A screenshot of a computer

Description automatically generated

A screenshot of a computer

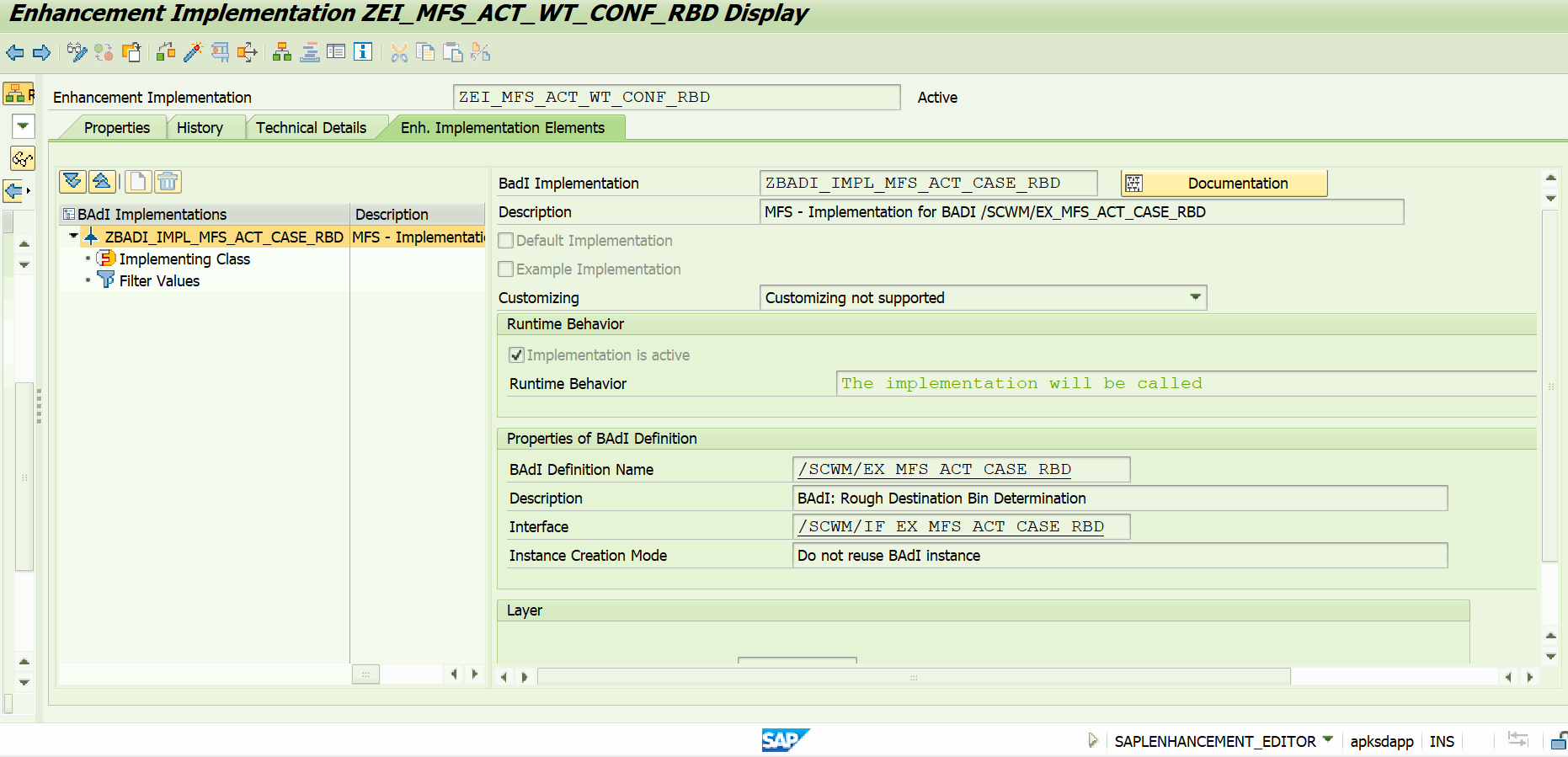
Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

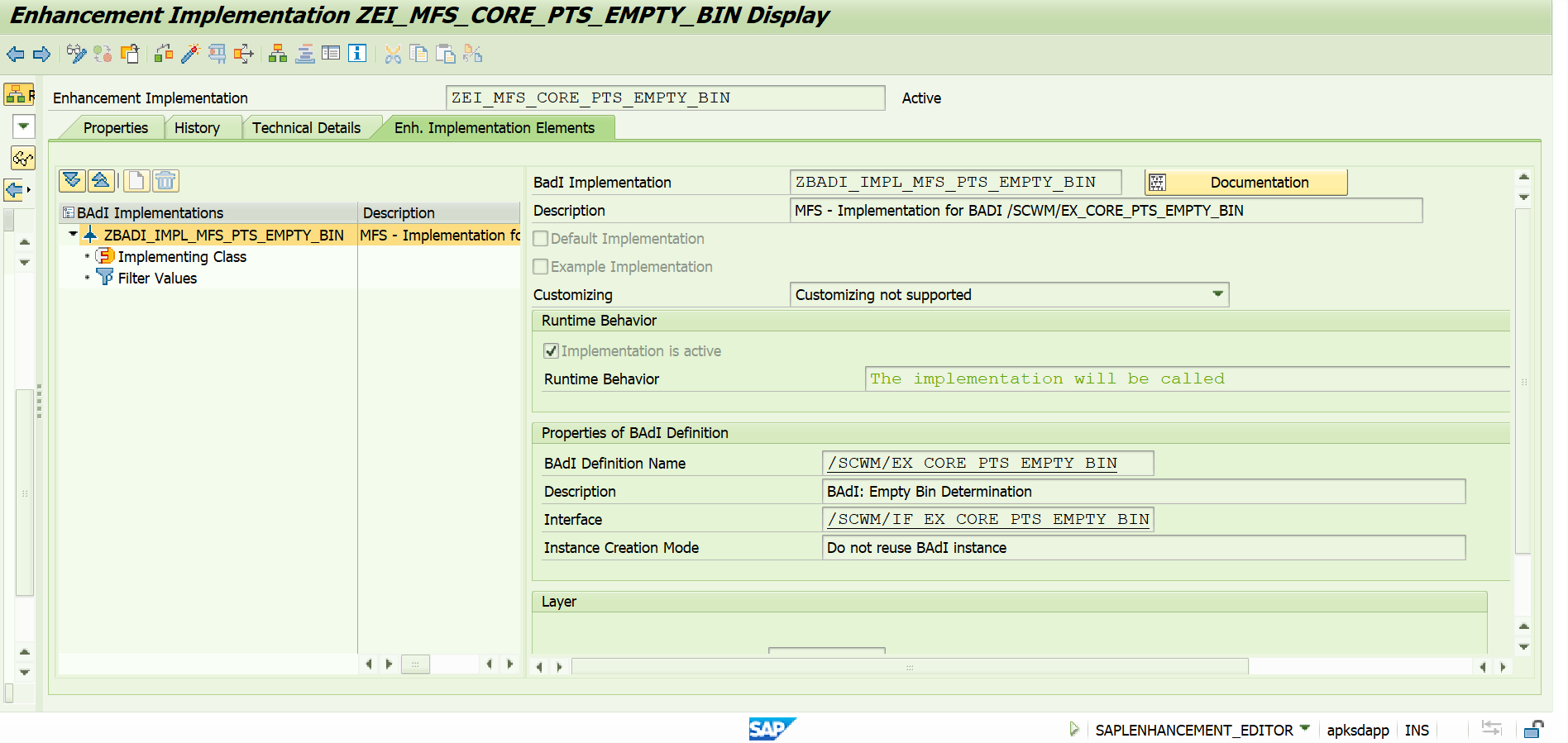
Description automatically generated



A screenshot of a computer

Description automatically generated

  METHOD /scwm/if\_ex\_mfs\_act\_case\_rbd~call\_rbd.  
\* This is the fallback implementation of the MFS-BAdI to determine  
\* the rough destination bin in the final storage type  
  
\* If you do not like this fallback implementation, please implement the  
\* BAdI on your own.  
\* You might reuse the the methods of class /scwm/cl\_core\_putstra or create  
\* your own class, which inherits from the class /scwm/cl\_core\_putstra.  
\* You can either redefine a public method or just some specific protetcted  
\* ones, if you like the general rough bin determination, but e.g. the  
\* sortation of aisles does not fit to your requirement.  
  
    DATA: lv\_no\_bin\_det\_reason TYPE /scwm/de\_rbd\_reason,  
          lv\_msgtext           TYPE bapi\_msg.               "#EC NEEDED  
  
\*    DATA: lo\_put\_stra  TYPE REF TO /scwm/cl\_core\_putstra.  
    DATA: lo\_put\_stra  TYPE REF TO zscwm\_cl\_core\_putstra.  
  
    gv\_rearr\_hu = cv\_no\_bin\_det\_behavior.  
  
    CLEAR: cv\_no\_bin\_det\_behavior, cs\_mfshumove.  
  
    TRY.  
        CREATE OBJECT lo\_put\_stra.  
  
        CALL METHOD lo\_put\_stra->init\_4\_rough\_destbin\_at\_adp  
          EXPORTING  
            iv\_lgnum       = iv\_lgnum  
            iv\_current\_plc = iv\_plc  
            iv\_current\_cp  = is\_tmfscp-cp.  
  
        CALL METHOD lo\_put\_stra->determine\_rough\_destbin\_at\_adp  
          EXPORTING  
            iv\_huident           = is\_telegram-huident  
          IMPORTING  
            es\_mfshumove         = cs\_mfshumove  
            ev\_no\_bin\_det\_reason = lv\_no\_bin\_det\_reason.  
  
        CASE lv\_no\_bin\_det\_reason.  
          WHEN wmegc\_rbd\_depth\_missing.  
\*           setup issue - bin depth  
            cv\_no\_bin\_det\_behavior = wmegc\_rbd\_behav\_hu2\_error.  
            MESSAGE w404(/scwm/mfs) INTO lv\_msgtext.  
            io\_log->add\_message( ip\_row = 0 ).  
          WHEN wmegc\_rbd\_undefined\_psv.  
\*           setup issue - put away strategy  
            cv\_no\_bin\_det\_behavior = wmegc\_rbd\_behav\_hu2\_error.  
            MESSAGE w415(/scwm/mfs) INTO lv\_msgtext.  
            io\_log->add\_message( ip\_row = 0 ).  
          WHEN wmegc\_rbd\_no\_lgber\_lptyp.  
\*           setup issue - storage section / storage bin type  
            cv\_no\_bin\_det\_behavior = wmegc\_rbd\_behav\_hu2\_error.  
            MESSAGE w416(/scwm/mfs) INTO lv\_msgtext.  
            io\_log->add\_message( ip\_row = 0 ).  
          WHEN wmegc\_rbd\_longterm\_block.  
\*           long term blocking issue  
            cv\_no\_bin\_det\_behavior = wmegc\_rbd\_behav\_hu2\_default.  
            MESSAGE w399(/scwm/mfs) INTO lv\_msgtext.  
            io\_log->add\_message( ip\_row = 0 ).  
          WHEN wmegc\_rbd\_no\_free\_pup\_capa.  
\*           no capacity at resource pick-up points  
            MESSAGE w279(/scwm/put\_bin\_det) INTO lv\_msgtext.  
            io\_log->add\_message( ip\_row = 0 ).  
            IF is\_tmfscp-retry\_wait IS NOT INITIAL.  
              cv\_no\_bin\_det\_behavior = wmegc\_rbd\_behav\_tele\_retry.  
            ELSE.  
              cv\_no\_bin\_det\_behavior = wmegc\_rbd\_behav\_hu2\_default.  
              MESSAGE w398(/scwm/mfs) WITH is\_tmfscp-retry\_wait  
                                      INTO lv\_msgtext.  
              io\_log->add\_message( ip\_row = 0 ).  
            ENDIF.  
          WHEN wmegc\_rbd\_cpgrp\_enqueue.  
\*           enqueue fails for CP-Group  
            MESSAGE w424(/scwm/mfs) WITH is\_tmfscp-cpgrp  
                                    INTO lv\_msgtext.  
            io\_log->add\_message( ip\_row = 0 ).  
            IF is\_tmfscp-retry\_wait IS NOT INITIAL.  
              cv\_no\_bin\_det\_behavior = wmegc\_rbd\_behav\_tele\_retry.  
            ELSE.  
              cv\_no\_bin\_det\_behavior = wmegc\_rbd\_behav\_hu2\_default.  
              MESSAGE w398(/scwm/mfs) WITH is\_tmfscp-retry\_wait  
                                      INTO lv\_msgtext.  
              io\_log->add\_message( ip\_row = 0 ).  
            ENDIF.  
          WHEN wmegc\_rbd\_no\_empty\_bin.  
\*           long term blocking issue  
            cv\_no\_bin\_det\_behavior = wmegc\_rbd\_behav\_hu2\_default.  
            MESSAGE w258(/scwm/put\_bin\_det) INTO lv\_msgtext.  
            io\_log->add\_message( ip\_row = 0 ).  
          WHEN OTHERS.  
        ENDCASE.  
  
      CATCH /scwm/cx\_core.  " configuration issue - set error  
        cv\_no\_bin\_det\_behavior = wmegc\_rbd\_behav\_tele\_error.  
        "caller will issue an error message for this behavior  
    ENDTRY.  
  
  ENDMETHOD.



A screenshot of a computer

Description automatically generated

  METHOD /scwm/if\_ex\_core\_pts\_empty\_bin~delete\_empty\_bin\_buffer.  
  
\*--------------------------------------------------------------------  
\* Data Definition  
\*--------------------------------------------------------------------  
    data: lv\_relevant      type xfeld.  
    DATA: lv\_delete\_buffer TYPE xfeld.  
    DATA: lo\_putstra       TYPE REF TO /scwm/cl\_core\_putstra.  
  
\*--------------------------------------------------------------------  
\* Program Logic  
\*--------------------------------------------------------------------  
\* 1) Check changing parameter, if allready set  
\* 2) Check storage type setting  
\*    - destination storage type settings  
\*       - HU requirement (HUOBL): ' ' - HU obligatory  
\*       - Storage Behaviour (BEHAV):  
\*         - ' ' - Standard Warehouse  
\*       - Putaway Rule (PUT\_RULE):  
\*         - '2' - Addition to Existing Stock/Empty Bin  
\*         - '5' - Empty Bin  
\*       - Use Fix bins (FIXBIN): No  
\* 3) Check scenario relevance, due to existance of min 1 entry in  
\*    new customizing  
\* 3) Initialize implementation logic  
\* 4) Determine Destination Bin depending on  
\*    - source storage bin is Communication Point OR  
\*      source-resource is a MFS Resource  
\*--------------------------------------------------------------------  
  
\* 1) Check changing parameter, if allready set  
\*--------------------------------------------------------------------  
  
\* 2) Check storage type setting  
\*--------------------------------------------------------------------  
    IF is\_t331-st\_role =  wmegc\_strole\_mfs\_ctrl AND  
       is\_t331-huobl   =  wmegc\_huobl\_obl       AND  
       is\_t331-behav   =  wmegc\_behav\_normal    AND  
       is\_t331-fixbin  =  wmegc\_nofbin.  
  
\* 3) Check scenario relevance, due to existance of min 1 entry in  
\*    new customizing  
\*--------------------------------------------------------------------  
      CALL FUNCTION '/SCWM/MFS\_AISLE\_LEVEL\_RELEVANT'  
        EXPORTING  
          iv\_lgnum    = iv\_lgnum  
        IMPORTING  
          ev\_relevant = lv\_relevant.  
  
      IF lv\_relevant IS NOT INITIAL.  
  
\* 3) Initialize implementation logic  
\*--------------------------------------------------------------------  
        CREATE OBJECT lo\_putstra.  
        TRY.  
  
            CALL METHOD lo\_putstra->init\_4\_destbin\_at\_wt\_crea  
              EXPORTING  
                iv\_lgnum   = iv\_lgnum  
                iv\_vlpla   = is\_ltap-vlpla  
                iv\_srsrc   = is\_ltap-srsrc  
                io\_log     = io\_log  
                iv\_log\_row = iv\_log\_row. "request line  
  
\* 4) Decide for deletion of empty bin buffer  
\*--------------------------------------------------------------------  
            CALL METHOD lo\_putstra->det\_ebin\_buf\_del\_at\_wt\_crea  
              EXPORTING  
                iv\_nltyp      = iv\_nltyp  
                is\_request    = is\_ltap  
                is\_t331       = is\_t331  
                it\_eb\_buffer  = it\_eb\_buffer  
              IMPORTING  
                ev\_del\_buffer = lv\_delete\_buffer.  
  
          CATCH /scwm/cx\_core.  
\*       not interested in error, go ahead  
        ENDTRY.  
  
        cv\_delete\_buffer = lv\_delete\_buffer.  
  
      ENDIF. "IF lv\_relevant IS NOT INITIAL.  
  
    ENDIF. "IF is\_t331-st\_role =  wmegc\_strole\_mfs\_ctrl AND  
  
  ENDMETHOD.

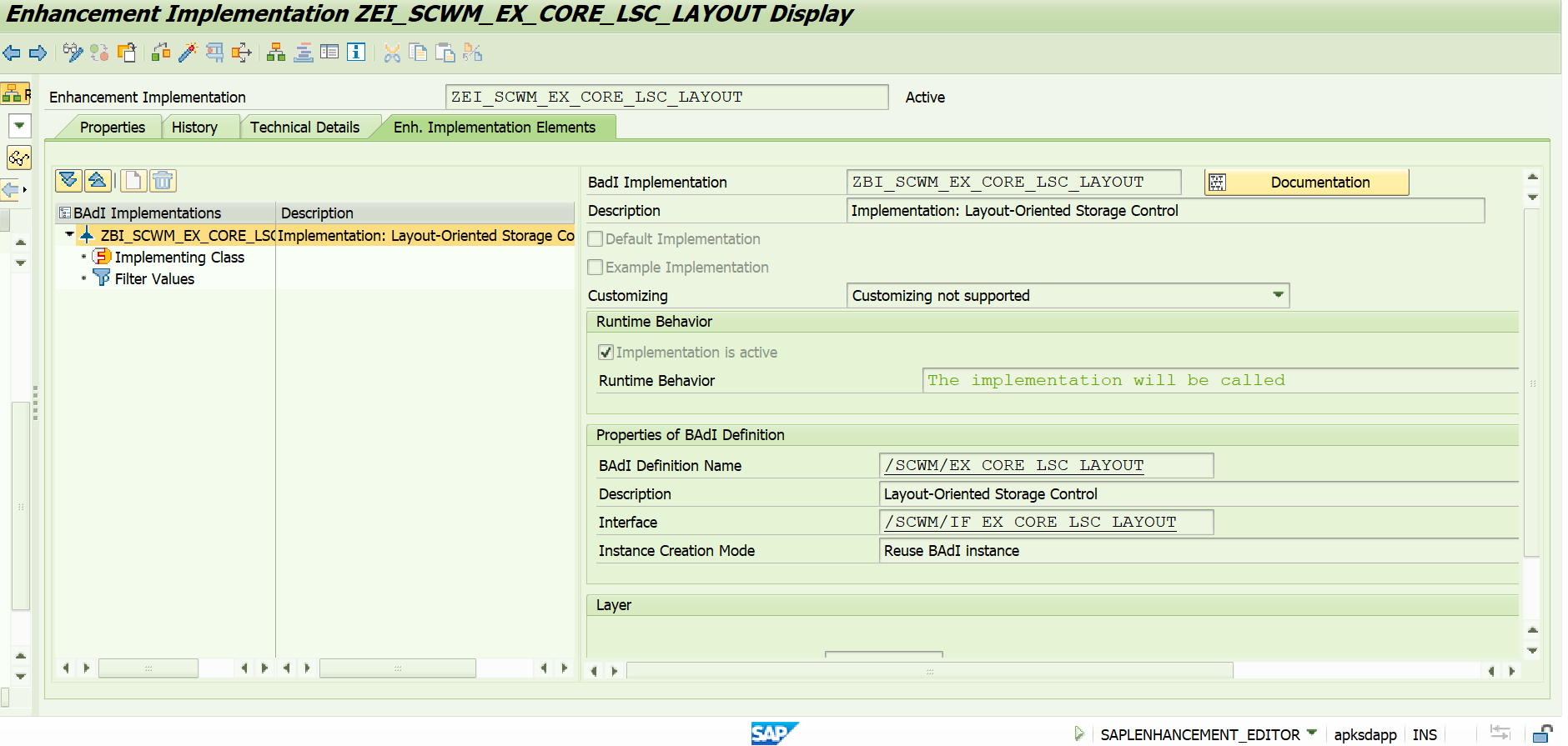
  METHOD /scwm/if\_ex\_core\_pts\_empty\_bin~determine\_empty\_bins.  
\*--------------------------------------------------------------------  
\* Data Definition  
\*--------------------------------------------------------------------  
    DATA: lv\_relevant TYPE xfeld.  
    DATA: lt\_lagpl    TYPE /scwm/tt\_lagpl.  
    DATA: lo\_putstra  TYPE REF TO zscwm\_cl\_core\_putstra.  "/scwm/cl\_core\_putstra.  
  
\*--------------------------------------------------------------------  
\* Program Logic  
\*--------------------------------------------------------------------  
\* 1) Check changing parameters are empty  
\* 2) Check storage type settings, if BAdI Implementation is relevant  
\* 3) Check scenario relevance, due to existance of min 1 entry in  
\*    new customizing  
\* 4) Search for empty bins according:  
\*    - reachable Storage Type/Aisle/Level combination for given  
\*      CP (Resource Pick-up Point) or Resource  
\*    - with special DB Index for faster search  
\* 5) Decide, if standard empty bin search should run  
\*  
\* A T T E N T I O N:  
\* Activate the DB index "N - empty bins for aisles/level"  
\* on DB Table /SCWM/LAGP to achieve faster response times.  
\* This index is not "active" by default  
\*--------------------------------------------------------------------  
  
\* 1) Check changing parameters are empty  
\*--------------------------------------------------------------------  
    IF ct\_lagpl[] IS NOT INITIAL.  
      RETURN.  
    ENDIF.  
  
\* 2) Check storage type settings, if BAdI Implementation is relevant  
\*--------------------------------------------------------------------  
    IF is\_t331-st\_role =  wmegc\_strole\_mfs\_ctrl AND  
       is\_t331-huobl   =  wmegc\_huobl\_obl       AND  
       is\_t331-behav   =  wmegc\_behav\_normal    AND  
       is\_t331-avqlvl  =  wmegc\_avqlvl\_hu       AND  
       is\_t331-fixbin  =  wmegc\_nofbin.  
  
\* 3) Check scenario relevance, due to existance of min 1 entry in  
\*    new customizing  
\*--------------------------------------------------------------------  
      CALL FUNCTION '/SCWM/MFS\_AISLE\_LEVEL\_RELEVANT'  
        EXPORTING  
          iv\_lgnum    = iv\_lgnum  
        IMPORTING  
          ev\_relevant = lv\_relevant.  
  
      IF lv\_relevant IS NOT INITIAL.  
  
\* 5) Search for empty bins  
\*--------------------------------------------------------------------  
        CREATE OBJECT lo\_putstra.  
        TRY.  
            CALL METHOD lo\_putstra->init\_4\_destbin\_at\_wt\_crea  
              EXPORTING  
                iv\_lgnum   = iv\_lgnum  
                iv\_vlpla   = is\_ltap-vlpla  
                iv\_srsrc   = is\_ltap-srsrc  
                io\_log     = io\_log  
                iv\_log\_row = iv\_log\_row. "request line  
  
\*            CALL METHOD lo\_putstra->det\_empty\_bin\_at\_wt\_crea  
\*              EXPORTING  
\*                iv\_nltyp     = iv\_nltyp  
\*                iv\_nlber     = iv\_nlber  
\*                iv\_nptyp     = iv\_nptyp  
\*                is\_request   = is\_ltap  
\*                is\_t331      = is\_t331  
\*                it\_lagp\_excl = it\_lagp\_excl  
\*              IMPORTING  
\*                et\_lagpl     = lt\_lagpl.  
            CALL METHOD lo\_putstra->zdet\_empty\_bin\_at\_wt\_crea\_khfg  
              EXPORTING  
                iv\_nltyp     = iv\_nltyp  
                iv\_nlber     = iv\_nlber  
                iv\_nptyp     = iv\_nptyp  
                is\_request   = is\_ltap  
                is\_t331      = is\_t331  
                it\_lagp\_excl = it\_lagp\_excl  
              IMPORTING  
                et\_lagpl     = lt\_lagpl.  
          CATCH /scwm/cx\_core.  
\*           if error occurs, run the functionality < EWM 9.2  
            cv\_do\_not\_run\_std = abap\_false.  
            RETURN.  
        ENDTRY.  
  
\* 6) Decide, if standard empty bin search should run  
\*--------------------------------------------------------------------  
        IF lt\_lagpl[] IS NOT INITIAL.  
          APPEND LINES OF lt\_lagpl TO ct\_lagpl.  
        ENDIF.  
\*         if no bin was found with our own logic, then we also do NOT want to  
\*         run the standard empty bin determination, because this would  
\*         not consider the aisles/levels....  
        cv\_do\_not\_run\_std = abap\_true.  
  
      ENDIF. "IF lv\_relevant IS NOT INITIAL.  
    ENDIF. "IF is\_t331-st\_role =  wmegc\_strole\_mfs\_ctrl AND...  
  
  ENDMETHOD.



A screenshot of a computer

Description automatically generated

  METHOD /scwm/if\_ex\_core\_pts\_md\_addbin~determine\_bin.  
\*--------------------------------------------------------------------  
\* Data Definition  
\*--------------------------------------------------------------------  
    DATA: lo\_putstra TYPE REF TO /scwm/cl\_core\_putstra.  
  
\*--------------------------------------------------------------------  
\* Program Logic  
\*--------------------------------------------------------------------  
\* 1) Only determine destination bin, if destination bin is empty  
\* 2) Check relevance via destination storage type settings  
\*   - Role (ST\_ROLE): 'J' - Automatic Warehouse (MFS controlled)  
\*   - HU requirement (HUOBL): ' ' - HU obligatory  
\*   - Storage Behaviour (BEHAV):  
\*     - ' ' - Standard Warehouse  
\*   - Putaway Rule (PUT\_RULE):  
\*     - '2' - Addition to Existing Stock/Empty Bin  
\*             Addition of HU to a bin, where HU is already stored is only  
\*             done, if storage type is set to Multi-depth (MULTIDEEP)  
\*   - Level of Available Quantity (AVQLVL)  
\*     - ' ' - Highest Level HU  
\*   - Use Fix bins (FIXBIN): No  
\*   - Level of Addition to Stock (ADD\_LEVEL): ' ' - Bin  
\* 3) Initialize implementation logic  
\* 4) Determine Destination Bin depending on  
\*    - source storage bin is Communication Point OR  
\*      source-resource is a MFS Resource  
\*--------------------------------------------------------------------  
  
\* 1) Only determine destination bin, if destination bin is empty  
\*--------------------------------------------------------------------  
    IF cv\_nlpla IS NOT INITIAL.  
      RETURN.  
    ENDIF.  
  
\* 2) Check relevance via storage type settings  
\*--------------------------------------------------------------------  
    IF is\_t331-st\_role   = wmegc\_strole\_mfs\_ctrl AND  
       is\_t331-huobl     = wmegc\_huobl\_obl       AND  
       is\_t331-behav     = wmegc\_behav\_normal    AND  
       is\_t331-avqlvl    = wmegc\_avqlvl\_hu       AND  
       is\_t331-add\_level = wmegc\_add\_to\_bin      AND  
       is\_t331-fixbin    = wmegc\_nofbin.  
  
\* 3) Initialize implementation logic  
\*--------------------------------------------------------------------  
      CREATE OBJECT lo\_putstra.  
      TRY.  
          CALL METHOD lo\_putstra->init\_4\_destbin\_at\_wt\_crea  
            EXPORTING  
              iv\_lgnum   = iv\_lgnum  
              iv\_vlpla   = is\_ltap-vlpla  
              iv\_srsrc   = is\_ltap-srsrc  
              io\_log     = io\_log  
              iv\_log\_row = iv\_log\_row. "request line  
  
\* 4) Determine Destination Bin  
\*--------------------------------------------------------------------  
          CALL METHOD lo\_putstra->det\_partly\_occ\_bin\_at\_wt\_crea  
            EXPORTING  
              iv\_nltyp     = iv\_nltyp  
              iv\_nlber     = iv\_nlber  
              iv\_lptyp     = iv\_nptyp  
              is\_request   = is\_ltap  
              it\_lagp\_excl = it\_lagp\_excl  
            IMPORTING  
              ev\_nlpla     = cv\_nlpla.  
  
        CATCH /scwm/cx\_core ##NO\_HANDLER .  
\*       we are not interested in errors, as there might be an  
\*       empty bin found  
      ENDTRY.  
  
    ENDIF. "IF is\_t331-st\_role =  wmegc\_strole\_mfs\_ctrl AND..  
  
    IF is\_ltap-procty = '3090' AND is\_ltap-act\_type = 'PTWY'.  
      DATA:lx\_selcrit TYPE /scwm/s\_to\_selcrit\_mon,  
           lt\_to      TYPE /scwm/tt\_to\_det\_mon.  
  
      INSERT VALUE #( sign   = wmegc\_sign\_inclusive  
                      option = wmegc\_option\_eq  
                      low    =  cv\_nlpla ) INTO TABLE lx\_selcrit-r\_nlpla.  
  
      INSERT VALUE #( sign   = wmegc\_sign\_inclusive  
                      option = wmegc\_option\_eq  
                      low    = iv\_nltyp ) INTO TABLE lx\_selcrit-r\_lgtyp.  
  
  
      CALL FUNCTION '/SCWM/TO\_GET\_WIP'  
        EXPORTING  
          iv\_lgnum   = iv\_lgnum  
          iv\_open    = abap\_true  
          is\_selcrit = lx\_selcrit  
        IMPORTING  
          et\_to      = lt\_to.  
  
      IF lt\_to IS NOT INITIAL.  
  
        READ TABLE lt\_to ASSIGNING FIELD-SYMBOL(<fs\_to>) WITH KEY dlogpos = '02'.  
        IF sy-subrc = 0.  
          cv\_dlogpos = '01'.  
        ENDIF.  
  
      ENDIF.  
    ENDIF.  
  
  ENDMETHOD.

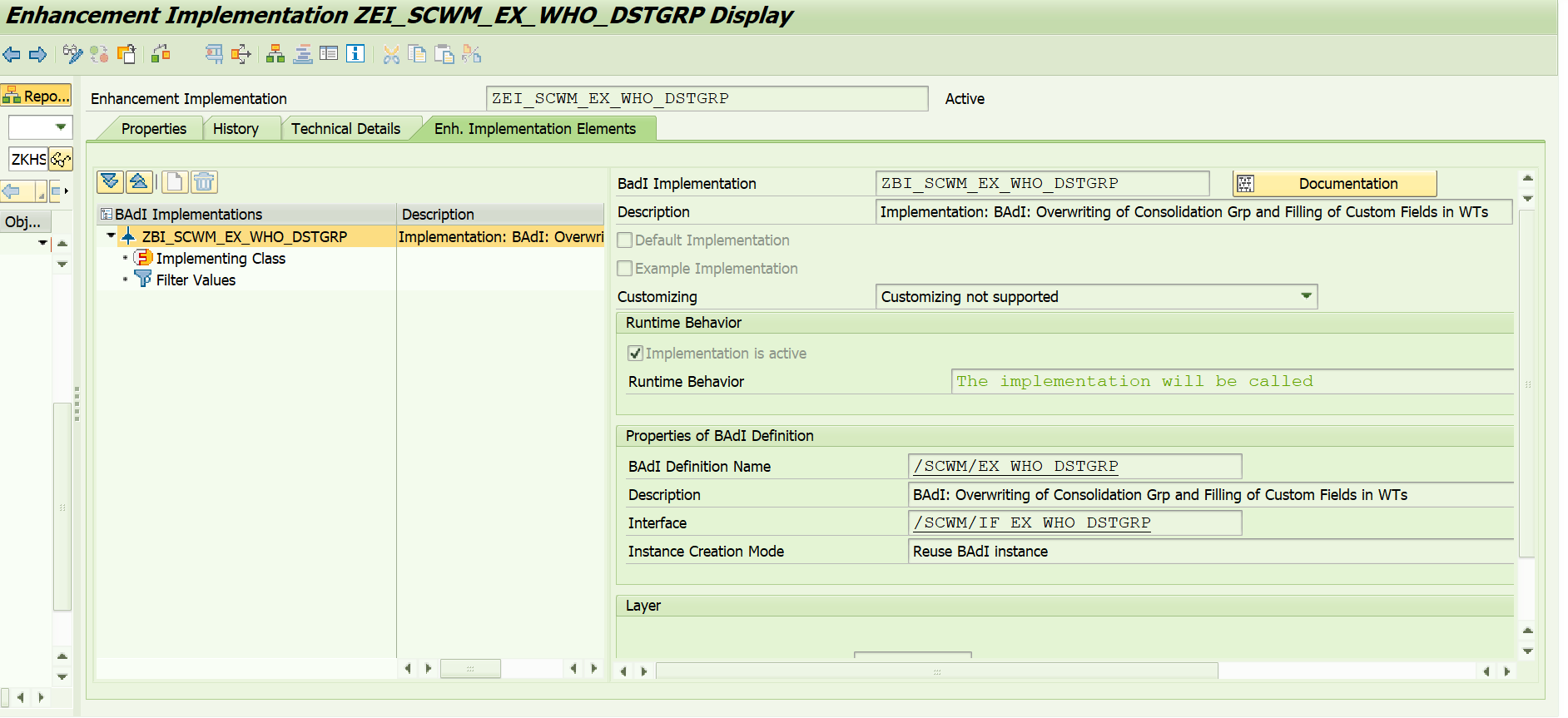


A screenshot of a computer

Description automatically generated

  METHOD /scwm/if\_ex\_core\_lsc\_layout~layout.  
  
\* Determine Daifuku Sorting station  
\* and Distribute Deliveries across Active Stations  
    determine\_dfku\_sortstation(  
      EXPORTING  
        iv\_lgnum   = iv\_lgnum  
        iv\_vltyp   = iv\_vltyp  
        io\_log     = io\_log  
        is\_ordim\_o = is\_ordim\_o  
      CHANGING  
        cv\_iltyp   = cv\_iltyp  
        cv\_ilber   = cv\_ilber  
        cv\_ilpla   = cv\_ilpla ).  
  
