASK'  
        CHANGING  
          cs\_zpast2 = cs\_zpast2.

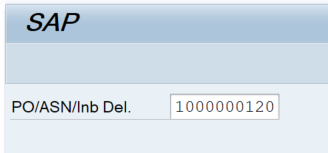
FUNCTION zewm\_rf\_crt\_and\_conf\_task.  
\*"----------------------------------------------------------------------  
\*"\*"Local Interface:  
\*"  CHANGING  
\*"     REFERENCE(CS\_ZPAST2) TYPE  ZEWM\_S\_ZPAST2  
\*"----------------------------------------------------------------------  
  TYPES:BEGIN OF ty\_tanum,  
          tanum TYPE /scwm/tanum,  
        END OF ty\_tanum.  
  
  DATA:lv\_inb\_dl   TYPE /scdl/dl\_docno\_int,  
       lv\_tanum    TYPE  /scwm/tanum,  
       lv\_severity TYPE bapi\_mtype.  
  
  DATA:lo\_pack    TYPE REF TO /scwm/cl\_dlv\_pack\_ibdl,  
       lo\_packing TYPE REF TO /scwm/cl\_wm\_packing.  
  
  DATA:ls\_huhdr TYPE /scwm/s\_huhdr\_int.  
  
  DATA:lt\_docid   TYPE /scwm/tt\_docid,  
       lt\_crea\_hu TYPE /scwm/tt\_to\_prep\_hu\_int,  
       lt\_tanum   TYPE STANDARD TABLE OF ty\_tanum,  
       lt\_ltap    TYPE /scwm/tt\_ltap\_vb,  
       lt\_bapiret TYPE bapirettab,  
       lt\_conf    TYPE /scwm/to\_conf\_tt,  
       lt\_return  TYPE bapirettab.  
  
\*------------------------------------------------------------------------------------------------------  
\*Create warehouse task  
\*------------------------------------------------------------------------------------------------------  
  
  DATA(ls\_process) = gt\_process\_data[ itemno = cs\_zpast2-line\_item ].  
  
  DATA(lv\_lgnum) = /scwm/cl\_rf\_bll\_srvc=>get\_lgnum( ).  
  
  IF lo\_pack IS NOT BOUND.  
    lo\_pack = NEW #( ).  
  ENDIF.  
  
  IF lo\_packing IS NOT BOUND.  
    lo\_packing = NEW #( ).  
  ENDIF.  
  
\*Set warehouse number  
  lo\_pack->gv\_lgnum = lv\_lgnum.  
  
\*Initialize By Delivery  
  lt\_docid = VALUE #( ( docid = ls\_process-docid ) ).  
  
  lo\_pack->init(  
    EXPORTING  
      iv\_lgnum = lv\_lgnum  
      it\_docid = lt\_docid ).  
  
\*Get Handling unit for the current inbound delivery line item  
  lv\_inb\_dl = cs\_zpast2-po\_asn\_inbdl.  
  CALL FUNCTION 'CONVERSION\_EXIT\_ALPHA\_INPUT'  
    EXPORTING  
      input  = lv\_inb\_dl  
    IMPORTING  
      output = lv\_inb\_dl.  
  
  SELECT a~docid,  
         a~itemid,  
         b~guid\_parent,  
         b~qdocid,  
         b~qitmid,  
         c~guid\_hu,  
         c~huident  
  FROM /scdl/db\_proci\_i AS a INNER JOIN /scwm/quan AS b ON b~qdocid  EQ a~docid  AND  
                                                           b~qitmid  EQ a~itemid AND  
                                                           b~qdoccat EQ 'PDI'  
                             INNER JOIN /scwm/huhdr AS c ON c~guid\_hu EQ b~guid\_parent  
  WHERE a~docno  EQ @lv\_inb\_dl AND  
        a~itemno EQ @cs\_zpast2-line\_item  
  INTO TABLE @DATA(lt\_hu\_data).  
  IF sy-subrc EQ 0.  
  
    LOOP AT lt\_hu\_data ASSIGNING FIELD-SYMBOL(<lfs\_hu\_data>).  
  
