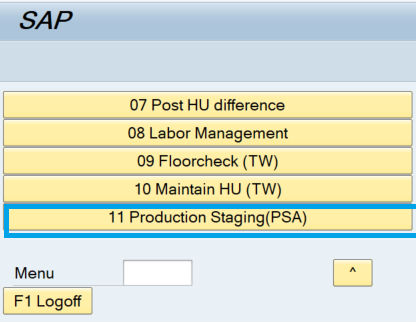
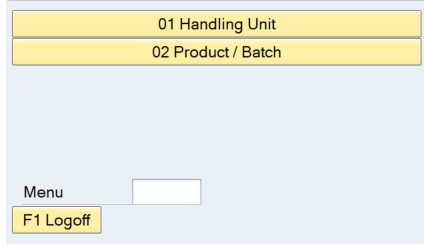
SAP EWM – RF Screen Framework

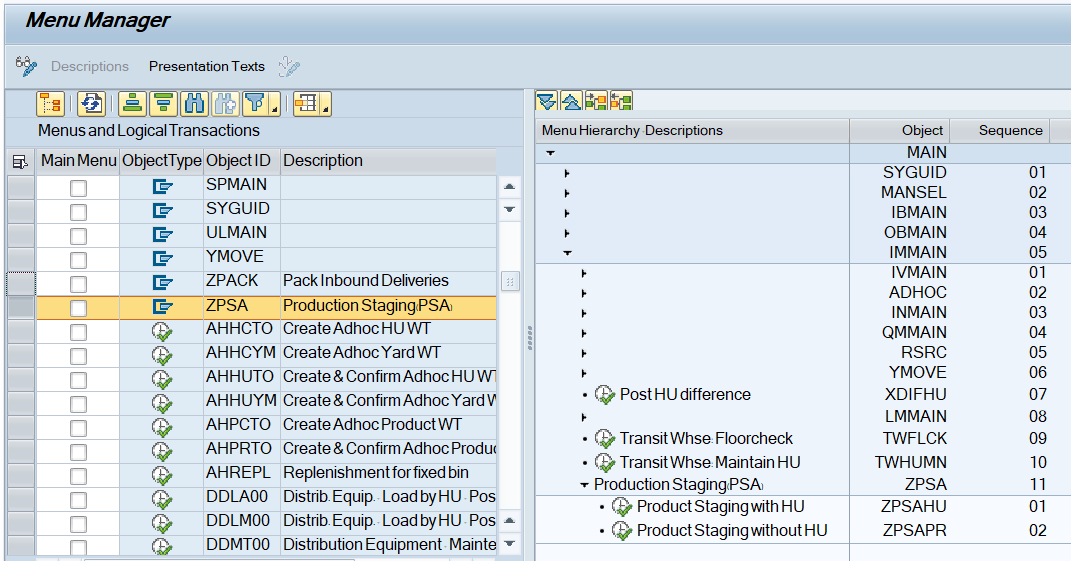
Case1: Create and confirm Warehouse task for the movement of HU to Final Bin.

Below shown are the custom added in the standard RF screen menu:

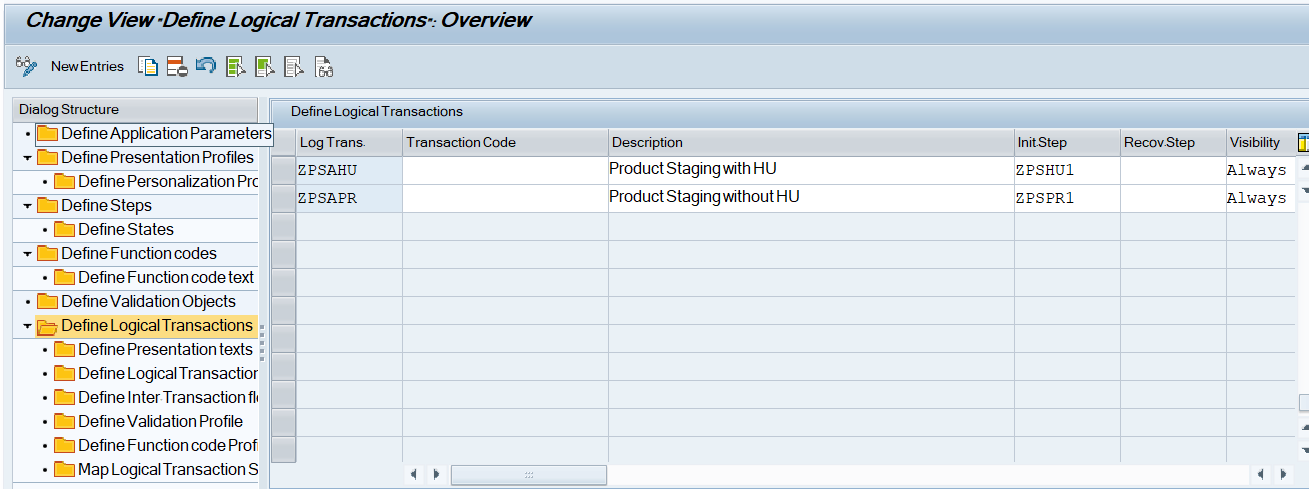
     

Below explained are the configurations done to add above custom options:

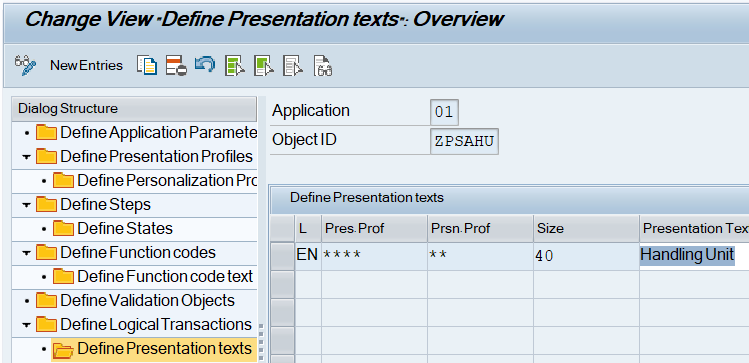
RF Menu Manager



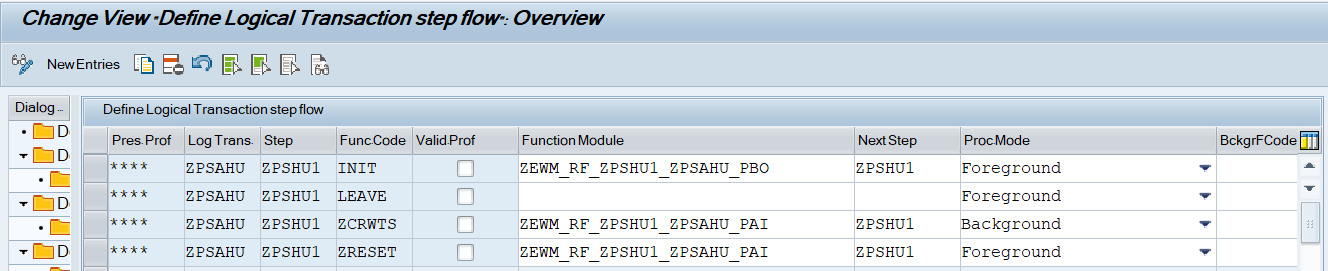
Define Steps in Logical Transactions



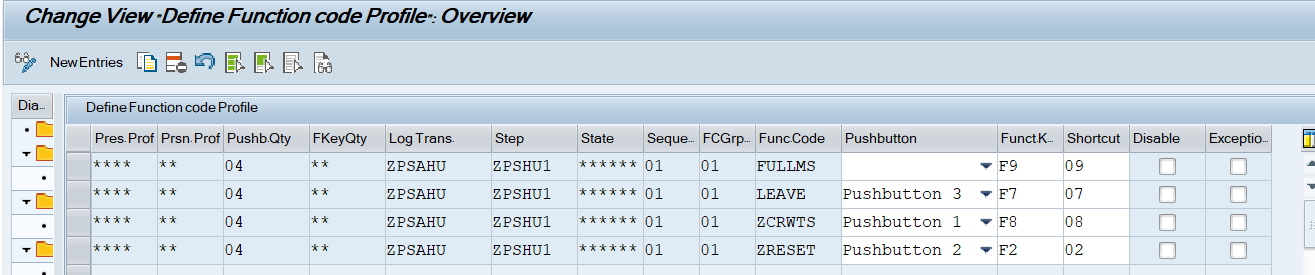
Logical Transaction: ZPSAHU

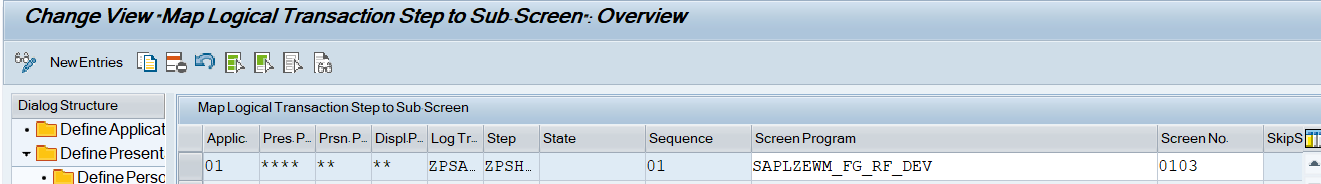


Define Logical Transaction step flow



Define Function code Profile.