\*Restricting LOSC for full pallet merging task.  
    restrict\_taskcrea\_leftsec(  
          EXPORTING  
            iv\_lgnum   = iv\_lgnum  
            iv\_vltyp   = iv\_vltyp  
            io\_log     = io\_log  
            is\_ordim\_o = is\_ordim\_o  
          CHANGING  
            cv\_iltyp   = cv\_iltyp  
            cv\_ilber   = cv\_ilber  
            cv\_ilpla   = cv\_ilpla ).  
  
  ENDMETHOD.

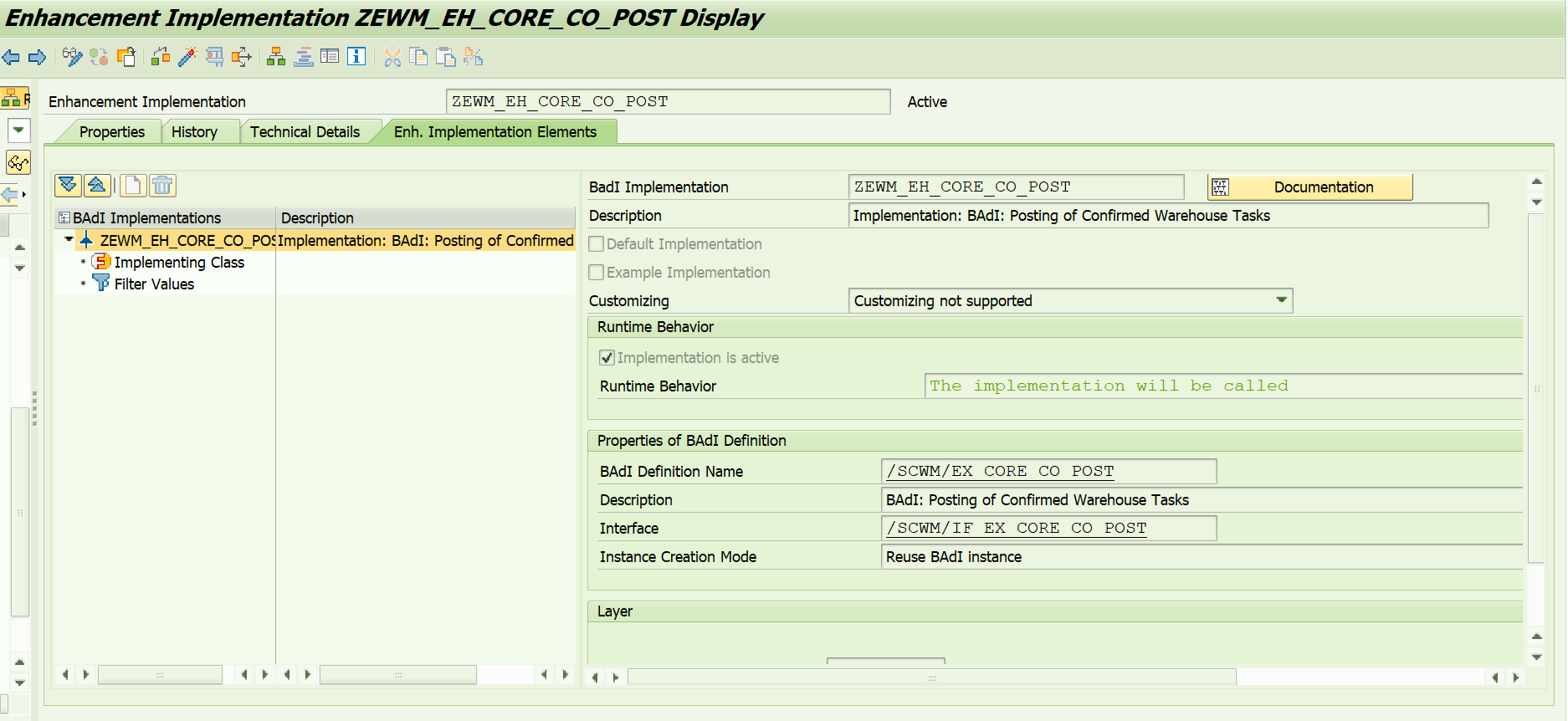
  METHOD DFKU\_SORTSTATION\_DET.  
  
    CONSTANTS: lc\_vltyp\_relevant      TYPE rvari\_vnam VALUE 'ZPICKING\_SORTING\_DAIFUKU'.  
    CONSTANTS: lc\_daifuku\_station     TYPE rvari\_vnam VALUE 'ZDFKU\_SORTATION\_STATION',  
               lc\_dfku\_toatl\_stations TYPE rvari\_vnam VALUE 'ZDFKU\_TOTAL\_SORTSTATIONS'.  
    DATA: lv\_total\_stations  TYPE i,  
          lv\_current\_station TYPE i.  
  
    DATA: lv\_matid      TYPE /scwm/de\_matid,  
          ls\_mat\_global TYPE /scwm/s\_material\_global,  
          lt\_mat\_uom    TYPE /scwm/tt\_material\_uom.  
    DATA: ls\_content\_query TYPE /scwm/s\_ps\_content\_query,  
          lt\_matid         TYPE /scmb/mdl\_matid\_tab,  
          ls\_matid         TYPE /scmb/mdl\_matid\_str,  
          lt\_ps\_keys       TYPE /scwm/tt\_ps\_header\_key,  
          lv\_severity      TYPE bapi\_mtype,  
          lt\_return        TYPE bapirettab,  
          lt\_ps\_object     TYPE /scwm/tt\_ps\_object.  
    DATA ls\_merger\_cond TYPE zpack\_merge\_cond.  
    DATA lt\_huitm TYPE /scwm/tt\_huitm\_int.  
    DATA: lv\_station TYPE /scwm/de\_workstation.  
  
\* Verify source storage types  
    SELECT sign, opti AS option, low, high  
      FROM tvarvc  
      INTO TABLE @DATA(lt\_vltyp\_relevant)  
      WHERE name = @lc\_vltyp\_relevant.  
  
    CHECK iv\_vltyp IN lt\_vltyp\_relevant. "9001-9005"  
  
\*1) check for empty station or station to be allocated (new table with Dlvs assigned)  
\* Verify Delivery assignment first  
    IF ms\_dlv\_sortstation\_ass-delivery IS INITIAL.  
      SELECT SINGLE a~docid, a~docno, b~refdocno  
        FROM /scdl/db\_proch\_o AS a  
        INNER JOIN /scdl/db\_refdoc AS b  
        ON b~docid = a~docid  
        WHERE a~docid = @is\_ordim\_o-rdocid  
        AND b~itemid IS INITIAL  
        AND b~refdoccat = 'ERP'  
        AND b~doccat = 'PDO'  
        INTO @DATA(ls\_docid).  
      IF sy-subrc = 0.  
  
        SELECT SINGLE \* FROM zdlv\_sortst\_ass  
          INTO @DATA(ls\_dlv\_sort\_ass)  
          WHERE lgnum = @iv\_lgnum  
          AND delivery = @ls\_docid-docno.  
  
        IF sy-subrc = 0.  
          lv\_current\_station = ls\_dlv\_sort\_ass-station.  
        ELSE.  
          DATA(lv\_first) = abap\_true.  
          ls\_dlv\_sort\_ass-lgnum = iv\_lgnum.  
          ls\_dlv\_sort\_ass-vbeln = ls\_docid-refdocno.  
          ls\_dlv\_sort\_ass-delivery = ls\_docid-docno.  
          ls\_dlv\_sort\_ass-created\_by = sy-uname.  
          CONVERT DATE sy-datum TIME sy-uzeit INTO TIME STAMP  ls\_dlv\_sort\_ass-created\_at TIME ZONE sy-zonlo.  
        ENDIF.  
      ENDIF.  
    ELSE.  
      lv\_current\_station = ms\_dlv\_sortstation\_ass-station.  
    ENDIF.  
  
\* Verify Sation counter next  
    SELECT \* FROM tvarvc  
      INTO TABLE @DATA(lt\_tvarvc\_station)  
      WHERE name IN ( @lc\_daifuku\_station, @lc\_dfku\_toatl\_stations ).  
    IF sy-subrc = 0.  
  
      READ TABLE lt\_tvarvc\_station ASSIGNING FIELD-SYMBOL(<fs\_tvarvc>) WITH KEY name = lc\_dfku\_toatl\_stations.  
      IF sy-subrc = 0.  
        lv\_total\_stations = <fs\_tvarvc>-low.  
      ENDIF.  
  
      CLEAR <fs\_tvarvc>.  
      IF lv\_current\_station IS INITIAL.  
        READ TABLE lt\_tvarvc\_station ASSIGNING <fs\_tvarvc> WITH KEY name = lc\_daifuku\_station.  
        IF sy-subrc = 0.  
          lv\_current\_station = <fs\_tvarvc>-low.  
          mv\_update = abap\_true.  
        ENDIF.  
\* calculate station number  
        IF lv\_current\_station GE lv\_total\_stations.  
          lv\_current\_station = 1.  
        ELSE.  
          lv\_current\_station = lv\_current\_station + 1.  
        ENDIF.  
      ENDIF.  
  
\* get relevant Sort station  
      SELECT \* FROM zdlv\_sortst\_stat  
        INTO TABLE @DATA(lt\_sort\_station)  
        WHERE lgnum = @iv\_lgnum.  
\*        AND station = @lv\_current\_station.  
\*        AND active = @abap\_true.  
      IF sy-subrc = 0.  
        SORT lt\_sort\_station BY station.  
        DO 2 TIMES.  
          LOOP AT lt\_sort\_station INTO DATA(ls\_sort\_station).  
            IF ls\_sort\_station-station = lv\_current\_station.  
              IF ls\_sort\_station-active = abap\_true.  
                DATA(lv\_stationf) = abap\_true.  
                EXIT.  
              ELSE.  
                IF lv\_current\_station GE lv\_total\_stations.  
                  lv\_current\_station = 1.  
                ELSE.  
                  lv\_current\_station = lv\_current\_station + 1.  
                ENDIF.  
              ENDIF.  
            ENDIF.  
          ENDLOOP.  
  
          IF lv\_stationf = abap\_true.  
            EXIT.  
          ENDIF.  
        ENDDO.  
  
        <fs\_tvarvc>-low = lv\_current\_station.  
      ENDIF.  
    ENDIF.  
  
\*2) check for merging logic - Identify to allocate odd number or even number station based on threshold qty(cross check with pick qty)  
\*  Get pack size from pack spec. consider product volume as packsize and total qty for packing material as Full pallet qty.  
\*  UOM for Full pallet and threshold are same.  
\* If pick qty < threshold qty, consider Odd number storage section  
\* if pick qty >=  threshold qty, consider Even number storage section  
    lv\_matid = is\_ordim\_o-matid.  
\* get Product base unit volume  
    TRY.  
        CALL FUNCTION '/SCWM/MATERIAL\_READ\_SINGLE'  
          EXPORTING  
            iv\_matid      = lv\_matid  
          IMPORTING  
            es\_mat\_global = ls\_mat\_global  
            et\_mat\_uom    = lt\_mat\_uom.  
  
      CATCH /scwm/cx\_md.  
        IF ls\_mat\_global-matnr IS INITIAL.  
          MESSAGE 'Product not found' TYPE 'E'.  
          RETURN.  
        ENDIF.  
    ENDTRY.  
  
    READ TABLE lt\_mat\_uom INTO DATA(ls\_mat\_uom) WITH KEY meinh = ls\_mat\_global-meins.  
    IF sy-subrc = 0.  
      ls\_merger\_cond-lgnum = iv\_lgnum.  
      ls\_merger\_cond-pack\_size = ls\_mat\_uom-volum.  
      ls\_merger\_cond-pack\_size\_uom = ls\_mat\_uom-voleh.  
  
\* get Total/ Full Pallet Qty from Product packspec  
      ls\_matid-matid = is\_ordim\_o-matid.  
      APPEND ls\_matid TO lt\_matid.  
      ls\_content\_query-matid = lt\_matid.  
  
      CALL FUNCTION '/SCWM/API\_PACKSPEC\_GETLIST'  
        EXPORTING  
          is\_content\_query = ls\_content\_query  
        IMPORTING  
          et\_ps\_keys       = lt\_ps\_keys  
          ev\_severity      = lv\_severity  
          et\_return        = lt\_return.  
      IF lv\_severity <> 'E'.  
  
        CALL FUNCTION '/SCWM/API\_PACKSPEC\_READ'  
          EXPORTING  
            it\_ps\_id    = lt\_ps\_keys  
          IMPORTING  
            et\_object   = lt\_ps\_object  
            ev\_severity = lv\_severity  
            et\_return   = lt\_return.  
        IF lv\_severity <> 'E'  
          AND lt\_ps\_object IS NOT INITIAL.  
          IF lt\_ps\_object[ 1 ]-levels IS NOT INITIAL.  
            DATA(ls\_ps\_level) = lt\_ps\_object[ 1 ]-levels[ 1 ].  
            ls\_merger\_cond-full\_pallet = ls\_ps\_level-total\_quan.  
            ls\_merger\_cond-uom = ls\_ps\_level-total\_unit.  
          ENDIF.  
        ENDIF.  
      ENDIF.  
      IF lv\_severity = 'E'.  
        LOOP AT lt\_return INTO DATA(ls\_return) WHERE type CA 'EAX'.  
          MESSAGE ID ls\_return-id TYPE ls\_return-type NUMBER ls\_return-number  
          WITH ls\_return-message\_v1 ls\_return-message\_v2 ls\_return-message\_v3 ls\_return-message\_v4 DISPLAY LIKE 'E'.  
          RETURN.  
        ENDLOOP.  
      ENDIF.  
  
\* get Merger condition  
      SELECT SINGLE \* FROM zpack\_merge\_cond  
        INTO ls\_merger\_cond  
        WHERE lgnum = ls\_merger\_cond-lgnum  
        AND pack\_size = ls\_merger\_cond-pack\_size  
        AND pack\_size\_uom = ls\_merger\_cond-pack\_size\_uom  
        AND full\_pallet = ls\_merger\_cond-full\_pallet  
        AND uom = ls\_merger\_cond-uom.  
      IF sy-subrc = 0.  
        lv\_station = lv\_current\_station.  
        CONDENSE lv\_station.  
  
        IF is\_ordim\_o-vsolm < ls\_merger\_cond-thresh\_qty.  
          READ TABLE lt\_sort\_station INTO ls\_sort\_station WITH KEY station = lv\_station bin\_position = 'Left'.  
          IF sy-subrc = 0.  
            IF cv\_iltyp <> 'PSWC'.  
              cv\_iltyp = 'PSWC'.  
            ENDIF.  
            cv\_ilber = ls\_sort\_station-lgber.  
            mv\_uptick = abap\_true.  
          ENDIF.  
  
        ELSE. ">= thqty  
\*In case of full pallet from mini ASRS consider storage type = "CPOT"  
\*  consider pick qty = physical pallet qty(calculate HU qty)  
\*  and pick qty should be > than threshold qty.  
          CALL FUNCTION '/SCWM/HU\_READ'  
            EXPORTING  
              iv\_appl    = 'WME'  
              iv\_lgnum   = iv\_lgnum  
              iv\_huident = is\_ordim\_o-vlenr  
            IMPORTING  
              et\_huitm   = lt\_huitm  
            EXCEPTIONS  
              deleted    = 1  
              not\_found  = 2  
              error      = 3  
              OTHERS     = 4.  
          IF sy-subrc = 0.  
\*            DELETE lt\_huitm WHERE guid\_stock = is\_ordim\_o-guid\_stock.  
\*            IF lines( lt\_huitm ) = 0.  
            READ TABLE lt\_huitm INTO DATA(ls\_huitm) WITH KEY guid\_stock = is\_ordim\_o-guid\_stock.  
            IF sy-subrc = 0  
              AND ls\_huitm-quan = is\_ordim\_o-vsolm.  
\* Bin determination for CPOT  
\* Get bin section for the destination bin of the task,  
              IF is\_ordim\_o-nlpla IS NOT INITIAL.  
                SELECT SINGLE \* FROM /scwm/lagp  
                  INTO @DATA(ls\_lagp)  
                  WHERE lgnum = @iv\_lgnum  
                  AND lgpla = @is\_ordim\_o-nlpla  
                  AND lgtyp = @is\_ordim\_o-nltyp.  
                IF sy-subrc = 0.  
                  SELECT SINGLE \* FROM /scwm/troutl  
                    INTO @DATA(ls\_troutl)  
                    WHERE lgnum =  @iv\_lgnum  
                    AND vltyp = 'PSWC'  
                    AND nltyp = @is\_ordim\_o-nltyp  
                    AND nlber = @ls\_lagp-lgber\_lgst.  
                  IF sy-subrc = 0.  
                    cv\_iltyp = ls\_troutl-iltyp. "'CPOT'.  
                    cv\_ilber = ls\_troutl-ilber.  
                    cv\_ilpla = ls\_troutl-ilpla.  
                  ENDIF.  
                ENDIF.  
              ENDIF.  
  
            ELSE.  
              READ TABLE lt\_sort\_station INTO ls\_sort\_station WITH KEY station = lv\_station bin\_position = 'Right'.  
              IF sy-subrc = 0.  
                IF cv\_iltyp <> 'PSWC'.  
                  cv\_iltyp = 'PSWC'.  
                ENDIF.  
                cv\_ilber = ls\_sort\_station-lgber.  
                mv\_uptick = abap\_true.  
              ENDIF.  
            ENDIF.  
          ENDIF.  
        ENDIF.  
  
      ELSE.  
        MESSAGE 'Merger condition is missing' TYPE 'E'.  
        RETURN.  
      ENDIF.  
    ENDIF.  
  
    IF mv\_uptick = abap\_true  
      AND mv\_update = abap\_true.  
      SHIFT <fs\_tvarvc>-low LEFT DELETING LEADING space.  
      UPDATE tvarvc SET low = <fs\_tvarvc>-low  
      WHERE name = lc\_daifuku\_station.  
    ENDIF.  
  
\* Delivery assignment  
\* Cross check the picking status for active Dlv for this Destination bin/.  
    IF lv\_first = abap\_true  
      AND ls\_dlv\_sort\_ass-delivery IS NOT INITIAL  
      AND lv\_station IS NOT INITIAL.  
  
      ls\_dlv\_sort\_ass-station = lv\_station.  
      MODIFY zdlv\_sortst\_ass FROM ls\_dlv\_sort\_ass.  
      ms\_dlv\_sortstation\_ass = ls\_dlv\_sort\_ass.  
    ENDIF.  
  
    CLEAR: lt\_vltyp\_relevant,  
           lt\_tvarvc\_station,  
           lv\_total\_stations,  
           lv\_current\_station,  
           lt\_sort\_station,  
           lv\_matid,  
           ls\_mat\_global,  
           lt\_mat\_uom,  
           ls\_merger\_cond,  
           ls\_matid,  
           lt\_matid,  
           ls\_content\_query,  
           lt\_ps\_keys,  
           lt\_ps\_object,  
           lv\_severity,  
           lt\_return,  
           ls\_ps\_level,  
           ls\_sort\_station,  
           lt\_huitm,  
\*           lv\_uptick,  
\*           lv\_update,  
           ls\_docid,  
           ls\_dlv\_sort\_ass,  
           ls\_lagp,  
           ls\_troutl.  
  
  
  ENDMETHOD.



A screenshot of a computer

Description automatically generated

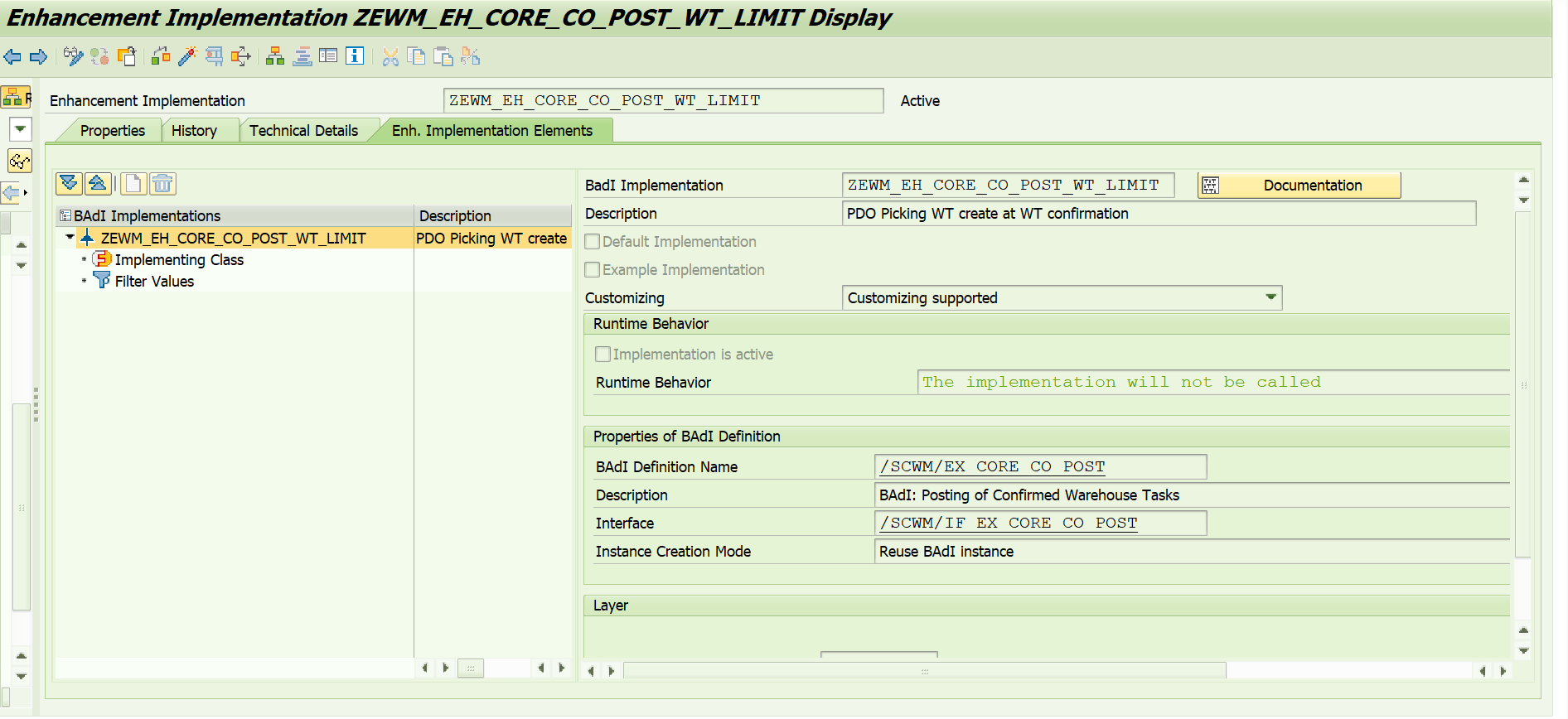
  METHOD /scwm/if\_ex\_who\_dstgrp~dstgrp.  
  
\* Update Delivery-Sort station assignment table when saving the task...  
    IF zcl\_scwm\_ex\_core\_lsc\_layout=>mt\_dlv\_sortstation\_ass[] IS NOT INITIAL.  
      SORT zcl\_scwm\_ex\_core\_lsc\_layout=>mt\_dlv\_sortstation\_ass BY vbeln.  
      DELETE ADJACENT DUPLICATES FROM zcl\_scwm\_ex\_core\_lsc\_layout=>mt\_dlv\_sortstation\_ass COMPARING vbeln.  
  
      MODIFY zdlv\_sortst\_ass FROM TABLE zcl\_scwm\_ex\_core\_lsc\_layout=>mt\_dlv\_sortstation\_ass.  "update table  
      CLEAR zcl\_scwm\_ex\_core\_lsc\_layout=>mt\_dlv\_sortstation\_ass.  
    ENDIF.  
  
  ENDMETHOD.



A screenshot of a computer

Description automatically generated

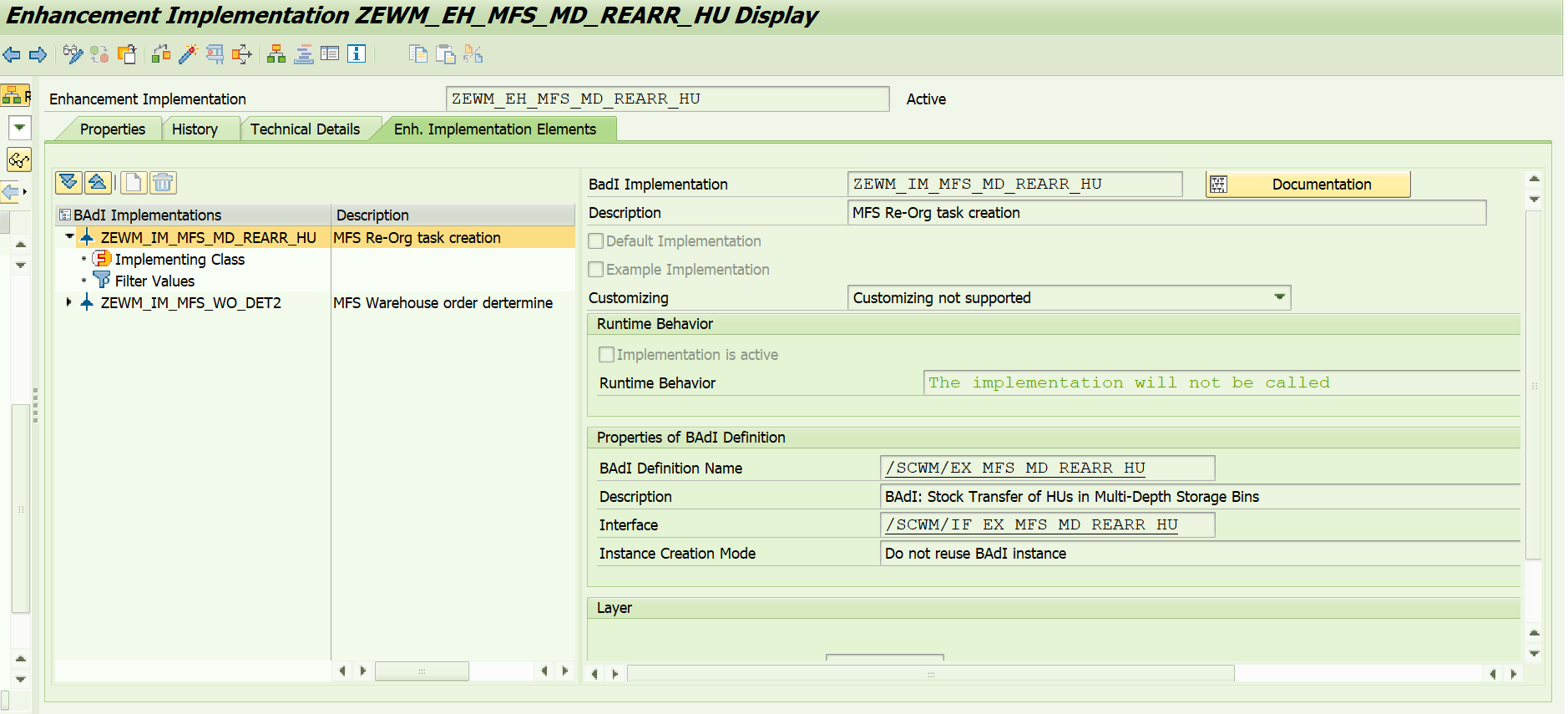
  METHOD /scwm/if\_ex\_core\_co\_post~post.  
  
    DATA: lt\_dlv\_sortst\_ass TYPE TABLE OF zdlv\_sortst\_ass,  
          ls\_dlv\_sortst\_ass TYPE zdlv\_sortst\_ass.  
    DATA lt\_dlv\_tanum TYPE /scwm/tt\_wip\_wt4dlv.  
    CONSTANTS: lc\_vltyp\_relevant      TYPE rvari\_vnam VALUE 'ZPICKING\_SORTING\_DAIFUKU'.  
  
  
\* Verify source storage types  
    SELECT sign, opti AS option, low, high  
      FROM tvarvc  
      INTO TABLE @DATA(lt\_vltyp\_relevant)  
      WHERE name = @lc\_vltyp\_relevant.  
  
    DATA(lt\_ordim\_c) = it\_ordim\_c[].  
  
    DATA(lr\_vlenr) = VALUE rseloption( FOR lr\_ordim\_c IN lt\_ordim\_c sign = wmegc\_sign\_inclusive  
                                                              option = wmegc\_option\_eq  
                                                              ( low  = lr\_ordim\_c-vlenr ) ).  
    IF lt\_ordim\_c IS NOT INITIAL.  
      SELECT \* FROM /scwm/ordim\_o  
        INTO TABLE @DATA(lt\_ordim\_o)  
        FOR ALL ENTRIES IN @lt\_ordim\_c  
        WHERE lgnum = @lt\_ordim\_c-lgnum  
        AND vlenr = @lt\_ordim\_c-vlenr.  
      IF sy-subrc = 0.  
        DATA(lt\_relevant\_wts) = lt\_ordim\_o[].  
  
      ELSE.  
        MOVE-CORRESPONDING lt\_ordim\_c[] TO lt\_relevant\_wts.  
      ENDIF.  
  
      IF lt\_relevant\_wts IS NOT INITIAL.  
\* Get Delivery & Reference document  
        SELECT a~docid, a~docno, b~refdocno  
          FROM /scdl/db\_proch\_o AS a  
          INNER JOIN /scdl/db\_refdoc AS b  
          ON b~docid = a~docid  
          FOR ALL ENTRIES IN @lt\_relevant\_wts  
          WHERE a~docid = @lt\_relevant\_wts-rdocid  
          AND b~itemid IS INITIAL  
          AND b~refdoccat = 'ERP'  
          AND b~doccat = 'PDO'  
          INTO TABLE @DATA(lt\_refdoc).  
        IF sy-subrc = 0.  
  
          DATA(lt\_docno) = VALUE /scwm/dlv\_docno\_tab( FOR li\_refdoc IN lt\_refdoc  
                                                      doccat = /scdl/if\_dl\_doc\_c=>sc\_doccat\_out\_prd  
                                                      ( docno  = li\_refdoc-docno ) ).  
          IF lt\_docno IS NOT INITIAL.  
\* Get all warehouse task relevant to Delivery  
            CALL FUNCTION '/SCWM/WIP\_GETWT4DLV'  
              EXPORTING  
                iv\_lgnum  = iv\_lgnum  
                iv\_doccat = /scdl/if\_dl\_doc\_c=>sc\_doccat\_out\_prd  
                it\_docno  = lt\_docno  
              IMPORTING  
                et\_wt4dlv = lt\_dlv\_tanum.  
            IF lt\_dlv\_tanum IS NOT INITIAL.  
              CLEAR: lt\_ordim\_o,  
                     lt\_dlv\_sortst\_ass.  
  
\* Read only Open tasks of the delivery  
              SELECT  \* FROM /scwm/ordim\_o AS to  
                  INTO CORRESPONDING FIELDS OF TABLE lt\_ordim\_o  
                  FOR ALL ENTRIES IN lt\_dlv\_tanum  
                  WHERE to~lgnum        = iv\_lgnum  
                    AND to~tanum        = lt\_dlv\_tanum-tanum.  
\*                    AND to~flghuto      = abap\_true.  
              IF sy-subrc <> 0  
\* If no Open tasks are available for any of the deliveries...  
                AND lt\_ordim\_o IS INITIAL.  
  
                IF lt\_docno IS NOT  INITIAL.  
\* Read/ Verify Delivery - station entries  
                  SELECT \* FROM zdlv\_sortst\_ass  
                    INTO TABLE lt\_dlv\_sortst\_ass  
                    FOR ALL ENTRIES IN lt\_docno  
                    WHERE lgnum = iv\_lgnum  
                    AND delivery = lt\_docno-docno.  
                ENDIF.  
              ENDIF.  
            ENDIF.  
          ENDIF.  
        ENDIF.  
      ENDIF.  
    ENDIF.  
  
\* Verify Open task for the Deliveries of some are available...  
    IF lt\_ordim\_o IS NOT INITIAL.  
      DELETE lt\_ordim\_c WHERE flghuto = abap\_false.  
  
      LOOP AT lt\_ordim\_c INTO DATA(ls\_ordim\_c) WHERE vltyp IN lt\_vltyp\_relevant AND flghuto = abap\_true.  
        "delete current HU task  
        DELETE lt\_ordim\_o WHERE tanum = ls\_ordim\_c-tanum.  
  
        READ TABLE lt\_ordim\_o INTO DATA(ls\_ordim\_o) WITH KEY vlenr = ls\_ordim\_c-vlenr.  "identify product task of source HU  
        IF sy-subrc = 0.  
          "identify and delete Product task relevant to current HU task  
          DELETE lt\_ordim\_o WHERE tanum = ls\_ordim\_o-tanum.  
  
          LOOP AT lt\_ordim\_o INTO ls\_ordim\_o WHERE flghuto = abap\_true AND nltyp = 'PSWC'.  
            "Identify Open Product task for the same delivery  
            "Verify if any open HU tasks are there for the same delivery  
            READ TABLE lt\_ordim\_o INTO ls\_ordim\_o WITH KEY rdocid = ls\_ordim\_o-rdocid vlenr = ls\_ordim\_o-vlenr.  
            IF sy-subrc = 0.  
              DATA(lv\_open\_huwt) = abap\_true.  
              EXIT.  
            ENDIF.  
          ENDLOOP.  
  
          IF lv\_open\_huwt = abap\_true.  
            "still some open tasks are there for the delivery  
          ELSE.  
            "this is the last HU task... being confirmed now...  
            READ TABLE lt\_refdoc INTO DATA(ls\_refdoc) WITH KEY docid = ls\_ordim\_o-rdocid.  
            IF sy-subrc = 0.  
\*          Map table record to be deleted  
              ls\_dlv\_sortst\_ass-lgnum = iv\_lgnum.  
              ls\_dlv\_sortst\_ass-vbeln = ls\_refdoc-refdocno.  
              APPEND ls\_dlv\_sortst\_ass TO lt\_dlv\_sortst\_ass.  
              CLEAR ls\_dlv\_sortst\_ass.  
            ENDIF.  
          ENDIF.  
  
        ENDIF.  
        CLEAR: ls\_ordim\_c, ls\_ordim\_o, lv\_open\_huwt, ls\_refdoc.  
      ENDLOOP.  
  
      IF lt\_dlv\_sortst\_ass IS NOT INITIAL.  
\* Read/ Verify Delivery - station entries  
        SELECT \* FROM zdlv\_sortst\_ass  
          INTO TABLE lt\_dlv\_sortst\_ass  
          FOR ALL ENTRIES IN lt\_dlv\_sortst\_ass  
          WHERE lgnum = lt\_dlv\_sortst\_ass-lgnum  
          AND vbeln = lt\_dlv\_sortst\_ass-vbeln.  
      ENDIF.  
    ENDIF.  
  