      CLEAR ls\_huhdr.  
      CALL METHOD lo\_packing->get\_hu  
        EXPORTING  
          iv\_huident = <lfs\_hu\_data>-huident  
        IMPORTING  
          es\_huhdr   = ls\_huhdr  
        EXCEPTIONS  
          not\_found  = 1  
          OTHERS     = 2.  
      IF sy-subrc <> 0.  
        MESSAGE ID sy-msgid TYPE sy-msgty NUMBER sy-msgno WITH sy-msgv1 sy-msgv2 sy-msgv3 sy-msgv4.  
      ENDIF.  
  
      CLEAR lt\_crea\_hu.  
      APPEND INITIAL LINE TO lt\_crea\_hu ASSIGNING FIELD-SYMBOL(<lfs\_crea\_hu>).  
      IF <lfs\_crea\_hu> IS ASSIGNED.  
        <lfs\_crea\_hu>-guid\_hu = <lfs\_hu\_data>-guid\_hu.  
        <lfs\_crea\_hu>-procty  = ls\_huhdr-procty.  
      ENDIF.  
  
      CLEAR:lv\_tanum,lv\_severity,lt\_ltap,lt\_bapiret.  
      CALL FUNCTION '/SCWM/TO\_CREATE\_WHR'  
        EXPORTING  
          iv\_lgnum       = lv\_lgnum  
          iv\_process     = wmegc\_whr\_proc\_pu  
          iv\_mve\_hu      = 'X'  
          it\_crea\_hu     = lt\_crea\_hu  
          iv\_commit\_work = ' '  
        IMPORTING  
          ev\_tanum       = lv\_tanum  
          et\_ltap\_vb     = lt\_ltap  
          et\_bapiret     = lt\_bapiret  
          ev\_severity    = lv\_severity.  
  
      IF lv\_severity CA 'EA'.  
        ROLLBACK WORK.  
  
        IF line\_exists( lt\_bapiret[ type = wmegc\_severity\_abort ] ).  
          DATA(ls\_bapiret) = lt\_bapiret[ type = wmegc\_severity\_abort ].  
        ELSE.  
          IF line\_exists( lt\_bapiret[ type = wmegc\_severity\_err ] ).  
            ls\_bapiret = lt\_bapiret[ type = wmegc\_severity\_err ].  
          ENDIF.  
        ENDIF.  
  
        IF ls\_bapiret IS NOT INITIAL.  
          MESSAGE ID ls\_bapiret-id TYPE ls\_bapiret-type NUMBER ls\_bapiret-number  
          WITH ls\_bapiret-message\_v1 ls\_bapiret-message\_v2 ls\_bapiret-message\_v3 ls\_bapiret-message\_v4.  
        ENDIF.  
      ENDIF.  
  
      APPEND INITIAL LINE TO lt\_tanum ASSIGNING FIELD-SYMBOL(<lfs\_tanum>).  
      IF <lfs\_tanum> IS ASSIGNED.  
        <lfs\_tanum>-tanum = lv\_tanum.  
      ENDIF.  
  
      COMMIT WORK AND WAIT.  
    ENDLOOP.  
  
  ENDIF.  
  
\*------------------------------------------------------------------------------------------------------  
\*Confirm warehouse task  
\*------------------------------------------------------------------------------------------------------  
  IF lt\_tanum IS NOT INITIAL AND  
     lv\_lgnum IS NOT INITIAL.  
  
\*Get quantity in the task  
    SELECT lgnum,  
           tanum,  
           meins,  
           vsolm  
    FROM /scwm/ordim\_o  
    FOR ALL ENTRIES IN @lt\_tanum  
    WHERE lgnum EQ @lv\_lgnum AND  
          tanum EQ @lt\_tanum-tanum  
    INTO TABLE @DATA(lt\_open\_task).  
    IF sy-subrc EQ 0.  
      LOOP AT lt\_tanum ASSIGNING <lfs\_tanum>.  
        APPEND INITIAL LINE TO lt\_conf ASSIGNING FIELD-SYMBOL(<lfs\_conf>).  
        IF <lfs\_conf> IS ASSIGNED.  
          <lfs\_conf>-tanum = <lfs\_tanum>-tanum.  
  
          IF line\_exists( lt\_open\_task[ lgnum = lv\_lgnum  
                                        tanum = <lfs\_tanum>-tanum ] ).  
            DATA(ls\_open\_task) = lt\_open\_task[ lgnum = lv\_lgnum  
                                               tanum = <lfs\_tanum>-tanum ].  
  