RF Configuration:

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

Logical Transaction: ZPSAHU

Initial Step: ZPSHU1

Screen Parameters Structure: ZEWM\_S\_ZPSHU1

FM:

ZEWM\_RF\_ZPSHU1\_ZPSAHU\_PBO

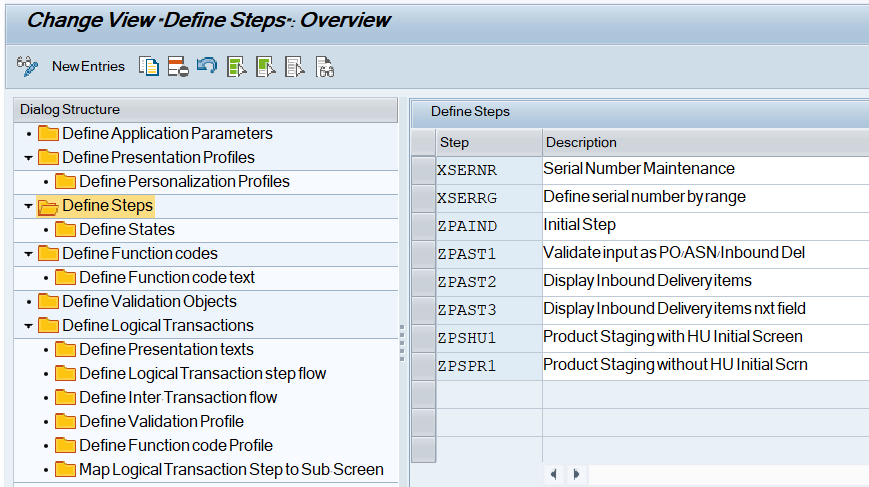
ZEWM\_RF\_ZPSHU1\_ZPSAHU\_PAI

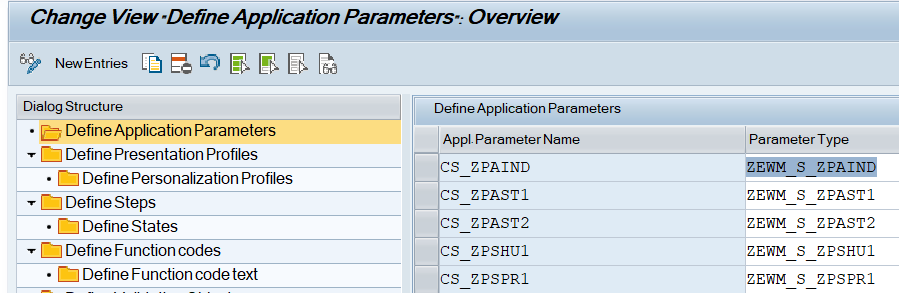
Function Codes:

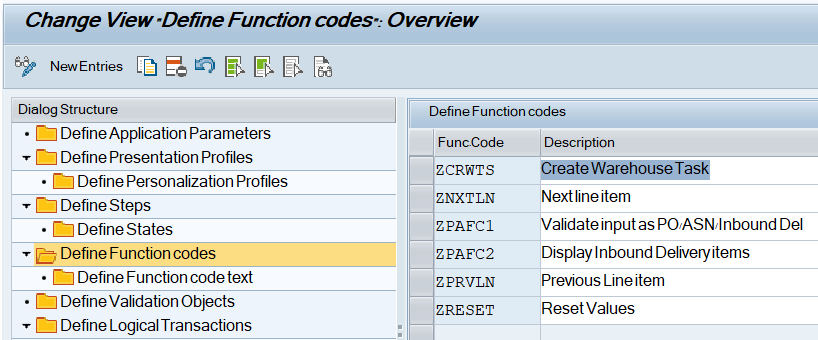
ZCRWTS      Create Warehouse Task

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

Define Steps







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

FUNCTION zewm\_rf\_zpshu1\_zpsahu\_pbo.  
\*"----------------------------------------------------------------------  
\*"\*"Local Interface:  
\*"  CHANGING  
\*"     REFERENCE(CS\_ZPSHU1) TYPE  ZEWM\_S\_ZPSHU1  
\*"----------------------------------------------------------------------  
  
\*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\_zpshu1.  
  
ENDFUNCTION.

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

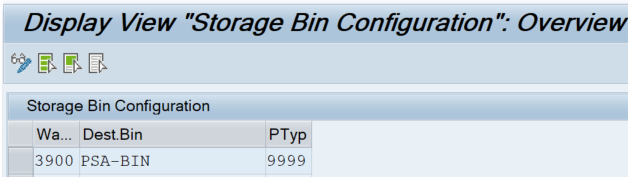
-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------

FUNCTION zewm\_rf\_zpshu1\_zpsahu\_pai.  
\*"----------------------------------------------------------------------  
\*"\*"Local Interface:  
\*"  CHANGING  
\*"     REFERENCE(CS\_ZPSHU1) TYPE  ZEWM\_S\_ZPSHU1  
\*"----------------------------------------------------------------------  
  
  DATA:lv\_wtcode   TYPE /scwm/de\_wtcode,  
       lv\_tanum    TYPE /scwm/tanum,  
       lv\_severity TYPE bapi\_mtype,  
       lv\_ok       TYPE xfeld,  
       lv\_who      TYPE /scwm/de\_who.  
  
  DATA:ls\_huhdr    TYPE /scwm/s\_huhdr\_int,  
       ls\_huheader TYPE /scwm/s\_huhdr\_int,  
       ls\_lagp     TYPE /scwm/lagp,  
       ls\_rsrc     TYPE /scwm/rsrc,  
       ls\_ordim\_o  TYPE /scwm/ordim\_o.  
  
  DATA:lt\_ordim\_o   TYPE /scwm/tt\_ordim\_o,  
       lt\_whoid     TYPE /scwm/tt\_whoid,  
       lt\_who       TYPE /scwm/tt\_who\_int,  
       lt\_create\_hu TYPE /scwm/tt\_to\_crea\_hu,  
       lt\_ltap\_vb   TYPE /scwm/tt\_ltap\_vb,  
       lt\_conf      TYPE /scwm/to\_conf\_tt,  
       lt\_conf\_exc  TYPE /scwm/tt\_conf\_exc,  
       lt\_bapiret   TYPE bapiret2\_t.  
  
  DATA:lo\_hu TYPE REF TO /scwm/cl\_wm\_packing.  
  
  DATA(lv\_fcode) = /scwm/cl\_rf\_bll\_srvc=>get\_fcode( ).  
  
  CASE lv\_fcode.  
    WHEN 'ZCRWTS'.  
  
      CALL FUNCTION '/SCWM/RSRC\_RESOURCE\_MEMORY'  
        EXPORTING  
          iv\_uname = sy-uname  
        CHANGING  
          cs\_rsrc  = ls\_rsrc.  
  
      /scwm/cl\_rf\_bll\_srvc=>set\_lgnum( iv\_lgnum = ls\_rsrc-lgnum ).  
  
\*--------------------------------------------------------------------  
\*Validate if HU exists against the supplier provided HU ID  
\*Validate if multiple HU found against the supplier provided HU ID  
\*--------------------------------------------------------------------  
      IF cs\_zpshu1-hu\_id IS NOT INITIAL.  
  
        SELECT a~guid\_hu,  
               a~huident AS huid\_ext,  
               b~huident AS huid\_int  
        FROM /scwm/hu\_ident AS a INNER JOIN /scwm/huhdr AS b ON b~guid\_hu EQ a~guid\_hu  
        INTO TABLE @DATA(lt\_huid)  
        WHERE a~huident EQ @cs\_zpshu1-hu\_id.  
        IF sy-subrc EQ 0.  
          IF lines( lt\_huid ) GT 1.  
            MESSAGE e015(zewm01) WITH cs\_zpshu1-hu\_id.  
          ENDIF.  
        ELSE.  
          MESSAGE e014(zewm01) WITH cs\_zpshu1-hu\_id.  
        ENDIF.  
      ENDIF.  
\*--------------------------------------------------------------------  
  
\*--------------------------------------------------------------------  
\*Validate WO exists with in process status for HU  
\*--------------------------------------------------------------------  
      DATA(lv\_huid) = lt\_huid[ 1 ]-huid\_int.  
  
      CALL FUNCTION '/SCWM/TO\_READ\_HU'  
        EXPORTING  
          iv\_lgnum       = /scwm/cl\_rf\_bll\_srvc=>get\_lgnum( )  
          iv\_huident     = lv\_huid  
        IMPORTING  
          et\_ordim\_o\_src = lt\_ordim\_o  
        EXCEPTIONS  
          OTHERS         = 0.  
      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\_ordim\_o ASSIGNING FIELD-SYMBOL(<lfs\_ordim\_o>).  
        APPEND INITIAL LINE TO lt\_whoid ASSIGNING FIELD-SYMBOL(<lfs\_whoid>).  
        IF <lfs\_whoid> IS ASSIGNED.  
          <lfs\_whoid>-who = <lfs\_ordim\_o>-who.  
        ENDIF.  
      ENDLOOP.  
  
      IF lt\_whoid IS NOT INITIAL.  
        TRY.  
            CALL FUNCTION '/SCWM/WHO\_SELECT'  
              EXPORTING  
                iv\_lgnum = /scwm/cl\_rf\_bll\_srvc=>get\_lgnum( )  
                it\_who   = lt\_whoid  
              IMPORTING  
                et\_who   = lt\_who.  
          CATCH /scwm/cx\_core.  
            MESSAGE ID sy-msgid TYPE sy-msgty NUMBER sy-msgno WITH sy-msgv1 sy-msgv2 sy-msgv3 sy-msgv4.  
        ENDTRY.  
  