    IF lt\_dlv\_sortst\_ass IS NOT INITIAL.  
      SORT lt\_dlv\_sortst\_ass.  
\* Delete Delivery - station records from assignment table  
      DELETE zdlv\_sortst\_ass FROM TABLE lt\_dlv\_sortst\_ass[].  
      CLEAR: lt\_dlv\_sortst\_ass.  
    ENDIF.  
  
  
  ENDMETHOD.

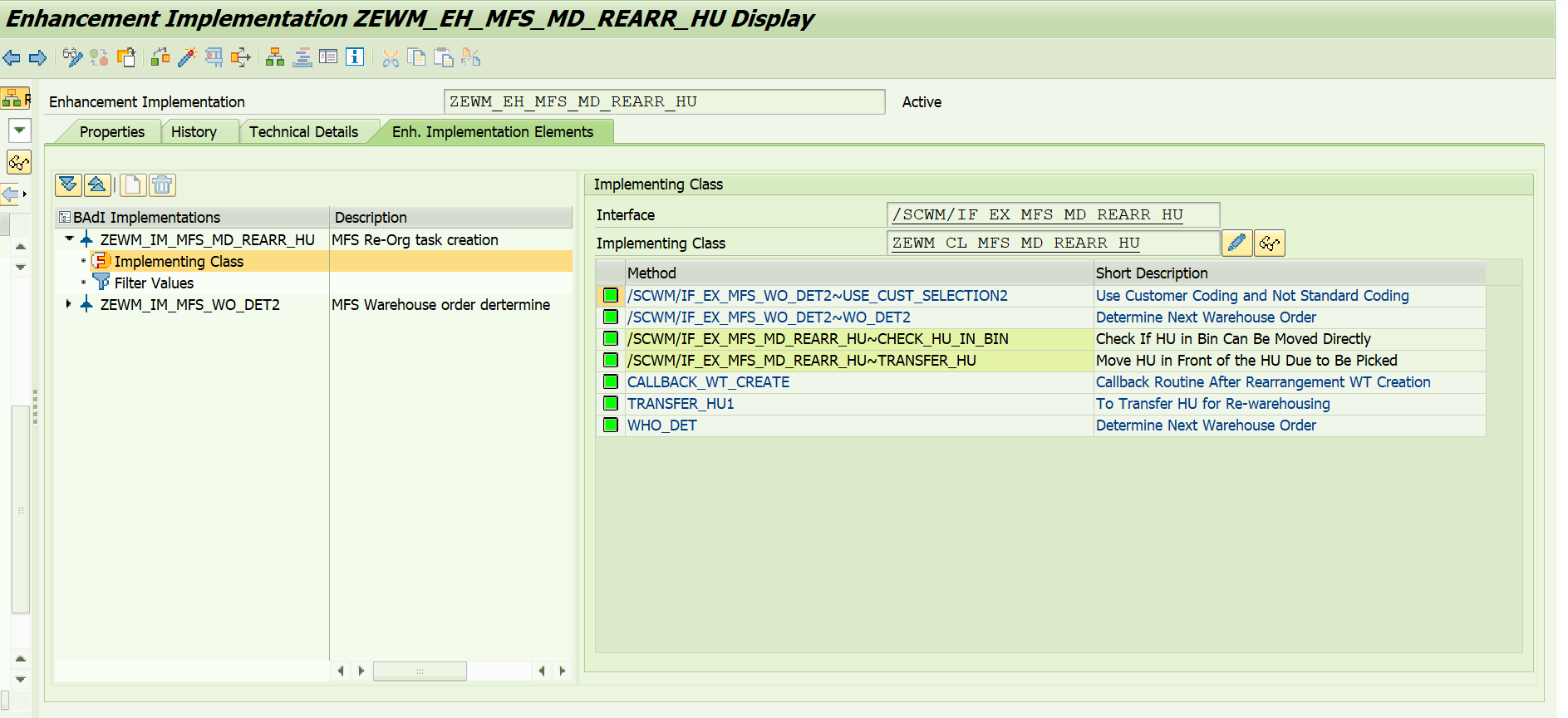


A screenshot of a computer

Description automatically generated

  METHOD /scwm/if\_ex\_core\_co\_post~post.  
    DATA : lt\_con\_param TYPE TABLE OF zcon\_param\_val.  
  
    DATA: lt\_ordim\_c TYPE /scwm/tt\_ordim\_c,  
          ls\_docid   TYPE /scwm/dlv\_docid\_item\_str.  
  
    DATA : lc\_requirement\_name TYPE zde\_req\_name VALUE 'PDO\_PICK\_TASK\_LIMIT',  
           lc\_para\_name1       TYPE zde\_param    VALUE 'ENH\_ACTIVE',  
           lc\_para\_name2       TYPE zde\_param    VALUE 'VLTYP'.  
  
    READ TABLE it\_ordim\_c ASSIGNING FIELD-SYMBOL(<lfs\_ordim\_c>) INDEX 1.  
    CHECK sy-subrc IS INITIAL.  
\*get all parametric values  
    SELECT \* FROM zcon\_param\_val INTO TABLE lt\_con\_param  
             WHERE lgnum     = <lfs\_ordim\_c>-lgnum  
             AND zreq\_name = lc\_requirement\_name.  
\* condense variable in case any gap while maintaining  
    LOOP AT lt\_con\_param ASSIGNING FIELD-SYMBOL(<lfs\_con\_param>).  
      CONDENSE <lfs\_con\_param>-zvalue NO-GAPS.  
    ENDLOOP.  
    SORT lt\_con\_param BY zparam.  
\* check active parameter for enhancement  
    READ TABLE lt\_con\_param ASSIGNING <lfs\_con\_param>  
                            WITH KEY zparam    = lc\_para\_name1  
                                     zvalue    = abap\_true  
                                     zactive   = abap\_true.  
    CHECK sy-subrc = 0.  
\* collect information on cancelled WTs  
    LOOP AT it\_ordim\_c ASSIGNING <lfs\_ordim\_c>.  
      IF <lfs\_ordim\_c>-trart    = wmegc\_trart\_pick AND   "2  
      <lfs\_ordim\_c>-rdoccat  = wmegc\_doccat\_pdo." AND   "PDO  
\* checking for 3000 storage type  
        READ TABLE lt\_con\_param ASSIGNING <lfs\_con\_param>  
                                WITH KEY zparam    = lc\_para\_name2  
                                         zvalue    = <lfs\_ordim\_c>-vltyp  
                                         zactive   = abap\_true.  
        IF sy-subrc = 0.  
        IF <lfs\_ordim\_c>-rdoccat IS NOT INITIAL AND  
           <lfs\_ordim\_c>-rdocid  IS NOT INITIAL.  
          APPEND <lfs\_ordim\_c> TO lt\_ordim\_c.  
        ENDIF.  
        ENDIF.  
      ENDIF.  
    ENDLOOP.  
    CHECK lt\_ordim\_c IS NOT INITIAL.  
    SORT lt\_ordim\_c BY rdoccat.  
    DELETE lt\_ordim\_c WHERE rdoccat IS INITIAL.  
    SORT lt\_ordim\_c BY rdocid.  
    DELETE lt\_ordim\_c WHERE rdocid IS INITIAL.  
    LOOP AT lt\_ordim\_c ASSIGNING <lfs\_ordim\_c>.  
      CLEAR : ls\_docid.  
      ls\_docid-docid  = <lfs\_ordim\_c>-rdocid.  
      ls\_docid-doccat = <lfs\_ordim\_c>-rdoccat.  
      CALL FUNCTION 'ZEWM\_PDO\_WT\_CREATE' IN UPDATE TASK  
        EXPORTING  
          iv\_lgnum = <lfs\_ordim\_c>-lgnum  
          is\_docid = ls\_docid.  
      EXIT.  
    ENDLOOP.  
  
  ENDMETHOD.





  METHOD /scwm/if\_ex\_mfs\_wo\_det2~wo\_det2.  
  
  
  
    CALL METHOD who\_det  
      EXPORTING  
        iv\_lgnum            = iv\_lgnum                 " Warehouse Number/Warehouse Complex  
        iv\_queue            = iv\_queue                 " Queue  
        is\_who              = is\_who                 " Warehouse Order Determined by Standard  
        iv\_intleav          = iv\_intleav                  " Interleaving is Active  
        iv\_who\_inproc       = iv\_who\_inproc                 " Warehouse Order in Process  
        iv\_current\_wo       = iv\_current\_wo                 " Warehouse Order Number  
        iv\_current\_wt       = iv\_current\_wt                 " Warehouse Task  
        iv\_caller\_of\_badi   = iv\_caller\_of\_badi                 " Context of program which calls a BAdI  
        is\_rsrc             = is\_rsrc                 " Resources  
        it\_who              = it\_who                 " Table of Internal Warehouse Orders  
        it\_ordim\_o          = it\_ordim\_o                 " Table: Warehouse Tasks Open, Internal  
        it\_who\_original     = it\_who\_original                 " WO before BAdI is called  
        it\_ordim\_o\_original = it\_ordim\_o\_original                 " Open WT before BAdI is called  
      CHANGING  
        ct\_who              = ct\_who                 " Warehouse Order  
        ct\_ordim\_o          = ct\_ordim\_o                 " Table: Warehouse Tasks Open  
        cv\_rearr\_done       = cv\_rearr\_done                 " Stock Transfer Done inside BAdI  
        cv\_call\_wt\_det\_prep = cv\_call\_wt\_det\_prep                 " Flag if /SCWM/MFS\_WT\_DET\_PREP should be called  
        co\_log              = co\_log.               " Log  
  
  
    DATA ls\_selcrit   TYPE /scwm/s\_to\_selcrit\_mon.  
    DATA ltr\_who      TYPE rseloption.  
    DATA lsr\_who      TYPE rsdsselopt.  
    DATA lt\_to        TYPE /scwm/tt\_to\_det\_mon.  
  
    lsr\_who-sign    = wmegc\_sign\_inclusive.  
    lsr\_who-option  = wmegc\_option\_eq.  
    lsr\_who-low     = iv\_current\_wo.  
    APPEND lsr\_who TO ltr\_who.  
  
    LOOP AT ct\_who ASSIGNING FIELD-SYMBOL(<fs\_who>).  
      lsr\_who-low = <fs\_who>-who.  
      APPEND lsr\_who TO ltr\_who.  
    ENDLOOP.  
  
    ls\_selcrit-r\_who = ltr\_who.  
  
\*  GET all wt details for who in asrs Queue  
     CALL FUNCTION '/SCWM/TO\_GET\_WIP'  
       EXPORTING  
         iv\_lgnum   = iv\_lgnum  
         iv\_open    = abap\_true  
         iv\_conf    = abap\_true  
         is\_selcrit = ls\_selcrit  
       IMPORTING  
           et\_to      = lt\_to.  
    IF sy-subrc = 0 AND lt\_to IS NOT INITIAL.  
  
      MOVE-CORRESPONDING lt\_to TO ct\_ordim\_o.  
  
    ENDIF.  
  