            <lfs\_conf>-nista = ls\_open\_task-vsolm.  
            <lfs\_conf>-altme = ls\_open\_task-meins.  
          ENDIF.  
        ENDIF.  
      ENDLOOP.  
    ENDIF.  
  ENDIF.  
  
  IF lt\_conf IS NOT INITIAL.  
    CALL FUNCTION '/SCWM/TO\_CONFIRM'  
      EXPORTING  
        iv\_lgnum       = /scwm/cl\_rf\_bll\_srvc=>get\_lgnum( )  
        iv\_qname       = sy-uname  
        iv\_commit\_work = 'X'  
        it\_conf        = lt\_conf  
      IMPORTING  
        et\_bapiret     = lt\_return.  
  
    IF lt\_return IS NOT INITIAL.  
      READ TABLE lt\_return ASSIGNING FIELD-SYMBOL(<lfs\_return>) WITH KEY type = 'E'.  
      IF sy-subrc EQ 0.  
        MESSAGE ID <lfs\_return>-id TYPE <lfs\_return>-type NUMBER <lfs\_return>-number  
        WITH <lfs\_return>-message\_v1 <lfs\_return>-message\_v2  
             <lfs\_return>-message\_v3 <lfs\_return>-message\_v4 INTO DATA(lv\_error).  
      ENDIF.  
    ENDIF.  
  ENDIF.  
  
ENDFUNCTION.

-------------------------------------------------------------------------------------------------------------------------------------------

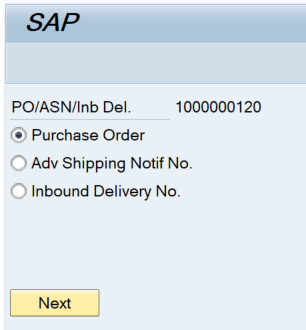
* Create Inbound Delivery



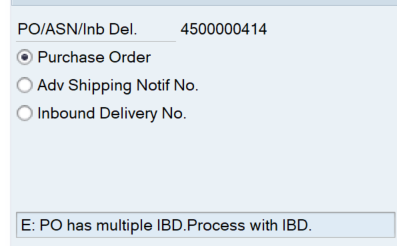
* Logon in the RF screen framework using transaction code /SCWM/RFUI and click on new added custom button   
  
* Enter either PO/ASN or Inbound Delivery and press enter.



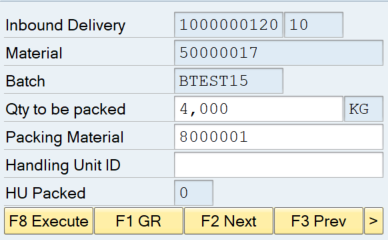
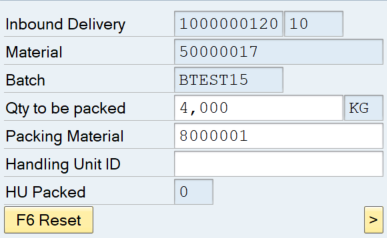
* Now on enter press if the provided input found as duplicate then below screen will appear to check on the provided input from the user.



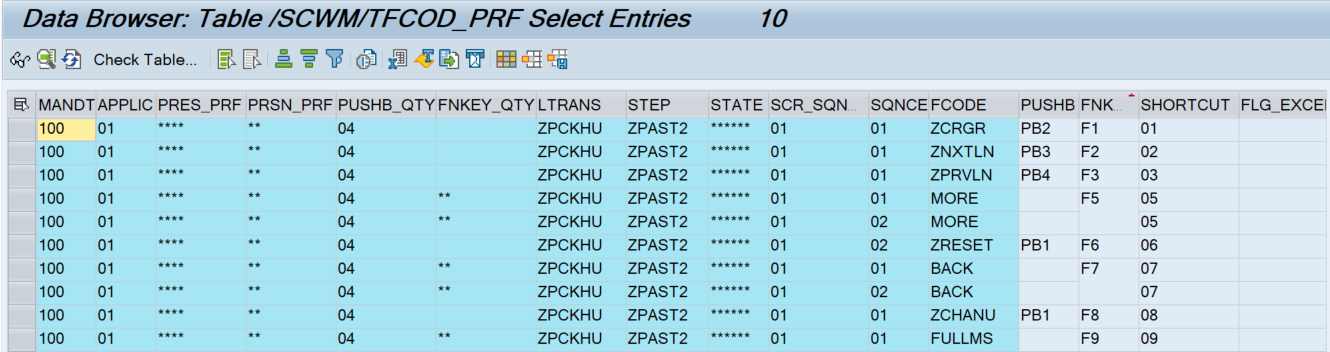
* If multiple inbound deliveries found against the PO or ASN number, then user is asked to provide the appropriate IBD to be processed.