        LOOP AT lt\_who ASSIGNING FIELD-SYMBOL(<ls\_who>) WHERE rsrc IS NOT INITIAL.  
          MESSAGE e016(zewm01) WITH lv\_huid <ls\_who>-rsrc.  
        ENDLOOP.  
  
      ENDIF.  
\*--------------------------------------------------------------------  
  
\*--------------------------------------------------------------------  
\*Validate if source bin is blocked  
\*--------------------------------------------------------------------  
  
      CALL FUNCTION '/SCWM/HU\_READ'  
        EXPORTING  
          iv\_huident = lv\_huid  
          iv\_lgnum   = /scwm/cl\_rf\_bll\_srvc=>get\_lgnum( )  
        IMPORTING  
          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.  
  
      IF ls\_huhdr-lgpla IS NOT INITIAL.  
        CALL FUNCTION '/SCWM/LAGP\_READ\_SINGLE'  
          EXPORTING  
            iv\_lgnum      = ls\_huhdr-lgnum  
            iv\_lgpla      = ls\_huhdr-lgpla  
          IMPORTING  
            es\_lagp       = ls\_lagp  
          EXCEPTIONS  
            wrong\_input   = 1  
            not\_found     = 2  
            enqueue\_error = 3  
            OTHERS        = 4.  
        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.  
  
      IF ls\_lagp-skzua = abap\_true.  
        MESSAGE e017(zewm01) WITH ls\_lagp-lgpla.  
      ENDIF.  
  
\*--------------------------------------------------------------------  
  
\*--------------------------------------------------------------------  
\*Read storage bin configuration  
\*--------------------------------------------------------------------  
      SELECT SINGLE \*  
      FROM zewm\_t\_bin\_conf  
      INTO @DATA(ls\_bin\_conf)  
      WHERE warehouse\_no EQ @ls\_rsrc-lgnum.  
      IF sy-subrc EQ 0.  
  
        IF ls\_bin\_conf-dest\_bin IS INITIAL.  
          MESSAGE e029(zewm01) WITH ls\_rsrc-lgnum.  
        ENDIF.  
  
        IF ls\_bin\_conf-process\_type IS INITIAL.  
          MESSAGE e030(zewm01) WITH ls\_rsrc-lgnum.  
        ENDIF.  
  
        DATA(lv\_dest\_bin)     = ls\_bin\_conf-dest\_bin.  
        DATA(lv\_process\_type) = ls\_bin\_conf-process\_type.  
      ELSE.  
        MESSAGE e029(zewm01) WITH ls\_rsrc-lgnum.  
      ENDIF.  
\*--------------------------------------------------------------------  
  
\*--------------------------------------------------------------------  
\*Create Warehouse task for HU movement to Final Bin  
\*--------------------------------------------------------------------  
      lv\_wtcode = wmegc\_wtcode\_adhoc\_hu.  
  
      lo\_hu = NEW #( ).  
  
      CALL METHOD lo\_hu->get\_hu  
        EXPORTING  
          iv\_guid\_hu = lt\_huid[ 1 ]-guid\_hu  
          iv\_lock    = 'X'  
        IMPORTING  
          es\_huhdr   = ls\_huheader  
        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.  
  
      lt\_create\_hu = VALUE #( ( huident  = lv\_huid  
                                guid\_hu = lt\_huid[ 1 ]-guid\_hu  
                                procty  = lv\_process\_type  
                                nlpla   = lv\_dest\_bin  
                                nlenr   = lv\_huid  
                                seqno   = 1  
                                dstgrp  = ls\_huheader-dstgrp ) ).  
  
      CALL FUNCTION '/SCWM/RF\_PRINT\_GLOBAL\_DATA'.  
  
      CALL FUNCTION '/SCWM/TO\_CREATE\_MOVE\_HU'  
        EXPORTING  
          iv\_lgnum       = ls\_huhdr-lgnum  
          it\_create\_hu   = lt\_create\_hu  
          iv\_wtcode      = lv\_wtcode  
          iv\_commit\_work = ' '  
        IMPORTING  
          ev\_tanum       = lv\_tanum  
          et\_ltap\_vb     = lt\_ltap\_vb  
          et\_bapiret     = lt\_bapiret  
          ev\_severity    = lv\_severity.  
  
      IF lv\_severity CA wmegc\_severity\_eax.  
        IF lt\_bapiret IS NOT INITIAL.  
          LOOP AT lt\_bapiret INTO DATA(ls\_bapiret).  
            IF ls\_bapiret-type CA wmegc\_severity\_ea.  
              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.  
              EXIT.  
            ENDIF.  
          ENDLOOP.  
        ENDIF.  
      ENDIF.  
  
      COMMIT WORK AND WAIT.  
      IF sy-subrc IS NOT INITIAL.  
        MESSAGE e018(zewm01).  
      ENDIF.  
  
\*--------------------------------------------------------------------  
\*Validate if warehouse order designed for resource  
\*--------------------------------------------------------------------  
      IF line\_exists( lt\_ltap\_vb[ tanum = lv\_tanum ] ).  
        DATA(ls\_ltap\_vb) = lt\_ltap\_vb[ tanum = lv\_tanum ].  
  
        PERFORM who\_rsrc\_allowed USING ls\_huhdr-lgnum  
                                       ls\_ltap\_vb-who  
                                       ls\_ltap\_vb-queue  
                                 CHANGING lv\_ok.  
  
        IF lv\_ok IS INITIAL.  
          MESSAGE e019(zewm01).  
        ENDIF.  
      ENDIF.  
  
      CALL FUNCTION '/SCWM/REC\_WT\_CHANGE'  
        EXPORTING  
          it\_ltap\_vb        = lt\_ltap\_vb  
        EXCEPTIONS  
          no\_free\_wt\_in\_who = 1  
          internal\_error    = 2  
          OTHERS            = 3.  
      IF sy-subrc <> 0.  
        MESSAGE ID sy-msgid TYPE sy-msgty NUMBER sy-msgno WITH sy-msgv1 sy-msgv2 sy-msgv3 sy-msgv4.  
      ENDIF.  
  
\*--------------------------------------------------------------------  
\*Confirm Warehouse Task  
\*--------------------------------------------------------------------  
  
\*Update WHO with resource and processor and start bin  
      lv\_who = ls\_ltap\_vb-who.  
      CALL FUNCTION '/SCWM/RF\_WHO\_UPD'  
        EXPORTING  
          iv\_lgnum = ls\_huhdr-lgnum  
          iv\_who   = lv\_who.  
  
      CLEAR:lt\_bapiret,lv\_severity.  
  
      CALL FUNCTION '/SCWM/RSRC\_RESOURCE\_MEMORY'  
        EXPORTING  
          iv\_uname = sy-uname  
        CHANGING  
          cs\_rsrc  = ls\_rsrc.  
  
      APPEND INITIAL LINE TO lt\_conf ASSIGNING FIELD-SYMBOL(<lfs\_conf>).  
      IF <lfs\_conf> IS ASSIGNED.  
        <lfs\_conf>-tanum = lv\_tanum.  
        <lfs\_conf>-squit = abap\_true.  
        <lfs\_conf>-nlenr = lv\_huid.  
  
        CALL FUNCTION 'CONVERSION\_EXIT\_HUID\_INPUT'  
          EXPORTING  
            input  = <lfs\_conf>-nlenr  
          IMPORTING  
            output = <lfs\_conf>-nlenr.  
  
        <lfs\_conf>-drsrc = ls\_rsrc-rsrc.  
      ENDIF.  
  
      CALL FUNCTION '/SCWM/TO\_CONFIRM'  
        EXPORTING  
          iv\_lgnum         = ls\_huhdr-lgnum  
          iv\_wtcode        = wmegc\_wtcode\_rsrc  
          it\_conf          = lt\_conf  
          iv\_commit\_work   = ' '  
          iv\_update\_task   = ' '  
          iv\_processor\_det = 'X'  
        IMPORTING  
          et\_bapiret       = lt\_bapiret  
          ev\_severity      = lv\_severity.  
  
      IF lv\_severity CA wmegc\_severity\_eax.  
        LOOP AT lt\_bapiret INTO ls\_bapiret.  
          IF ls\_bapiret-type CA wmegc\_severity\_ea.  
            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.  
            EXIT.  
          ENDIF.  
        ENDLOOP.  
      ENDIF.  
  
      COMMIT WORK AND WAIT.  
      IF sy-subrc IS NOT INITIAL.  
        MESSAGE e020(zewm01).  
      ENDIF.  
  
\*Determination of new TO number  
      CLEAR lt\_ordim\_o.  
      CALL FUNCTION '/SCWM/TO\_READ\_WHO'  
        EXPORTING  
          iv\_lgnum     = ls\_huhdr-lgnum  
          iv\_who       = lv\_who  
        IMPORTING  
          et\_ordim\_o   = lt\_ordim\_o  
        EXCEPTIONS  
          wrong\_input  = 1  
          not\_found    = 2  
          foreign\_lock = 3  
          error        = 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.  
  
      IF lt\_ordim\_o IS NOT INITIAL.  
  
        IF line\_exists( lt\_ordim\_o[ who = lv\_who ] ).  
          ls\_ordim\_o = lt\_ordim\_o[ who = lv\_who ].  
        ENDIF.  
  