\* Start Below code moved to the mthod who\_det 7/6/2023 by Santha  
\*\*  
\*\*    DATA: lt\_cancl  TYPE /scwm/tt\_cancl,  
\*\*          ls\_cancel LIKE LINE OF lt\_cancl.  
\*\*    DATA: lt\_create\_hu      TYPE /scwm/tt\_to\_crea\_hu,  
\*\*          ls\_create\_hu      TYPE /scwm/s\_to\_crea\_hu,  
\*\*          lt\_rewarehouse\_fg TYPE TABLE OF zrewareho\_fg\_log,  
\*\*          ls\_rewarehouse\_fg LIKE LINE OF lt\_rewarehouse\_fg,  
\*\*          lt\_wo\_info        TYPE TABLE OF /scwm/ordim\_o,  
\*\*          ls\_wo\_info        LIKE LINE OF lt\_wo\_info,  
\*\*          ls\_log            TYPE bal\_s\_log,  
\*\*  
\*\*          lv\_severity       TYPE bapi\_mtype.  
\*\*  
\*\*    CONSTANTS:  
\*\*              wmegc\_severity\_eax(3)   TYPE c          VALUE 'EAX'.  
\*\*\*  
\*\*    DATA: lt\_who                 TYPE TABLE OF /scwm/who,  
\*\*          lt\_dir\_procty          TYPE TABLE OF /scwm/who,  
\*\*          lt\_rtv\_procty          TYPE TABLE OF /scwm/who,  
\*\*          lt\_repl\_procty         TYPE TABLE OF /scwm/who,  
\*\*          lt\_str\_procty          TYPE TABLE OF /scwm/who,  
\*\*          lt\_str\_who             TYPE /scwm/tt\_who,  
\*\*          ls\_str\_who             LIKE LINE OF lt\_str\_who,  
\*\*          lt\_rtv\_who             TYPE   /scwm/tt\_who,  
\*\*          lt\_rtv\_who1            TYPE   /scwm/tt\_who,  
\*\*          lt\_repl\_who            TYPE  /scwm/tt\_who,  
\*\*          ls\_repl\_who            LIKE LINE OF lt\_repl\_who,  
\*\*          ls\_rtv\_who             LIKE LINE OF lt\_rtv\_who,  
\*\*          lt\_dir\_who             TYPE /scwm/tt\_who,  
\*\*          lt\_param\_agc\_schd      TYPE zapl\_tt\_param,  
\*\*          lt\_param\_mfs\_tele\_send TYPE zapl\_tt\_param,  
\*\*          lt\_crane\_check         TYPE TABLE OF zcon\_param\_val,  
\*\*          ls\_crane\_check         LIKE LINE OF lt\_crane\_check.  
\*\*    DATA: cs\_who LIKE LINE OF ct\_who.  
\*\*    DATA : lv\_count\_who TYPE i.  
\*\*  
\*\*    DATA : ls\_param      TYPE zapl\_param,  
\*\*           ls\_hdr\_procty TYPE /scwm/who.  
\*\*\*           ls\_who        TYPE /scwm/s\_who\_int.  
\*\*  
\*\*    DATA : lv\_id1          TYPE zapl\_de\_param\_id1,  
\*\*           lv\_id2          TYPE zapl\_de\_param\_id2,  
\*\*           lv\_msgtext      TYPE bapi\_msg,  
\*\*           lv\_count        TYPE i,  
\*\*           lv\_direct       TYPE i,  
\*\*           lv\_storage      TYPE i,  
\*\*           lv\_retrieve     TYPE i,  
\*\*           lv\_max\_direct   TYPE i,  
\*\*           lv\_max\_storage  TYPE i,  
\*\*           lv\_max\_retrieve TYPE i,  
\*\*           iv\_wo\_count     TYPE i.  
\*\*    DATA:lt\_error TYPE TABLE OF zcrane\_dashboard,  
\*\*         ls\_error LIKE LINE OF lt\_error.  
\*\*  
\*\*    FIELD-SYMBOLS : <ls\_who> TYPE /scwm/who.  
\*\*    DATA: lv\_rewarehouse TYPE zcon\_param\_val-zvalue.  
\*\*    DATA: lt\_ordim\_o TYPE /scwm/tt\_ordim\_o.  
\*\*    DATA: lv\_open\_task      TYPE i,  
\*\*          lv\_stn\_pallets    TYPE i,  
\*\*          lv\_total\_pallets  TYPE i,  
\*\*          lv\_rem\_pallets    TYPE i,  
\*\*          lv\_buffer\_pallets TYPE i.  
\*\*    DATA: lt\_stock\_bin TYPE /scwm/tt\_stock\_bin,  
\*\*          lt\_bapiret   TYPE bapiret2\_t.  
\*\*    DATA: lt\_132\_error TYPE TABLE OF z132\_error\_log,  
\*\*          ls\_132\_error LIKE LINE OF lt\_132\_error.  
\*\*    DATA:iv\_who TYPE /scwm/ordim\_o-who.  
\*\*    DATA :gv\_msgtext      TYPE bapi\_msg,  
\*\*          lo\_log          TYPE REF TO /scwm/cl\_log,  
\*\*          lv\_tstamp\_start TYPE timestampl,  
\*\*          iv\_caller       TYPE /scwm/de\_seqnr,  
\*\*          iv\_check\_future TYPE xfeld,  
\*\*          lv\_who          TYPE /scwm/de\_who,  
\*\*          ls\_who          TYPE /scwm/who,  
\*\*          lv\_time         TYPE symsgv,  
\*\*          lv\_tstamp       TYPE timestampl,  
\*\*  
\*\*          ls\_who\_int      TYPE /scwm/s\_who\_int,  
\*\*          lv\_no\_new\_task  TYPE xfeld,  
\*\*          lt\_rearrange\_hu TYPE /scwm/tt\_huident,  
\*\*          go\_log          TYPE REF TO /scwm/cl\_log.  
\*\*    CONSTANTS:lv\_load\_in\_rack TYPE i VALUE '132'.  
\*\*\*    FIELD-SYMBOLS: ls\_ordim\_open     TYPE /scwm/ordim\_o.  
\*\*  
\*\*    DATA: lo\_badi\_rearr TYPE REF TO zewm\_cl\_mfs\_md\_rearr\_hu,  
\*\*          lt\_inter\_who  TYPE TABLE OF zinter\_leave,  
\*\*          ls\_inter\_who  LIKE LINE OF lt\_inter\_who,  
\*\*          lv\_multideep  TYPE /scwm/de\_multideep,  
\*\*          ls\_prcty      TYPE  rsdsselopt,  
\*\*          lt\_prcty      TYPE rseloption.  
\*\*    DATA:lt\_interleave TYPE TABLE OF zinter\_leave,  
\*\*         ls\_interleave LIKE LINE OF lt\_interleave.  
\*\*  
\*\*    DATA: lt\_wo\_list  TYPE TABLE OF /scwm/s\_wo\_det\_mon\_out,  
\*\*          ls\_wo\_list  TYPE /scwm/s\_wo\_det\_mon\_out,  
\*\*          cs\_who\_list LIKE LINE OF ct\_who,  
\*\*          lt\_who\_to   TYPE  /scwm/tt\_to\_det\_mon\_out,  
\*\*          ls\_who\_to   LIKE LINE OF  lt\_who\_to.  
\*\*    TYPES:BEGIN OF ty\_hu\_tanum,  
\*\*            vlenr TYPE /scwm/ordim\_o-vlenr,  
\*\*            tanum TYPE /scwm/ordim\_o-tanum,  
\*\*          END OF ty\_hu\_tanum.  
\*\*    DATA:lt\_hu\_tanum        TYPE TABLE OF ty\_hu\_tanum,  
\*\*         ls\_hu\_tanum        LIKE LINE OF lt\_hu\_tanum,  
\*\*         lt\_det2\_fg         TYPE TABLE OF zdet2\_fg\_log,  
\*\*         ls\_det2\_fg         LIKE LINE OF lt\_det2\_fg,  
\*\*         lt\_warehouse\_order TYPE TABLE OF zwarehouse\_order,  
\*\*         ls\_warehouse\_order LIKE LINE OF lt\_warehouse\_order.  
\*\*    DATA : lv\_missing\_case TYPE i.  
\*\*    DATA : lv\_active1 TYPE zcon\_param\_val-zactive.  
\*\*  
\*\*    lv\_missing\_case = 1.  
\*\*    CONSTANTS: lc\_ptwy TYPE /scwm/ordim\_o-procty VALUE '3090'.  
\*\*  
\*\*    DATA:lo\_badi\_transfer TYPE REF TO zewm\_cl\_mfs\_md\_rearr\_hu.  
\*\*    CONSTANTS: wmegc\_apl\_object\_wme     TYPE balobj\_d  VALUE '/SCWM/WME'.  
\*\*    CREATE OBJECT lo\_badi\_transfer.  
\*\*  
\*\*    DATA(lv\_queue) = iv\_queue+0(2).  
\*\*    DATA(lv\_crane\_no) = iv\_queue+6(2).  
\*\*  
\*\*    MOVE-CORRESPONDING ct\_who TO lt\_warehouse\_order.  
\*\*    MODIFY zwarehouse\_order FROM TABLE lt\_warehouse\_order.  
\*\*    COMMIT WORK.  
\*\*  
\*\*    SELECT \* INTO TABLE lt\_crane\_check FROM zcon\_param\_val  
\*\*                         WHERE lgnum =  iv\_lgnum  
\*\*                         AND zreq\_name = 'CRANE\_STATUS'.  
\*\*    IF sy-subrc EQ 0.  
\*\*  
\*\*    ENDIF.  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  START OF  LOAD IN rack 132  
\*\*  
\*\*    READ TABLE lt\_crane\_check INTO ls\_crane\_check WITH KEY zvalue = lv\_crane\_no.  
\*\*    IF sy-subrc EQ 0.  
\*\*      IF ls\_crane\_check-zactive EQ abap\_true.  
\*\*        REFRESH:ct\_who.  
\*\*        RETURN.  
\*\*      ENDIF.  
\*\*    ENDIF.  
\*\*  
\*\*  
\*\*    SELECT SINGLE zvalue INTO @DATA(lv\_wo\_value)  
\*\*                              FROM zcon\_param\_val  
\*\*                              WHERE lgnum = @iv\_lgnum  
\*\*                              AND zreq\_name = 'WO\_COUNT\_REQ'  
\*\*                              AND zparam = 'WO\_COUNT\_PARAM'  
\*\*                              AND zactive = @abap\_true.  
\*\*    IF sy-subrc EQ 0.  
\*\*      iv\_wo\_count = lv\_wo\_value.  
\*\*    ENDIF.  
\*\*  
\*\*  
\*\*    MOVE: wmegc\_apl\_object\_wme TO ls\_log-object,  
\*\*    zapl\_cl\_common\_constants=>gc\_det2 TO ls\_log-subobject.  
\*\*    CREATE OBJECT lo\_log.  
\*\*  
\*\*  
\*\*    IF lv\_queue EQ 'SW'.  
\*\*      IF ct\_who[] IS NOT INITIAL.  
\*\*  
\*\*        SELECT SINGLE zvalue INTO @DATA(lv\_det2) FROM zcon\_param\_val  
\*\*                             WHERE lgnum = @iv\_lgnum  
\*\*                             AND zreq\_name = 'DET2\_NEW\_132'  
\*\*                             AND zparam = 'DET2\_NEW\_132\_P'  
\*\*                             AND zactive EQ @abap\_true.  
\*\*        IF sy-subrc = 0.  
\*\*          lv\_id1 = zapl\_cl\_common\_constants=>gc\_agc\_scheduler.  
\*\*          CALL METHOD zapl\_cl\_common\_constants=>get\_param\_value  
\*\*            EXPORTING  
\*\*              iv\_lgnum = iv\_lgnum  
\*\*              iv\_id1   = lv\_id1  
\*\*            IMPORTING  
\*\*              et\_param = lt\_param\_agc\_schd.  
\*\*          LOOP AT lt\_param\_agc\_schd INTO ls\_param WHERE id2 = zapl\_cl\_common\_constants=>gc\_storage\_interleave.  
\*\*            ls\_hdr\_procty-hdr\_procty = ls\_param-value.  
\*\*            APPEND ls\_hdr\_procty TO lt\_str\_procty.  
\*\*          ENDLOOP.  
\*\*  
\*\*          SELECT \* FROM /scwm/who INTO TABLE lt\_str\_who UP TO iv\_wo\_count ROWS  
\*\*          FOR ALL ENTRIES IN lt\_str\_procty  
\*\*            WHERE lgnum = iv\_lgnum  
\*\*              AND hdr\_procty = lt\_str\_procty-hdr\_procty  
\*\*              AND status = wmegc\_wo\_open  
\*\*              AND queue = iv\_queue.  
\*\*          IF sy-subrc EQ 0.  
\*\*            READ TABLE lt\_str\_who INTO ls\_str\_who WITH KEY hdr\_procty = '5010'. " DE task  
\*\*  
\*\*            IF sy-subrc EQ 0 .  
\*\*              INSERT ls\_str\_who INTO ct\_who INDEX 1.  
\*\*              ls\_det2\_fg-lgnum = iv\_lgnum.  
\*\*              ls\_det2\_fg-crane\_no = lv\_crane\_no.  
\*\*              ls\_det2\_fg-who = ls\_str\_who-who.  
\*\*              ls\_det2\_fg-description = '5010 process type task sent to subsystem '.  
\*\*              ls\_det2\_fg-log\_date = sy-datum.  
\*\*              ls\_det2\_fg-log\_time = sy-uzeit.  
\*\*              APPEND ls\_det2\_fg TO lt\_det2\_fg.  
\*\*              APPEND ls\_det2\_fg TO lt\_det2\_fg.  
\*\*              MODIFY zdet2\_fg\_log FROM TABLE lt\_det2\_fg.  
\*\*              RETURN.  
\*\*              lv\_missing\_case = 2.  
\*\*            ELSE.  
\*\*              READ TABLE lt\_str\_who INTO ls\_str\_who WITH KEY hdr\_procty = '5020'. " ADHOC task  
\*\*              IF sy-subrc EQ 0.  
\*\*                INSERT ls\_str\_who INTO ct\_who INDEX 1.  
\*\*                ls\_det2\_fg-lgnum = iv\_lgnum.  
\*\*                ls\_det2\_fg-crane\_no = lv\_crane\_no.  
\*\*                ls\_det2\_fg-who = ls\_str\_who-who.  
\*\*                ls\_det2\_fg-description = '5020 process type task sent to subsystem '.  
\*\*                ls\_det2\_fg-log\_date = sy-datum.  
\*\*                ls\_det2\_fg-log\_time = sy-uzeit.  
\*\*                APPEND ls\_det2\_fg TO lt\_det2\_fg.  
\*\*                APPEND ls\_det2\_fg TO lt\_det2\_fg.  
\*\*                MODIFY zdet2\_fg\_log FROM TABLE lt\_det2\_fg.  
\*\*                RETURN.  
\*\*                lv\_missing\_case = 2.  
\*\*              ENDIF.  
\*\*            ENDIF.  
\*\*          ENDIF.  
\*\*  
\*\*  
\*\*          SELECT SINGLE zvalue INTO lv\_rewarehouse FROM zcon\_param\_val  
\*\*                               WHERE lgnum EQ iv\_lgnum  
\*\*                               AND zreq\_name EQ 'REWAREHOUSE\_NEW\_ENABLE'  
\*\*                               AND zparam EQ 'REWAREHOUSE\_NEW\_PARAM'  
\*\*                               AND zactive EQ abap\_true.  
\*\*          IF sy-subrc = 0.  
\*\*            lv\_id1 = zapl\_cl\_common\_constants=>gc\_agc\_scheduler.  
\*\*            CALL METHOD zapl\_cl\_common\_constants=>get\_param\_value  
\*\*              EXPORTING  
\*\*                iv\_lgnum = iv\_lgnum  
\*\*                iv\_id1   = lv\_id1  
\*\*              IMPORTING  
\*\*                et\_param = lt\_param\_agc\_schd.  
\*\*            LOOP AT lt\_param\_agc\_schd INTO ls\_param WHERE id2 = zapl\_cl\_common\_constants=>gc\_retrieval\_interleave.  
\*\*              ls\_hdr\_procty-hdr\_procty = ls\_param-value.  
\*\*              APPEND ls\_hdr\_procty TO lt\_rtv\_procty.  
\*\*              CLEAR ls\_hdr\_procty.  
\*\*            ENDLOOP.  
\*\*  
\*\*            SELECT \* FROM /scwm/who INTO TABLE lt\_rtv\_who UP TO iv\_wo\_count ROWS  
\*\*            FOR ALL ENTRIES IN lt\_rtv\_procty  
\*\*            WHERE lgnum = iv\_lgnum  
\*\*              AND hdr\_procty = lt\_rtv\_procty-hdr\_procty  
\*\*              AND status = wmegc\_wo\_open  
\*\*              AND queue = iv\_queue.  
\*\*  
\*\*            IF sy-subrc NE 0.  
\*\*  
\*\*            ENDIF.  
\*\*  
\*\*            LOOP AT lt\_param\_agc\_schd INTO ls\_param WHERE id2 = zapl\_cl\_common\_constants=>gc\_storage\_interleave.  
\*\*              ls\_hdr\_procty-hdr\_procty = ls\_param-value.  
\*\*              APPEND ls\_hdr\_procty TO lt\_str\_procty.  
\*\*              CLEAR ls\_hdr\_procty.  
\*\*            ENDLOOP.  
\*\*  
\*\*            SELECT \* FROM /scwm/who INTO TABLE lt\_str\_who UP TO iv\_wo\_count ROWS  
\*\*            FOR ALL ENTRIES IN lt\_str\_procty  
\*\*            WHERE lgnum = iv\_lgnum  
\*\*              AND hdr\_procty = lt\_str\_procty-hdr\_procty  
\*\*              AND status = wmegc\_wo\_open  
\*\*              AND queue = iv\_queue.  
\*\*            IF sy-subrc NE 0.  
\*\*  
\*\*            ENDIF.  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* When we have only retrival tasks  
\*\*            IF lt\_str\_who[] IS INITIAL AND lt\_rtv\_who[] IS NOT INITIAL.  
\*\*              READ TABLE lt\_rtv\_who INTO ls\_rtv\_who WITH KEY hdr\_procty = '3010'.  
\*\*              IF sy-subrc = 0.  
\*\*                INSERT ls\_rtv\_who INTO ct\_who INDEX 1.  
\*\*                READ TABLE ct\_who INTO cs\_who INDEX 1.  
\*\*                iv\_who = cs\_who-who.  
\*\*                ls\_det2\_fg-lgnum = iv\_lgnum.  
\*\*                ls\_det2\_fg-crane\_no = lv\_crane\_no.  
\*\*                ls\_det2\_fg-who = iv\_who.  
\*\*                ls\_det2\_fg-description = 'checking for picking task need rewarehousing '.  
\*\*                ls\_det2\_fg-log\_date = sy-datum.  
\*\*                ls\_det2\_fg-log\_time = sy-uzeit.  
\*\*                APPEND ls\_det2\_fg TO lt\_det2\_fg.  
\*\*                MODIFY zdet2\_fg\_log FROM TABLE lt\_det2\_fg.  
\*\*                COMMIT WORK.  
\*\*                lv\_missing\_case = 2.  
\*\*                CALL METHOD lo\_badi\_transfer->transfer\_hu1  
\*\*                  EXPORTING  
\*\*                    iv\_lgnum   = iv\_lgnum  
\*\*                    is\_whorder = is\_who  
\*\*                    iv\_who     = iv\_who  
\*\*                    iv\_rsrc    = is\_rsrc  
\*\*                    it\_ordim\_o = ct\_ordim\_o  
\*\*                  CHANGING  
\*\*                    ev\_who     = lv\_who.  
\*\*  
\*\*                CLEAR ls\_who.  
\*\*                IF lv\_who+0(1) NE 0.  
\*\*                  IF iv\_who NE lv\_who.  
\*\*                    "if current WO already contains WT for HU in front of HU to be picked => do nothing  
\*\*                    ls\_who-who = lv\_who.  
\*\*                    INSERT ls\_who INTO ct\_who INDEX 1.  
\*\*                    ls\_det2\_fg-lgnum = iv\_lgnum.  
\*\*                    ls\_det2\_fg-crane\_no = lv\_crane\_no.  
\*\*                    ls\_det2\_fg-who = lv\_who.  
\*\*                    ls\_det2\_fg-description = 'Rewarehousing task sent to subsystem Picking task'.  
\*\*                    ls\_det2\_fg-log\_date = sy-datum.  
\*\*                    ls\_det2\_fg-log\_time = sy-uzeit.  
\*\*                    APPEND ls\_det2\_fg TO lt\_det2\_fg.  
\*\*                    MODIFY zdet2\_fg\_log FROM TABLE lt\_det2\_fg.  
\*\*                    COMMIT WORK.  
\*\*  
\*\*                    UPDATE zinter\_leave SET active = abap\_true  
\*\*                                            who\_no = ls\_who-who  
\*\*                                        WHERE lgnum EQ iv\_lgnum AND resource\_name = iv\_queue.  
\*\*                    COMMIT WORK.  
\*\*                  ENDIF.  
\*\*                  cv\_rearr\_done = abap\_true.  
\*\*                  RETURN.  
\*\*                ELSE.  
\*\*                  SELECT \* INTO TABLE lt\_inter\_who FROM zinter\_leave  
\*\*                                                    WHERE lgnum = iv\_lgnum  
\*\*                                                    AND resource\_name = iv\_queue .  
\*\*                  IF sy-subrc = 0.  
\*\*                    LOOP AT lt\_inter\_who INTO ls\_inter\_who.  
\*\*                      READ TABLE ct\_who INTO cs\_who WITH KEY who = ls\_inter\_who-who\_no.  
\*\*                      IF sy-subrc = 0.  
\*\*                        "do nothing  
\*\*                      ELSE.  
\*\*                        UPDATE zinter\_leave SET active = abap\_false  
\*\*                                                who\_no = ls\_rtv\_who-who  
\*\*                                            WHERE lgnum EQ iv\_lgnum  
\*\*                                            AND resource\_name = iv\_queue.  
\*\*                        COMMIT WORK.  
\*\*                      ENDIF.  
\*\*                    ENDLOOP.  
\*\*                  ENDIF.  
\*\*                  RETURN.  
\*\*                ENDIF.  
\*\*  
\*\*              ENDIF."sy-subrc = 0 end  
\*\*  
\*\*              READ TABLE lt\_rtv\_who INTO ls\_rtv\_who WITH KEY hdr\_procty = '2010'.  
\*\*              IF sy-subrc = 0.  
\*\*                INSERT ls\_rtv\_who INTO ct\_who INDEX 1.  
\*\*                READ TABLE ct\_who INTO cs\_who INDEX 1.  
\*\*                iv\_who = cs\_who-who.  
\*\*                ls\_det2\_fg-lgnum = iv\_lgnum.  
\*\*                ls\_det2\_fg-crane\_no = lv\_crane\_no.  
\*\*                ls\_det2\_fg-who = iv\_who.  
\*\*                ls\_det2\_fg-description = 'checking for picking task need rewarehousing '.  
\*\*                ls\_det2\_fg-log\_date = sy-datum.  
\*\*                ls\_det2\_fg-log\_time = sy-uzeit.  
\*\*                APPEND ls\_det2\_fg TO lt\_det2\_fg.  
\*\*                MODIFY zdet2\_fg\_log FROM TABLE lt\_det2\_fg.  
\*\*  
\*\*                COMMIT WORK.  
\*\*                lv\_missing\_case = 2.  
\*\*                CALL METHOD lo\_badi\_transfer->transfer\_hu1  
\*\*                  EXPORTING  
\*\*                    iv\_lgnum   = iv\_lgnum  
\*\*                    is\_whorder = is\_who  
\*\*                    iv\_who     = iv\_who  
\*\*                    iv\_rsrc    = is\_rsrc  
\*\*                    it\_ordim\_o = ct\_ordim\_o  
\*\*                  CHANGING  
\*\*                    ev\_who     = lv\_who.  
\*\*  
\*\*                CLEAR ls\_who.  
\*\*                IF lv\_who+0(1) NE 0.  
\*\*                  IF iv\_who NE lv\_who.  
\*\*                    "if current WO already contains WT for HU in front of HU to be picked => do nothing  
\*\*                    ls\_who-who = lv\_who.  
\*\*                    INSERT ls\_who INTO ct\_who INDEX 1.  
\*\*                    ls\_det2\_fg-lgnum = iv\_lgnum.  
\*\*                    ls\_det2\_fg-crane\_no = lv\_crane\_no.  
\*\*                    ls\_det2\_fg-who = lv\_who.  
\*\*                    ls\_det2\_fg-description = 'Rewarehousing task sent to subsystem Picking task'.  
\*\*                    ls\_det2\_fg-log\_date = sy-datum.  
\*\*                    ls\_det2\_fg-log\_time = sy-uzeit.  
\*\*                    APPEND ls\_det2\_fg TO lt\_det2\_fg.  
\*\*                    MODIFY zdet2\_fg\_log FROM TABLE lt\_det2\_fg.  
\*\*                    COMMIT WORK.  
\*\*  
\*\*                    UPDATE zinter\_leave SET active = abap\_true  
\*\*                                            who\_no = ls\_who-who  
\*\*                                       WHERE lgnum EQ iv\_lgnum  
\*\*                                       AND resource\_name = iv\_queue.  
\*\*                    COMMIT WORK.  
\*\*                  ENDIF.  
\*\*                  cv\_rearr\_done = abap\_true.  
\*\*                  RETURN.  
\*\*                ELSE.  
\*\*                  SELECT \*  INTO TABLE lt\_inter\_who FROM zinter\_leave  
\*\*                                                    WHERE lgnum = iv\_lgnum  
\*\*                                                    AND resource\_name = iv\_queue .  
\*\*                  IF sy-subrc = 0.  
\*\*                    LOOP AT lt\_inter\_who INTO ls\_inter\_who.  
\*\*                      READ TABLE ct\_who INTO cs\_who WITH KEY who = ls\_inter\_who-who\_no.  
\*\*                      IF sy-subrc = 0.  
\*\*                        "do nothing  
\*\*                      ELSE.  
\*\*                        UPDATE zinter\_leave SET active = abap\_false  
\*\*                                                who\_no = ls\_rtv\_who-who  
\*\*                                            WHERE lgnum EQ iv\_lgnum  
\*\*                                            AND resource\_name = iv\_queue.  
\*\*                        COMMIT WORK.  
\*\*                      ENDIF.  
\*\*                    ENDLOOP.  
\*\*                  ENDIF.  
\*\*                  RETURN.  
\*\*                ENDIF.  
\*\*              ENDIF."sy-subrc = 0 end  
\*\*\*              ENDIF.  
\*\*            ENDIF." lt\_str\_who is initial and lt\_str\_who is not intial  
\*\*\*\*\*\*\*End of changes when we have only retrival tasks  
\*\*  
\*\*            READ TABLE lt\_rtv\_who INTO ls\_rtv\_who WITH KEY hdr\_procty = '3010'.  
\*\*            IF sy-subrc = 0.  
\*\*              SELECT \* INTO TABLE lt\_interleave FROM zinter\_leave  
\*\*                                                WHERE lgnum = iv\_lgnum  
\*\*                                                AND resource\_name = iv\_queue  
\*\*                                                AND active EQ abap\_true.  
\*\*              IF sy-subrc = 0.  
\*\*                READ TABLE lt\_rtv\_who INTO ls\_rtv\_who WITH KEY hdr\_procty = '3010'.  
\*\*                IF sy-subrc = 0.  
\*\*                  ls\_rewarehouse\_fg-lgnum = iv\_lgnum.  
\*\*                  ls\_rewarehouse\_fg-warehouse\_order = ls\_rtv\_who-who.  
\*\*                  SELECT \* INTO TABLE lt\_wo\_info FROM /scwm/ordim\_o  
\*\*                                                 WHERE lgnum = iv\_lgnum  
\*\*                                                 AND who = ls\_rtv\_who-who.  
\*\*                  IF sy-subrc = 0.  
\*\*                    READ TABLE lt\_wo\_info INTO ls\_wo\_info INDEX 1.  
\*\*                    ls\_rewarehouse\_fg-bin\_name = ls\_wo\_info-vlpla.  
\*\*                    ls\_rewarehouse\_fg-handling\_unit = ls\_wo\_info-vlenr.  
\*\*                    ls\_rewarehouse\_fg-crane\_number = ls\_wo\_info-vlpla+1(2).  
\*\*                    ls\_rewarehouse\_fg-log\_date = sy-datum.  
\*\*                    ls\_rewarehouse\_fg-log\_time = sy-uzeit.  
\*\*                    APPEND ls\_rewarehouse\_fg TO lt\_rewarehouse\_fg.  
\*\*                    MODIFY zrewareho\_fg\_log FROM TABLE lt\_rewarehouse\_fg.  
\*\*                    CLEAR:lt\_rewarehouse\_fg, lt\_wo\_info.  
\*\*                  ENDIF.  
\*\*                  INSERT ls\_rtv\_who INTO ct\_who INDEX 1.  
\*\*                  READ TABLE ct\_who INTO cs\_who INDEX 1.  
\*\*                  iv\_who = cs\_who-who.  
\*\*                  ls\_det2\_fg-lgnum = iv\_lgnum.  
\*\*                  ls\_det2\_fg-crane\_no = lv\_crane\_no.  
\*\*\*                    ls\_det2\_fg-error\_no = ls\_final\_error-error.  
\*\*                  ls\_det2\_fg-who = iv\_who.  
\*\*                  ls\_det2\_fg-description = 'checking for picking task need rewarehousing repl '.  
\*\*                  ls\_det2\_fg-log\_date = sy-datum.  
\*\*                  ls\_det2\_fg-log\_time = sy-uzeit.  
\*\*                  APPEND ls\_det2\_fg TO lt\_det2\_fg.  
\*\*                  MODIFY zdet2\_fg\_log FROM TABLE lt\_det2\_fg.  
\*\*                  COMMIT WORK.  
\*\*                  lv\_missing\_case = 2.  
\*\*                  CALL METHOD lo\_badi\_transfer->transfer\_hu1  
\*\*                    EXPORTING  
\*\*                      iv\_lgnum   = iv\_lgnum  
\*\*                      is\_whorder = is\_who  
\*\*                      iv\_who     = iv\_who  
\*\*                      iv\_rsrc    = is\_rsrc  
\*\*                      it\_ordim\_o = ct\_ordim\_o  
\*\*                    CHANGING  
\*\*                      ev\_who     = lv\_who.  
\*\*  
\*\*                  CLEAR ls\_who.  
\*\*                  IF lv\_who+0(1) NE 0.  
\*\*                    IF iv\_who NE lv\_who.  
\*\*                      "if current WO already contains WT for HU in front of HU to be picked => do nothing  
\*\*                      ls\_who-who = lv\_who.  
\*\*                      INSERT ls\_who INTO ct\_who INDEX 1.  
\*\*                      ls\_det2\_fg-lgnum = iv\_lgnum.  
\*\*                      ls\_det2\_fg-crane\_no = lv\_crane\_no.  
\*\*                      ls\_det2\_fg-tanum = ls\_str\_who-who.  
\*\*                      ls\_det2\_fg-description = 'Rewarehousing task sent to subsystem Picking task repl'.  
\*\*                      ls\_det2\_fg-log\_date = sy-datum.  
\*\*                      ls\_det2\_fg-log\_time = sy-uzeit.  
\*\*                      APPEND ls\_det2\_fg TO lt\_det2\_fg.  
\*\*                      MODIFY zdet2\_fg\_log FROM TABLE lt\_det2\_fg.  
\*\*                      COMMIT WORK.  
\*\*  
\*\*                      UPDATE zinter\_leave SET active = abap\_true  
\*\*                                              who\_no = ls\_who-who  
\*\*                                          WHERE lgnum EQ iv\_lgnum  
\*\*                                          AND resource\_name = iv\_queue.  
\*\*                      COMMIT WORK.  
\*\*                    ENDIF.  
\*\*                    cv\_rearr\_done = abap\_true.  
\*\*                    RETURN.  
\*\*                  ELSE.  
\*\*                    SELECT \*  INTO TABLE lt\_inter\_who FROM zinter\_leave  
\*\*                                                      WHERE lgnum = iv\_lgnum  
\*\*                                                      AND resource\_name = iv\_queue .  
\*\*                    IF sy-subrc = 0.  
\*\*                      LOOP AT lt\_inter\_who INTO ls\_inter\_who.  
\*\*                        READ TABLE ct\_who INTO cs\_who WITH KEY who = ls\_inter\_who-who\_no.  
\*\*                        IF sy-subrc = 0.  
\*\*                          "do nothing  
\*\*                        ELSE.  
\*\*                          UPDATE zinter\_leave SET active = abap\_false  
\*\*                                                  who\_no = ls\_rtv\_who-who  
\*\*                                              WHERE lgnum EQ iv\_lgnum  
\*\*                                              AND resource\_name = iv\_queue.  
\*\*                          COMMIT WORK.  
\*\*                        ENDIF.  
\*\*                      ENDLOOP.  
\*\*                    ENDIF.  
\*\*                    RETURN.  
\*\*                  ENDIF.  
\*\*  
\*\*                ENDIF."sy-subrc = 0 end  
\*\*  
\*\*              ENDIF.  
\*\*            ENDIF.  
\*\*  
\*\*            READ TABLE lt\_rtv\_who INTO ls\_rtv\_who WITH KEY hdr\_procty = '2010'.  
\*\*            IF sy-subrc = 0.  
\*\*              SELECT \* INTO TABLE lt\_interleave FROM zinter\_leave  
\*\*                                                WHERE lgnum = iv\_lgnum  
\*\*                                                AND resource\_name = iv\_queue  
\*\*                                                AND active EQ abap\_true.  
\*\*              IF sy-subrc = 0.  
\*\*                READ TABLE lt\_rtv\_who INTO ls\_rtv\_who WITH KEY hdr\_procty = '2010'.  
\*\*                IF sy-subrc = 0.  
\*\*                  INSERT ls\_rtv\_who INTO ct\_who INDEX 1.  
\*\*                  READ TABLE ct\_who INTO cs\_who INDEX 1.  
\*\*                  iv\_who = cs\_who-who.  
\*\*                  ls\_det2\_fg-lgnum = iv\_lgnum.  
\*\*                  ls\_det2\_fg-crane\_no = lv\_crane\_no.  
\*\*\*                    ls\_det2\_fg-error\_no = ls\_final\_error-error.  
\*\*                  ls\_det2\_fg-who = iv\_who.  
\*\*                  ls\_det2\_fg-description = 'checking for picking task need rewarehousing '.  
\*\*                  ls\_det2\_fg-log\_date = sy-datum.  
\*\*                  ls\_det2\_fg-log\_time = sy-uzeit.  
\*\*                  APPEND ls\_det2\_fg TO lt\_det2\_fg.  
\*\*                  MODIFY zdet2\_fg\_log FROM TABLE lt\_det2\_fg.  
\*\*                  COMMIT WORK.  
\*\*                  lv\_missing\_case = 2.  
\*\*                  CALL METHOD lo\_badi\_transfer->transfer\_hu1  
\*\*                    EXPORTING  
\*\*                      iv\_lgnum   = iv\_lgnum  
\*\*                      is\_whorder = is\_who  
\*\*                      iv\_who     = iv\_who  
\*\*                      iv\_rsrc    = is\_rsrc  
\*\*                      it\_ordim\_o = ct\_ordim\_o  
\*\*                    CHANGING  
\*\*                      ev\_who     = lv\_who.  
\*\*  
\*\*                  CLEAR ls\_who.  
\*\*                  IF lv\_who+0(1) NE 0.  
\*\*                    IF iv\_who NE lv\_who.  
\*\*                      "if current WO already contains WT for HU in front of HU to be picked => do nothing  
\*\*                      ls\_who-who = lv\_who.  
\*\*                      INSERT ls\_who INTO ct\_who INDEX 1.  
\*\*                      ls\_det2\_fg-lgnum = iv\_lgnum.  
\*\*                      ls\_det2\_fg-crane\_no = lv\_crane\_no.  
\*\*                      ls\_det2\_fg-who = lv\_who.  
\*\*                      ls\_det2\_fg-description = 'Rewarehousing task sent to subsystem Picking task'.  
\*\*                      ls\_det2\_fg-log\_date = sy-datum.  
\*\*                      ls\_det2\_fg-log\_time = sy-uzeit.  
\*\*                      APPEND ls\_det2\_fg TO lt\_det2\_fg.  
\*\*                      MODIFY zdet2\_fg\_log FROM TABLE lt\_det2\_fg.  
\*\*                      COMMIT WORK.  
\*\*  
\*\*                      UPDATE zinter\_leave SET active = abap\_true  
\*\*                                              who\_no = ls\_who-who  
\*\*                                          WHERE lgnum EQ iv\_lgnum  
\*\*                                          AND resource\_name = iv\_queue.  
\*\*                      COMMIT WORK.  
\*\*                    ENDIF.  
\*\*                    cv\_rearr\_done = abap\_true.  
\*\*                    RETURN.  
\*\*                  ELSE.  
\*\*                    SELECT \*  INTO TABLE lt\_inter\_who FROM zinter\_leave  
\*\*                                                      WHERE lgnum = iv\_lgnum  
\*\*                                                      AND resource\_name = iv\_queue .  
\*\*                    IF sy-subrc = 0.  
\*\*                      LOOP AT lt\_inter\_who INTO ls\_inter\_who.  
\*\*                        READ TABLE ct\_who INTO cs\_who WITH KEY who = ls\_inter\_who-who\_no.  
\*\*                        IF sy-subrc = 0.  
\*\*                          "do nothing  
\*\*                        ELSE.  
\*\*                          UPDATE zinter\_leave SET active = abap\_false  
\*\*                                                  who\_no = ls\_rtv\_who-who  
\*\*                                              WHERE lgnum EQ iv\_lgnum  
\*\*                                              AND resource\_name = iv\_queue.  
\*\*                          COMMIT WORK.  
\*\*                        ENDIF.  
\*\*                      ENDLOOP.  
\*\*                    ENDIF.  
\*\*                    RETURN.  
\*\*                  ENDIF.  
\*\*                ENDIF."sy-subrc = 0 end  
\*\*  
\*\*              ENDIF.  
\*\*            ENDIF.  
\*\*  
\*\*\*            ENDIF.  
\*\*          ENDIF.  
\*\*        ENDIF."sy-subrc  EQ 0  
\*\*  
\*\*        SELECT \* INTO TABLE @DATA(lt\_dprcty) FROM zcon\_param\_val  
\*\*                            WHERE lgnum = @iv\_lgnum  
\*\*                            AND zreq\_name = 'DET2\_PRCTY'  
\*\*                            AND zparam = 'DET2\_PRCTY\_PARAM'  
\*\*                            AND zactive = @abap\_true.  
\*\*        IF sy-subrc EQ 0.  
\*\*          LOOP AT lt\_dprcty INTO DATA(ls\_dprcty).  
\*\*            ls\_prcty-sign = 'I'.  
\*\*            ls\_prcty-option = 'EQ'.  
\*\*            ls\_prcty-low = ls\_dprcty-zvalue.  
\*\*            APPEND ls\_prcty TO lt\_prcty.  
\*\*          ENDLOOP.  
\*\*        ENDIF.  
\*\*        SELECT vlenr tanum INTO TABLE lt\_hu\_tanum FROM /scwm/ordim\_o  
\*\*                                                  FOR ALL ENTRIES IN ct\_who  
\*\*                                                  WHERE lgnum = iv\_lgnum  
\*\*                                                  AND queue = ct\_who-queue  
\*\*                                                  AND  who = ct\_who-who  
\*\*                                                  AND procty IN lt\_prcty.  
\*\*        IF sy-subrc EQ 0.  
\*\*          CLEAR:ls\_who.  
\*\*        ENDIF.  
\*\*      ENDIF."ct\_who[] is not initial  
\*\*  
\*\*      IF lt\_hu\_tanum[] IS NOT INITIAL.  
\*\*        LOOP AT lt\_hu\_tanum ASSIGNING FIELD-SYMBOL(<fs\_hu\_tanum>).  
\*\*          DATA(lv\_hu) = <fs\_hu\_tanum>-vlenr.  
\*\*          SHIFT lv\_hu LEFT DELETING LEADING '0'.    "to delete leading zero from handling unit  
\*\*          <fs\_hu\_tanum>-vlenr = lv\_hu.  
\*\*        ENDLOOP.  
\*\*        SELECT SINGLE zvalue INTO @DATA(lv\_seconds)  
\*\*                             FROM zcon\_param\_val  
\*\*                             WHERE lgnum = @iv\_lgnum  
\*\*                             AND zreq\_name = 'TASK\_DELAY\_REQ'  
\*\*                             AND zparam = 'TASK\_DELAY\_PARAM'  
\*\*                             AND zactive = @abap\_true.  
\*\*        IF sy-subrc EQ 0.  
\*\*          SELECT \* INTO TABLE lt\_error  FROM zcrane\_dashboard  
\*\*                        FOR ALL ENTRIES IN lt\_hu\_tanum  
\*\*                        WHERE lgnum = iv\_lgnum  
\*\*                        AND crane\_no  = lv\_crane\_no  
\*\*                        AND huident = lt\_hu\_tanum-vlenr  
\*\*                        AND error = lv\_load\_in\_rack .  
\*\*          IF sy-subrc EQ 0.  
\*\*            WAIT UP TO lv\_seconds SECONDS.  
\*\*          ENDIF.  
\*\*        ENDIF.  
\*\*      ENDIF.  
\*\*  
\*\*      LOOP AT lt\_hu\_tanum INTO ls\_hu\_tanum.  
\*\*        READ TABLE lt\_error INTO ls\_error WITH KEY huident = ls\_hu\_tanum-vlenr.  
\*\*        IF sy-subrc EQ 0.  
\*\*          ls\_det2\_fg-lgnum = iv\_lgnum.  
\*\*          ls\_det2\_fg-crane\_no = lv\_crane\_no.  
\*\*          ls\_det2\_fg-handling\_unit = ls\_hu\_tanum-vlenr.  
\*\*          ls\_det2\_fg-error\_no = ls\_error-error.  
\*\*          ls\_det2\_fg-tanum = ls\_hu\_tanum-tanum.  
\*\*          ls\_det2\_fg-description = '132 LOAD RACK OCCURED'.  
\*\*          ls\_det2\_fg-log\_date = sy-datum.  
\*\*          ls\_det2\_fg-log\_time = sy-uzeit.  
\*\*          APPEND ls\_det2\_fg TO lt\_det2\_fg.  
\*\*          MODIFY zdet2\_fg\_log FROM TABLE lt\_det2\_fg.  
\*\*          COMMIT WORK.  
\*\*  
\*\*          MOVE-CORRESPONDING ls\_who\_to TO ls\_who.  
\*\*          INSERT ls\_who INTO ct\_who INDEX 1.  
\*\*  
\*\*          CLEAR: ls\_132\_error,lv\_msgtext,lt\_132\_error.  
\*\*          ls\_det2\_fg-lgnum = iv\_lgnum.  
\*\*          ls\_det2\_fg-crane\_no = lv\_crane\_no.  
\*\*          ls\_det2\_fg-handling\_unit = ls\_hu\_tanum-vlenr.  
\*\*          ls\_det2\_fg-error\_no = ls\_error-error.  
\*\*          ls\_det2\_fg-tanum = ls\_who-who.  
\*\*          ls\_det2\_fg-description = 'Det2 132 task sent to subsystem'.  
\*\*          ls\_det2\_fg-log\_date = sy-datum.  
\*\*          ls\_det2\_fg-log\_time = sy-uzeit.  
\*\*          APPEND ls\_det2\_fg TO lt\_det2\_fg.  
\*\*          MODIFY zdet2\_fg\_log FROM TABLE lt\_det2\_fg.  
\*\*          COMMIT WORK.  
\*\*          CONCATENATE ls\_who-who 'DET2 132 trigger task sent' INTO lv\_msgtext.  
\*\*          ls\_132\_error-lgnum = iv\_lgnum.  
\*\*          ls\_132\_error-crane\_no = ls\_who-queue+7(2).  
\*\*          ls\_132\_error-log\_date = sy-datum.  
\*\*          ls\_132\_error-log\_time = sy-uzeit.  
\*\*          ls\_132\_error-description = lv\_msgtext.  
\*\*          APPEND ls\_132\_error TO lt\_132\_error.  
\*\*          MODIFY z132\_error\_log FROM TABLE lt\_132\_error.  
\*\*          COMMIT WORK.  
\*\*\*          lo\_log->save\_applog2db( ).  
\*\*          RETURN.  
\*\*        ENDIF.  
\*\*      ENDLOOP.  
\*\*  
\*\*    ENDIF.  
\*\*  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*END OF LOAD IN RACK 132  
\*\*  
\*\*  
\*\*    IF lv\_queue EQ 'SW'. "ALLOWED INTERLEAVING FOR Swislog only  
\*\*      IF lt\_hu\_tanum[] IS NOT INITIAL.  
\*\*        SELECT \* INTO TABLE @DATA(lt\_final\_error) FROM zcrane\_dashboard  
\*\*                            FOR ALL ENTRIES IN @lt\_hu\_tanum  
\*\*                            WHERE lgnum = @iv\_lgnum  
\*\*                            AND crane\_no = @lv\_crane\_no  
\*\*                            AND huident = @lt\_hu\_tanum-vlenr.  
\*\*        IF lt\_final\_error[] IS INITIAL.  
\*\*          LOOP AT lt\_final\_error INTO DATA(ls\_final\_error).  
\*\*            READ TABLE lt\_hu\_tanum INTO ls\_hu\_tanum WITH KEY vlenr = ls\_final\_error-huident.  
\*\*            IF sy-subrc NE 0.  
\*\*              MOVE ls\_hu\_tanum-vlenr  TO ls\_log-extnumber.  
\*\*\*              CALL METHOD lo\_log->create\_log  
\*\*\*                EXPORTING  
\*\*\*                  is\_log = ls\_log.  
\*\*\*              MESSAGE s032(zapl\_det) WITH ls\_final\_error-huident  INTO lv\_msgtext.  
\*\*\*              lo\_log->add\_message2log( ).  
\*\*              ls\_det2\_fg-lgnum = iv\_lgnum.  
\*\*              ls\_det2\_fg-crane\_no = lv\_crane\_no.  
\*\*              ls\_det2\_fg-handling\_unit = ls\_final\_error-huident.  
\*\*              ls\_det2\_fg-error\_no = ls\_final\_error-error.  
\*\*              ls\_det2\_fg-tanum = ls\_final\_error-open\_task.  
\*\*              ls\_det2\_fg-description = 'HU has no error'.  
\*\*              APPEND ls\_det2\_fg TO lt\_det2\_fg.  
\*\*              MODIFY zdet2\_fg\_log FROM TABLE lt\_det2\_fg.  
\*\*              COMMIT WORK.  
\*\*            ENDIF.  
\*\*          ENDLOOP.  
\*\*  
\*\*          SELECT SINGLE zvalue INTO @DATA(ls\_zvalue)  
\*\*                      FROM zcon\_param\_val  
\*\*                      WHERE lgnum = @iv\_lgnum  
\*\*                      AND zreq\_name = 'INTERLEAVING'  
\*\*                      AND zparam = 'INTERLEAVE\_PARAM'  
\*\*                      AND zactive = @abap\_true.  
\*\*          IF sy-subrc EQ 0.  
\*\*\*    REFRESH: ct\_who[].  
\*\*\*\*-- MFS Whse process type constants  
\*\*            lv\_id1 = zapl\_cl\_common\_constants=>gc\_mfs\_tele\_send.  
\*\*            CALL METHOD zapl\_cl\_common\_constants=>get\_param\_value  
\*\*              EXPORTING  
\*\*                iv\_lgnum = iv\_lgnum  
\*\*                iv\_id1   = lv\_id1  
\*\*              IMPORTING  
\*\*                et\_param = lt\_param\_mfs\_tele\_send.  
\*\*  
\*\*\*-- Scheduler constants  
\*\*            lv\_id1 = zapl\_cl\_common\_constants=>gc\_agc\_scheduler.  
\*\*            CALL METHOD zapl\_cl\_common\_constants=>get\_param\_value  
\*\*              EXPORTING  
\*\*                iv\_lgnum = iv\_lgnum  
\*\*                iv\_id1   = lv\_id1  
\*\*              IMPORTING  
\*\*                et\_param = lt\_param\_agc\_schd.  
\*\*  
\*\*            CLEAR ls\_param.  
\*\*            READ TABLE lt\_param\_agc\_schd INTO ls\_param WITH KEY id2 = zapl\_cl\_common\_constants=>gc\_max\_direct\_command.  
\*\*            IF sy-subrc = 0.  
\*\*              MOVE ls\_param-value TO lv\_max\_direct .  
\*\*            ENDIF.  
\*\*  
\*\*            CLEAR ls\_param.  
\*\*            READ TABLE lt\_param\_agc\_schd INTO ls\_param WITH KEY id2 = zapl\_cl\_common\_constants=>gc\_max\_storage\_command.  
\*\*            IF sy-subrc = 0.  
\*\*              MOVE ls\_param-value TO lv\_max\_storage .  
\*\*            ENDIF.  
\*\*  
\*\*            CLEAR ls\_param.  
\*\*            READ TABLE lt\_param\_agc\_schd INTO ls\_param WITH KEY id2 = zapl\_cl\_common\_constants=>gc\_max\_retrieve\_command.  
\*\*            IF sy-subrc = 0.  
\*\*              MOVE ls\_param-value TO lv\_max\_retrieve .  
\*\*            ENDIF.  
\*\*  
\*\*\*-- Read Whse procty for Storage, retrieval and Direct transfer operations  
\*\*            CLEAR : ls\_param.  
\*\*            READ TABLE lt\_param\_agc\_schd INTO ls\_param WITH KEY id2 = zapl\_cl\_common\_constants=>gc\_direct\_transfer.  
\*\*            IF sy-subrc = 0.  
\*\*              ls\_hdr\_procty-hdr\_procty = ls\_param-value .  
\*\*              APPEND ls\_hdr\_procty TO lt\_dir\_procty.  
\*\*            ENDIF.  
\*\*  
\*\*            CLEAR : ls\_param.  
\*\*            READ TABLE lt\_param\_agc\_schd INTO ls\_param WITH KEY id2 = zapl\_cl\_common\_constants=>gc\_retrieval\_interleave.  
\*\*            IF sy-subrc = 0.  
\*\*              ls\_hdr\_procty-hdr\_procty = ls\_param-value.  
\*\*              APPEND ls\_hdr\_procty TO lt\_rtv\_procty.  
\*\*            ENDIF.  
\*\*            READ TABLE lt\_param\_agc\_schd INTO ls\_param WITH KEY id2 = zapl\_cl\_common\_constants=>gc\_repl\_interleave.  
\*\*            IF sy-subrc = 0.  
\*\*              ls\_hdr\_procty-hdr\_procty = ls\_param-value.  
\*\*              APPEND ls\_hdr\_procty TO lt\_repl\_procty.  
\*\*            ENDIF.  
\*\*  
\*\*            CLEAR : ls\_param.  
\*\*            LOOP AT lt\_param\_agc\_schd INTO ls\_param WHERE id2 = zapl\_cl\_common\_constants=>gc\_storage\_interleave.  
\*\*              ls\_hdr\_procty-hdr\_procty = ls\_param-value.  
\*\*              APPEND ls\_hdr\_procty TO lt\_str\_procty.  
\*\*            ENDLOOP.  
\*\*  
\*\*\*    CLEAR :  lt\_who[] , ct\_who[], lv\_count, lv\_retrieve, lv\_storage, lv\_direct.  
\*\*  
\*\*\*-- get all 'in process' WO, to check current WO count for each operation.Based on current count  
\*\*\*   and maximum allowed count, next set of WOs shall be decided and sent  
\*\*            SELECT \* FROM /scwm/who INTO TABLE lt\_who  
\*\*                      WHERE lgnum = iv\_lgnum  
\*\*                        AND status = wmegc\_wo\_open.  
\*\*  
\*\*            IF NOT lt\_who[] IS INITIAL.  
\*\*  
\*\*              LOOP AT lt\_who ASSIGNING <ls\_who>.  
\*\*  
\*\*  
\*\*\*-- Retrieval  
\*\*                READ TABLE lt\_rtv\_procty TRANSPORTING NO FIELDS WITH KEY hdr\_procty = <ls\_who>-hdr\_procty.  
\*\*                IF sy-subrc = 0.  
\*\*                  lv\_retrieve = lv\_retrieve + 1.  
\*\*                ENDIF.  
\*\*\*\*\*                Replineshment  
\*\*                READ TABLE lt\_repl\_procty TRANSPORTING NO FIELDS WITH KEY hdr\_procty = <ls\_who>-hdr\_procty.  
\*\*                IF sy-subrc = 0.  
\*\*                  lv\_retrieve = lv\_retrieve + 1.  
\*\*                ENDIF.  
\*\*  
\*\*\*-- Storage  
\*\*                READ TABLE lt\_str\_procty TRANSPORTING NO FIELDS WITH KEY hdr\_procty = <ls\_who>-hdr\_procty.  
\*\*                IF sy-subrc = 0.  
\*\*                  lv\_storage = lv\_storage + 1.  
\*\*                ENDIF.  
\*\*              ENDLOOP.  
\*\*            ENDIF.  
\*\*  
\*\*            IF lv\_max\_retrieve = lv\_retrieve.  
\*\*              CLEAR lv\_msgtext.  
\*\*              MESSAGE w432(zapl\_asrs) WITH  lv\_retrieve 'Retrieve' INTO lv\_msgtext.  
\*\*              lo\_log->add\_message( ).  
\*\*            ENDIF.  
\*\*  
\*\*            IF lv\_max\_storage = lv\_storage.  
\*\*              CLEAR lv\_msgtext.  
\*\*              MESSAGE w432(zapl\_asrs) WITH  lv\_storage 'Storage' INTO lv\_msgtext.  
\*\*              lo\_log->add\_message( ).  
\*\*            ENDIF.  
\*\*  
\*\*  
\*\*  
\*\*\*-- Validate maximum WO limit and select WO  acordingly  
\*\*  
\*\*            lv\_retrieve = lv\_max\_retrieve - lv\_retrieve.  
\*\*            lv\_storage = lv\_max\_storage - lv\_storage.  
\*\*            lv\_direct = lv\_max\_direct - lv\_direct.  
\*\*  
\*\*\*            IF lv\_storage > 0 .  
\*\*  
\*\*            SELECT \* FROM /scwm/who INTO TABLE lt\_str\_who UP TO iv\_wo\_count ROWS  
\*\*            FOR ALL ENTRIES IN lt\_str\_procty  
\*\*              WHERE lgnum = iv\_lgnum  
\*\*                AND hdr\_procty = lt\_str\_procty-hdr\_procty  
\*\*                AND status = wmegc\_wo\_open  
\*\*                AND queue = iv\_queue.  
\*\*  
\*\*\*            ENDIF.  
\*\*  
\*\*\*            IF lv\_retrieve > 0.  
\*\*            SELECT \* FROM /scwm/who INTO  TABLE  lt\_rtv\_who UP TO iv\_wo\_count ROWS  
\*\*              FOR ALL ENTRIES IN lt\_rtv\_procty  
\*\*                WHERE lgnum = iv\_lgnum  
\*\*                  AND hdr\_procty = lt\_rtv\_procty-hdr\_procty  
\*\*                  AND status = wmegc\_wo\_open  
\*\*                  AND queue = iv\_queue.  
\*\*\*\*\*\*\*\*\*\*                REplenishment  
\*\*            SELECT \* FROM /scwm/who INTO  TABLE lt\_repl\_who UP TO iv\_wo\_count ROWS  
\*\*               FOR ALL ENTRIES IN lt\_repl\_procty  
\*\*              WHERE lgnum = iv\_lgnum  
\*\*              AND hdr\_procty = lt\_repl\_procty-hdr\_procty  
\*\*              AND status = wmegc\_wo\_open  
\*\*              AND queue = iv\_queue.  
\*\*\*            ENDIF .  
\*\*  
\*\*            SELECT SINGLE zvalue INTO @DATA(lv\_repl)  
\*\*                    FROM zcon\_param\_val  
\*\*                    WHERE lgnum = @iv\_lgnum  
\*\*                    AND zreq\_name = 'REPL\_DET2\_REQ'  
\*\*                    AND zparam = 'REPL\_DET2\_PARAM'  
\*\*                    AND zactive = @abap\_true.  
\*\*  
\*\*            IF sy-subrc = 0.  
\*\*              IF lt\_repl\_who[] IS NOT INITIAL.  
\*\*                lt\_rtv\_who1[] = lt\_rtv\_who[].  
\*\*                lt\_rtv\_who[] = lt\_repl\_who[].  
\*\*                APPEND LINES OF lt\_rtv\_who1 TO lt\_rtv\_who.  
\*\*              ENDIF.  
\*\*            ENDIF.  
\*\*            DATA(lv\_str\_cnt) = lines( lt\_str\_who ) .  
\*\*            DATA(lv\_rtv\_cnt) = lines( lt\_rtv\_who ).  
\*\*  
\*\*            MOVE iv\_lgnum  TO ls\_log-extnumber.  
\*\*            CALL METHOD lo\_log->create\_log  
\*\*              EXPORTING  
\*\*                is\_log = ls\_log.  
\*\*  
\*\*            MESSAGE s033(zapl\_det) WITH 'Interleaving block started'  INTO lv\_msgtext.  
\*\*            lo\_log->add\_message2log( ).  
\*\*  
\*\*\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*When we have only putaway tasks  
\*\*            IF lt\_str\_who[] IS  NOT INITIAL AND lt\_rtv\_who[] IS INITIAL.  
\*\*              READ TABLE lt\_str\_who INTO ls\_str\_who WITH KEY hdr\_procty = '3090'.  
\*\*              IF sy-subrc = 0.  
\*\*                SELECT \* INTO TABLE lt\_interleave FROM zinter\_leave  
\*\*                         WHERE lgnum = iv\_lgnum  
\*\*                         AND resource\_name = iv\_queue  
\*\*                         AND active EQ abap\_true.  
\*\*                IF sy-subrc NE 0.  
\*\*                  INSERT ls\_str\_who INTO ct\_who INDEX 1.  
\*\*  
\*\*                  SELECT \*  INTO TABLE lt\_inter\_who  
\*\*                             FROM zinter\_leave  
\*\*                             WHERE lgnum = iv\_lgnum  
\*\*                             AND resource\_name = iv\_queue .  
\*\*                  IF sy-subrc = 0.  
\*\*                    READ TABLE lt\_inter\_who INTO ls\_inter\_who INDEX 1.  
\*\*                    IF ls\_str\_who-who NE ls\_inter\_who-who\_no.  
\*\*                      UPDATE zinter\_leave SET active = abap\_false  
\*\*                                              who\_no = ls\_str\_who-who  
\*\*                                             WHERE lgnum EQ iv\_lgnum AND resource\_name = iv\_queue.  
\*\*                      COMMIT WORK.  
\*\*                      lv\_missing\_case = 2.  
\*\*                    ENDIF.  
\*\*                  ENDIF.  
\*\*                  RETURN.  
\*\*                ENDIF.  
\*\*  
\*\*              ENDIF.  
\*\*  
\*\*            ENDIF.  
\*\*\*\*\*\*\*\*\*\*\*\* end of when we have only putaway tasks  
\*\*  
\*\*\*\*\*\*\*\*\*\*\*\*\* When we have only retrival tasks  
\*\*  
\*\*  
\*\*            SELECT SINGLE zactive INTO lv\_active1 FROM zcon\_param\_val  
\*\*                              WHERE lgnum = iv\_lgnum AND zreq\_name = 'ZINTER' AND zactive EQ abap\_true.  
\*\*            IF lv\_active1 EQ abap\_true.  
\*\*              IF lt\_str\_who[] IS INITIAL AND lt\_rtv\_who[] IS NOT INITIAL.  
\*\*                READ TABLE lt\_rtv\_who INTO ls\_rtv\_who WITH KEY hdr\_procty = '3010'.  
\*\*                IF sy-subrc = 0.  
\*\*  
\*\*  
\*\*                  INSERT ls\_rtv\_who INTO ct\_who INDEX 1.  
\*\*                  READ TABLE ct\_who INTO cs\_who INDEX 1.  
\*\*                  iv\_who = cs\_who-who.  
\*\*                  ls\_det2\_fg-lgnum = iv\_lgnum.  
\*\*                  ls\_det2\_fg-crane\_no = lv\_crane\_no.  
\*\*                  ls\_det2\_fg-error\_no = ls\_final\_error-error.  
\*\*                  ls\_det2\_fg-who = iv\_who.  
\*\*                  ls\_det2\_fg-description = 'checking for picking task need rewarehousing repl '.  
\*\*                  ls\_det2\_fg-log\_date = sy-datum.  
\*\*                  ls\_det2\_fg-log\_time = sy-uzeit.  
\*\*                  APPEND ls\_det2\_fg TO lt\_det2\_fg.  
\*\*                  MODIFY zdet2\_fg\_log FROM TABLE lt\_det2\_fg.  
\*\*                  COMMIT WORK.  
\*\*  
\*\*                  lv\_missing\_case = 2.  
\*\*                  CALL METHOD lo\_badi\_transfer->transfer\_hu1  
\*\*                    EXPORTING  
\*\*                      iv\_lgnum   = iv\_lgnum  
\*\*                      is\_whorder = is\_who  
\*\*                      iv\_who     = iv\_who  
\*\*                      iv\_rsrc    = is\_rsrc  
\*\*                      it\_ordim\_o = ct\_ordim\_o  
\*\*                    CHANGING  
\*\*                      ev\_who     = lv\_who.  
\*\*  
\*\*                  CLEAR ls\_who.  
\*\*                  IF lv\_who+0(1) NE 0.  
\*\*                    IF iv\_who NE lv\_who.  
\*\*                      "if current WO already contains WT for HU in front of HU to be picked => do nothing  
\*\*                      ls\_who-who = lv\_who.  
\*\*                      INSERT ls\_who INTO ct\_who INDEX 1.  
\*\*                      ls\_det2\_fg-lgnum = iv\_lgnum.  
\*\*                      ls\_det2\_fg-crane\_no = lv\_crane\_no.  
\*\*                      ls\_det2\_fg-who = lv\_who.  
\*\*                      ls\_det2\_fg-description = 'Rewarehousing task sent to subsystem Picking task repl'.  
\*\*                      ls\_det2\_fg-log\_date = sy-datum.  
\*\*                      ls\_det2\_fg-log\_time = sy-uzeit.  
\*\*                      APPEND ls\_det2\_fg TO lt\_det2\_fg.  
\*\*                      MODIFY zdet2\_fg\_log FROM TABLE lt\_det2\_fg.  
\*\*                      COMMIT WORK.  
\*\*  
\*\*                      UPDATE zinter\_leave SET active = abap\_true  
\*\*                                              who\_no = ls\_who-who  
\*\*                                           WHERE lgnum EQ iv\_lgnum  
\*\*                                           AND resource\_name = iv\_queue.  
\*\*                      COMMIT WORK.  
\*\*                    ENDIF.  
\*\*                    cv\_rearr\_done = abap\_true.  
\*\*                    RETURN.  
\*\*                  ELSE.  
\*\*                    SELECT \*  INTO TABLE lt\_inter\_who FROM zinter\_leave  
\*\*                               WHERE lgnum = iv\_lgnum  
\*\*                               AND resource\_name = iv\_queue .  
\*\*                    IF sy-subrc = 0.  
\*\*                      LOOP AT lt\_inter\_who INTO ls\_inter\_who.  
\*\*                        READ TABLE ct\_who INTO cs\_who WITH KEY who = ls\_inter\_who-who\_no.  
\*\*                        IF sy-subrc = 0.  
\*\*                          "do nothing  
\*\*                        ELSE.  
\*\*                          UPDATE zinter\_leave SET active = abap\_false  
\*\*                                                  who\_no = ls\_rtv\_who-who  
\*\*                                              WHERE lgnum EQ iv\_lgnum  
\*\*                                              AND resource\_name = iv\_queue.  
\*\*                          COMMIT WORK.  
\*\*                        ENDIF.  
\*\*                      ENDLOOP.  
\*\*                    ENDIF.  
\*\*                    RETURN.  
\*\*                  ENDIF.  
\*\*  
\*\*                  SELECT \*  INTO TABLE lt\_inter\_who FROM zinter\_leave  
\*\*                            WHERE lgnum = iv\_lgnum  
\*\*                           AND resource\_name = iv\_queue .  
\*\*                  IF sy-subrc = 0.  
\*\*                    LOOP AT lt\_inter\_who INTO ls\_inter\_who.  
\*\*                      READ TABLE ct\_who INTO cs\_who WITH KEY who = ls\_inter\_who-who\_no.  
\*\*                      IF sy-subrc = 0.  
\*\*                        "do nothing  
\*\*                      ELSE.  
\*\*                        UPDATE zinter\_leave SET active = abap\_true  
\*\*                                                who\_no = ls\_rtv\_who-who  
\*\*                                            WHERE lgnum EQ iv\_lgnum  
\*\*                                            AND resource\_name = iv\_queue.  
\*\*                        COMMIT WORK.  
\*\*                      ENDIF.  
\*\*                    ENDLOOP.  
\*\*                  ENDIF.  
\*\*                  RETURN.  
\*\*                ENDIF.  
\*\*  
\*\*                READ TABLE lt\_rtv\_who INTO ls\_rtv\_who WITH KEY hdr\_procty = '2010'.  
\*\*                IF sy-subrc = 0.  
\*\*  
\*\*                  INSERT ls\_rtv\_who INTO ct\_who INDEX 1.  
\*\*                  READ TABLE ct\_who INTO cs\_who INDEX 1.  
\*\*                  iv\_who = cs\_who-who.  
\*\*                  ls\_det2\_fg-lgnum = iv\_lgnum.  
\*\*                  ls\_det2\_fg-crane\_no = lv\_crane\_no.  
\*\*                  ls\_det2\_fg-error\_no = ls\_final\_error-error.  
\*\*                  ls\_det2\_fg-who = iv\_who.  
\*\*                  ls\_det2\_fg-description = 'checking for picking task need rewarehousing '.  
\*\*                  ls\_det2\_fg-log\_date = sy-datum.  
\*\*                  ls\_det2\_fg-log\_time = sy-uzeit.  
\*\*                  APPEND ls\_det2\_fg TO lt\_det2\_fg.  
\*\*                  MODIFY zdet2\_fg\_log FROM TABLE lt\_det2\_fg.  
\*\*  
\*\*                  COMMIT WORK.  
\*\*                  lv\_missing\_case = 2.  
\*\*                  CALL METHOD lo\_badi\_transfer->transfer\_hu1  
\*\*                    EXPORTING  
\*\*                      iv\_lgnum   = iv\_lgnum  
\*\*                      is\_whorder = is\_who  
\*\*                      iv\_who     = iv\_who  
\*\*                      iv\_rsrc    = is\_rsrc  
\*\*                      it\_ordim\_o = ct\_ordim\_o  
\*\*                    CHANGING  
\*\*                      ev\_who     = lv\_who.  
\*\*  
\*\*                  CLEAR ls\_who.  
\*\*                  IF lv\_who+0(1) NE 0.  
\*\*                    IF iv\_who NE lv\_who.  
\*\*                      "if current WO already contains WT for HU in front of HU to be picked => do nothing  
\*\*                      ls\_who-who = lv\_who.  
\*\*                      INSERT ls\_who INTO ct\_who INDEX 1.  
\*\*                      ls\_det2\_fg-lgnum = iv\_lgnum.  
\*\*                      ls\_det2\_fg-crane\_no = lv\_crane\_no.  
\*\*                      ls\_det2\_fg-who = lv\_who.  
\*\*                      ls\_det2\_fg-description = 'Rewarehousing task sent to subsystem Picking task'.  
\*\*                      ls\_det2\_fg-log\_date = sy-datum.  
\*\*                      ls\_det2\_fg-log\_time = sy-uzeit.  
\*\*                      APPEND ls\_det2\_fg TO lt\_det2\_fg.  
\*\*                      MODIFY zdet2\_fg\_log FROM TABLE lt\_det2\_fg.  
\*\*                      COMMIT WORK.  
\*\*  
\*\*                      UPDATE zinter\_leave SET active = abap\_true  
\*\*                                              who\_no = ls\_who-who  
\*\*                                          WHERE lgnum EQ iv\_lgnum AND resource\_name = iv\_queue.  
\*\*                      COMMIT WORK.  
\*\*                    ENDIF.  
\*\*                    cv\_rearr\_done = abap\_true.  
\*\*                    RETURN.  
\*\*                  ELSE.  
\*\*                    SELECT \*  INTO TABLE lt\_inter\_who FROM zinter\_leave  
\*\*                             WHERE lgnum = iv\_lgnum AND resource\_name = iv\_queue .  
\*\*                    IF sy-subrc = 0.  
\*\*                      LOOP AT lt\_inter\_who INTO ls\_inter\_who.  
\*\*                        READ TABLE ct\_who INTO cs\_who WITH KEY who = ls\_inter\_who-who\_no.  
\*\*                        IF sy-subrc = 0.  
\*\*                          "do nothing  
\*\*                        ELSE.  
\*\*                          UPDATE zinter\_leave SET active = abap\_false  
\*\*                                                  who\_no = ls\_rtv\_who-who  
\*\*                                              WHERE lgnum EQ iv\_lgnum  
\*\*                                              AND resource\_name = iv\_queue.  
\*\*                          COMMIT WORK.  
\*\*                        ENDIF.  
\*\*                      ENDLOOP.  
\*\*                    ENDIF.  
\*\*                    RETURN.  
\*\*                  ENDIF.  
\*\*  
\*\*                  SELECT \*  INTO TABLE lt\_inter\_who FROM zinter\_leave  
\*\*                            WHERE lgnum = iv\_lgnum  
\*\*                            AND resource\_name = iv\_queue .  
\*\*                  IF sy-subrc = 0.  
\*\*                    LOOP AT lt\_inter\_who INTO ls\_inter\_who.  
\*\*                      READ TABLE ct\_who INTO cs\_who WITH KEY who = ls\_inter\_who-who\_no.  
\*\*                      IF sy-subrc = 0.  
\*\*                        "do nothing  
\*\*                      ELSE.  
\*\*                        UPDATE zinter\_leave SET active = abap\_true  
\*\*                                                who\_no = ls\_rtv\_who-who  
\*\*                                            WHERE lgnum EQ iv\_lgnum AND resource\_name = iv\_queue.  
\*\*                        COMMIT WORK.  
\*\*                      ENDIF.  
\*\*                    ENDLOOP.  
\*\*                  ENDIF.  
\*\*                  RETURN.  
\*\*                ENDIF.  
\*\*  
\*\*              ENDIF.  
\*\*  
\*\*            ELSE.  
\*\*              IF lt\_str\_who[] IS INITIAL AND lt\_rtv\_who[] IS NOT INITIAL.  
\*\*                READ TABLE lt\_rtv\_who INTO ls\_rtv\_who WITH KEY hdr\_procty = '3010'.  
\*\*                IF sy-subrc = 0.  
\*\*                  SELECT \* INTO TABLE lt\_interleave FROM zinter\_leave  
\*\*                    WHERE lgnum = iv\_lgnum AND resource\_name = iv\_queue AND active EQ abap\_true.  
\*\*                  IF sy-subrc = 0.  
\*\*  
\*\*                    READ TABLE lt\_rtv\_who INTO ls\_rtv\_who WITH KEY hdr\_procty = '3010'.  
\*\*                    IF sy-subrc = 0.  
\*\*                      INSERT ls\_rtv\_who INTO ct\_who INDEX 1.  
\*\*                      READ TABLE ct\_who INTO cs\_who INDEX 1.  
\*\*                      iv\_who = cs\_who-who.  
\*\*                      ls\_det2\_fg-lgnum = iv\_lgnum.  
\*\*                      ls\_det2\_fg-crane\_no = lv\_crane\_no.  
\*\*                      ls\_det2\_fg-error\_no = ls\_final\_error-error.  
\*\*                      ls\_det2\_fg-who = iv\_who.  
\*\*                      ls\_det2\_fg-description = 'checking for picking task need rewarehousing repl '.  
\*\*                      ls\_det2\_fg-log\_date = sy-datum.  
\*\*                      ls\_det2\_fg-log\_time = sy-uzeit.  
\*\*                      APPEND ls\_det2\_fg TO lt\_det2\_fg.  
\*\*                      MODIFY zdet2\_fg\_log FROM TABLE lt\_det2\_fg.  
\*\*                      COMMIT WORK.  
\*\*  
\*\*                      lv\_missing\_case = 2.  
\*\*                      CALL METHOD lo\_badi\_transfer->transfer\_hu1  
\*\*                        EXPORTING  
\*\*                          iv\_lgnum   = iv\_lgnum  
\*\*                          is\_whorder = is\_who  
\*\*                          iv\_who     = iv\_who  
\*\*                          iv\_rsrc    = is\_rsrc  
\*\*                          it\_ordim\_o = ct\_ordim\_o  
\*\*                        CHANGING  
\*\*                          ev\_who     = lv\_who.  
\*\*  
\*\*                      CLEAR ls\_who.  
\*\*                      IF lv\_who+0(1) NE 0.  
\*\*                        IF iv\_who NE lv\_who.  
\*\*                          "if current WO already contains WT for HU in front of HU to be picked => do nothing  
\*\*                          ls\_who-who = lv\_who.  
\*\*                          INSERT ls\_who INTO ct\_who INDEX 1.  
\*\*                          ls\_det2\_fg-lgnum = iv\_lgnum.  
\*\*                          ls\_det2\_fg-crane\_no = lv\_crane\_no.  
\*\*                          ls\_det2\_fg-who = lv\_who.  
\*\*                          ls\_det2\_fg-description = 'Rewarehousing task sent to subsystem Picking task repl'.  
\*\*                          ls\_det2\_fg-log\_date = sy-datum.  
\*\*                          ls\_det2\_fg-log\_time = sy-uzeit.  
\*\*                          APPEND ls\_det2\_fg TO lt\_det2\_fg.  
\*\*                          MODIFY zdet2\_fg\_log FROM TABLE lt\_det2\_fg.  
\*\*                          COMMIT WORK.  
\*\*  
\*\*                          UPDATE zinter\_leave SET active = abap\_true  
\*\*                                                  who\_no = ls\_who-who  
\*\*                                              WHERE lgnum EQ iv\_lgnum  
\*\*                                              AND resource\_name = iv\_queue.  
\*\*                          COMMIT WORK.  
\*\*                        ENDIF.  
\*\*                        cv\_rearr\_done = abap\_true.  
\*\*                        RETURN.  
\*\*                      ELSE.  
\*\*                        SELECT \*  INTO TABLE lt\_inter\_who FROM zinter\_leave  
\*\*                                  WHERE lgnum = iv\_lgnum  
\*\*                                   AND resource\_name = iv\_queue .  
\*\*                        IF sy-subrc = 0.  
\*\*                          LOOP AT lt\_inter\_who INTO ls\_inter\_who.  
\*\*                            READ TABLE ct\_who INTO cs\_who WITH KEY who = ls\_inter\_who-who\_no.  
\*\*                            IF sy-subrc = 0.  
\*\*                              "do nothing  
\*\*                            ELSE.  
\*\*                              UPDATE zinter\_leave SET active = abap\_false  
\*\*                                                     who\_no = ls\_rtv\_who-who  
\*\*                                                  WHERE lgnum EQ iv\_lgnum AND resource\_name = iv\_queue.  
\*\*                              COMMIT WORK.  
\*\*                            ENDIF.  
\*\*                          ENDLOOP.  
\*\*                        ENDIF.  
\*\*                        RETURN.  
\*\*                      ENDIF.  
\*\*  
\*\*                      SELECT \*  INTO TABLE lt\_inter\_who FROM zinter\_leave  
\*\*                                 WHERE lgnum = iv\_lgnum AND resource\_name = iv\_queue .  
\*\*                      IF sy-subrc = 0.  
\*\*                        LOOP AT lt\_inter\_who INTO ls\_inter\_who.  
\*\*                          READ TABLE ct\_who INTO cs\_who WITH KEY who = ls\_inter\_who-who\_no.  
\*\*                          IF sy-subrc = 0.  
\*\*                            "do nothing  
\*\*                          ELSE.  
\*\*                            UPDATE zinter\_leave SET active = abap\_true  
\*\*                                                    who\_no = ls\_rtv\_who-who  
\*\*                                                WHERE lgnum EQ iv\_lgnum AND resource\_name = iv\_queue.  
\*\*                            COMMIT WORK.  
\*\*                          ENDIF.  
\*\*                        ENDLOOP.  
\*\*                      ENDIF.  
\*\*                      RETURN.  
\*\*                    ENDIF."sy-subrc = 0 end  
\*\*  
\*\*  
\*\*                  ENDIF.  
\*\*                ENDIF.  
\*\*  
\*\*                READ TABLE lt\_rtv\_who INTO ls\_rtv\_who WITH KEY hdr\_procty = '2010'.  
\*\*                IF sy-subrc = 0.  
\*\*                  SELECT \* INTO TABLE lt\_interleave FROM zinter\_leave WHERE  
\*\*                          lgnum = iv\_lgnum AND resource\_name = iv\_queue AND active EQ abap\_true.  
\*\*                  IF sy-subrc = 0.  
\*\*  
\*\*                    READ TABLE lt\_rtv\_who INTO ls\_rtv\_who WITH KEY hdr\_procty = '2010'.  
\*\*                    IF sy-subrc = 0.  
\*\*                      INSERT ls\_rtv\_who INTO ct\_who INDEX 1.  
\*\*                      READ TABLE ct\_who INTO cs\_who INDEX 1.  
\*\*                      iv\_who = cs\_who-who.  
\*\*                      ls\_det2\_fg-lgnum = iv\_lgnum.  
\*\*                      ls\_det2\_fg-crane\_no = lv\_crane\_no.  
\*\*                      ls\_det2\_fg-error\_no = ls\_final\_error-error.  
\*\*                      ls\_det2\_fg-who = iv\_who.  
\*\*                      ls\_det2\_fg-description = 'checking for picking task need rewarehousing '.  
\*\*                      ls\_det2\_fg-log\_date = sy-datum.  
\*\*                      ls\_det2\_fg-log\_time = sy-uzeit.  
\*\*                      APPEND ls\_det2\_fg TO lt\_det2\_fg.  
\*\*                      MODIFY zdet2\_fg\_log FROM TABLE lt\_det2\_fg.  
\*\*  
\*\*                      COMMIT WORK.  
\*\*                      lv\_missing\_case = 2.  
\*\*                      CALL METHOD lo\_badi\_transfer->transfer\_hu1  
\*\*                        EXPORTING  
\*\*                          iv\_lgnum   = iv\_lgnum  
\*\*                          is\_whorder = is\_who  
\*\*                          iv\_who     = iv\_who  
\*\*                          iv\_rsrc    = is\_rsrc  
\*\*                          it\_ordim\_o = ct\_ordim\_o  
\*\*                        CHANGING  
\*\*                          ev\_who     = lv\_who.  
\*\*  
\*\*                      CLEAR ls\_who.  
\*\*                      IF lv\_who+0(1) NE 0.  
\*\*                        IF iv\_who NE lv\_who.  
\*\*                          "if current WO already contains WT for HU in front of HU to be picked => do nothing  
\*\*                          ls\_who-who = lv\_who.  
\*\*                          INSERT ls\_who INTO ct\_who INDEX 1.  
\*\*                          ls\_det2\_fg-lgnum = iv\_lgnum.  
\*\*                          ls\_det2\_fg-crane\_no = lv\_crane\_no.  
\*\*                          ls\_det2\_fg-who = lv\_who.  
\*\*                          ls\_det2\_fg-description = 'Rewarehousing task sent to subsystem Picking task'.  
\*\*                          ls\_det2\_fg-log\_date = sy-datum.  
\*\*                          ls\_det2\_fg-log\_time = sy-uzeit.  
\*\*                          APPEND ls\_det2\_fg TO lt\_det2\_fg.  
\*\*                          MODIFY zdet2\_fg\_log FROM TABLE lt\_det2\_fg.  
\*\*                          COMMIT WORK.  
\*\*  
\*\*                          UPDATE zinter\_leave SET active = abap\_true  
\*\*                                                  who\_no = ls\_who-who  
\*\*                                                  WHERE lgnum EQ iv\_lgnum AND resource\_name = iv\_queue.  
\*\*                          COMMIT WORK.  
\*\*                        ENDIF.  
\*\*                        cv\_rearr\_done = abap\_true.  
\*\*                        RETURN.  
\*\*                      ELSE.  
\*\*                        SELECT \*  INTO TABLE lt\_inter\_who FROM zinter\_leave  
\*\*                                  WHERE lgnum = iv\_lgnum AND resource\_name = iv\_queue .  
\*\*                        IF sy-subrc = 0.  
\*\*                          LOOP AT lt\_inter\_who INTO ls\_inter\_who.  
\*\*                            READ TABLE ct\_who INTO cs\_who WITH KEY who = ls\_inter\_who-who\_no.  
\*\*                            IF sy-subrc = 0.  
\*\*                              "do nothing  
\*\*                            ELSE.  
\*\*                              UPDATE zinter\_leave SET active = abap\_false  
\*\*                                                      who\_no = ls\_rtv\_who-who  
\*\*                                                  WHERE lgnum EQ iv\_lgnum  
\*\*                                                  AND resource\_name = iv\_queue.  
\*\*                              COMMIT WORK.  
\*\*                            ENDIF.  
\*\*                          ENDLOOP.  
\*\*                        ENDIF.  
\*\*                        RETURN.  
\*\*                      ENDIF.  
\*\*  
\*\*                      SELECT \*  INTO TABLE lt\_inter\_who FROM zinter\_leave  
\*\*                                WHERE lgnum = iv\_lgnum AND resource\_name = iv\_queue .  
\*\*                      IF sy-subrc = 0.  
\*\*                        LOOP AT lt\_inter\_who INTO ls\_inter\_who.  
\*\*                          READ TABLE ct\_who INTO cs\_who WITH KEY who = ls\_inter\_who-who\_no.  
\*\*                          IF sy-subrc = 0.  
\*\*                            "do nothing  
\*\*                          ELSE.  
\*\*                            UPDATE zinter\_leave SET active = abap\_true  
\*\*                                                    who\_no = ls\_rtv\_who-who  
\*\*                                                WHERE lgnum EQ iv\_lgnum  
\*\*                                                AND resource\_name = iv\_queue.  
\*\*                            COMMIT WORK.  
\*\*                          ENDIF.  
\*\*                        ENDLOOP.  
\*\*                      ENDIF.  
\*\*                      RETURN.  
\*\*                    ENDIF."sy-subrc = 0 end  
\*\*  
\*\*                  ENDIF.  
\*\*                ENDIF.  
\*\*  
\*\*              ENDIF.  
\*\*  
\*\*            ENDIF.  
\*\*\*\*\*\*\*\*\*\*\*\* end of when we have retrival tasks  
\*\*  
\*\*  
\*\*            READ TABLE lt\_str\_who INTO ls\_str\_who WITH KEY hdr\_procty = '3090'.  
\*\*            IF sy-subrc = 0.  
\*\*              SELECT \* INTO TABLE lt\_interleave FROM zinter\_leave  
\*\*                     WHERE lgnum = iv\_lgnum AND resource\_name = iv\_queue AND active EQ abap\_true.  
\*\*              IF sy-subrc NE 0.  
\*\*                INSERT ls\_str\_who INTO ct\_who INDEX 1.  
\*\*  
\*\*                SELECT \*  INTO TABLE lt\_inter\_who FROM zinter\_leave  
\*\*                         WHERE lgnum = iv\_lgnum AND resource\_name = iv\_queue .  
\*\*                IF sy-subrc = 0.  
\*\*                  READ TABLE lt\_inter\_who INTO ls\_inter\_who INDEX 1.  
\*\*                  IF ls\_str\_who-who NE ls\_inter\_who-who\_no.  
\*\*                    UPDATE zinter\_leave SET active = abap\_true  
\*\*                                            who\_no = ls\_str\_who-who  
\*\*                                        WHERE lgnum EQ iv\_lgnum  
\*\*                                        AND resource\_name = iv\_queue.  
\*\*                    COMMIT WORK.  
\*\*                    lv\_missing\_case = 2.  
\*\*                  ENDIF.  
\*\*                ENDIF.  
\*\*  
\*\*                RETURN.  
\*\*              ELSEIF sy-subrc = 0.  
\*\*                READ TABLE lt\_rtv\_who INTO ls\_rtv\_who WITH KEY hdr\_procty = '3010'.  
\*\*                IF sy-subrc = 0.  
\*\*                  INSERT ls\_rtv\_who INTO ct\_who INDEX 1.  
\*\*                  READ TABLE ct\_who INTO cs\_who INDEX 1.  
\*\*                  iv\_who = cs\_who-who.  
\*\*                  ls\_det2\_fg-lgnum = iv\_lgnum.  
\*\*                  ls\_det2\_fg-crane\_no = lv\_crane\_no.  
\*\*                  ls\_det2\_fg-error\_no = ls\_final\_error-error.  
\*\*                  ls\_det2\_fg-who = iv\_who.  
\*\*                  ls\_det2\_fg-description = 'check for putaway task need rewarehousing when we have only putaway '.  
\*\*                  ls\_det2\_fg-log\_date = sy-datum.  
\*\*                  ls\_det2\_fg-log\_time = sy-uzeit.  
\*\*                  APPEND ls\_det2\_fg TO lt\_det2\_fg.  
\*\*                  MODIFY zdet2\_fg\_log FROM TABLE lt\_det2\_fg.  
\*\*                  COMMIT WORK.  
\*\*                  lv\_missing\_case = 2.  
\*\*                  CALL METHOD lo\_badi\_transfer->transfer\_hu1  
\*\*                    EXPORTING  
\*\*                      iv\_lgnum   = iv\_lgnum  
\*\*                      is\_whorder = is\_who  
\*\*                      iv\_who     = iv\_who  
\*\*                      iv\_rsrc    = is\_rsrc  
\*\*                      it\_ordim\_o = ct\_ordim\_o  
\*\*                    CHANGING  
\*\*                      ev\_who     = lv\_who.  
\*\*  
\*\*                  CLEAR ls\_who.  
\*\*                  IF lv\_who+0(1) NE 0.  
\*\*                    IF iv\_who NE lv\_who.  
\*\*                      "if current WO already contains WT for HU in front of HU to be picked => do nothing  
\*\*                      ls\_who-who = lv\_who.  
\*\*                      INSERT ls\_who INTO ct\_who INDEX 1.  
\*\*                      ls\_det2\_fg-lgnum = iv\_lgnum.  
\*\*                      ls\_det2\_fg-crane\_no = lv\_crane\_no.  
\*\*                      ls\_det2\_fg-who = lv\_who.  
\*\*                      ls\_det2\_fg-description = 'Rewarehouse task sent to subsystem Putaway tasks are present'.  
\*\*                      ls\_det2\_fg-log\_date = sy-datum.  
\*\*                      ls\_det2\_fg-log\_time = sy-uzeit.  
\*\*                      APPEND ls\_det2\_fg TO lt\_det2\_fg.  
\*\*                      MODIFY zdet2\_fg\_log FROM TABLE lt\_det2\_fg.  
\*\*                      COMMIT WORK.  
\*\*\*                      SELECT SINGLE who\_no INTO lv\_inter\_who FROM zinter\_leave WHERE lgnum = iv\_lgnum AND resource\_name = iv\_queue .  
\*\*\*                      IF sy-subrc = 0.  
\*\*\*                        IF lv\_inter\_who NE ls\_who-who.  
\*\*                      UPDATE zinter\_leave SET active = abap\_true  
\*\*                                              who\_no = ls\_who-who  
\*\*                                          WHERE lgnum EQ iv\_lgnum  
\*\*                                          AND resource\_name = iv\_queue.  
\*\*                      COMMIT WORK.  
\*\*                    ENDIF.  
\*\*                    cv\_rearr\_done = abap\_true.  
\*\*                    RETURN.  
\*\*                  ELSE.  
\*\*                    SELECT \*  INTO TABLE lt\_inter\_who FROM zinter\_leave  
\*\*                              WHERE lgnum = iv\_lgnum AND resource\_name = iv\_queue .  
\*\*                    IF sy-subrc = 0.  
\*\*                      LOOP AT lt\_inter\_who INTO ls\_inter\_who.  
\*\*                        READ TABLE ct\_who INTO cs\_who WITH KEY who = ls\_inter\_who-who\_no.  
\*\*                        IF sy-subrc = 0.  
\*\*                          "do nothing  
\*\*                        ELSE.  
\*\*                          UPDATE zinter\_leave SET active = abap\_false  
\*\*                                                  who\_no = ls\_rtv\_who-who  