* If valid Inbound delivery found, then below screen will appear.

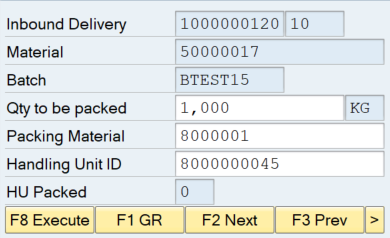
  Click on more button to see more available function codes

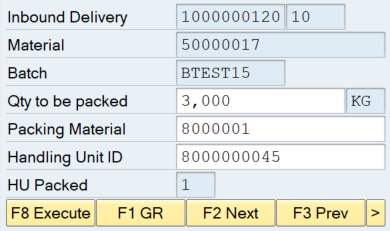
Below are the supported function codes in this screen.



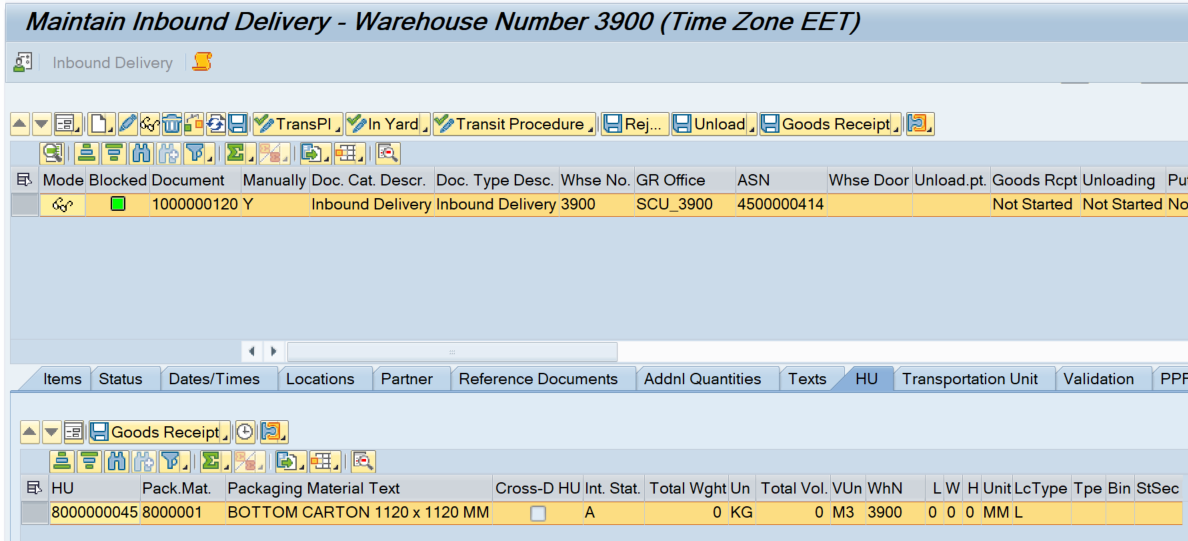
* Pack 1 KG quantity per HU out of total 4KG to be packed.

Change the quantity to be packed as 1 KG and provide the supplier handling unit number and click on Execute button to create the HU and pack the stock.