        CALL FUNCTION '/SCWM/REC\_WT\_CHANGE'  
          EXPORTING  
            it\_ltap\_vb        = lt\_ltap\_vb  
          CHANGING  
            cs\_ordim\_o        = ls\_ordim\_o  
            ct\_ordim\_o        = lt\_ordim\_o  
          EXCEPTIONS  
            no\_free\_wt\_in\_who = 1  
            internal\_error    = 2  
            OTHERS            = 3.  
        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.  
  
\*--------------------------------------------------------------------  
\*Confirm new TO on destination bin  
\*--------------------------------------------------------------------  
      CLEAR lt\_conf.  
      APPEND INITIAL LINE TO lt\_conf ASSIGNING <lfs\_conf>.  
      IF <lfs\_conf> IS ASSIGNED.  
        <lfs\_conf>-tanum = lv\_tanum.  
        <lfs\_conf>-nlpla = lv\_dest\_bin.  
        <lfs\_conf>-nlenr = lv\_huid.  
  
        CALL FUNCTION 'CONVERSION\_EXIT\_HUID\_INPUT'  
          EXPORTING  
            input  = <lfs\_conf>-nlenr  
          IMPORTING  
            output = <lfs\_conf>-nlenr.  
      ENDIF.  
  
      CLEAR:lv\_severity,lt\_bapiret.  
      CALL FUNCTION '/SCWM/TO\_CONF\_INT\_SIMULATE'  
        EXPORTING  
          iv\_lgnum                 = ls\_huhdr-lgnum  
          iv\_catch\_weight\_no\_check = abap\_true  
          it\_conf                  = lt\_conf  
          it\_conf\_exc              = lt\_conf\_exc  
        IMPORTING  
          et\_ltap\_vb               = lt\_ltap\_vb  
          et\_bapiret               = lt\_bapiret  
          ev\_severity              = lv\_severity  
          ev\_ok                    = lv\_ok.  
  
      IF lv\_severity CA 'EAX'.  
        CLEAR ls\_bapiret.  
        LOOP AT lt\_bapiret INTO ls\_bapiret.  
          IF ls\_bapiret-type CA wmegc\_severity\_ea.  
            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.  
            EXIT.  
          ENDIF.  
        ENDLOOP.  
      ENDIF.  
  
      CLEAR:lv\_severity,lt\_bapiret.  
      CALL FUNCTION '/SCWM/TO\_CONFIRM'  
        EXPORTING  
          iv\_lgnum         = ls\_huhdr-lgnum  
          it\_conf          = lt\_conf  
          iv\_commit\_work   = ' '  
          iv\_update\_task   = ' '  
          iv\_processor\_det = 'X'  
          it\_conf\_exc      = lt\_conf\_exc  
        IMPORTING  
          et\_bapiret       = lt\_bapiret  
          ev\_severity      = lv\_severity.  
  
      IF lv\_severity CA wmegc\_severity\_eax.  
        CLEAR ls\_bapiret.  
        LOOP AT lt\_bapiret INTO ls\_bapiret.  
          IF ls\_bapiret-type CA wmegc\_severity\_ea.  
            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.  
            EXIT.  
          ENDIF.  
        ENDLOOP.  
      ENDIF.  
  
      COMMIT WORK AND WAIT.  
      IF sy-subrc IS NOT INITIAL.  
        MESSAGE e020(zewm01).  
      ENDIF.  
  
      /scwm/cl\_rf\_bll\_srvc=>set\_prmod( '1' ).  
      /scwm/cl\_rf\_bll\_srvc=>set\_fcode( /scwm/cl\_rf\_bll\_srvc=>c\_fcode\_compl\_ltrans ).  
  
    WHEN 'ZRESET'.  
      CLEAR cs\_zpshu1-hu\_id.  
  
    WHEN OTHERS.  
  ENDCASE.  
  
ENDFUNCTION.

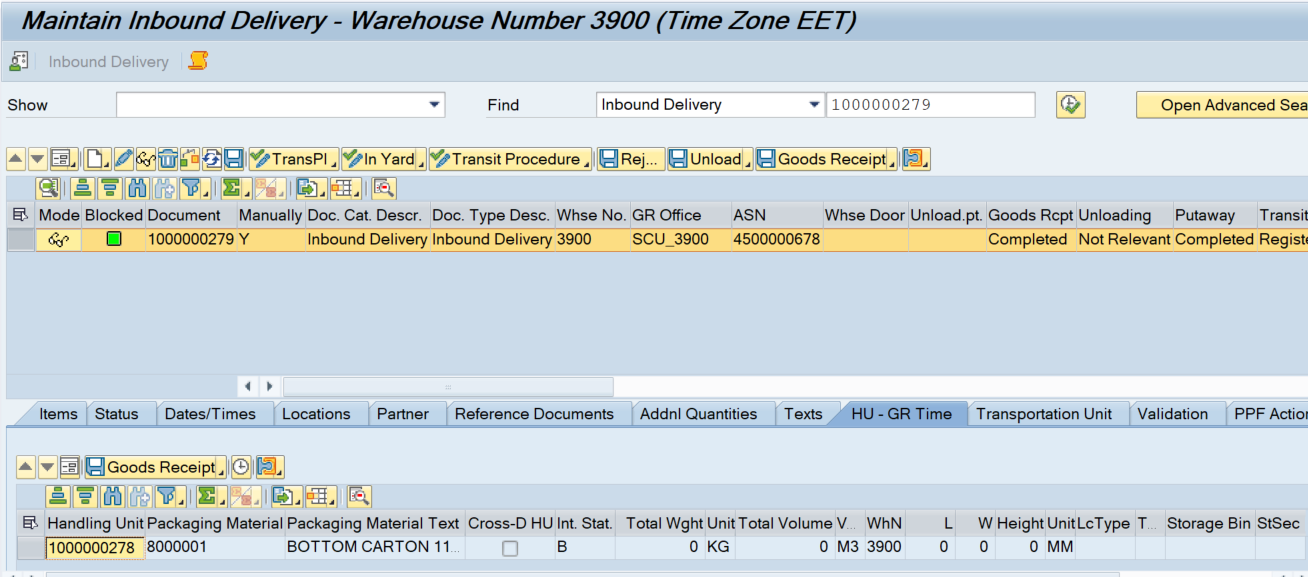
-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------

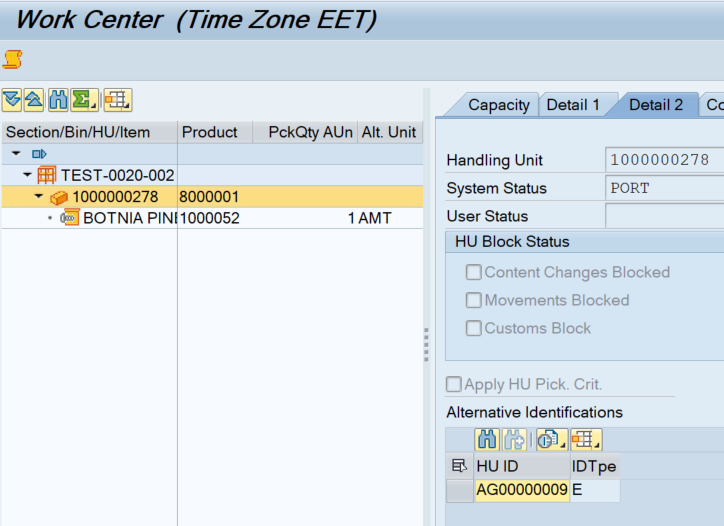
Unit test detail:

Prerequisite: Process Type and Destination Bin for the warehouse should be maintained as shown below using SM30 - ZEWM\_T\_BIN\_CONF:



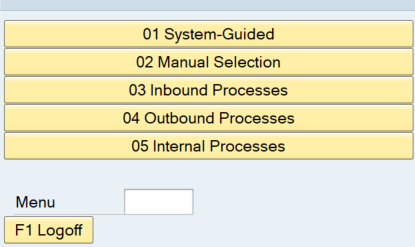
Case 1 : Perform PSA with Supplier provided Handling Unit.