\*\*                                              WHERE lgnum EQ iv\_lgnum  
\*\*                                              AND resource\_name = iv\_queue.  
\*\*                          COMMIT WORK.  
\*\*                        ENDIF.  
\*\*                      ENDLOOP.  
\*\*                    ENDIF.  
\*\*                    RETURN.  
\*\*                  ENDIF.  
\*\*  
\*\*                  SELECT \*  INTO TABLE lt\_inter\_who FROM zinter\_leave  
\*\*                             WHERE lgnum = iv\_lgnum AND resource\_name = iv\_queue .  
\*\*                  IF sy-subrc = 0.  
\*\*                    LOOP AT lt\_inter\_who INTO ls\_inter\_who.  
\*\*                      READ TABLE ct\_who INTO cs\_who WITH KEY who = ls\_inter\_who-who\_no.  
\*\*                      IF sy-subrc = 0.  
\*\*                        "do nothing  
\*\*                      ELSE.  
\*\*                        UPDATE zinter\_leave SET active = abap\_false  
\*\*                                                who\_no = ls\_rtv\_who-who  
\*\*                                            WHERE lgnum EQ iv\_lgnum AND resource\_name = iv\_queue.  
\*\*                        COMMIT WORK.  
\*\*                      ENDIF.  
\*\*                    ENDLOOP.  
\*\*                  ENDIF.  
\*\*                  RETURN.  
\*\*                ELSE."3010 task is not present  
\*\*                  READ TABLE lt\_rtv\_who INTO ls\_rtv\_who WITH KEY hdr\_procty = '2010'.  
\*\*                  IF sy-subrc = 0.  
\*\*                    INSERT ls\_rtv\_who INTO ct\_who INDEX 1.  
\*\*                    READ TABLE ct\_who INTO cs\_who INDEX 1.  
\*\*                    iv\_who = cs\_who-who.  
\*\*                    ls\_det2\_fg-lgnum = iv\_lgnum.  
\*\*                    ls\_det2\_fg-crane\_no = lv\_crane\_no.  
\*\*                    ls\_det2\_fg-error\_no = ls\_final\_error-error.  
\*\*                    ls\_det2\_fg-who = iv\_who.  
\*\*                    ls\_det2\_fg-description = 'checking for putaway task need rewarehousing when we have only putaway  '.  
\*\*                    ls\_det2\_fg-log\_date = sy-datum.  
\*\*                    ls\_det2\_fg-log\_time = sy-uzeit.  
\*\*                    APPEND ls\_det2\_fg TO lt\_det2\_fg.  
\*\*                    MODIFY zdet2\_fg\_log FROM TABLE lt\_det2\_fg.  
\*\*                    COMMIT WORK.  
\*\*                    lv\_missing\_case = 2.  
\*\*                    CALL METHOD lo\_badi\_transfer->transfer\_hu1  
\*\*                      EXPORTING  
\*\*                        iv\_lgnum   = iv\_lgnum  
\*\*                        is\_whorder = is\_who  
\*\*                        iv\_who     = iv\_who  
\*\*                        iv\_rsrc    = is\_rsrc  
\*\*                        it\_ordim\_o = ct\_ordim\_o  
\*\*                      CHANGING  
\*\*                        ev\_who     = lv\_who.  
\*\*  
\*\*                    CLEAR ls\_who.  
\*\*                    IF lv\_who+0(1) NE 0.  
\*\*                      IF iv\_who NE lv\_who.  
\*\*                        "if current WO already contains WT for HU in front of HU to be picked => do nothing  
\*\*                        ls\_who-who = lv\_who.  
\*\*                        INSERT ls\_who INTO ct\_who INDEX 1.  
\*\*                        ls\_det2\_fg-lgnum = iv\_lgnum.  
\*\*                        ls\_det2\_fg-crane\_no = lv\_crane\_no.  
\*\*                        ls\_det2\_fg-who = lv\_who.  
\*\*                        ls\_det2\_fg-description = 'Rewarehousing task sent to subsystem Putaway tasks are present'.  
\*\*                        ls\_det2\_fg-log\_date = sy-datum.  
\*\*                        ls\_det2\_fg-log\_time = sy-uzeit.  
\*\*                        APPEND ls\_det2\_fg TO lt\_det2\_fg.  
\*\*                        MODIFY zdet2\_fg\_log FROM TABLE lt\_det2\_fg.  
\*\*                        COMMIT WORK.  
\*\*  
\*\*                        UPDATE zinter\_leave SET active = abap\_true  
\*\*                                                who\_no = ls\_who-who  
\*\*                                            WHERE lgnum EQ iv\_lgnum  
\*\*                                            AND resource\_name = iv\_queue.  
\*\*                        COMMIT WORK.  
\*\*                      ENDIF.  
\*\*                      cv\_rearr\_done = abap\_true.  
\*\*                      RETURN.  
\*\*                    ELSE.  
\*\*                      SELECT \*  INTO TABLE lt\_inter\_who FROM zinter\_leave  
\*\*                                WHERE lgnum = iv\_lgnum AND resource\_name = iv\_queue .  
\*\*                      IF sy-subrc = 0.  
\*\*                        LOOP AT lt\_inter\_who INTO ls\_inter\_who.  
\*\*                          READ TABLE ct\_who INTO cs\_who WITH KEY who = ls\_inter\_who-who\_no.  
\*\*                          IF sy-subrc = 0.  
\*\*                            "do nothing  
\*\*                          ELSE.  
\*\*                            UPDATE zinter\_leave SET active = abap\_false  
\*\*                                                    who\_no = ls\_rtv\_who-who  
\*\*                                                WHERE lgnum EQ iv\_lgnum  
\*\*                                                AND resource\_name = iv\_queue.  
\*\*                            COMMIT WORK.  
\*\*                          ENDIF.  
\*\*                        ENDLOOP.  
\*\*                      ENDIF.  
\*\*                      RETURN.  
\*\*                    ENDIF.  
\*\*  
\*\*                    SELECT \*  INTO TABLE lt\_inter\_who FROM zinter\_leave  
\*\*                              WHERE lgnum = iv\_lgnum AND resource\_name = iv\_queue .  
\*\*                    IF sy-subrc = 0.  
\*\*                      LOOP AT lt\_inter\_who INTO ls\_inter\_who.  
\*\*                        READ TABLE ct\_who INTO cs\_who WITH KEY who = ls\_inter\_who-who\_no.  
\*\*                        IF sy-subrc = 0.  
\*\*                          "do nothing  
\*\*                        ELSE.  
\*\*                          UPDATE zinter\_leave SET active = abap\_false  
\*\*                                                  who\_no = ls\_rtv\_who-who  
\*\*                                              WHERE lgnum EQ iv\_lgnum  
\*\*                                              AND resource\_name = iv\_queue.  
\*\*                          COMMIT WORK.  
\*\*                        ENDIF.  
\*\*                      ENDLOOP.  
\*\*                    ENDIF.  
\*\*                    RETURN.  
\*\*                  ENDIF."sy-subrc = 0 end  
\*\*                ENDIF.  
\*\*              ENDIF.  
\*\*            ELSE." lt\_str\_who is initial  
\*\*              READ TABLE lt\_rtv\_who INTO ls\_rtv\_who WITH KEY hdr\_procty = '3010'.  
\*\*              IF sy-subrc = 0.  
\*\*                INSERT ls\_rtv\_who INTO ct\_who INDEX 1.  
\*\*                READ TABLE ct\_who INTO cs\_who INDEX 1.  
\*\*                iv\_who = cs\_who-who.  
\*\*                ls\_det2\_fg-lgnum = iv\_lgnum.  
\*\*                ls\_det2\_fg-crane\_no = lv\_crane\_no.  
\*\*                ls\_det2\_fg-error\_no = ls\_final\_error-error.  
\*\*                ls\_det2\_fg-who = iv\_who.  
\*\*                ls\_det2\_fg-description = 'check for putaway task need rewarehousing for picking  '.  
\*\*                ls\_det2\_fg-log\_date = sy-datum.  
\*\*                ls\_det2\_fg-log\_time = sy-uzeit.  
\*\*                APPEND ls\_det2\_fg TO lt\_det2\_fg.  
\*\*                MODIFY zdet2\_fg\_log FROM TABLE lt\_det2\_fg.  
\*\*                COMMIT WORK.  
\*\*                lv\_missing\_case = 2.  
\*\*                CALL METHOD lo\_badi\_transfer->transfer\_hu1  
\*\*                  EXPORTING  
\*\*                    iv\_lgnum   = iv\_lgnum  
\*\*                    is\_whorder = is\_who  
\*\*                    iv\_who     = iv\_who  
\*\*                    iv\_rsrc    = is\_rsrc  
\*\*                    it\_ordim\_o = ct\_ordim\_o  
\*\*                  CHANGING  
\*\*                    ev\_who     = lv\_who.  
\*\*  
\*\*                CLEAR ls\_who.  
\*\*                IF lv\_who+0(1) NE 0.  
\*\*                  IF iv\_who NE lv\_who.  
\*\*                    "if current WO already contains WT for HU in front of HU to be picked => do nothing  
\*\*                    ls\_who-who = lv\_who.  
\*\*                    INSERT ls\_who INTO ct\_who INDEX 1.  
\*\*                    ls\_det2\_fg-lgnum = iv\_lgnum.  
\*\*                    ls\_det2\_fg-crane\_no = lv\_crane\_no.  
\*\*                    ls\_det2\_fg-who = lv\_who.  
\*\*                    ls\_det2\_fg-description = 'Rewarehouse task sent to subsystem for picking'.  
\*\*                    ls\_det2\_fg-log\_date = sy-datum.  
\*\*                    ls\_det2\_fg-log\_time = sy-uzeit.  
\*\*                    APPEND ls\_det2\_fg TO lt\_det2\_fg.  
\*\*                    MODIFY zdet2\_fg\_log FROM TABLE lt\_det2\_fg.  
\*\*                    COMMIT WORK.  
\*\*  
\*\*                    UPDATE zinter\_leave SET active = abap\_true  
\*\*                                            who\_no = ls\_who-who  
\*\*                                        WHERE lgnum EQ iv\_lgnum  
\*\*                                        AND resource\_name = iv\_queue.  
\*\*                    COMMIT WORK.  
\*\*                  ENDIF.  
\*\*                  cv\_rearr\_done = abap\_true.  
\*\*                  RETURN.  
\*\*                ELSE.  
\*\*                  SELECT \*  INTO TABLE lt\_inter\_who FROM zinter\_leave  
\*\*                            WHERE lgnum = iv\_lgnum AND resource\_name = iv\_queue .  
\*\*                  IF sy-subrc = 0.  
\*\*                    LOOP AT lt\_inter\_who INTO ls\_inter\_who.  
\*\*                      READ TABLE ct\_who INTO cs\_who WITH KEY who = ls\_inter\_who-who\_no.  
\*\*                      IF sy-subrc = 0.  
\*\*                        "do nothing  
\*\*                      ELSE.  
\*\*                        UPDATE zinter\_leave SET active = abap\_false  
\*\*                                                who\_no = ls\_rtv\_who-who  
\*\*                                            WHERE lgnum EQ iv\_lgnum  
\*\*                                            AND resource\_name = iv\_queue.  
\*\*                        COMMIT WORK.  
\*\*                      ENDIF.  
\*\*                    ENDLOOP.  
\*\*                  ENDIF.  
\*\*                  RETURN.  
\*\*                ENDIF.  
\*\*  
\*\*                SELECT \*  INTO TABLE lt\_inter\_who FROM zinter\_leave  
\*\*                          WHERE lgnum = iv\_lgnum AND resource\_name = iv\_queue .  
\*\*                IF sy-subrc = 0.  
\*\*                  LOOP AT lt\_inter\_who INTO ls\_inter\_who.  
\*\*                    READ TABLE ct\_who INTO cs\_who WITH KEY who = ls\_inter\_who-who\_no.  
\*\*                    IF sy-subrc = 0.  
\*\*                      "do nothing  
\*\*                    ELSE.  
\*\*                      UPDATE zinter\_leave SET active = abap\_false  
\*\*                                              who\_no = ls\_rtv\_who-who  
\*\*                                          WHERE lgnum EQ iv\_lgnum  
\*\*                                          AND resource\_name = iv\_queue.  
\*\*                      COMMIT WORK.  
\*\*                    ENDIF.  
\*\*                  ENDLOOP.  
\*\*                ENDIF.  
\*\*                RETURN.  
\*\*              ELSE."3010 task is not present when lt\_str\_who is initial  
\*\*                READ TABLE lt\_rtv\_who INTO ls\_rtv\_who WITH KEY hdr\_procty = '2010'.  
\*\*                IF sy-subrc = 0.  
\*\*                  INSERT ls\_rtv\_who INTO ct\_who INDEX 1.  
\*\*                  READ TABLE ct\_who INTO cs\_who INDEX 1.  
\*\*                  iv\_who = cs\_who-who.  
\*\*                  ls\_det2\_fg-lgnum = iv\_lgnum.  
\*\*                  ls\_det2\_fg-crane\_no = lv\_crane\_no.  
\*\*                  ls\_det2\_fg-error\_no = ls\_final\_error-error.  
\*\*                  ls\_det2\_fg-who = iv\_who.  
\*\*                  ls\_det2\_fg-description = 'checking for putaway task need rewarehousing for picking  '.  
\*\*                  ls\_det2\_fg-log\_date = sy-datum.  
\*\*                  ls\_det2\_fg-log\_time = sy-uzeit.  
\*\*                  APPEND ls\_det2\_fg TO lt\_det2\_fg.  
\*\*                  MODIFY zdet2\_fg\_log FROM TABLE lt\_det2\_fg.  
\*\*                  COMMIT WORK.  
\*\*                  lv\_missing\_case = 2.  
\*\*                  CALL METHOD lo\_badi\_transfer->transfer\_hu1  
\*\*                    EXPORTING  
\*\*                      iv\_lgnum   = iv\_lgnum  
\*\*                      is\_whorder = is\_who  
\*\*                      iv\_who     = iv\_who  
\*\*                      iv\_rsrc    = is\_rsrc  
\*\*                      it\_ordim\_o = ct\_ordim\_o  
\*\*                    CHANGING  
\*\*                      ev\_who     = lv\_who.  
\*\*  
\*\*                  CLEAR ls\_who.  
\*\*                  IF lv\_who+0(1) NE 0.  
\*\*                    IF iv\_who NE lv\_who.  
\*\*                      "if current WO already contains WT for HU in front of HU to be picked => do nothing  
\*\*                      ls\_who-who = lv\_who.  
\*\*                      INSERT ls\_who INTO ct\_who INDEX 1.  
\*\*                      ls\_det2\_fg-lgnum = iv\_lgnum.  
\*\*                      ls\_det2\_fg-crane\_no = lv\_crane\_no.  
\*\*                      ls\_det2\_fg-who = lv\_who.  
\*\*                      ls\_det2\_fg-description = 'Rewarehousing task sent to subsystem Putaway tasks are present'.  
\*\*                      ls\_det2\_fg-log\_date = sy-datum.  
\*\*                      ls\_det2\_fg-log\_time = sy-uzeit.  
\*\*                      APPEND ls\_det2\_fg TO lt\_det2\_fg.  
\*\*                      MODIFY zdet2\_fg\_log FROM TABLE lt\_det2\_fg.  
\*\*                      COMMIT WORK.  
\*\*  
\*\*                      UPDATE zinter\_leave SET active = abap\_true  
\*\*                                              who\_no = ls\_who-who  
\*\*                                          WHERE lgnum EQ iv\_lgnum  
\*\*                                          AND resource\_name = iv\_queue.  
\*\*                      COMMIT WORK.  
\*\*  
\*\*                    ENDIF.  
\*\*                    cv\_rearr\_done = abap\_true.  
\*\*                    RETURN.  
\*\*                  ELSE.  
\*\*                    SELECT \*  INTO TABLE lt\_inter\_who FROM zinter\_leave  
\*\*                               WHERE lgnum = iv\_lgnum AND resource\_name = iv\_queue .  
\*\*                    IF sy-subrc = 0.  
\*\*                      LOOP AT lt\_inter\_who INTO ls\_inter\_who.  
\*\*                        READ TABLE ct\_who INTO cs\_who WITH KEY who = ls\_inter\_who-who\_no.  
\*\*                        IF sy-subrc = 0.  
\*\*                          "do nothing  
\*\*                        ELSE.  
\*\*                          UPDATE zinter\_leave SET active = abap\_false  
\*\*                                                  who\_no = ls\_rtv\_who-who  
\*\*                                              WHERE lgnum EQ iv\_lgnum  
\*\*                                              AND resource\_name = iv\_queue.  
\*\*                          COMMIT WORK.  
\*\*                        ENDIF.  
\*\*                      ENDLOOP.  
\*\*  
\*\*                    ENDIF.  
\*\*                    RETURN.  
\*\*                  ENDIF.  
\*\*  
\*\*                  SELECT \*  INTO TABLE lt\_inter\_who FROM zinter\_leave  
\*\*                            WHERE lgnum = iv\_lgnum AND resource\_name = iv\_queue .  
\*\*                  IF sy-subrc = 0.  
\*\*                    LOOP AT lt\_inter\_who INTO ls\_inter\_who.  
\*\*                      READ TABLE ct\_who INTO cs\_who WITH KEY who = ls\_inter\_who-who\_no.  
\*\*                      IF sy-subrc = 0.  
\*\*                        "do nothing  
\*\*                      ELSE.  
\*\*                        UPDATE zinter\_leave SET active = abap\_false  
\*\*                                                who\_no = ls\_rtv\_who-who  
\*\*                                            WHERE lgnum EQ iv\_lgnum  
\*\*                                            AND resource\_name = iv\_queue.  
\*\*                        COMMIT WORK.  
\*\*                      ENDIF.  
\*\*                    ENDLOOP.  
\*\*                  ENDIF.  
\*\*                  RETURN.  
\*\*                ENDIF."sy-subrc = 0  of 2010 read table end  
\*\*  
\*\*              ENDIF."3010 of read table end  
\*\*  
\*\*            ENDIF."3090 READ TABLE end  
\*\*  
\*\*          ENDIF." end of interleaving  
\*\*  
\*\*  
\*\*        ELSEIF lt\_final\_error[] IS NOT INITIAL. " if lt\_final\_error[] is not initial  
\*\*  
\*\*          LOOP AT lt\_final\_error INTO ls\_final\_error.  
\*\*            MOVE ls\_final\_error-huident  TO ls\_log-extnumber.  
\*\*            CALL METHOD lo\_log->create\_log  
\*\*              EXPORTING  
\*\*                is\_log = ls\_log.  
\*\*            MESSAGE s031(zapl\_det) WITH ls\_final\_error-error INTO lv\_msgtext.  
\*\*            lo\_log->add\_message2log( ).  
\*\*            IF ls\_final\_error-error IS INITIAL OR ls\_final\_error-error+0(2) EQ '00'.  
\*\*              SELECT SINGLE zvalue INTO ls\_zvalue FROM zcon\_param\_val  
\*\*                                   WHERE lgnum = iv\_lgnum  
\*\*                                   AND zreq\_name = 'INTERLEAVING'  
\*\*                                   AND zparam = 'INTERLEAVE\_PARAM'  
\*\*                                   AND zactive = abap\_true.  
\*\*              IF sy-subrc EQ 0.  
\*\*  
\*\*\*              refresh: ct\_who[].  
\*\*\*\*-- MFS Whse process type constants  
\*\*                lv\_id1 = zapl\_cl\_common\_constants=>gc\_mfs\_tele\_send.  
\*\*                CALL METHOD zapl\_cl\_common\_constants=>get\_param\_value  
\*\*                  EXPORTING  
\*\*                    iv\_lgnum = iv\_lgnum  
\*\*                    iv\_id1   = lv\_id1  
\*\*                  IMPORTING  
\*\*                    et\_param = lt\_param\_mfs\_tele\_send.  
\*\*  
\*\*\*-- Scheduler constants  
\*\*                lv\_id1 = zapl\_cl\_common\_constants=>gc\_agc\_scheduler.  
\*\*                CALL METHOD zapl\_cl\_common\_constants=>get\_param\_value  
\*\*                  EXPORTING  
\*\*                    iv\_lgnum = iv\_lgnum  
\*\*                    iv\_id1   = lv\_id1  
\*\*                  IMPORTING  
\*\*                    et\_param = lt\_param\_agc\_schd.  
\*\*  
\*\*                CLEAR ls\_param.  
\*\*                READ TABLE lt\_param\_agc\_schd INTO ls\_param WITH KEY id2 = zapl\_cl\_common\_constants=>gc\_max\_direct\_command.  
\*\*                IF sy-subrc = 0.  
\*\*                  MOVE ls\_param-value TO lv\_max\_direct .  
\*\*                ENDIF.  
\*\*  
\*\*                CLEAR ls\_param.  
\*\*                READ TABLE lt\_param\_agc\_schd INTO ls\_param WITH KEY id2 = zapl\_cl\_common\_constants=>gc\_max\_storage\_command.  
\*\*                IF sy-subrc = 0.  
\*\*                  MOVE ls\_param-value TO lv\_max\_storage .  
\*\*                ENDIF.  
\*\*  
\*\*                CLEAR ls\_param.  
\*\*                READ TABLE lt\_param\_agc\_schd INTO ls\_param WITH KEY id2 = zapl\_cl\_common\_constants=>gc\_max\_retrieve\_command.  
\*\*                IF sy-subrc = 0.  
\*\*                  MOVE ls\_param-value TO lv\_max\_retrieve .  
\*\*                ENDIF.  
\*\*  
\*\*\*-- Read Whse procty for Storage, retrieval and Direct transfer operations  
\*\*                CLEAR : ls\_param.  
\*\*                READ TABLE lt\_param\_agc\_schd INTO ls\_param WITH KEY id2 = zapl\_cl\_common\_constants=>gc\_direct\_transfer.  
\*\*                IF sy-subrc = 0.  
\*\*                  ls\_hdr\_procty-hdr\_procty = ls\_param-value .  
\*\*                  APPEND ls\_hdr\_procty TO lt\_dir\_procty.  
\*\*                ENDIF.  
\*\*  
\*\*                CLEAR : ls\_param.  
\*\*                READ TABLE lt\_param\_agc\_schd INTO ls\_param WITH KEY id2 = zapl\_cl\_common\_constants=>gc\_retrieval\_interleave.  
\*\*                IF sy-subrc = 0.  
\*\*                  ls\_hdr\_procty-hdr\_procty = ls\_param-value.  
\*\*                  APPEND ls\_hdr\_procty TO lt\_rtv\_procty.  
\*\*                ENDIF.  
\*\*                READ TABLE lt\_param\_agc\_schd INTO ls\_param WITH KEY id2 = zapl\_cl\_common\_constants=>gc\_repl\_interleave.  
\*\*                IF sy-subrc = 0.  
\*\*                  ls\_hdr\_procty-hdr\_procty = ls\_param-value.  
\*\*                  APPEND ls\_hdr\_procty TO lt\_repl\_procty.  
\*\*                ENDIF.  
\*\*  
\*\*                CLEAR : ls\_param.  
\*\*                LOOP AT lt\_param\_agc\_schd INTO ls\_param WHERE id2 = zapl\_cl\_common\_constants=>gc\_storage\_interleave.  
\*\*                  ls\_hdr\_procty-hdr\_procty = ls\_param-value.  
\*\*                  APPEND ls\_hdr\_procty TO lt\_str\_procty.  
\*\*                ENDLOOP.  
\*\*  
\*\*\*    CLEAR :  lt\_who[] , ct\_who[], lv\_count, lv\_retrieve, lv\_storage, lv\_direct.  
\*\*  
\*\*\*-- get all 'in process' WO, to check current WO count for each operation.Based on current count  
\*\*\*   and maximum allowed count, next set of WOs shall be decided and sent  
\*\*                SELECT \* FROM /scwm/who INTO TABLE lt\_who  
\*\*                          WHERE lgnum = iv\_lgnum  
\*\*                            AND status = wmegc\_wo\_open.  
\*\*  
\*\*                IF NOT lt\_who[] IS INITIAL.  
\*\*  
\*\*                  LOOP AT lt\_who ASSIGNING <ls\_who>.  
\*\*  
\*\*  
\*\*\*-- Retrieval  
\*\*                    READ TABLE lt\_rtv\_procty TRANSPORTING NO FIELDS WITH KEY hdr\_procty = <ls\_who>-hdr\_procty.  
\*\*                    IF sy-subrc = 0.  
\*\*                      lv\_retrieve = lv\_retrieve + 1.  
\*\*                    ENDIF.  
\*\*\*\*\*                Replineshment  
\*\*                    READ TABLE lt\_repl\_procty TRANSPORTING NO FIELDS WITH KEY hdr\_procty = <ls\_who>-hdr\_procty.  
\*\*                    IF sy-subrc = 0.  
\*\*                      lv\_retrieve = lv\_retrieve + 1.  
\*\*                    ENDIF.  
\*\*  
\*\*\*-- Storage  
\*\*                    READ TABLE lt\_str\_procty TRANSPORTING NO FIELDS WITH KEY hdr\_procty = <ls\_who>-hdr\_procty.  
\*\*                    IF sy-subrc = 0.  
\*\*                      lv\_storage = lv\_storage + 1.  
\*\*                    ENDIF.  
\*\*                  ENDLOOP.  
\*\*                ENDIF.  
\*\*  
\*\*                IF lv\_max\_retrieve = lv\_retrieve.  
\*\*                  CLEAR lv\_msgtext.  
\*\*                  MESSAGE w432(zapl\_asrs) WITH  lv\_retrieve 'Retrieve' INTO lv\_msgtext.  
\*\*                  lo\_log->add\_message( ).  
\*\*                ENDIF.  
\*\*  
\*\*                IF lv\_max\_storage = lv\_storage.  
\*\*                  CLEAR lv\_msgtext.  
\*\*                  MESSAGE w432(zapl\_asrs) WITH  lv\_storage 'Storage' INTO lv\_msgtext.  
\*\*                  lo\_log->add\_message( ).  
\*\*                ENDIF.  
\*\*  
\*\*  
\*\*  
\*\*\*-- Validate maximum WO limit and select WO  acordingly  
\*\*  
\*\*                lv\_retrieve = lv\_max\_retrieve - lv\_retrieve.  
\*\*                lv\_storage = lv\_max\_storage - lv\_storage.  
\*\*                lv\_direct = lv\_max\_direct - lv\_direct.  
\*\*  
\*\*\*            IF lv\_storage > 0 .  
\*\*  
\*\*                SELECT \* FROM /scwm/who INTO TABLE lt\_str\_who UP TO iv\_wo\_count ROWS  
\*\*                FOR ALL ENTRIES IN lt\_str\_procty  
\*\*                  WHERE lgnum = iv\_lgnum  
\*\*                    AND hdr\_procty = lt\_str\_procty-hdr\_procty  
\*\*                    AND status = wmegc\_wo\_open  
\*\*                    AND queue = iv\_queue.  
\*\*  
\*\*\*            ENDIF.  
\*\*  
\*\*\*            IF lv\_retrieve > 0.  
\*\*                SELECT \* FROM /scwm/who INTO  TABLE  lt\_rtv\_who UP TO iv\_wo\_count ROWS  
\*\*                  FOR ALL ENTRIES IN lt\_rtv\_procty  
\*\*                    WHERE lgnum = iv\_lgnum  
\*\*                      AND hdr\_procty = lt\_rtv\_procty-hdr\_procty  
\*\*                      AND status = wmegc\_wo\_open  
\*\*                      AND queue = iv\_queue.  
\*\*\*\*\*\*\*\*\*\*                REplenishment  
\*\*                SELECT \* FROM /scwm/who INTO  TABLE lt\_repl\_who UP TO iv\_wo\_count ROWS  
\*\*                   FOR ALL ENTRIES IN lt\_repl\_procty  
\*\*                  WHERE lgnum = iv\_lgnum  
\*\*                  AND hdr\_procty = lt\_repl\_procty-hdr\_procty  
\*\*                  AND status = wmegc\_wo\_open  
\*\*                  AND queue = iv\_queue.  
\*\*\*            ENDIF .  
\*\*  
\*\*                SELECT SINGLE zvalue INTO lv\_repl FROM zcon\_param\_val WHERE lgnum = iv\_lgnum AND zreq\_name = 'REPL\_DET2\_REQ' AND zparam = 'REPL\_DET2\_PARAM' AND zactive = abap\_true.  
\*\*                IF sy-subrc = 0.  
\*\*                  IF lt\_repl\_who[] IS NOT INITIAL.  
\*\*                    lt\_rtv\_who1[] = lt\_rtv\_who[].  
\*\*                    lt\_rtv\_who[] = lt\_repl\_who[].  
\*\*                    APPEND LINES OF lt\_rtv\_who1 TO lt\_rtv\_who.  
\*\*                  ENDIF.  
\*\*                ENDIF.  
\*\*                lv\_str\_cnt = lines( lt\_str\_who ) .  
\*\*                lv\_rtv\_cnt = lines( lt\_rtv\_who ).  
\*\*  
\*\*                MOVE iv\_lgnum  TO ls\_log-extnumber.  
\*\*                CALL METHOD lo\_log->create\_log  
\*\*                  EXPORTING  
\*\*                    is\_log = ls\_log.  
\*\*  
\*\*                MESSAGE s033(zapl\_det) WITH 'Interleaving block started'  INTO lv\_msgtext.  
\*\*                lo\_log->add\_message2log( ).  
\*\*  
\*\*\*\*\*\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*When we have only putaway tasks  
\*\*                IF lt\_str\_who[] IS  NOT INITIAL AND lt\_rtv\_who[] IS INITIAL.  
\*\*                  READ TABLE lt\_str\_who INTO ls\_str\_who WITH KEY hdr\_procty = '3090'.  
\*\*                  IF sy-subrc = 0.  
\*\*                    SELECT \* INTO TABLE lt\_interleave FROM zinter\_leave  
\*\*                             WHERE lgnum = iv\_lgnum AND resource\_name = iv\_queue AND active EQ abap\_true.  
\*\*                    IF sy-subrc NE 0.  
\*\*                      INSERT ls\_str\_who INTO ct\_who INDEX 1.  
\*\*  
\*\*                      SELECT \*  INTO TABLE lt\_inter\_who FROM zinter\_leave  
\*\*                                WHERE lgnum = iv\_lgnum AND resource\_name = iv\_queue .  
\*\*                      IF sy-subrc = 0.  
\*\*                        READ TABLE lt\_inter\_who INTO ls\_inter\_who INDEX 1.  
\*\*                        IF ls\_str\_who-who NE ls\_inter\_who-who\_no.  
\*\*                          UPDATE zinter\_leave SET active = abap\_false  
\*\*                                                  who\_no = ls\_str\_who-who  
\*\*                                              WHERE lgnum EQ iv\_lgnum  
\*\*                                              AND resource\_name = iv\_queue.  
\*\*                          COMMIT WORK.  
\*\*                          lv\_missing\_case = 2.  
\*\*                        ENDIF.  
\*\*                      ENDIF.  
\*\*                      RETURN.  
\*\*                    ENDIF.  
\*\*  
\*\*                  ENDIF.  
\*\*  
\*\*                ENDIF.  
\*\*\*\*\*\*\*\*\*\*\*\* end of when we have only putaway tasks  
\*\*  
\*\*\*\*\*\*\*\*\*\*\*\*\* When we have only retrival tasks  
\*\*  
\*\*                SELECT SINGLE zactive INTO lv\_active1 FROM zcon\_param\_val  
\*\*                                      WHERE lgnum = iv\_lgnum  
\*\*                                      AND zreq\_name = 'ZINTER'  
\*\*                                      AND zactive EQ abap\_true.  
\*\*                IF lv\_active1 EQ abap\_true.  
\*\*                  IF lt\_str\_who[] IS INITIAL AND lt\_rtv\_who[] IS NOT INITIAL.  
\*\*  
\*\*                    READ TABLE lt\_rtv\_who INTO ls\_rtv\_who WITH KEY hdr\_procty = '3010'.  
\*\*                    IF sy-subrc = 0.  
\*\*  
\*\*                      INSERT ls\_rtv\_who INTO ct\_who INDEX 1.  
\*\*                      READ TABLE ct\_who INTO cs\_who INDEX 1.  
\*\*                      iv\_who = cs\_who-who.  
\*\*                      ls\_det2\_fg-lgnum = iv\_lgnum.  
\*\*                      ls\_det2\_fg-crane\_no = lv\_crane\_no.  
\*\*                      ls\_det2\_fg-error\_no = ls\_final\_error-error.  
\*\*                      ls\_det2\_fg-who = iv\_who.  
\*\*                      ls\_det2\_fg-description = 'checking for picking task need rewarehousing repl '.  
\*\*                      ls\_det2\_fg-log\_date = sy-datum.  
\*\*                      ls\_det2\_fg-log\_time = sy-uzeit.  
\*\*                      APPEND ls\_det2\_fg TO lt\_det2\_fg.  
\*\*                      MODIFY zdet2\_fg\_log FROM TABLE lt\_det2\_fg.  
\*\*                      COMMIT WORK.  
\*\*                      lv\_missing\_case = 2.  
\*\*                      CALL METHOD lo\_badi\_transfer->transfer\_hu1  
\*\*                        EXPORTING  
\*\*                          iv\_lgnum   = iv\_lgnum  
\*\*                          is\_whorder = is\_who  
\*\*                          iv\_who     = iv\_who  
\*\*                          iv\_rsrc    = is\_rsrc  
\*\*                          it\_ordim\_o = ct\_ordim\_o  
\*\*                        CHANGING  
\*\*                          ev\_who     = lv\_who.  
\*\*  
\*\*                      CLEAR ls\_who.  
\*\*                      IF lv\_who+0(1) NE 0.  
\*\*                        IF iv\_who NE lv\_who.  
\*\*                          "if current WO already contains WT for HU in front of HU to be picked => do nothing  
\*\*                          ls\_who-who = lv\_who.  
\*\*                          INSERT ls\_who INTO ct\_who INDEX 1.  
\*\*                          ls\_det2\_fg-lgnum = iv\_lgnum.  
\*\*                          ls\_det2\_fg-crane\_no = lv\_crane\_no.  
\*\*                          ls\_det2\_fg-who = lv\_who.  
\*\*                          ls\_det2\_fg-description = 'Rewarehousing task sent to subsystem Picking task repl'.  
\*\*                          ls\_det2\_fg-log\_date = sy-datum.  
\*\*                          ls\_det2\_fg-log\_time = sy-uzeit.  
\*\*                          APPEND ls\_det2\_fg TO lt\_det2\_fg.  
\*\*                          MODIFY zdet2\_fg\_log FROM TABLE lt\_det2\_fg.  
\*\*                          COMMIT WORK.  
\*\*  
\*\*                          UPDATE zinter\_leave SET active = abap\_true  
\*\*                                                  who\_no = ls\_who-who  
\*\*                                              WHERE lgnum EQ iv\_lgnum  
\*\*                                              AND resource\_name = iv\_queue.  
\*\*                          COMMIT WORK.  
\*\*                        ENDIF.  
\*\*                        cv\_rearr\_done = abap\_true.  
\*\*                        RETURN.  
\*\*                      ELSE.  
\*\*                        SELECT \*  INTO TABLE lt\_inter\_who FROM zinter\_leave  
\*\*                                   WHERE lgnum = iv\_lgnum  
\*\*                                   AND resource\_name = iv\_queue .  
\*\*                        IF sy-subrc = 0.  
\*\*                          LOOP AT lt\_inter\_who INTO ls\_inter\_who.  
\*\*                            READ TABLE ct\_who INTO cs\_who WITH KEY who = ls\_inter\_who-who\_no.  
\*\*                            IF sy-subrc = 0.  
\*\*                              "do nothing  
\*\*                            ELSE.  
\*\*                              UPDATE zinter\_leave SET active = abap\_false  
\*\*                                                      who\_no = ls\_rtv\_who-who  
\*\*                                                  WHERE lgnum EQ iv\_lgnum  
\*\*                                                  AND resource\_name = iv\_queue.  
\*\*                              COMMIT WORK.  
\*\*                            ENDIF.  
\*\*                          ENDLOOP.  
\*\*                        ENDIF.  
\*\*                        RETURN.  
\*\*  
\*\*  
\*\*                      ENDIF.  
\*\*  
\*\*                      SELECT \*  INTO TABLE lt\_inter\_who FROM zinter\_leave  
\*\*                                WHERE lgnum = iv\_lgnum AND resource\_name = iv\_queue .  
\*\*                      IF sy-subrc = 0.  
\*\*                        LOOP AT lt\_inter\_who INTO ls\_inter\_who.  
\*\*                          READ TABLE ct\_who INTO cs\_who WITH KEY who = ls\_inter\_who-who\_no.  
\*\*                          IF sy-subrc = 0.  
\*\*                            "do nothing  
\*\*                          ELSE.  
\*\*                            UPDATE zinter\_leave SET active = abap\_true  
\*\*                                                    who\_no = ls\_rtv\_who-who  
\*\*                                                WHERE lgnum EQ iv\_lgnum  
\*\*                                                AND resource\_name = iv\_queue.  
\*\*                            COMMIT WORK.  
\*\*                          ENDIF.  
\*\*                        ENDLOOP.  
\*\*                      ENDIF.  
\*\*                      RETURN.  
\*\*                    ENDIF.  
\*\*  
\*\*                    READ TABLE lt\_rtv\_who INTO ls\_rtv\_who WITH KEY hdr\_procty = '2010'.  
\*\*                    IF sy-subrc = 0.  
\*\*  
\*\*                      INSERT ls\_rtv\_who INTO ct\_who INDEX 1.  
\*\*                      READ TABLE ct\_who INTO cs\_who INDEX 1.  
\*\*                      iv\_who = cs\_who-who.  
\*\*                      ls\_det2\_fg-lgnum = iv\_lgnum.  
\*\*                      ls\_det2\_fg-crane\_no = lv\_crane\_no.  
\*\*                      ls\_det2\_fg-error\_no = ls\_final\_error-error.  
\*\*                      ls\_det2\_fg-who = iv\_who.  
\*\*                      ls\_det2\_fg-description = 'checking for picking task need rewarehousing '.  
\*\*                      ls\_det2\_fg-log\_date = sy-datum.  
\*\*                      ls\_det2\_fg-log\_time = sy-uzeit.  
\*\*                      APPEND ls\_det2\_fg TO lt\_det2\_fg.  
\*\*                      MODIFY zdet2\_fg\_log FROM TABLE lt\_det2\_fg.  
\*\*                      COMMIT WORK.  
\*\*                      lv\_missing\_case = 2.  
\*\*                      CALL METHOD lo\_badi\_transfer->transfer\_hu1  
\*\*                        EXPORTING  
\*\*                          iv\_lgnum   = iv\_lgnum  
\*\*                          is\_whorder = is\_who  
\*\*                          iv\_who     = iv\_who  
\*\*                          iv\_rsrc    = is\_rsrc  
\*\*                          it\_ordim\_o = ct\_ordim\_o  
\*\*                        CHANGING  
\*\*                          ev\_who     = lv\_who.  
\*\*  
\*\*                      CLEAR ls\_who.  
\*\*                      IF lv\_who+0(1) NE 0.  
\*\*                        IF iv\_who NE lv\_who.  
\*\*                          "if current WO already contains WT for HU in front of HU to be picked => do nothing  
\*\*                          ls\_who-who = lv\_who.  
\*\*                          INSERT ls\_who INTO ct\_who INDEX 1.  
\*\*                          ls\_det2\_fg-lgnum = iv\_lgnum.  
\*\*                          ls\_det2\_fg-crane\_no = lv\_crane\_no.  
\*\*                          ls\_det2\_fg-who = lv\_who.  
\*\*                          ls\_det2\_fg-description = 'Rewarehousing task sent to subsystem Picking task'.  
\*\*                          ls\_det2\_fg-log\_date = sy-datum.  
\*\*                          ls\_det2\_fg-log\_time = sy-uzeit.  
\*\*                          APPEND ls\_det2\_fg TO lt\_det2\_fg.  
\*\*                          MODIFY zdet2\_fg\_log FROM TABLE lt\_det2\_fg.  
\*\*                          COMMIT WORK.  
\*\*  
\*\*                          UPDATE zinter\_leave SET active = abap\_true  
\*\*                                                  who\_no = ls\_who-who  
\*\*                                              WHERE lgnum EQ iv\_lgnum  
\*\*                                              AND resource\_name = iv\_queue.  
\*\*                          COMMIT WORK.  
\*\*                        ENDIF.  
\*\*                        cv\_rearr\_done = abap\_true.  
\*\*                        RETURN.  
\*\*                      ELSE.  
\*\*                        SELECT \*  INTO TABLE lt\_inter\_who FROM zinter\_leave  
\*\*                                  WHERE lgnum = iv\_lgnum AND resource\_name = iv\_queue .  
\*\*                        IF sy-subrc = 0.  
\*\*                          LOOP AT lt\_inter\_who INTO ls\_inter\_who.  
\*\*                            READ TABLE ct\_who INTO cs\_who WITH KEY who = ls\_inter\_who-who\_no.  
\*\*                            IF sy-subrc = 0.  
\*\*                              "do nothing  
\*\*                            ELSE.  
\*\*                              UPDATE zinter\_leave SET active = abap\_false  
\*\*                                                      who\_no = ls\_rtv\_who-who  
\*\*                                                  WHERE lgnum EQ iv\_lgnum  
\*\*                                                  AND resource\_name = iv\_queue.  
\*\*                              COMMIT WORK.  
\*\*                            ENDIF.  
\*\*                          ENDLOOP.  
\*\*                        ENDIF.  
\*\*                        RETURN.  
\*\*                      ENDIF.  
\*\*  
\*\*                      SELECT \*  INTO TABLE lt\_inter\_who FROM zinter\_leave  
\*\*                                WHERE lgnum = iv\_lgnum AND resource\_name = iv\_queue .  
\*\*                      IF sy-subrc = 0.  
\*\*                        LOOP AT lt\_inter\_who INTO ls\_inter\_who.  
\*\*                          READ TABLE ct\_who INTO cs\_who WITH KEY who = ls\_inter\_who-who\_no.  
\*\*                          IF sy-subrc = 0.  
\*\*                            "do nothing  
\*\*                          ELSE.  
\*\*                            UPDATE zinter\_leave SET active = abap\_true  
\*\*                                                    who\_no = ls\_rtv\_who-who  
\*\*                                                WHERE lgnum EQ iv\_lgnum  
\*\*                                                AND resource\_name = iv\_queue.  
\*\*                            COMMIT WORK.  
\*\*                          ENDIF.  
\*\*                        ENDLOOP.  
\*\*                      ENDIF.  
\*\*                      RETURN.  
\*\*                    ENDIF.  
\*\*                  ENDIF.  
\*\*                ELSE.  
\*\*                  IF lt\_str\_who[] IS INITIAL AND lt\_rtv\_who[] IS NOT INITIAL.  
\*\*  
\*\*                    READ TABLE lt\_rtv\_who INTO ls\_rtv\_who WITH KEY hdr\_procty = '3010'.  
\*\*                    IF sy-subrc = 0.  
\*\*                      SELECT \* INTO TABLE lt\_interleave FROM zinter\_leave  
\*\*                               WHERE lgnum = iv\_lgnum  
\*\*                               AND resource\_name = iv\_queue  
\*\*                               AND active EQ abap\_true.  
\*\*                      IF sy-subrc = 0.  
\*\*  
\*\*                        READ TABLE lt\_rtv\_who INTO ls\_rtv\_who WITH KEY hdr\_procty = '3010'.  
\*\*                        IF sy-subrc = 0.  
\*\*                          INSERT ls\_rtv\_who INTO ct\_who INDEX 1.  
\*\*                          READ TABLE ct\_who INTO cs\_who INDEX 1.  
\*\*                          iv\_who = cs\_who-who.  
\*\*                          ls\_det2\_fg-lgnum = iv\_lgnum.  
\*\*                          ls\_det2\_fg-crane\_no = lv\_crane\_no.  
\*\*                          ls\_det2\_fg-error\_no = ls\_final\_error-error.  
\*\*                          ls\_det2\_fg-who = iv\_who.  
\*\*                          ls\_det2\_fg-description = 'checking for picking task need rewarehousing repl '.  
\*\*                          ls\_det2\_fg-log\_date = sy-datum.  
\*\*                          ls\_det2\_fg-log\_time = sy-uzeit.  
\*\*                          APPEND ls\_det2\_fg TO lt\_det2\_fg.  
\*\*                          MODIFY zdet2\_fg\_log FROM TABLE lt\_det2\_fg.  
\*\*                          COMMIT WORK.  
\*\*                          lv\_missing\_case = 2.  
\*\*                          CALL METHOD lo\_badi\_transfer->transfer\_hu1  
\*\*                            EXPORTING  
\*\*                              iv\_lgnum   = iv\_lgnum  
\*\*                              is\_whorder = is\_who  
\*\*                              iv\_who     = iv\_who  
\*\*                              iv\_rsrc    = is\_rsrc  
\*\*                              it\_ordim\_o = ct\_ordim\_o  
\*\*                            CHANGING  
\*\*                              ev\_who     = lv\_who.  
\*\*  
\*\*                          CLEAR ls\_who.  
\*\*                          IF lv\_who+0(1) NE 0.  
\*\*                            IF iv\_who NE lv\_who.  
\*\*                              "if current WO already contains WT for HU in front of HU to be picked => do nothing  
\*\*                              ls\_who-who = lv\_who.  
\*\*                              INSERT ls\_who INTO ct\_who INDEX 1.  
\*\*                              ls\_det2\_fg-lgnum = iv\_lgnum.  
\*\*                              ls\_det2\_fg-crane\_no = lv\_crane\_no.  
\*\*                              ls\_det2\_fg-who = lv\_who.  
\*\*                              ls\_det2\_fg-description = 'Rewarehousing task sent to subsystem Picking task repl'.  
\*\*                              ls\_det2\_fg-log\_date = sy-datum.  
\*\*                              ls\_det2\_fg-log\_time = sy-uzeit.  
\*\*                              APPEND ls\_det2\_fg TO lt\_det2\_fg.  
\*\*                              MODIFY zdet2\_fg\_log FROM TABLE lt\_det2\_fg.  
\*\*                              COMMIT WORK.  
\*\*  
\*\*                              UPDATE zinter\_leave SET active = abap\_true  
\*\*                                                      who\_no = ls\_who-who  
\*\*                                                  WHERE lgnum EQ iv\_lgnum  
\*\*                                                  AND resource\_name = iv\_queue.  
\*\*                              COMMIT WORK.  
\*\*                            ENDIF.  
\*\*                            cv\_rearr\_done = abap\_true.  
\*\*                            RETURN.  
\*\*                          ELSE.  
\*\*                            SELECT \*  INTO TABLE lt\_inter\_who FROM zinter\_leave  
\*\*                                      WHERE lgnum = iv\_lgnum AND resource\_name = iv\_queue .  
\*\*                            IF sy-subrc = 0.  
\*\*                              LOOP AT lt\_inter\_who INTO ls\_inter\_who.  
\*\*                                READ TABLE ct\_who INTO cs\_who WITH KEY who = ls\_inter\_who-who\_no.  
\*\*                                IF sy-subrc = 0.  
\*\*                                  "do nothing  
\*\*                                ELSE.  
\*\*                                  UPDATE zinter\_leave SET active = abap\_false  
\*\*                                                          who\_no = ls\_rtv\_who-who  
\*\*                                                      WHERE lgnum EQ iv\_lgnum AND resource\_name = iv\_queue.  
\*\*                                  COMMIT WORK.  
\*\*                                ENDIF.  
\*\*                              ENDLOOP.  
\*\*                            ENDIF.  
\*\*                            RETURN.  
\*\*  
\*\*  
\*\*                          ENDIF.  
\*\*  
\*\*                          SELECT \*  INTO TABLE lt\_inter\_who FROM zinter\_leave  
\*\*                                    WHERE lgnum = iv\_lgnum AND resource\_name = iv\_queue .  
\*\*                          IF sy-subrc = 0.  
\*\*                            LOOP AT lt\_inter\_who INTO ls\_inter\_who.  
\*\*                              READ TABLE ct\_who INTO cs\_who WITH KEY who = ls\_inter\_who-who\_no.  
\*\*                              IF sy-subrc = 0.  
\*\*                                "do nothing  
\*\*                              ELSE.  
\*\*                                UPDATE zinter\_leave SET active = abap\_true  
\*\*                                                        who\_no = ls\_rtv\_who-who  
\*\*                                                    WHERE lgnum EQ iv\_lgnum  
\*\*                                                    AND resource\_name = iv\_queue.  
\*\*                                COMMIT WORK.  
\*\*                              ENDIF.  
\*\*                            ENDLOOP.  
\*\*                          ENDIF.  
\*\*                          RETURN.  
\*\*                        ENDIF."sy-subrc = 0 end  
\*\*  
\*\*                      ENDIF.  
\*\*                    ENDIF.  
\*\*  
\*\*                    READ TABLE lt\_rtv\_who INTO ls\_rtv\_who WITH KEY hdr\_procty = '2010'.  
\*\*                    IF sy-subrc = 0.  
\*\*                      SELECT \* INTO TABLE lt\_interleave FROM zinter\_leave  
\*\*                               WHERE lgnum = iv\_lgnum AND resource\_name = iv\_queue AND active EQ abap\_true.  
\*\*                      IF sy-subrc = 0.  
\*\*  
\*\*                        READ TABLE lt\_rtv\_who INTO ls\_rtv\_who WITH KEY hdr\_procty = '2010'.  
\*\*                        IF sy-subrc = 0.  
\*\*                          INSERT ls\_rtv\_who INTO ct\_who INDEX 1.  
\*\*                          READ TABLE ct\_who INTO cs\_who INDEX 1.  
\*\*                          iv\_who = cs\_who-who.  
\*\*                          ls\_det2\_fg-lgnum = iv\_lgnum.  
\*\*                          ls\_det2\_fg-crane\_no = lv\_crane\_no.  
\*\*                          ls\_det2\_fg-error\_no = ls\_final\_error-error.  
\*\*                          ls\_det2\_fg-who = iv\_who.  
\*\*                          ls\_det2\_fg-description = 'checking for picking task need rewarehousing '.  
\*\*                          ls\_det2\_fg-log\_date = sy-datum.  
\*\*                          ls\_det2\_fg-log\_time = sy-uzeit.  
\*\*                          APPEND ls\_det2\_fg TO lt\_det2\_fg.  
\*\*                          MODIFY zdet2\_fg\_log FROM TABLE lt\_det2\_fg.  
\*\*                          COMMIT WORK.  
\*\*                          lv\_missing\_case = 2.  
\*\*                          CALL METHOD lo\_badi\_transfer->transfer\_hu1  
\*\*                            EXPORTING  
\*\*                              iv\_lgnum   = iv\_lgnum  
\*\*                              is\_whorder = is\_who  
\*\*                              iv\_who     = iv\_who  
\*\*                              iv\_rsrc    = is\_rsrc  
\*\*                              it\_ordim\_o = ct\_ordim\_o  
\*\*                            CHANGING  
\*\*                              ev\_who     = lv\_who.  
\*\*  
\*\*                          CLEAR ls\_who.  
\*\*                          IF lv\_who+0(1) NE 0.  
\*\*                            IF iv\_who NE lv\_who.  
\*\*                              "if current WO already contains WT for HU in front of HU to be picked => do nothing  
\*\*                              ls\_who-who = lv\_who.  
\*\*                              INSERT ls\_who INTO ct\_who INDEX 1.  
\*\*                              ls\_det2\_fg-lgnum = iv\_lgnum.  
\*\*                              ls\_det2\_fg-crane\_no = lv\_crane\_no.  
\*\*                              ls\_det2\_fg-who = lv\_who.  
\*\*                              ls\_det2\_fg-description = 'Rewarehousing task sent to subsystem Picking task'.  
\*\*                              ls\_det2\_fg-log\_date = sy-datum.  
\*\*                              ls\_det2\_fg-log\_time = sy-uzeit.  
\*\*                              APPEND ls\_det2\_fg TO lt\_det2\_fg.  
\*\*                              MODIFY zdet2\_fg\_log FROM TABLE lt\_det2\_fg.  
\*\*                              COMMIT WORK.  
\*\*  
\*\*                              UPDATE zinter\_leave SET active = abap\_true  
\*\*                                                      who\_no = ls\_who-who  
\*\*                                                  WHERE lgnum EQ iv\_lgnum  
\*\*                                                  AND resource\_name = iv\_queue.  
\*\*                              COMMIT WORK.  
\*\*                            ENDIF.  
\*\*                            cv\_rearr\_done = abap\_true.  
\*\*                            RETURN.  
\*\*                          ELSE.  
\*\*                            SELECT \*  INTO TABLE lt\_inter\_who FROM zinter\_leave  
\*\*                                      WHERE lgnum = iv\_lgnum AND resource\_name = iv\_queue .  
\*\*                            IF sy-subrc = 0.  
\*\*                              LOOP AT lt\_inter\_who INTO ls\_inter\_who.  
\*\*                                READ TABLE ct\_who INTO cs\_who WITH KEY who = ls\_inter\_who-who\_no.  
\*\*                                IF sy-subrc = 0.  
\*\*                                  "do nothing  
\*\*                                ELSE.  
\*\*                                  UPDATE zinter\_leave SET active = abap\_false  
\*\*                                                          who\_no = ls\_rtv\_who-who  
\*\*                                                      WHERE lgnum EQ iv\_lgnum  
\*\*                                                      AND resource\_name = iv\_queue.  
\*\*                                  COMMIT WORK.  
\*\*                                ENDIF.  
\*\*                              ENDLOOP.  
\*\*                            ENDIF.  
\*\*                            RETURN.  
\*\*                          ENDIF.  
\*\*  
\*\*                          SELECT \*  INTO TABLE lt\_inter\_who FROM zinter\_leave  
\*\*                                    WHERE lgnum = iv\_lgnum AND resource\_name = iv\_queue .  
\*\*                          IF sy-subrc = 0.  
\*\*                            LOOP AT lt\_inter\_who INTO ls\_inter\_who.  
\*\*                              READ TABLE ct\_who INTO cs\_who WITH KEY who = ls\_inter\_who-who\_no.  
\*\*                              IF sy-subrc = 0.  
\*\*                                "do nothing  
\*\*                              ELSE.  
\*\*                                UPDATE zinter\_leave SET active = abap\_true  
\*\*                                                        who\_no = ls\_rtv\_who-who  
\*\*                                                    WHERE lgnum EQ iv\_lgnum  
\*\*                                                    AND resource\_name = iv\_queue.  
\*\*                                COMMIT WORK.  
\*\*                              ENDIF.  
\*\*                            ENDLOOP.  
\*\*                          ENDIF.  
\*\*                          RETURN.  
\*\*                        ENDIF."sy-subrc = 0 end  
\*\*  
\*\*                      ENDIF.  
\*\*  
\*\*                    ENDIF.  
\*\*                  ENDIF.  
\*\*                ENDIF.  
\*\*\*\*\*\*\*\*\*\*\*\* end of when we have retrival tasks  
\*\*  
\*\*                READ TABLE lt\_str\_who INTO ls\_str\_who WITH KEY hdr\_procty = '3090'.  
\*\*                IF sy-subrc = 0.  
\*\*                  SELECT \* INTO TABLE lt\_interleave FROM zinter\_leave  
\*\*                           WHERE lgnum = iv\_lgnum  
\*\*                           AND resource\_name = iv\_queue  
\*\*                           AND active EQ abap\_true.  
\*\*                  IF sy-subrc NE 0.  
\*\*                    INSERT ls\_str\_who INTO ct\_who INDEX 1.  
\*\*  
\*\*                    SELECT \*  INTO TABLE lt\_inter\_who FROM zinter\_leave  
\*\*                              WHERE lgnum = iv\_lgnum AND resource\_name = iv\_queue .  
\*\*                    IF sy-subrc = 0.  
\*\*                      READ TABLE lt\_inter\_who INTO ls\_inter\_who INDEX 1.  
\*\*                      IF ls\_str\_who-who NE ls\_inter\_who-who\_no.  
\*\*                        UPDATE zinter\_leave SET active = abap\_true  
\*\*                                                who\_no = ls\_str\_who-who  
\*\*                                            WHERE lgnum EQ iv\_lgnum  
\*\*                                            AND resource\_name = iv\_queue.  
\*\*                        COMMIT WORK.  
\*\*                        lv\_missing\_case = 2.  
\*\*                      ENDIF.  
\*\*                    ENDIF.  
\*\*                    COMMIT WORK.  
\*\*                    RETURN.  
\*\*                  ELSEIF sy-subrc = 0.  
\*\*                    READ TABLE lt\_rtv\_who INTO ls\_rtv\_who WITH KEY hdr\_procty = '3010'.  
\*\*                    IF sy-subrc = 0.  
\*\*                      INSERT ls\_rtv\_who INTO ct\_who INDEX 1.  
\*\*                      READ TABLE ct\_who INTO cs\_who INDEX 1.  
\*\*                      iv\_who = cs\_who-who.  
\*\*                      ls\_det2\_fg-lgnum = iv\_lgnum.  
\*\*                      ls\_det2\_fg-crane\_no = lv\_crane\_no.  
\*\*                      ls\_det2\_fg-error\_no = ls\_final\_error-error.  
\*\*                      ls\_det2\_fg-who = iv\_who.  
\*\*                      ls\_det2\_fg-description = 'checking for putaway task need rewarehousing for picking repl '.  
\*\*                      ls\_det2\_fg-log\_date = sy-datum.  
\*\*                      ls\_det2\_fg-log\_time = sy-uzeit.  
\*\*                      APPEND ls\_det2\_fg TO lt\_det2\_fg.  
\*\*                      MODIFY zdet2\_fg\_log FROM TABLE lt\_det2\_fg.  
\*\*                      COMMIT WORK.  
\*\*                      lv\_missing\_case = 2.  
\*\*                      CALL METHOD lo\_badi\_transfer->transfer\_hu1  
\*\*                        EXPORTING  
\*\*                          iv\_lgnum   = iv\_lgnum  
\*\*                          is\_whorder = is\_who  
\*\*                          iv\_who     = iv\_who  
\*\*                          iv\_rsrc    = is\_rsrc  
\*\*                          it\_ordim\_o = ct\_ordim\_o  
\*\*                        CHANGING  
\*\*                          ev\_who     = lv\_who.  
\*\*  
\*\*                      CLEAR ls\_who.  
\*\*                      IF lv\_who+0(1) NE 0.  
\*\*                        IF iv\_who NE lv\_who.  
\*\*                          "if current WO already contains WT for HU in front of HU to be picked => do nothing  
\*\*                          ls\_who-who = lv\_who.  
\*\*                          INSERT ls\_who INTO ct\_who INDEX 1.  
\*\*                          ls\_det2\_fg-lgnum = iv\_lgnum.  
\*\*                          ls\_det2\_fg-crane\_no = lv\_crane\_no.  
\*\*                          ls\_det2\_fg-who = lv\_who.  
\*\*                          ls\_det2\_fg-description = 'Rewarehousing task sent to subsystem for picking repl'.  
\*\*                          ls\_det2\_fg-log\_date = sy-datum.  
\*\*                          ls\_det2\_fg-log\_time = sy-uzeit.  
\*\*                          APPEND ls\_det2\_fg TO lt\_det2\_fg.  
\*\*                          MODIFY zdet2\_fg\_log FROM TABLE lt\_det2\_fg.  
\*\*                          COMMIT WORK.  
\*\*                          UPDATE zinter\_leave SET active = abap\_true  
\*\*                                                  who\_no = ls\_who-who  
\*\*                                              WHERE lgnum EQ iv\_lgnum  
\*\*                                              AND resource\_name = iv\_queue.  
\*\*                          COMMIT WORK.  
\*\*                        ENDIF.  
\*\*                        cv\_rearr\_done = abap\_true.  
\*\*                        RETURN.  
\*\*                      ELSE.  
\*\*                        SELECT \*  INTO TABLE lt\_inter\_who FROM zinter\_leave  
\*\*                                  WHERE lgnum = iv\_lgnum AND resource\_name = iv\_queue .  
\*\*                        IF sy-subrc = 0.  
\*\*                          LOOP AT lt\_inter\_who INTO ls\_inter\_who.  
\*\*                            READ TABLE ct\_who INTO cs\_who WITH KEY who = ls\_inter\_who-who\_no.  
\*\*                            IF sy-subrc = 0.  
\*\*                              "do nothing  
\*\*                            ELSE.  
\*\*                              UPDATE zinter\_leave SET active = abap\_false  
\*\*                                                      who\_no = ls\_rtv\_who-who  
\*\*                                                  WHERE lgnum EQ iv\_lgnum  
\*\*                                                  AND resource\_name = iv\_queue.  
\*\*                              COMMIT WORK.  
\*\*                            ENDIF.  
\*\*                          ENDLOOP.  
\*\*                        ENDIF.  
\*\*                        RETURN.  
\*\*                      ENDIF.  
\*\*  
\*\*  
\*\*  
\*\*                      SELECT \*  INTO TABLE lt\_inter\_who FROM zinter\_leave  
\*\*                                WHERE lgnum = iv\_lgnum AND resource\_name = iv\_queue .  
\*\*                      IF sy-subrc = 0.  
\*\*                        LOOP AT lt\_inter\_who INTO ls\_inter\_who.  
\*\*                          READ TABLE ct\_who INTO cs\_who WITH KEY who = ls\_inter\_who-who\_no.  
\*\*                          IF sy-subrc = 0.  
\*\*                            "do nothing  
\*\*                          ELSE.  
\*\*                            UPDATE zinter\_leave SET active = abap\_false  
\*\*                                                    who\_no = ls\_rtv\_who-who  
\*\*                                                WHERE lgnum EQ iv\_lgnum  
\*\*                                                AND resource\_name = iv\_queue.  
\*\*                            COMMIT WORK.  
\*\*                          ENDIF.  
\*\*                        ENDLOOP.  
\*\*                      ENDIF.  
\*\*                      RETURN.  
\*\*                    ELSE."3010 task is not present  
\*\*                      READ TABLE lt\_rtv\_who INTO ls\_rtv\_who WITH KEY hdr\_procty = '2010'.  
\*\*                      IF sy-subrc = 0.  
\*\*                        INSERT ls\_rtv\_who INTO ct\_who INDEX 1.  
\*\*                        READ TABLE ct\_who INTO cs\_who INDEX 1.  
\*\*                        iv\_who = cs\_who-who.  
\*\*                        ls\_det2\_fg-lgnum = iv\_lgnum.  
\*\*                        ls\_det2\_fg-crane\_no = lv\_crane\_no.  
\*\*                        ls\_det2\_fg-error\_no = ls\_final\_error-error.  
\*\*                        ls\_det2\_fg-who = iv\_who.  
\*\*                        ls\_det2\_fg-description = 'checking for putaway task need rewarehousing for picking '.  
\*\*                        ls\_det2\_fg-log\_date = sy-datum.  
\*\*                        ls\_det2\_fg-log\_time = sy-uzeit.  
\*\*                        APPEND ls\_det2\_fg TO lt\_det2\_fg.  
\*\*                        MODIFY zdet2\_fg\_log FROM TABLE lt\_det2\_fg.  
\*\*                        COMMIT WORK.  
\*\*                        lv\_missing\_case = 2.  
\*\*                        CALL METHOD lo\_badi\_transfer->transfer\_hu1  
\*\*                          EXPORTING  
\*\*                            iv\_lgnum   = iv\_lgnum  
\*\*                            is\_whorder = is\_who  
\*\*                            iv\_who     = iv\_who  
\*\*                            iv\_rsrc    = is\_rsrc  
\*\*                            it\_ordim\_o = ct\_ordim\_o  
\*\*                          CHANGING  
\*\*                            ev\_who     = lv\_who.  
\*\*  
\*\*                        CLEAR ls\_who.  
\*\*                        IF lv\_who+0(1) NE 0.  
\*\*                          IF iv\_who NE lv\_who.  
\*\*                            "if current WO already contains WT for HU in front of HU to be picked => do nothing  
\*\*                            ls\_who-who = lv\_who.  
\*\*                            INSERT ls\_who INTO ct\_who INDEX 1.  
\*\*                            ls\_det2\_fg-lgnum = iv\_lgnum.  
\*\*                            ls\_det2\_fg-crane\_no = lv\_crane\_no.  
\*\*                            ls\_det2\_fg-who = lv\_who.  
\*\*                            ls\_det2\_fg-description = 'Rewarehousing task sent to subsystem for picking'.  
\*\*                            ls\_det2\_fg-log\_date = sy-datum.  
\*\*                            ls\_det2\_fg-log\_time = sy-uzeit.  
\*\*                            APPEND ls\_det2\_fg TO lt\_det2\_fg.  
\*\*                            MODIFY zdet2\_fg\_log FROM TABLE lt\_det2\_fg.  
\*\*                            COMMIT WORK.  
\*\*  
\*\*  
\*\*                            UPDATE zinter\_leave SET active = abap\_true  
\*\*                                                    who\_no = ls\_who-who  
\*\*                                                WHERE lgnum EQ iv\_lgnum  
\*\*                                                AND resource\_name = iv\_queue.  
\*\*                            COMMIT WORK.  
\*\*                          ENDIF.  
\*\*                          cv\_rearr\_done = abap\_true.  
\*\*                          RETURN.  
\*\*                        ELSE.  
\*\*                          SELECT \*  INTO TABLE lt\_inter\_who FROM zinter\_leave  
\*\*                                    WHERE lgnum = iv\_lgnum AND resource\_name = iv\_queue .  
\*\*                          IF sy-subrc = 0.  
\*\*                            LOOP AT lt\_inter\_who INTO ls\_inter\_who.  
\*\*                              READ TABLE ct\_who INTO cs\_who WITH KEY who = ls\_inter\_who-who\_no.  
\*\*                              IF sy-subrc = 0.  
\*\*                                "do nothing  
\*\*                              ELSE.  
\*\*                                UPDATE zinter\_leave SET active = abap\_false  
\*\*                                                        who\_no = ls\_rtv\_who-who  
\*\*                                                    WHERE lgnum EQ iv\_lgnum  
\*\*                                                    AND resource\_name = iv\_queue.  
\*\*                                COMMIT WORK.  
\*\*                              ENDIF.  
\*\*                            ENDLOOP.  
\*\*                          ENDIF.  
\*\*                          RETURN.  
\*\*                        ENDIF.  
\*\*  
\*\*  
\*\*                        SELECT \*  INTO TABLE lt\_inter\_who FROM zinter\_leave  
\*\*                                  WHERE lgnum = iv\_lgnum AND resource\_name = iv\_queue .  
\*\*                        IF sy-subrc = 0.  
\*\*                          LOOP AT lt\_inter\_who INTO ls\_inter\_who.  
\*\*                            READ TABLE ct\_who INTO cs\_who WITH KEY who = ls\_inter\_who-who\_no.  
\*\*                            IF sy-subrc = 0.  
\*\*                              "do nothing  
\*\*                            ELSE.  
\*\*                              UPDATE zinter\_leave SET active = abap\_false  
\*\*                                                      who\_no = ls\_rtv\_who-who  
\*\*                                                  WHERE lgnum EQ iv\_lgnum  
\*\*                                                  AND resource\_name = iv\_queue.  
\*\*                              COMMIT WORK.  
\*\*                            ENDIF.  
\*\*                          ENDLOOP.  
\*\*                        ENDIF.  
\*\*  
\*\*                        RETURN.  
\*\*                      ENDIF."sy-subrc = 0 end  
\*\*                    ENDIF.  
\*\*                  ENDIF.  
\*\*                ELSE." lt\_str\_who is initial  
\*\*                  READ TABLE lt\_rtv\_who INTO ls\_rtv\_who WITH KEY hdr\_procty = '3010'.  
\*\*                  IF sy-subrc = 0.  
\*\*                    INSERT ls\_rtv\_who INTO ct\_who INDEX 1.  
\*\*                    READ TABLE ct\_who INTO cs\_who INDEX 1.  
\*\*                    iv\_who = cs\_who-who.  
\*\*                    ls\_det2\_fg-lgnum = iv\_lgnum.  
\*\*                    ls\_det2\_fg-crane\_no = lv\_crane\_no.  
\*\*                    ls\_det2\_fg-error\_no = ls\_final\_error-error.  
\*\*                    ls\_det2\_fg-who = iv\_who.  
\*\*                    ls\_det2\_fg-description = 'checking for putaway task need rewarehousing for picking repl '.  
\*\*                    ls\_det2\_fg-log\_date = sy-datum.  
\*\*                    ls\_det2\_fg-log\_time = sy-uzeit.  
\*\*                    APPEND ls\_det2\_fg TO lt\_det2\_fg.  
\*\*                    MODIFY zdet2\_fg\_log FROM TABLE lt\_det2\_fg.  
\*\*                    COMMIT WORK.  
\*\*                    lv\_missing\_case = 2.  
\*\*                    CALL METHOD lo\_badi\_transfer->transfer\_hu1  
\*\*                      EXPORTING  
\*\*                        iv\_lgnum   = iv\_lgnum  
\*\*                        is\_whorder = is\_who  
\*\*                        iv\_who     = iv\_who  
\*\*                        iv\_rsrc    = is\_rsrc  
\*\*                        it\_ordim\_o = ct\_ordim\_o  
\*\*                      CHANGING  
\*\*                        ev\_who     = lv\_who.  
\*\*  
\*\*                    CLEAR ls\_who.  
\*\*                    IF lv\_who+0(1) NE 0.  
\*\*                      IF iv\_who NE lv\_who.  
\*\*                        "if current WO already contains WT for HU in front of HU to be picked => do nothing  
\*\*                        ls\_who-who = lv\_who.  
\*\*                        INSERT ls\_who INTO ct\_who INDEX 1.  
\*\*                        ls\_det2\_fg-lgnum = iv\_lgnum.  
\*\*                        ls\_det2\_fg-crane\_no = lv\_crane\_no.  
\*\*                        ls\_det2\_fg-who = lv\_who.  
\*\*                        ls\_det2\_fg-description = 'Rewarehousing task sent to subsystem for picking repl'.  
\*\*                        ls\_det2\_fg-log\_date = sy-datum.  
\*\*                        ls\_det2\_fg-log\_time = sy-uzeit.  
\*\*                        APPEND ls\_det2\_fg TO lt\_det2\_fg.  
\*\*                        MODIFY zdet2\_fg\_log FROM TABLE lt\_det2\_fg.  
\*\*                        COMMIT WORK.  
\*\*  
\*\*  
\*\*                        UPDATE zinter\_leave SET active = abap\_true  
\*\*                                                who\_no = ls\_who-who  
\*\*                                            WHERE lgnum EQ iv\_lgnum  
\*\*                                            AND resource\_name = iv\_queue.  
\*\*                        COMMIT WORK.  
\*\*  
\*\*  
\*\*                      ENDIF.  
\*\*                      cv\_rearr\_done = abap\_true.  
\*\*                      RETURN.  
\*\*                    ELSE.  
\*\*                      SELECT \*  INTO TABLE lt\_inter\_who FROM zinter\_leave  
\*\*                                WHERE lgnum = iv\_lgnum AND resource\_name = iv\_queue .  
\*\*                      IF sy-subrc = 0.  
\*\*                        LOOP AT lt\_inter\_who INTO ls\_inter\_who.  
\*\*                          READ TABLE ct\_who INTO cs\_who WITH KEY who = ls\_inter\_who-who\_no.  
\*\*                          IF sy-subrc = 0.  
\*\*                            "do nothing  
\*\*                          ELSE.  
\*\*                            UPDATE zinter\_leave SET active = abap\_false  
\*\*                                                    who\_no = ls\_rtv\_who-who  
\*\*                                                WHERE lgnum EQ iv\_lgnum  
\*\*                                                AND resource\_name = iv\_queue.  
\*\*                            COMMIT WORK.  
\*\*                          ENDIF.  
\*\*                        ENDLOOP.  
\*\*                      ENDIF.  
\*\*                      RETURN.  
\*\*                    ENDIF.  
\*\*  
\*\*                    SELECT \*  INTO TABLE lt\_inter\_who FROM zinter\_leave  
\*\*                              WHERE lgnum = iv\_lgnum AND resource\_name = iv\_queue .  
\*\*                    IF sy-subrc = 0.  
\*\*                      LOOP AT lt\_inter\_who INTO ls\_inter\_who.  
\*\*                        READ TABLE ct\_who INTO cs\_who WITH KEY who = ls\_inter\_who-who\_no.  
\*\*                        IF sy-subrc = 0.  
\*\*                          "do nothing  
\*\*                        ELSE.  
\*\*                          UPDATE zinter\_leave SET active = abap\_false  
\*\*                                                  who\_no = ls\_rtv\_who-who  
\*\*                                              WHERE lgnum EQ iv\_lgnum  
\*\*                                              AND resource\_name = iv\_queue.  
\*\*                          COMMIT WORK.  
\*\*                        ENDIF.  
\*\*                      ENDLOOP.  
\*\*                    ENDIF.  
\*\*  
\*\*                    RETURN.  
\*\*                  ELSE."3010 task is not present when lt\_str\_who is initial  
\*\*                    READ TABLE lt\_rtv\_who INTO ls\_rtv\_who WITH KEY hdr\_procty = '2010'.  
\*\*                    IF sy-subrc = 0.  
\*\*                      INSERT ls\_rtv\_who INTO ct\_who INDEX 1.  
\*\*                      READ TABLE ct\_who INTO cs\_who INDEX 1.  
\*\*                      iv\_who = cs\_who-who.  
\*\*                      ls\_det2\_fg-lgnum = iv\_lgnum.  
\*\*                      ls\_det2\_fg-crane\_no = lv\_crane\_no.  
\*\*                      ls\_det2\_fg-error\_no = ls\_final\_error-error.  
\*\*                      ls\_det2\_fg-who = iv\_who.  
\*\*                      ls\_det2\_fg-description = 'checking for putaway task need rewarehousing for picking  '.  
\*\*                      ls\_det2\_fg-log\_date = sy-datum.  
\*\*                      ls\_det2\_fg-log\_time = sy-uzeit.  
\*\*  
\*\*                      APPEND ls\_det2\_fg TO lt\_det2\_fg.  
\*\*                      MODIFY zdet2\_fg\_log FROM TABLE lt\_det2\_fg.  
\*\*                      COMMIT WORK.  
\*\*                      CALL METHOD lo\_badi\_transfer->transfer\_hu1  
\*\*                        EXPORTING  
\*\*                          iv\_lgnum   = iv\_lgnum  
\*\*                          is\_whorder = is\_who  
\*\*                          iv\_who     = iv\_who  
\*\*                          iv\_rsrc    = is\_rsrc  
\*\*                          it\_ordim\_o = ct\_ordim\_o  
\*\*                        CHANGING  
\*\*                          ev\_who     = lv\_who.  
\*\*  
\*\*                      CLEAR ls\_who.  
\*\*                      IF lv\_who+0(1) NE 0.  
\*\*                        IF iv\_who NE lv\_who.  
\*\*                          "if current WO already contains WT for HU in front of HU to be picked => do nothing  
\*\*                          ls\_who-who = lv\_who.  
\*\*                          INSERT ls\_who INTO ct\_who INDEX 1.  
\*\*                          ls\_det2\_fg-lgnum = iv\_lgnum.  
\*\*                          ls\_det2\_fg-crane\_no = lv\_crane\_no.  
\*\*                          ls\_det2\_fg-who = lv\_who.  
\*\*                          ls\_det2\_fg-description = 'Rewarehousing task sent to subsystemfor picking'.  
\*\*                          ls\_det2\_fg-log\_date = sy-datum.  
\*\*                          ls\_det2\_fg-log\_time = sy-uzeit.  
\*\*                          APPEND ls\_det2\_fg TO lt\_det2\_fg.  
\*\*                          MODIFY zdet2\_fg\_log FROM TABLE lt\_det2\_fg.  
\*\*                          COMMIT WORK.  
\*\*  
\*\*  
\*\*                          UPDATE zinter\_leave SET active = abap\_true  
\*\*                                                  who\_no = ls\_who-who  
\*\*                                              WHERE lgnum EQ iv\_lgnum  
\*\*                                              AND resource\_name = iv\_queue.  
\*\*                          COMMIT WORK.  
\*\*  
\*\*                        ENDIF.  
\*\*                        cv\_rearr\_done = abap\_true.  
\*\*                        RETURN.  
\*\*                      ELSE.  
\*\*                        SELECT \*  INTO TABLE lt\_inter\_who FROM zinter\_leave  
\*\*                                  WHERE lgnum = iv\_lgnum AND resource\_name = iv\_queue .  
\*\*                        IF sy-subrc = 0.  
\*\*                          LOOP AT lt\_inter\_who INTO ls\_inter\_who.  
\*\*                            READ TABLE ct\_who INTO cs\_who WITH KEY who = ls\_inter\_who-who\_no.  
\*\*                            IF sy-subrc = 0.  
\*\*                              "do nothing  
\*\*                            ELSE.  
\*\*                              UPDATE zinter\_leave SET active = abap\_false  
\*\*                                                      who\_no = ls\_rtv\_who-who  
\*\*                                                  WHERE lgnum EQ iv\_lgnum  
\*\*                                                  AND resource\_name = iv\_queue.  
\*\*                              COMMIT WORK.  
\*\*                            ENDIF.  
\*\*                          ENDLOOP.  
\*\*                        ENDIF.  
\*\*                        RETURN.  
\*\*                      ENDIF.  
\*\*  
\*\*                      SELECT \*  INTO TABLE lt\_inter\_who FROM zinter\_leave  
\*\*                                WHERE lgnum = iv\_lgnum AND resource\_name = iv\_queue .  
\*\*                      IF sy-subrc = 0.  
\*\*                        LOOP AT lt\_inter\_who INTO ls\_inter\_who.  
\*\*                          READ TABLE ct\_who INTO cs\_who WITH KEY who = ls\_inter\_who-who\_no.  
\*\*                          IF sy-subrc = 0.  
\*\*                            "do nothing  
\*\*                          ELSE.  
\*\*                            UPDATE zinter\_leave SET active = abap\_false  
\*\*                                                    who\_no = ls\_rtv\_who-who  
\*\*                                                WHERE lgnum EQ iv\_lgnum  
\*\*                                                AND resource\_name = iv\_queue.  
\*\*                            COMMIT WORK.  
\*\*                          ENDIF.  
\*\*                        ENDLOOP.  
\*\*                      ENDIF.  
\*\*                      RETURN.  
\*\*                    ENDIF."sy-subrc = 0  of 2010 read table end  
\*\*  
\*\*                  ENDIF."3010 of read table end  
\*\*  
\*\*                ENDIF."3090 READ TABLE end  
\*\*  
\*\*  
\*\*              ENDIF."end of sy-subrc interleave  
\*\*  
\*\*            ENDIF."ls\_final\_error end  
\*\*  
\*\*  
\*\*          ENDLOOP.  
\*\*        ENDIF."lt\_final\_error  
\*\*      ENDIF.  
\*\*\*      lo\_log->save\_applog2db( ).  
\*\*    ENDIF.  
\*\*  
\*\*    IF lv\_missing\_case = 1 AND lt\_rtv\_who IS NOT INITIAL.  
\*\*  
\*\*      ls\_det2\_fg-lgnum = iv\_lgnum.  
\*\*      ls\_det2\_fg-crane\_no = lv\_crane\_no.  
\*\*      ls\_det2\_fg-description = 'Missing case '.  
\*\*      ls\_det2\_fg-log\_date = sy-datum.  
\*\*      ls\_det2\_fg-log\_time = sy-uzeit.  
\*\*      APPEND ls\_det2\_fg TO lt\_det2\_fg.  
\*\*      MODIFY zdet2\_fg\_log FROM TABLE lt\_det2\_fg.  
\*\*      COMMIT WORK.  
\*\*    ENDIF.  
\* END Above code moved to the mthod who\_det 7/6/2023 by Santha  
  