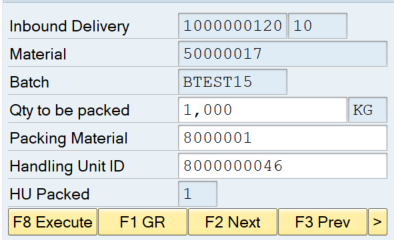
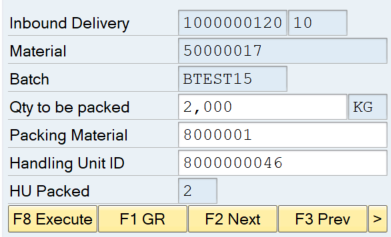


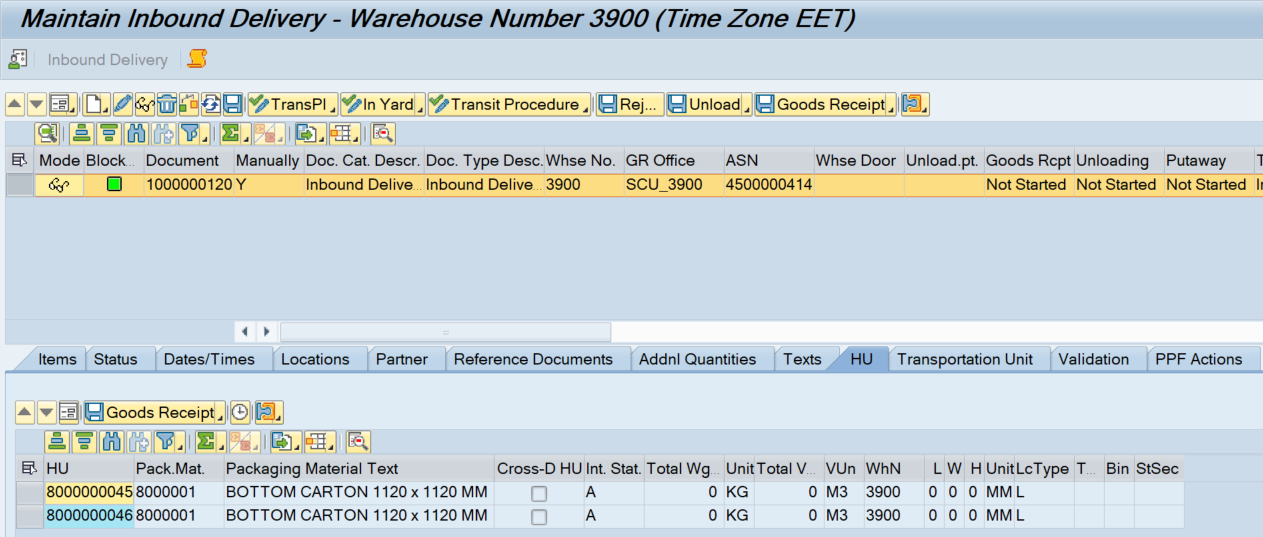
1 HU successfully created and the quantity to be packed is reduced to 3 KG  


Tcode:/SCWM/PRDI

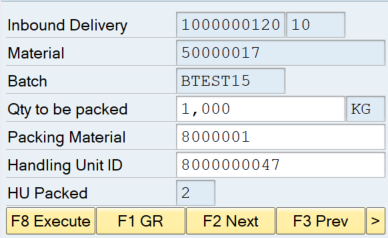
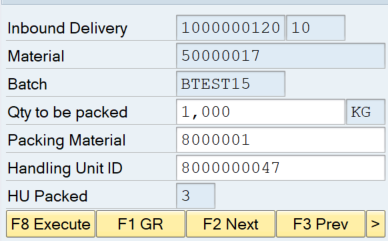