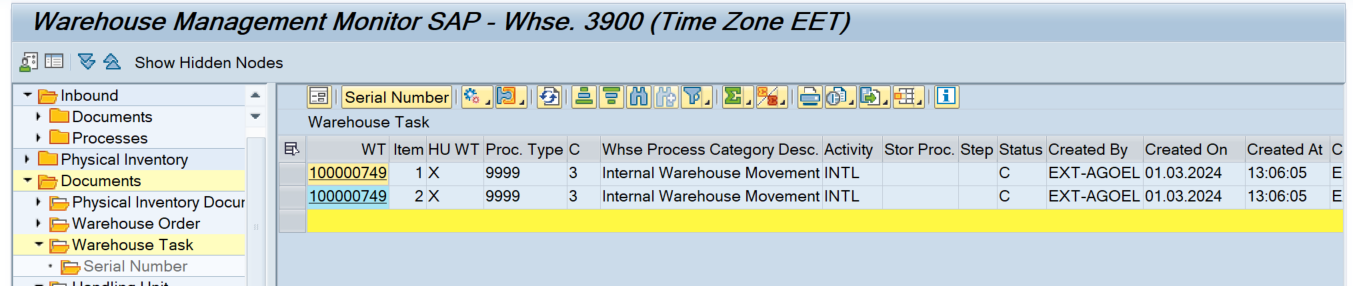


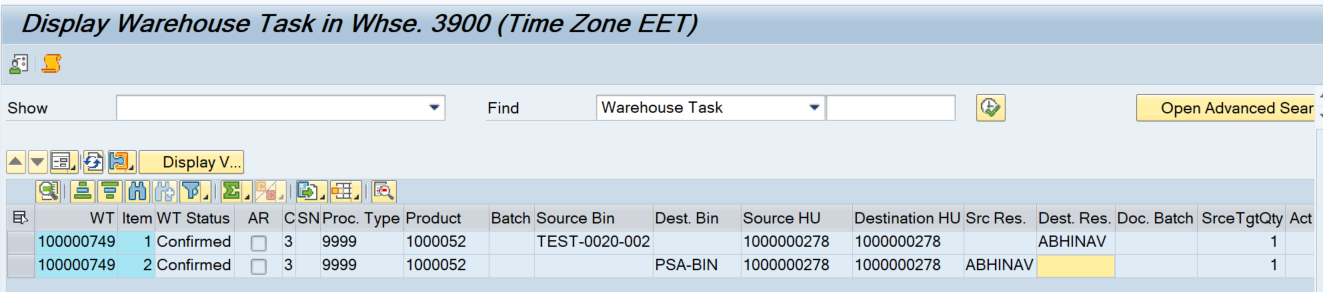
Click on Handling Unit Button and enter supplier provided HU and click on Execute button.



Same as SAP standard behavior, screen flow will come back to initial menu if successful.  


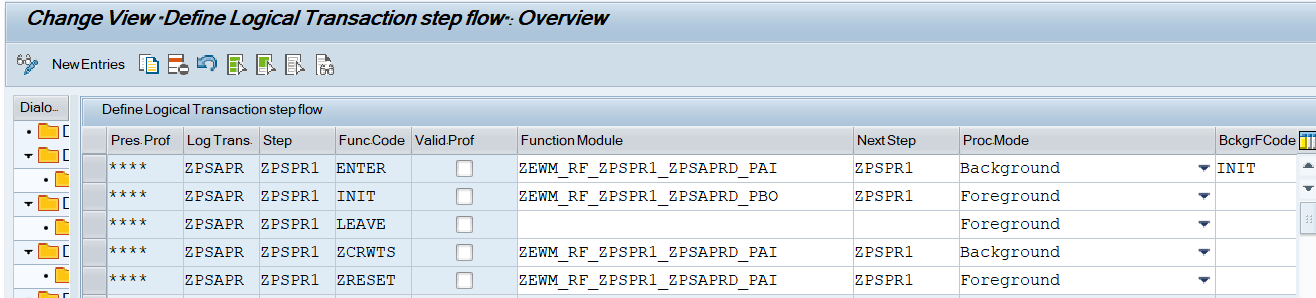
Go to /SCWM/MON transaction code and check the warehouse task for the movement to final bin.

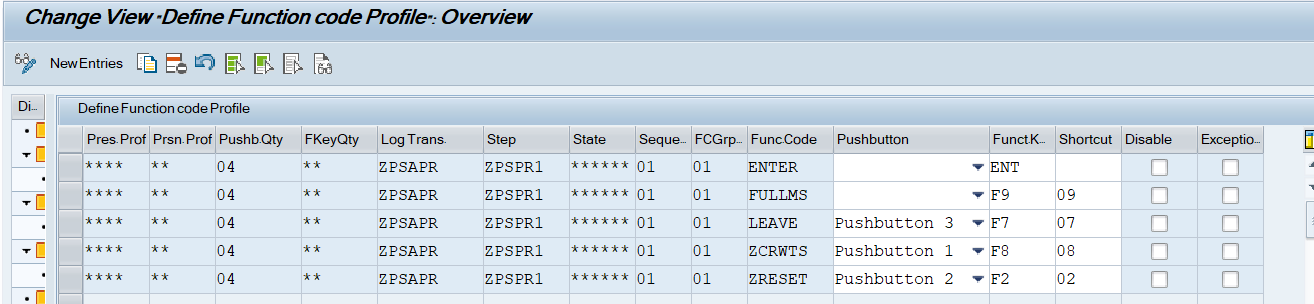


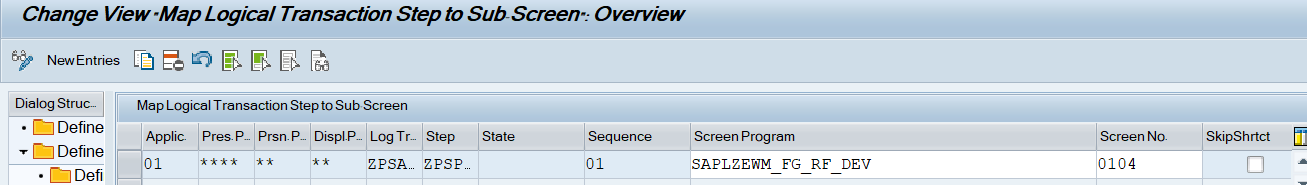


Case 2:Create and confirm Warehouse task for the movement of Product/Batch to Final Bin.

Logical Transaction: ZPSAPR







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

FUNCTION zewm\_rf\_zpspr1\_zpsaprd\_pbo.  
\*"----------------------------------------------------------------------  
\*"\*"Local Interface:  
\*"  CHANGING  
\*"     REFERENCE(CS\_ZPSPR1) TYPE  ZEWM\_S\_ZPSPR1  
\*"----------------------------------------------------------------------  
  
\*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\_zpspr1.  
  
ENDFUNCTION.

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

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

FUNCTION zewm\_rf\_zpspr1\_zpsaprd\_pai.  
\*"----------------------------------------------------------------------  
\*"\*"Local Interface:  
\*"  CHANGING  
\*"     REFERENCE(CS\_ZPSPR1) TYPE  ZEWM\_S\_ZPSPR1  
\*"----------------------------------------------------------------------  
  
  DATA:lv\_matnr40       TYPE /scwm/de\_rf\_matnr40,  
       lv\_filtered\_auth TYPE xfeld,  
       lv\_matid         TYPE /scwm/de\_matid,  
       lv\_papos         TYPE /scwm/ltap\_hu\_papos,  
       lv\_severity      TYPE bapi\_mtype,  
       lv\_ok            TYPE xfeld.  
  
  DATA:ls\_rsrc       TYPE /scwm/rsrc,  
       ls\_ordim\_o    TYPE /scwm/ordim\_o,  
       ls\_mat\_global TYPE /scwm/s\_material\_global.  
  
  DATA:lt\_range\_matid TYPE rseloption,  
       lt\_range\_lgpla TYPE rseloption,  
       lt\_range\_matnr TYPE rseloption,  
       lt\_conf        TYPE /scwm/to\_conf\_tt,  
       lt\_aqua\_int    TYPE /scwm/tt\_aqua\_int2,  
       lt\_ordim\_c     TYPE /scwm/tt\_ordim\_c,  
       lt\_conf\_exc    TYPE /scwm/tt\_conf\_exc,  
       lt\_conf\_serid  TYPE /scwm/tt\_conf\_serid,  
       lt\_ltap\_vb     TYPE /scwm/tt\_ltap\_vb,  
       lt\_bapiret     TYPE bapiret2\_t.  
  
  DATA(lv\_fcode) = /scwm/cl\_rf\_bll\_srvc=>get\_fcode( ).  
  
  CASE lv\_fcode.  
    WHEN 'ENTER'.  
\*--------------------------------------------------------  
\*Validate manadatory input received  
\*--------------------------------------------------------  
      IF cs\_zpspr1-product IS INITIAL.  
        MESSAGE e021(zewm01).  
      ENDIF.  
  
      IF cs\_zpspr1-source\_bin IS INITIAL.  
        MESSAGE e022(zewm01).  
      ENDIF.  
  
      IF gv\_batch\_req EQ abap\_true.  
        IF cs\_zpspr1-batch IS INITIAL.  
          MESSAGE e028(zewm01).  
        ENDIF.  
      ENDIF.  
  
\*--------------------------------------------------------  
  
      CALL FUNCTION '/SCWM/RSRC\_RESOURCE\_MEMORY'  
        EXPORTING  
          iv\_uname = sy-uname  
        CHANGING  
          cs\_rsrc  = ls\_rsrc.  
  
\*--------------------------------------------------------  
\*Validate if product exists  
\*--------------------------------------------------------  
      lv\_matnr40 = cs\_zpspr1-product.  
  
      CALL FUNCTION '/SCWM/RF\_PRODUCT\_INPUT'  
        EXPORTING  
          input    = lv\_matnr40  
        IMPORTING  
          ev\_matid = lv\_matid.  
  
      IF lv\_matid IS INITIAL.  
        MESSAGE e023(zewm01) WITH cs\_zpspr1-product.  
      ENDIF.  
\*--------------------------------------------------------  
  
\*--------------------------------------------------------  
\*Validate if the product is batch managed  
\*--------------------------------------------------------  
      IF gv\_batch\_req EQ abap\_false.  
  
        TRY.  
            CALL FUNCTION '/SCWM/MATERIAL\_READ\_SINGLE'  
              EXPORTING  
                iv\_matid      = lv\_matid  
                iv\_lgnum      = ls\_rsrc-lgnum  
              IMPORTING  
                es\_mat\_global = ls\_mat\_global.  
          CATCH /scwm/cx\_md.  
        ENDTRY.  
  