\*\*  
\*\*  
\*\*  
\*\*  
\*\*  
\*\*  
\*\*  
\*\*  
\*\*  
\*\*  
\*\*  
\*\*  
\*\*\*  
\*\*\*    DATA ls\_selcrit   TYPE /scwm/s\_to\_selcrit\_mon.  
\*\*\*    DATA ltr\_who      TYPE rseloption.  
\*\*\*    DATA lsr\_who      TYPE rsdsselopt.  
\*\*\*    DATA lt\_to        TYPE /scwm/tt\_to\_det\_mon.  
\*\*\*  
\*\*\*    lsr\_who-sign    = wmegc\_sign\_inclusive.  
\*\*\*    lsr\_who-option  = wmegc\_option\_eq.  
\*\*\*    lsr\_who-low     = iv\_current\_wo.  
\*\*\*    APPEND lsr\_who TO ltr\_who.  
\*\*\*  
\*\*\*    LOOP AT ct\_who ASSIGNING FIELD-SYMBOL(<fs\_who>).  
\*\*\*      lsr\_who-low = <fs\_who>-who.  
\*\*\*      APPEND lsr\_who TO ltr\_who.  
\*\*\*    ENDLOOP.  
\*\*\*  
\*\*\*    ls\_selcrit-r\_who = ltr\_who.  
\*\*\* Get all WT details for WHO in ASRS Queue  
\*\*\*    CALL FUNCTION '/SCWM/TO\_GET\_WIP'  
\*\*\*      EXPORTING  
\*\*\*        iv\_lgnum   = iv\_lgnum  
\*\*\*        iv\_open    = abap\_true  
\*\*\*        iv\_conf    = abap\_true  
\*\*\*        is\_selcrit = ls\_selcrit  
\*\*\*      IMPORTING  
\*\*\*        et\_to      = lt\_to.  
\*\*\*    IF sy-subrc = 0 AND lt\_to IS NOT INITIAL.  
\*\*\*  
\*\*\*      MOVE-CORRESPONDING lt\_to to ct\_ordim\_o.  
\*\*\*  
\*\*\*    ENDIF.  
\*\*\*  
\*\*\*  
  