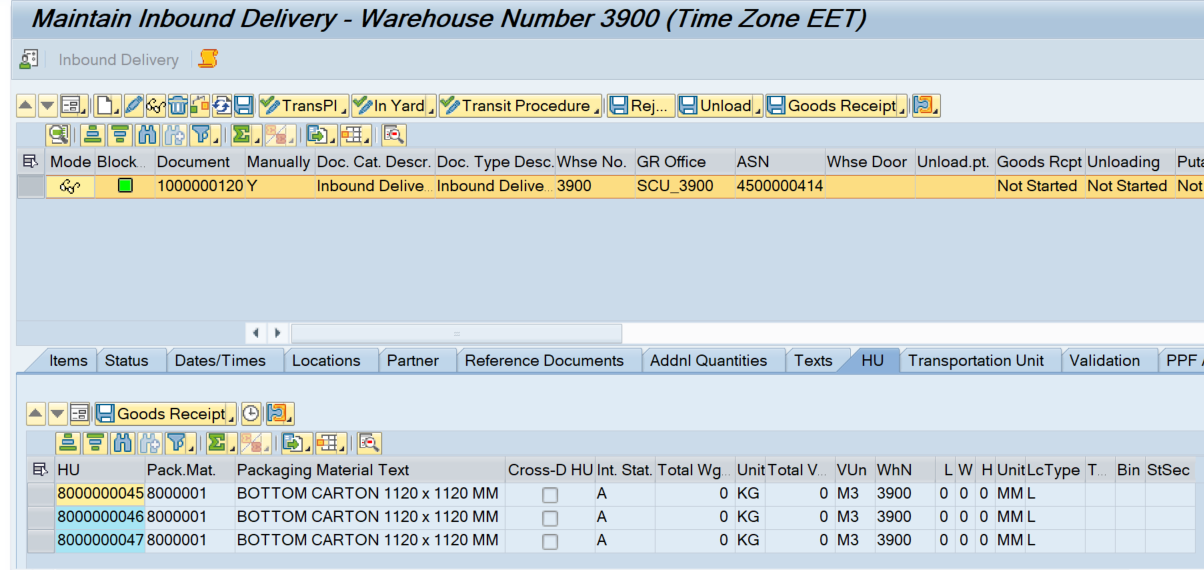
* Create 2nd HU

  HU created and packed  

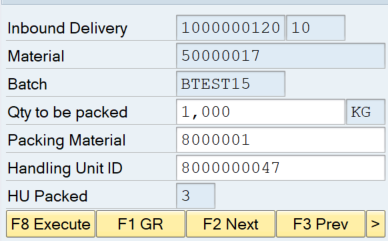
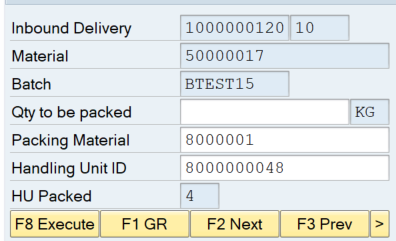