        IF  ls\_mat\_global-batch\_req EQ abap\_true.  
          /scwm/cl\_rf\_bll\_srvc=>set\_screlm\_input\_on( 'ZEWM\_S\_ZPSPR1-BATCH' ).  
          gv\_batch\_req = abap\_true.  
          EXIT.  
        ENDIF.  
  
      ENDIF.  
\*--------------------------------------------------------  
  
\*--------------------------------------------------------  
\*Read data to process  
\*--------------------------------------------------------  
      lt\_range\_lgpla = VALUE #( ( sign   = 'I'  
                                  option = 'EQ'  
                                  low    = cs\_zpspr1-source\_bin ) ).  
  
      lt\_range\_matid = VALUE #( ( sign   = 'I'  
                                  option = 'EQ'  
                                  low    = lv\_matid ) ).  
  
      CALL FUNCTION '/SCWM/AQUA\_SELECT'  
        EXPORTING  
          iv\_lgnum         = ls\_rsrc-lgnum  
          ir\_lgpla         = lt\_range\_lgpla  
          ir\_matid         = lt\_range\_matid  
        IMPORTING  
          et\_aqua\_int      = lt\_aqua\_int  
          ev\_filtered\_auth = lv\_filtered\_auth  
        EXCEPTIONS  
          wrong\_input      = 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.  
  
      IF lv\_filtered\_auth IS NOT INITIAL.  
        MESSAGE e025(zewm01).  
      ENDIF.  
  
      cs\_zpspr1-quantum = lines( lt\_aqua\_int ).  
      DELETE lt\_aqua\_int WHERE huident IS NOT INITIAL.  
  
      IF gv\_batch\_req EQ abap\_true.  
        DELETE lt\_aqua\_int WHERE charg NE cs\_zpspr1-batch.  
      ENDIF.  
  
      IF gv\_batch\_req EQ abap\_false.  
        IF line\_exists( lt\_aqua\_int[ lgnum = ls\_rsrc-lgnum  
                                     lgpla = cs\_zpspr1-source\_bin  
                                     matid = lv\_matid ] ).  
  
          DATA(ls\_aqua\_int) = lt\_aqua\_int[ lgnum = ls\_rsrc-lgnum  
                                           lgpla = cs\_zpspr1-source\_bin  
                                           matid = lv\_matid ].  
  
        ENDIF.  
      ELSE.  
        IF line\_exists( lt\_aqua\_int[ lgnum = ls\_rsrc-lgnum  
                                     lgpla = cs\_zpspr1-source\_bin  
                                     matid = lv\_matid  
                                     charg = cs\_zpspr1-batch ] ).  
  
          ls\_aqua\_int = lt\_aqua\_int[ lgnum = ls\_rsrc-lgnum  
                                     lgpla = cs\_zpspr1-source\_bin  
                                     matid = lv\_matid  
                                     charg = cs\_zpspr1-batch ].  
  
        ENDIF.  
      ENDIF.  
  
      IF ls\_aqua\_int IS NOT INITIAL.  
        cs\_zpspr1-avail\_quan    = ls\_aqua\_int-quan.  
        cs\_zpspr1-opunit        = ls\_aqua\_int-opunit.  
        cs\_zpspr1-uom           = ls\_aqua\_int-unit.  
        cs\_zpspr1-matid         = ls\_aqua\_int-matid.  
        cs\_zpspr1-batchid       = ls\_aqua\_int-batchid.  
        cs\_zpspr1-owner         = ls\_aqua\_int-owner.  
        cs\_zpspr1-entitled      = ls\_aqua\_int-entitled.  
        cs\_zpspr1-guid\_stock    = ls\_aqua\_int-guid\_stock.  
        cs\_zpspr1-lgtyp         = ls\_aqua\_int-lgtyp.  
        cs\_zpspr1-owner\_role    = ls\_aqua\_int-owner\_role.  
        cs\_zpspr1-entitled\_role = ls\_aqua\_int-entitled\_role.  
        cs\_zpspr1-stock\_type    = ls\_aqua\_int-cat.  
        cs\_zpspr1-guid\_parent   = ls\_aqua\_int-guid\_parent.  
        cs\_zpspr1-wdatu         = ls\_aqua\_int-wdatu.  
  
        /scwm/cl\_rf\_bll\_srvc=>set\_screlm\_input\_on( 'ZEWM\_S\_ZPSPR1-SOURCE\_QUAN' ).  
      ENDIF.  
  
\*--------------------------------------------------------  
  
    WHEN 'ZCRWTS'.  
  
      IF cs\_zpspr1-source\_quan IS INITIAL.  
        MESSAGE e024(zewm01).  
      ENDIF.  
  
      IF cs\_zpspr1-avail\_quan IS INITIAL.  
        MESSAGE e026(zewm01).  
      ENDIF.  
  
      IF gv\_batch\_req EQ abap\_true.  
        IF cs\_zpspr1-batch IS INITIAL.  
          MESSAGE e028(zewm01).  
        ENDIF.  
      ENDIF.  
  
\*--------------------------------------------------------  
\*Create and confirm WT Source bin to resource  
\*--------------------------------------------------------  
  
      CALL FUNCTION '/SCWM/RSRC\_RESOURCE\_MEMORY'  
        EXPORTING  
          iv\_uname = sy-uname  
        CHANGING  
          cs\_rsrc  = ls\_rsrc.  
  
\*Read storage bin configuration  
      SELECT SINGLE \*  
      FROM zewm\_t\_bin\_conf  
      INTO @DATA(ls\_bin\_conf)  
      WHERE warehouse\_no EQ @ls\_rsrc-lgnum.  
      IF sy-subrc EQ 0.  
  
        IF ls\_bin\_conf-dest\_bin IS INITIAL.  
          MESSAGE e029(zewm01) WITH ls\_rsrc-lgnum.  
        ENDIF.  
  
        IF ls\_bin\_conf-process\_type IS INITIAL.  
          MESSAGE e030(zewm01) WITH ls\_rsrc-lgnum.  
        ENDIF.  
  
        cs\_zpspr1-dest\_bin     = ls\_bin\_conf-dest\_bin.  
        cs\_zpspr1-process\_type = ls\_bin\_conf-process\_type.  
      ELSE.  
        MESSAGE e029(zewm01) WITH ls\_rsrc-lgnum.  
      ENDIF.  
  
      PERFORM wt\_crea\_src\_resource CHANGING cs\_zpspr1.  
  
\*--------------------------------------------------------  
\*Confirm WT To Final Bin  
\*--------------------------------------------------------  
      CALL FUNCTION '/SCWM/RF\_PRINT\_GLOBAL\_DATA'.  
  
      CALL FUNCTION '/SCWM/RSRC\_RESOURCE\_MEMORY'  
        EXPORTING  
          iv\_uname = sy-uname  
        CHANGING  
          cs\_rsrc  = ls\_rsrc.  
  
      CALL FUNCTION '/SCWM/TO\_READ\_SINGLE'  
        EXPORTING  
          iv\_lgnum   = ls\_rsrc-lgnum  
          iv\_tanum   = cs\_zpspr1-task\_number  
        IMPORTING  
          es\_ordim\_o = ls\_ordim\_o  
          et\_ordim\_c = lt\_ordim\_c  
        EXCEPTIONS  
          OTHERS     = 5.  
      IF sy-subrc = 0.  
        lv\_papos = lines( lt\_ordim\_c ) + 1.  
      ELSE.  
        lv\_papos = 1.  
      ENDIF.  
  
      lt\_conf = VALUE #( ( tanum      = cs\_zpspr1-task\_number  
                           nista      = cs\_zpspr1-source\_quan  
                           altme      = cs\_zpspr1-uom  
                           nlpla      = cs\_zpspr1-dest\_bin  
                           started\_at = cs\_zpspr1-started\_at  
                           papos      = lv\_papos  
                           wdatu      = cs\_zpspr1-wdatu ) ).  
  
\*Update WHO with resource and processor and start bin  
      CALL FUNCTION '/SCWM/RF\_WHO\_UPD'  
        EXPORTING  
          iv\_lgnum = ls\_rsrc-lgnum  
          iv\_who   = ls\_ordim\_o-who.  
  
\*Simulate TO confirmation  
      CLEAR:lv\_severity,lt\_bapiret.  
      CALL FUNCTION '/SCWM/TO\_CONF\_INT\_SIMULATE'  
        EXPORTING  
          iv\_lgnum                 = ls\_rsrc-lgnum  
          iv\_catch\_weight\_no\_check = abap\_true  
          it\_conf                  = lt\_conf  
          it\_conf\_exc              = lt\_conf\_exc  
          it\_conf\_serid            = lt\_conf\_serid  
        IMPORTING  
          et\_ltap\_vb               = lt\_ltap\_vb  
          et\_bapiret               = lt\_bapiret  
          ev\_severity              = lv\_severity  
          ev\_ok                    = lv\_ok.  
  
      IF lv\_severity CA 'EAX'.  
        LOOP AT lt\_bapiret INTO DATA(ls\_bapiret).  
          IF ls\_bapiret-type CA wmegc\_severity\_ea.  
            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.  
            EXIT.  
          ENDIF.  
        ENDLOOP.  
      ENDIF.  
  
      IF lv\_ok = abap\_true.  
  