  
  ENDMETHOD..

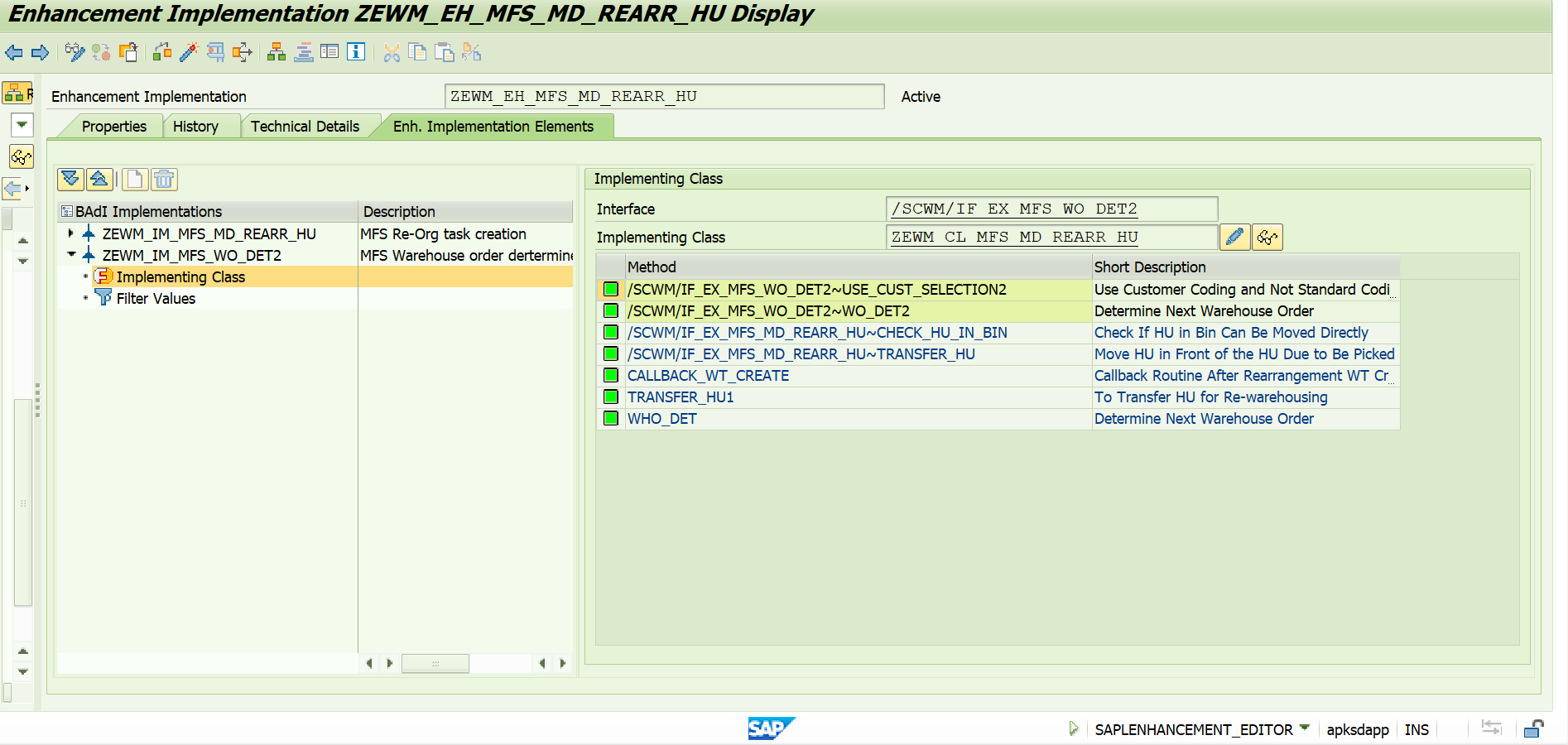
  METHOD /scwm/if\_ex\_mfs\_md\_rearr\_hu~check\_hu\_in\_bin.  
\* Check if the passed HU is in front of the storage bin and can be moved.  
\* Position of HU is stored in LOGPOS. Whereas position 1 is always at the  
\*   front of the storage bin.  
\* In standard only 1 HU can be rearranged at once.  
  