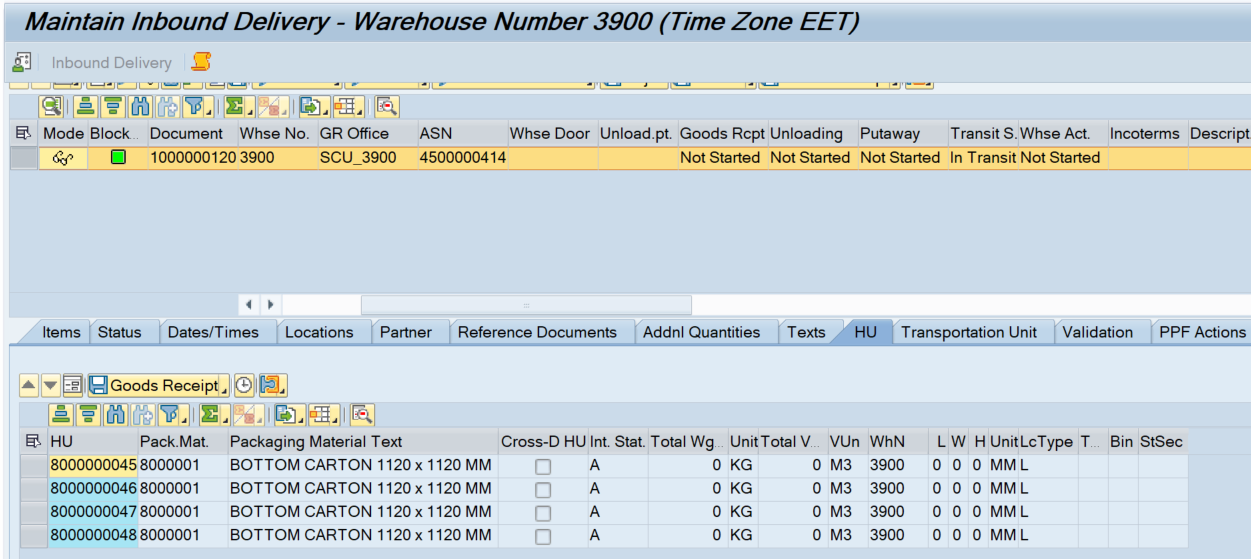
* Create 3rd HU

 HU created and packed 

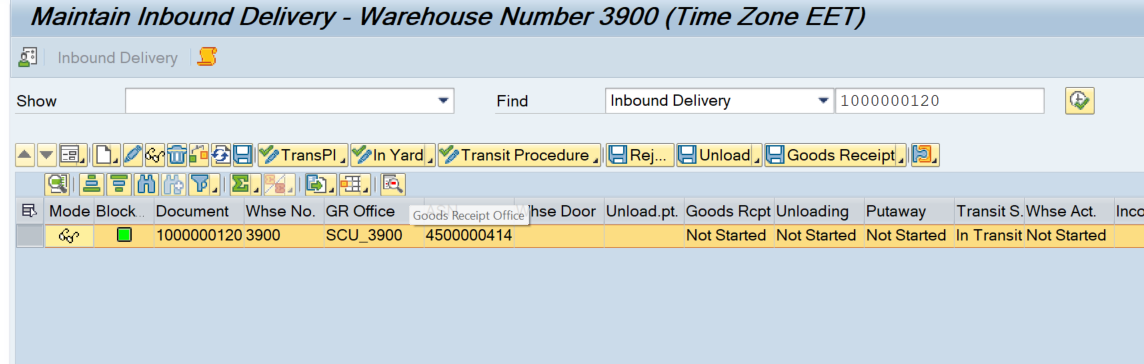


* Create 4th HU

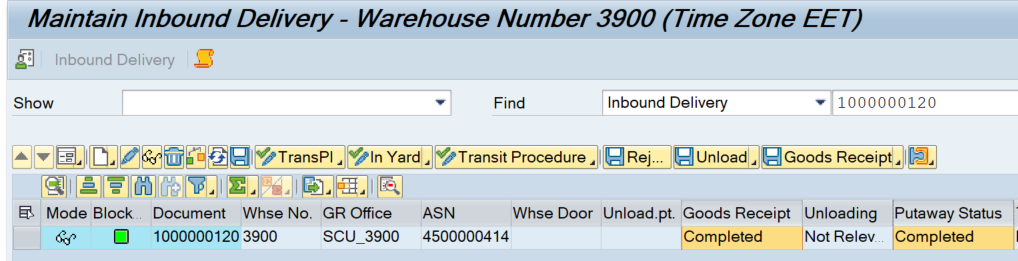
 HU created and packed  

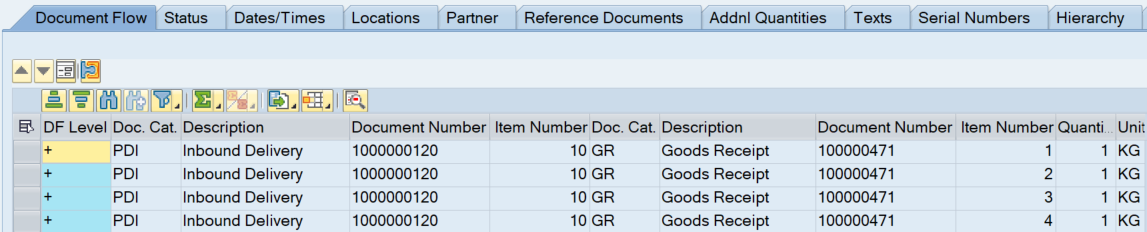
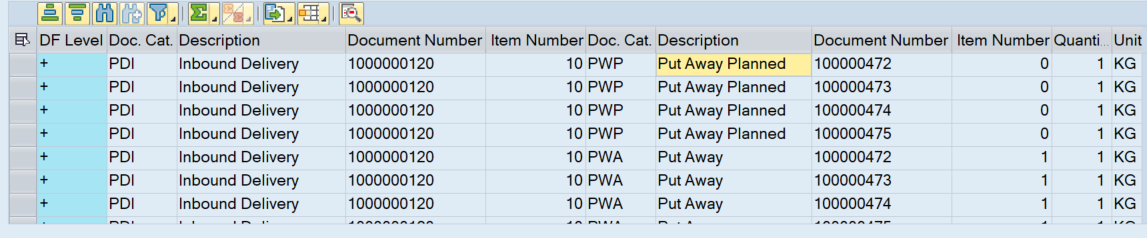


* Click on GR button to post the GR followed by creation and confirmation of Warehouse task.

Before status  


After status



GR documents created  
  
  
Put away documents created  
  
  
If user again try to create the HU below error message will come from standard as shown below  