        CLEAR:lv\_severity,lt\_bapiret.  
        CALL FUNCTION '/SCWM/TO\_CONFIRM'  
          EXPORTING  
            iv\_commit\_work   = ' '  
            iv\_update\_task   = ' '  
            iv\_lgnum         = ls\_rsrc-lgnum  
            it\_conf          = lt\_conf  
            it\_conf\_exc      = lt\_conf\_exc  
            it\_conf\_serid    = lt\_conf\_serid  
            iv\_processor\_det = 'X'  
          IMPORTING  
            et\_bapiret       = lt\_bapiret  
            ev\_severity      = lv\_severity.  
  
        IF lv\_severity CA 'EAX'.  
          LOOP AT lt\_bapiret INTO ls\_bapiret.  
            IF ls\_bapiret-type CA wmegc\_severity\_ea.  
              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.  
              EXIT.  
            ENDIF.  
          ENDLOOP.  
        ENDIF.  
  
        COMMIT WORK AND WAIT.  
        IF sy-subrc NE 0.  
          MESSAGE e020(zewm01).  
        ENDIF.  
  
      ENDIF.  
  
      CLEAR gv\_batch\_req.  
      /scwm/cl\_rf\_bll\_srvc=>set\_prmod( '1' ).  
      /scwm/cl\_rf\_bll\_srvc=>set\_fcode( /scwm/cl\_rf\_bll\_srvc=>c\_fcode\_compl\_ltrans ).  
  
    WHEN 'ZRESET'.  
      CLEAR cs\_zpspr1-source\_quan.  
  
    WHEN OTHERS.  
  ENDCASE.  
  
ENDFUNCTION.

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

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

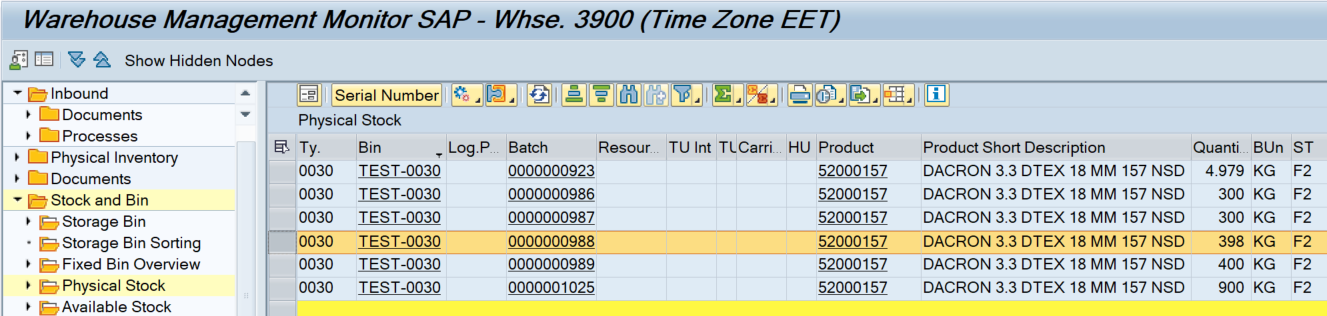
\*----------------------------------------------------------------------\*  
\*\*\*INCLUDE LZEWM\_FG\_RF\_DEVF01.  
\*----------------------------------------------------------------------\*  
\*&---------------------------------------------------------------------\*  
\*& Form who\_rsrc\_allowed  
\*&---------------------------------------------------------------------\*  
\*& text  
\*&---------------------------------------------------------------------\*  
\*&      --> LGNUM  
\*&      --> LS\_LTAP\_VB\_WHO  
\*&      --> LS\_LTAP\_VB\_QUEUE  
\*&      <-- LV\_OK  
\*&---------------------------------------------------------------------\*  
FORM who\_rsrc\_allowed USING iv\_lgnum    TYPE /scwm/lgnum  
                            iv\_who      TYPE /scwm/de\_who  
                            iv\_queue    TYPE /scwm/de\_queue  
                      CHANGING ev\_ok TYPE xfeld.  
  
  DATA:ls\_rsrc    TYPE /scwm/rsrc,  
       lt\_wo\_rsrc TYPE /scwm/tt\_wo\_rsrc\_ty,  
       ls\_ltap\_vb TYPE /scwm/ltap.  
  
  CLEAR ev\_ok.  
  
  CALL FUNCTION '/SCWM/RSRC\_RESOURCE\_MEMORY'  
    EXPORTING  
      iv\_uname = sy-uname  
    CHANGING  
      cs\_rsrc  = ls\_rsrc.  
  
  ls\_rsrc-rfind = abap\_true.  
  
  APPEND INITIAL LINE TO lt\_wo\_rsrc ASSIGNING FIELD-SYMBOL(<lfs\_wo\_rsrc>).  
  IF <lfs\_wo\_rsrc> IS ASSIGNED.  
    <lfs\_wo\_rsrc>-lgnum = iv\_lgnum.  
    <lfs\_wo\_rsrc>-queue = iv\_queue.  
    <lfs\_wo\_rsrc>-who   = iv\_who.  
  ENDIF.  
  
  CALL FUNCTION '/SCWM/RSRC\_QUALIF\_QUEUE\_CHECK'  
    EXPORTING  
      iv\_who        = iv\_who  
    CHANGING  
      cs\_rsrc       = ls\_rsrc  
      ct\_wo\_rsrc\_ty = lt\_wo\_rsrc.  
  
  IF line\_exists( lt\_wo\_rsrc[ who = iv\_who ] ).  
    ev\_ok = abap\_true.  
  ENDIF.  
  
ENDFORM.  
\*&---------------------------------------------------------------------\*  
\*& Form wt\_crea\_src\_resource  
\*&---------------------------------------------------------------------\*  
\*& text  
\*&---------------------------------------------------------------------\*  
\*&      --> CS\_ZPSPR1  
\*&---------------------------------------------------------------------\*  
FORM wt\_crea\_src\_resource CHANGING cs\_zpspr1 TYPE zewm\_s\_zpspr1.  
  
  DATA:lv\_tanum    TYPE /scwm/tanum,  
       lv\_severity TYPE bapi\_mtype,  
       lv\_ok       TYPE xfeld,  
       lv\_huent    TYPE /scwm/ltap\_hu\_huent.  
  
  DATA:ls\_rsrc    TYPE /scwm/rsrc,  
       ls\_ordim\_o TYPE /scwm/ordim\_o,  
       ls\_t331    TYPE /scwm/t331.  
  
  DATA:lt\_create     TYPE /scwm/tt\_to\_create\_int,  
       lt\_ltap\_vb    TYPE /scwm/tt\_ltap\_vb,  
       lt\_bapiret    TYPE bapiret2\_t,  
       lt\_conf       TYPE /scwm/to\_conf\_tt,  
       lt\_conf\_serid TYPE /scwm/tt\_conf\_serid,  
       lt\_ordim\_o    TYPE /scwm/tt\_ordim\_o.  
  
  CALL FUNCTION '/SCWM/RSRC\_RESOURCE\_MEMORY'  
    EXPORTING  
      iv\_uname = sy-uname  
    CHANGING  
      cs\_rsrc  = ls\_rsrc.  
  
  CALL FUNCTION '/SCWM/T331\_READ\_SINGLE'  
    EXPORTING  
      iv\_lgnum  = ls\_rsrc-lgnum  
      iv\_lgtyp  = cs\_zpspr1-lgtyp  
    IMPORTING  
      es\_t331   = ls\_t331  
    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.  
  
\*Check against the indicator for available quantity (Bin or HU level)  
  IF cs\_zpspr1-quantum > 1 AND ls\_t331-avqlvl NE wmegc\_avqlvl\_bin.  
    DATA(lv\_guid\_parent) = cs\_zpspr1-guid\_parent.  
  ENDIF.  
  
  lt\_create = VALUE #( ( procty        = cs\_zpspr1-process\_type  
                         matid         = cs\_zpspr1-matid  
                         batchid       = cs\_zpspr1-batchid  
                         cat           = cs\_zpspr1-stock\_type  
                         anfme         = cs\_zpspr1-source\_quan  
                         nlpla         = cs\_zpspr1-dest\_bin  
                         single\_to     = abap\_true  
                         seqno         = 1  
                         altme         = cs\_zpspr1-uom  
                         opunit        = cs\_zpspr1-opunit  
                         kompl         = abap\_true  
                         no\_stock\_det  = abap\_true  
                         owner         = cs\_zpspr1-owner  
                         owner\_role    = cs\_zpspr1-owner\_role  
                         entitled      = cs\_zpspr1-entitled  
                         entitled\_role = cs\_zpspr1-entitled\_role  
                         guid\_stock    = cs\_zpspr1-guid\_stock  
                         vltyp         = cs\_zpspr1-lgtyp  
                         vlpla         = cs\_zpspr1-source\_bin  
                         sguid\_hu      = lv\_guid\_parent ) ).  
  
  CALL FUNCTION '/SCWM/TO\_CREATE'  
    EXPORTING  
      iv\_lgnum       = ls\_rsrc-lgnum  
      it\_create      = lt\_create  
      iv\_wtcode      = wmegc\_wtcode\_adhoc\_prod  
      iv\_update\_task = ' '  
      iv\_commit\_work = ' '  
    IMPORTING  
      ev\_tanum       = lv\_tanum  
      et\_ltap\_vb     = lt\_ltap\_vb  
      et\_bapiret     = lt\_bapiret  
      ev\_severity    = lv\_severity.  
  
  IF lv\_severity CA 'EAX'.  
    LOOP AT lt\_bapiret INTO DATA(ls\_bapiret).  
      IF ls\_bapiret-type CA wmegc\_severity\_ea.  
        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.  
        EXIT.  
      ENDIF.  
    ENDLOOP.  
  ENDIF.  
  
  cs\_zpspr1-task\_number = lv\_tanum.  
  
  COMMIT WORK AND WAIT.  
  IF sy-subrc IS NOT INITIAL.  
    MESSAGE e018(zewm01).  
  ENDIF.  
  
  CLEAR lv\_ok.  
  IF line\_exists( lt\_ltap\_vb[ tanum = lv\_tanum ] ).  
    DATA(ls\_ltap\_vb) = lt\_ltap\_vb[ tanum = lv\_tanum ].  
    cs\_zpspr1-started\_at = ls\_ltap\_vb-created\_at.  
  
    PERFORM who\_rsrc\_allowed USING    ls\_rsrc-lgnum  
                                      ls\_ltap\_vb-who  
                                      ls\_ltap\_vb-queue  
                             CHANGING lv\_ok.  
    IF lv\_ok IS INITIAL.  
      MESSAGE e019(zewm01).  
    ENDIF.  
  ENDIF.  
  
  CALL FUNCTION '/SCWM/REC\_WT\_CHANGE'  
    EXPORTING  
      it\_ltap\_vb        = lt\_ltap\_vb  
    EXCEPTIONS  
      no\_free\_wt\_in\_who = 1  
      internal\_error    = 2  
      OTHERS            = 3.  
  IF sy-subrc <> 0.  
    MESSAGE ID sy-msgid TYPE sy-msgty NUMBER sy-msgno WITH sy-msgv1 sy-msgv2 sy-msgv3 sy-msgv4.  
  ENDIF.  
  
\*---------------------------------------------------------  
\*Confirm TO on resource  
\*---------------------------------------------------------  
  IF ls\_ltap\_vb IS NOT INITIAL.  
    MOVE-CORRESPONDING ls\_ltap\_vb TO ls\_ordim\_o.  
  ENDIF.  
  
  CALL FUNCTION '/SCWM/HUENT\_DET'  
    EXPORTING  
      is\_ordim\_o = ls\_ordim\_o  
    IMPORTING  
      ev\_huent   = lv\_huent  
    EXCEPTIONS  
      wrong\_data = 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.  
  
  lt\_conf = VALUE #( ( tanum = lv\_tanum  
                       squit = abap\_true  
                       altme = cs\_zpspr1-uom  
                       drsrc = ls\_rsrc-rsrc  
                       huent = lv\_huent ) ).  
  
\*Update WHO with resource and processor and start bin  
  CALL FUNCTION '/SCWM/RF\_WHO\_UPD'  
    EXPORTING  
      iv\_lgnum = ls\_rsrc-lgnum  
      iv\_who   = ls\_ltap\_vb-who.  
  
  CLEAR:lv\_severity,lt\_bapiret.  
  CALL FUNCTION '/SCWM/TO\_CONFIRM'  
    EXPORTING  
      iv\_lgnum         = ls\_rsrc-lgnum  
      iv\_wtcode        = wmegc\_wtcode\_rsrc  
      it\_conf          = lt\_conf  
      it\_conf\_serid    = lt\_conf\_serid  
      iv\_commit\_work   = ' '  
      iv\_update\_task   = ' '  
      iv\_processor\_det = 'X'  
    IMPORTING  
      et\_ltap\_vb       = lt\_ltap\_vb  
      et\_bapiret       = lt\_bapiret  
      ev\_severity      = lv\_severity.  
  
  IF lv\_severity CA 'EAX'.  
    LOOP AT lt\_bapiret INTO ls\_bapiret.  
      IF ls\_bapiret-type CA wmegc\_severity\_ea.  
        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.  
        EXIT.  
      ENDIF.  
    ENDLOOP.  
  ENDIF.  
  
  COMMIT WORK AND WAIT.  
  IF sy-subrc IS NOT INITIAL.  
    MESSAGE e020(zewm01).  
  ENDIF.  
  
\*Determination of new TO number  
  CALL FUNCTION '/SCWM/TO\_READ\_WHO'  
    EXPORTING  
      iv\_lgnum     = ls\_rsrc-lgnum  
      iv\_who       = ls\_ltap\_vb-who  
    IMPORTING  
      et\_ordim\_o   = lt\_ordim\_o  
    EXCEPTIONS  
      wrong\_input  = 1  
      not\_found    = 2  
      foreign\_lock = 3  
      error        = 4  
      OTHERS       = 5.  
  IF sy-subrc IS NOT INITIAL.  
    MESSAGE ID sy-msgid TYPE sy-msgty NUMBER sy-msgno WITH sy-msgv1 sy-msgv2 sy-msgv3 sy-msgv4.  
  ENDIF.  
  
\*Call Resource Execution Control  
  IF lt\_ordim\_o IS NOT INITIAL.  
  
    IF line\_exists( lt\_ordim\_o[ who = ls\_ltap\_vb-who ] ).  
      ls\_ordim\_o = lt\_ordim\_o[ who = ls\_ltap\_vb-who ].  
    ENDIF.  
  
    CALL FUNCTION '/SCWM/REC\_WT\_CHANGE'  
      CHANGING  
        cs\_ordim\_o        = ls\_ordim\_o  
        ct\_ordim\_o        = lt\_ordim\_o  
      EXCEPTIONS  
        no\_free\_wt\_in\_who = 1  
        internal\_error    = 2  
        OTHERS            = 3.  
    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.  
  
ENDFORM.

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

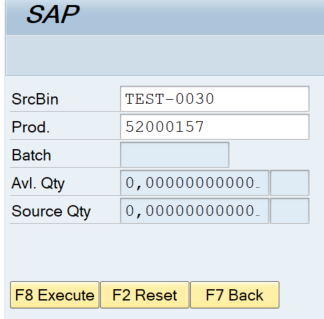
Unit test detail:

Perform PSA with Product/Batch.

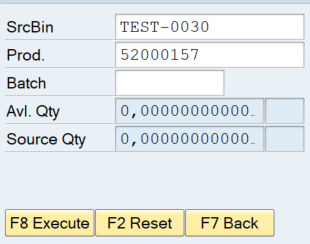
Before Quantity:



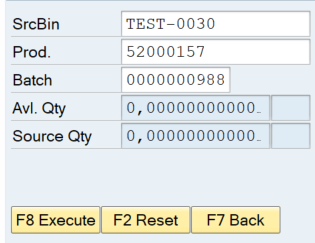
Click on Product / Batch button and enter Source Bin and Product.

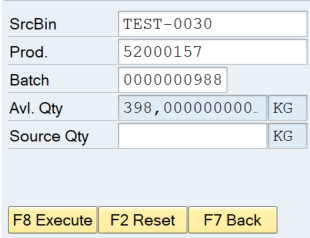


Press Enter to check if the product is batch managed then Batch field will change as input field.

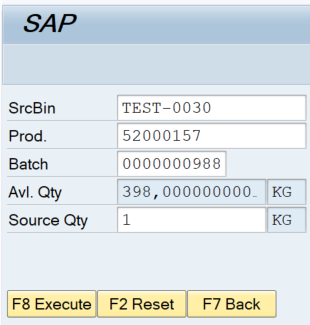


Input batch and press enter

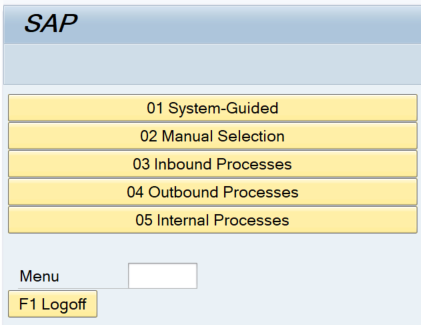




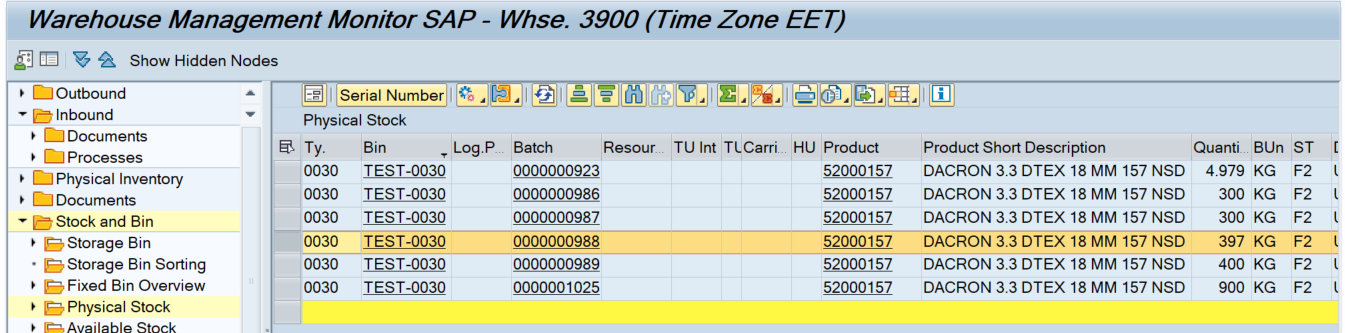
Now input the source quantity and press execute button to create the warehouse task for final bin movement



Same as SAP standard behavior, screen flow will come back to initial menu if successful.



After Quantity:



Go to /SCWM/MON transaction code and check the warehouse task for the movement to final bin.