    DATA: lv\_num\_rearrange\_hu TYPE numc2,  
          lv\_msgtext          TYPE string.  
    DATA: ls\_lagp      TYPE /scwm/lagp,  
          ls\_selopt    TYPE rsdsselopt,  
          ls\_t\_huident TYPE /scwm/s\_huident.  
    DATA: lt\_huhdr TYPE /scwm/tt\_huhdr\_int,  
          lr\_lgpla TYPE rseloption,  
          lr\_vhi   TYPE rseloption.  
  
    FIELD-SYMBOLS: <huhdr> TYPE /scwm/s\_huhdr\_int.  
    FIELD-SYMBOLS: <lagp>  TYPE /scwm/lagp.  
  
    IF iv\_lgpla IS INITIAL. "No bin -> no result  
      MESSAGE w368(/scwm/mfs) INTO lv\_msgtext.  
      co\_log->add\_message( ip\_row = 0 ).  
      RETURN.  
    ENDIF.  
  
    IF co\_log IS NOT BOUND.  
      CREATE OBJECT co\_log.  
    ENDIF.  
  
    CALL FUNCTION '/SCWM/LAGP\_READ\_SINGLE'  
      EXPORTING  
        iv\_lgnum      = iv\_lgnum  
        iv\_lgpla      = iv\_lgpla  
        iv\_nobuf      = 'X'  
      IMPORTING  
        es\_lagp       = ls\_lagp  
      EXCEPTIONS  
        wrong\_input   = 1  
        not\_found     = 2  
        enqueue\_error = 3  
        OTHERS        = 4.  
  
    IF sy-subrc <> 0.  
      co\_log->add\_message( ip\_row = 0 ).  
      RETURN.  
    ENDIF.  
  
    IF ls\_lagp-anzle < 2 OR  
        ls\_lagp-maxle < 2.  
\*      There is only one HU in the bin (ANZLE = 1) or  
\*        the bin can contain only one HU (MAXLE = 1)  
      RETURN.  
    ENDIF.  
  
\*   Get all the HU in the bin.  
    ls\_selopt-sign   = wmegc\_sign\_inclusive.  
    ls\_selopt-option = wmegc\_option\_eq.  
    ls\_selopt-low    = iv\_lgpla.  
    APPEND ls\_selopt TO lr\_lgpla.  
\*   only consider real HUs  
    CLEAR: ls\_selopt.  
    ls\_selopt-sign   = wmegc\_sign\_inclusive.  
    ls\_selopt-option = wmegc\_option\_eq.  
    ls\_selopt-low    = wmegc\_vhi\_real.  
    APPEND ls\_selopt TO lr\_vhi.  
  
  
    CALL FUNCTION '/SCWM/HU\_SELECT\_GEN'  
      EXPORTING  
        iv\_lgnum     = iv\_lgnum  
        ir\_lgpla     = lr\_lgpla  
        ir\_vhi       = lr\_vhi  
      IMPORTING  
        et\_huhdr     = lt\_huhdr  
      EXCEPTIONS  
        wrong\_input  = 1  
        not\_possible = 2  
        error        = 3  
        OTHERS       = 4.  
  
    IF sy-subrc <> 0.  
      co\_log->add\_message( ip\_row = 0 ).  
      RETURN.  
    ENDIF.  
  
    IF lt\_huhdr IS NOT INITIAL.  
\*     Sort HU by LOGPOS. LOGPOS = 1 is the one in the front of the bin  
      SORT lt\_huhdr BY logpos.  
      LOOP AT lt\_huhdr ASSIGNING <huhdr>.  
        IF <huhdr>-huident = iv\_huident.  
          EXIT.  
        ENDIF.  
        "We are only interested in the top HU  
        IF <huhdr>-top IS INITIAL.  
          CONTINUE.  
        ENDIF.  
        ls\_t\_huident-lgnum = <huhdr>-lgnum.  
        ls\_t\_huident-huident = <huhdr>-huident.  
        ls\_t\_huident-vhi = <huhdr>-vhi.  
  
        APPEND ls\_t\_huident TO ct\_rearrange\_hu.  
        "HU &1 Logpos &2 need to be rearranged  
        MESSAGE i367(/scwm/mfs) WITH <huhdr>-huident <huhdr>-logpos iv\_huident iv\_logpos INTO lv\_msgtext.  
        co\_log->add\_message( ip\_row = 0 ).  
  
        ADD 1 TO lv\_num\_rearrange\_hu.  
  
        IF lv\_num\_rearrange\_hu = 1.  
\*         In standard we only rearrange 1 HU  
          EXIT.  
        ENDIF.  
      ENDLOOP.  
    ENDIF.  
  ENDMETHOD.

  METHOD /scwm/if\_ex\_mfs\_md\_rearr\_hu~transfer\_hu.  
  
\* Rearrange one HU. The WT is created in a separate task to work on proper memory.  
\* The result (created WO) is received in method CALLBACK\_WT\_CREATE and stored in GV\_WHO.  
\* If the new WO could not be created, the HU is moved to clarification bin of resource.  
\* If this also fails, the original WO can't be processed. This must be handled  
\*   in the calling programm.  
\* If a HU is moved several times within an aisle, an adjustment of the HU is necessary.  
\*   We check counter MFS\_STOCKTRANS\_CNT and compare it with maximum value of resource type.  
\*  
\* By the way, term "rearrangement" is replaced by term "stock transfer" during development.  
  
\* IV\_LGTYP is not used in default implementation.  
  
    DATA: lv\_msgtext      TYPE bapi\_msg,                    "#EC NEEDED  
          lv\_mfs\_err\_dest TYPE /scwm/de\_mfs\_err\_dest,  
          lv\_lines        TYPE i,  
          lv\_severity     TYPE bapi\_mtype,  
          lv\_lsd          TYPE /scwm/who-lsd,  
          lv\_orig\_lsd     TYPE /scwm/who-lsd,  
          lv\_do\_realign   TYPE xfeld.  
    DATA: ls\_create\_hu  TYPE /scwm/s\_to\_crea\_hu,  
          ls\_ordim\_o\_db TYPE /scwm/ordim\_o,  
          ls\_huhdr      TYPE /scwm/s\_huhdr\_int,  
          ls\_t346       TYPE /scwm/t346,  
          ls\_tmfsplc    TYPE /scwm/tmfsplc,  
          ls\_rsrc       TYPE /scwm/rsrc,  
          ls\_ordim\_o\_vb TYPE /scwm/ltap,  
          ls\_who\_int    TYPE /scwm/s\_who\_int,  
          ls\_attributes TYPE /scwm/s\_who\_att,  
          ls\_trsrc\_typ  TYPE /scwm/trsrc\_typ.  
  
    DATA: lt\_create\_hu  TYPE /scwm/tt\_to\_crea\_hu,  
          lt\_ordim\_o\_db TYPE /scwm/tt\_ordim\_o,  
          lt\_ordim\_o    TYPE /scwm/tt\_ordim\_o,  
          lt\_ordim\_o\_vb TYPE /scwm/tt\_ltap\_vb,  
          lt\_bapiret    TYPE bapirettab.  
  
    FIELD-SYMBOLS: <ls\_ordim2>  TYPE /scwm/ordim\_o,  
                   <ls\_huident> TYPE /scwm/s\_huident.  
  
    DESCRIBE TABLE it\_huident LINES lv\_lines.  
    IF lv\_lines > 1.  
      "Standard implementation can only rearrangement one HU  
      MESSAGE w370(/scwm/mfs) WITH '1' lv\_lines INTO lv\_msgtext.  
      co\_log->add\_message( ip\_row = 0 ).  
      RETURN.  
    ENDIF.  
    IF lv\_lines = 0.  
      "No HU for rearrangement transfered  
      MESSAGE w369(/scwm/mfs) INTO lv\_msgtext.  
      co\_log->add\_message( ip\_row = 0 ).  
      RETURN.  
    ENDIF.  
  
\*   Get warehouse process type for HU rearrangement  
    CALL FUNCTION '/SCWM/T346\_READ\_SINGLE'  
      EXPORTING  
        iv\_lgnum  = iv\_lgnum  
        iv\_queue  = iv\_queue  
      IMPORTING  
        es\_t346   = ls\_t346  
      EXCEPTIONS  
        not\_found = 1  
        OTHERS    = 2.  
    IF sy-subrc <> 0.  
      co\_log->add\_message( ip\_row = 0 ).  
      RETURN.  
    ENDIF.  
  
    CALL FUNCTION '/SCWM/TMFSPLC\_READ\_SINGLE'  
      EXPORTING  
        iv\_lgnum   = iv\_lgnum  
        iv\_plc     = ls\_t346-plc  
      IMPORTING  
        es\_tmfsplc = ls\_tmfsplc  
      EXCEPTIONS  
        error      = 1  
        OTHERS     = 2.  
    IF sy-subrc <> 0.  
      co\_log->add\_message( ip\_row = 0 ).  
      RETURN.  
    ENDIF.  
  
    IF ls\_tmfsplc-proctype\_stock\_trans IS INITIAL.  
      "No WPT for rearrangement maintained  
      MESSAGE e376(/scwm/mfs) WITH iv\_lgnum ls\_t346-plc INTO lv\_msgtext.  
      co\_log->add\_message( ip\_row = 0 ).  
      RETURN.  
    ENDIF.  
  
\*   Read resource type to get maximum number of stock transfer  
    CALL FUNCTION '/SCWM/RSRC\_TRSRC\_TYP\_READ'  
      EXPORTING  
        iv\_lgnum     = iv\_lgnum  
        iv\_rsrc\_type = is\_rsrc-rsrc\_type  
      IMPORTING  
        es\_trsrc\_typ = ls\_trsrc\_typ  
      EXCEPTIONS  
        not\_found    = 1  
        OTHERS       = 2.  
    IF sy-subrc <> 0.  
      co\_log->add\_message( ip\_row = 0 ).  
    ENDIF.  
  
    IF ls\_trsrc\_typ-mfs\_max\_stocktrans IS INITIAL.  
\*     No maximum number of stock transfer maintained -> Do rearrangement  
    ENDIF.  
  
\*   select resource and clearing bin  
    IF is\_rsrc-rsrc IS INITIAL.  
      CALL FUNCTION '/SCWM/MFS\_GET\_RSRC\_BY\_QUEUE'  
        EXPORTING  
          iv\_lgnum = iv\_lgnum  
          iv\_queue = iv\_queue  
        IMPORTING  
          es\_rsrc  = ls\_rsrc.  
      lv\_mfs\_err\_dest = ls\_rsrc-mfs\_err\_dest.  
    ELSE.  
      lv\_mfs\_err\_dest = is\_rsrc-mfs\_err\_dest.  
    ENDIF.  
  
    READ TABLE it\_huident ASSIGNING <ls\_huident> INDEX 1.  
\*   Read HU  
    CALL FUNCTION '/SCWM/HUHEADER\_READ'  
      EXPORTING  
        iv\_appl      = wmegc\_huappl\_wme  
        iv\_huident   = <ls\_huident>-huident  
        iv\_db\_select = 'X'  
        iv\_nobuff    = ' '  
        iv\_lock      = ' '  
        iv\_wait      = ' '  
      IMPORTING  
        es\_huheader  = ls\_huhdr  
      EXCEPTIONS  
        OTHERS       = 99.  
  
    IF sy-subrc <> 0.  
      "HU &1 does not exist anymore; data inconsistency  
      MESSAGE e360(/scwm/mfs) WITH <ls\_huident>-huident INTO lv\_msgtext.  
      co\_log->add\_message( ip\_row = 0 ).  
      RETURN.  
    ENDIF.  
  
    IF ls\_huhdr-flgmove IS NOT INITIAL.  
\*     There is an open WT for this HU - use this first.  
\*     We assume that this is an open picking task.  
      CALL FUNCTION '/SCWM/TO\_READ\_SRC'  
        EXPORTING  
          iv\_lgnum         = iv\_lgnum  
          iv\_huident       = ls\_huhdr-huident  
          iv\_nobuf         = 'X'  
          iv\_add\_to\_memory = ' '  
          iv\_no\_waiting    = ' '  
        IMPORTING  
          et\_ordim\_o       = lt\_ordim\_o\_db  
        EXCEPTIONS  
          OTHERS           = 0.  
  
      READ TABLE lt\_ordim\_o\_db INTO ls\_ordim\_o\_db  
                                WITH KEY tostat = wmegc\_to\_open.  
  
      cv\_who = ls\_ordim\_o\_db-who.              " use WHO  
  
      "Warehouse order &1 already exists for HU &2  
      MESSAGE i361(/scwm/mfs) WITH ls\_ordim\_o\_db-who <ls\_huident>-huident  
                          INTO lv\_msgtext.  
      co\_log->add\_message( ip\_row = 0 ).  
      RETURN.  
    ENDIF.  
  
    "HU &1 will be rearranged from storage bin &2  
    MESSAGE i362(/scwm/mfs) WITH <ls\_huident>-huident ls\_huhdr-lgpla INTO lv\_msgtext.  
    co\_log->add\_message( ip\_row = 0 ).  
  
    DATA: ls\_telegram  TYPE  /scwm/s\_mfs\_teletotal,  
          ls\_tmfscp    TYPE /scwm/tmfscp,  
          lo\_log       TYPE REF TO  /scwm/cl\_log,  
          lv\_rbd\_behav TYPE /scwm/de\_rbd\_behavior,  
          ls\_humove    TYPE /scwm/mfshumove,  
          lo\_badi      TYPE REF TO /scwm/ex\_mfs\_act\_case\_rbd,  
          lv\_channel   TYPE /scwm/de\_mfscch VALUE '1',  
          lv\_plc       TYPE /scwm/de\_mfsplc.  
  
    SELECT SINGLE \* INTO @DATA(ls\_param) FROM zcon\_param\_val  
                                         WHERE lgnum = @iv\_lgnum  
                                         AND zreq\_name = 'COMMUNICATION\_POINT'  
                                         AND zparam = @iv\_queue  
                                         AND zactive = @abap\_true.  
  
    IF sy-subrc = 0.  
      ls\_telegram-huident  = <ls\_huident>-huident.  
      ls\_telegram-sender   = 'MIIFGIN'.  
      ls\_telegram-receiver = 'EWMK'.  
      lv\_plc               = 'MIIFGIN'.  
      lv\_channel           = '1'.  
  
      ls\_telegram-cp = ls\_param-zvalue.  
  
      CALL FUNCTION '/SCWM/TMFSCP\_READ\_SINGLE'  
        EXPORTING  
          iv\_lgnum  = iv\_lgnum  
          iv\_plc    = lv\_plc  
          iv\_cp     = ls\_telegram-cp  
        IMPORTING  
          es\_tmfscp = ls\_tmfscp  
        EXCEPTIONS  
          OTHERS    = 0.  
  
\*    Run rough bin determination in BAdI  
      TRY .  
          GET BADI lo\_badi  
            FILTERS  
              lgnum = iv\_lgnum.  
        CATCH cx\_badi.                                  "#EC NO\_HANDLER  
      ENDTRY.  
  
      lv\_rbd\_behav = 'X'.  
      DATA lv\_case\_aisle\_det TYPE c.  
  
      "Capacity check should not be considered  
      EXPORT lv\_rbd\_behav FROM lv\_rbd\_behav TO MEMORY ID 'MFSACT\_RBD\_BEHAV'.   " EXPORT to ZSCWM\_CL\_CORE\_PUTSTRA->SORT\_AISLES  
  
      "HU Move should not be considered  
  
      lv\_case\_aisle\_det = 'X'.  
      EXPORT lv\_case\_aisle\_det FROM lv\_case\_aisle\_det TO MEMORY ID 'MFSACT\_CASE\_AISLE\_DET'.   " EXPORT to ZSCWM\_CL\_CORE\_PUTSTRA->ZDET\_BIN\_FOR\_PRODHU  
  
      IF lo\_badi IS BOUND.  
        CALL BADI lo\_badi->call\_rbd  
          EXPORTING  
            iv\_lgnum               = iv\_lgnum  
            iv\_plc                 = lv\_plc  
            iv\_channel             = lv\_channel  
            is\_telegram            = ls\_telegram  
            is\_tmfscp              = ls\_tmfscp  
            io\_log                 = co\_log  
          CHANGING  
            cv\_no\_bin\_det\_behavior = lv\_rbd\_behav       "Capacity check should not be considered  
            cs\_mfshumove           = ls\_humove.  
  
        FREE MEMORY ID 'MFSACT\_CASE\_AISLE\_DET'.  
        FREE MEMORY ID 'MFSACT\_RBD\_BEHAV'.  
  
      ENDIF.  
      ls\_create\_hu-nlpla     = ls\_humove-nlpla.  
  
    ENDIF.  
  
\*   Set up structure for WT-creation FM  
    ls\_create\_hu-huident   = <ls\_huident>-huident.  
    ls\_create\_hu-procty    = ls\_tmfsplc-proctype\_stock\_trans.  
    ls\_create\_hu-creto     = wmegc\_creto\_no\_mfs\_trig.  
  
  
    APPEND ls\_create\_hu TO lt\_create\_hu.  
  
    IF ls\_trsrc\_typ-mfs\_max\_stocktrans <= ls\_huhdr-mfs\_stocktrans\_cnt AND  
       ls\_trsrc\_typ-mfs\_max\_stocktrans IS NOT INITIAL.  
      lv\_do\_realign = abap\_true.  
      "Realignment for HU &1 necessary  
      MESSAGE i421(/scwm/mfs)  
        WITH <ls\_huident>-huident ls\_huhdr-mfs\_stocktrans\_cnt ls\_trsrc\_typ-mfs\_max\_stocktrans  
        INTO lv\_msgtext.  
      co\_log->add\_message( ip\_row = 0 ).  
    ELSE.  
      IF ls\_trsrc\_typ-mfs\_max\_stocktrans IS NOT INITIAL.  
        "No realignment for HU &1 necessary  
        MESSAGE i423(/scwm/mfs)  
          WITH <ls\_huident>-huident ls\_huhdr-mfs\_stocktrans\_cnt ls\_trsrc\_typ-mfs\_max\_stocktrans  
          INTO lv\_msgtext.  
        co\_log->add\_message( ip\_row = 0 ).  
      ENDIF.  
      lv\_do\_realign = abap\_false.  
    ENDIF.  
    IF lv\_mfs\_err\_dest IS INITIAL.  
\*     Without error bin, a rearrangement is not possible  
      CLEAR lv\_do\_realign.  
    ENDIF.  
  
    IF lv\_do\_realign IS INITIAL.  
\*      IF iv\_no\_new\_task IS INITIAL.  
\*        gv\_wt\_created = ' '.  
  
      "prevent trigger of /SCWM/MFS\_WT\_DET during WT creation  
\*        CALL FUNCTION '/SCWM/TO\_CREATE\_MOVE\_HU'  
\*          STARTING NEW TASK 'WT\_CREATE'  
\*          CALLING callback\_wt\_create ON END OF TASK  
\*          EXPORTING  
\*            iv\_lgnum       = iv\_lgnum  
\*            iv\_update\_task = ' '  
\*            iv\_commit\_work = 'X'  
\*            it\_create\_hu   = lt\_create\_hu.  
\*  
\*        WAIT UNTIL gv\_wt\_created = 'X'.  
  
\* WT creation was successful - delete HU\_MOVE entry  
  
\*          CALL FUNCTION '/SCWM/MFS\_HUMOVE\_UPD'  
\*            IN UPDATE TASK  
\*            EXPORTING  
\*              iv\_lgnum   = iv\_lgnum  
\*              iv\_huident = ls\_create\_hu-huident  
\*              iv\_action  = wmegc\_delete.  
\*  
\*  
\*          COMMIT WORK AND WAIT.  
\*          /scwm/cl\_tm=>cleanup( ).  
\*        ENDIF.  
\*      ELSE.  
      "prevent trigger of /SCWM/MFS\_WT\_DET during WT creation  
      CALL FUNCTION '/SCWM/TO\_CREATE\_MOVE\_HU'  
        EXPORTING  
          iv\_lgnum       = iv\_lgnum  
          iv\_update\_task = ' '  
          iv\_commit\_work = 'X'  
          it\_create\_hu   = lt\_create\_hu  
        IMPORTING  
          et\_ltap\_vb     = lt\_ordim\_o\_vb  
          et\_bapiret     = lt\_bapiret  
          ev\_severity    = lv\_severity.  
  
\*       Get warehouse order of created WT  
      READ TABLE lt\_ordim\_o\_vb INTO ls\_ordim\_o\_vb  
                           WITH KEY tostat = wmegc\_to\_open.  
  
      IF sy-subrc = 0.  
        gv\_who = ls\_ordim\_o\_vb-who.  
\*          gv\_wt\_created = 'X'.  
  
\* WT creation was successful - delete HU\_MOVE entry  
  
        CALL FUNCTION '/SCWM/MFS\_HUMOVE\_UPD'  
          IN UPDATE TASK  
          EXPORTING  
            iv\_lgnum   = iv\_lgnum  
            iv\_huident = ls\_ordim\_o\_vb-vlenr  
            iv\_action  = wmegc\_delete.  
  
  
        COMMIT WORK AND WAIT.  
        /scwm/cl\_tm=>cleanup( ).  
      ENDIF.  
  
\*      ENDIF.  
    ENDIF.  
  
  
    IF gv\_who IS INITIAL.  
      IF lv\_do\_realign IS INITIAL.  
        "Warehouse Order could not be created for HU &1  
        MESSAGE i363(/scwm/mfs) WITH <ls\_huident>-huident INTO lv\_msgtext.  
        co\_log->add\_message( ip\_row = 0 ).  
      ENDIF.  
  
      CLEAR: ls\_create\_hu, lt\_create\_hu.  
  
      IF lv\_mfs\_err\_dest IS INITIAL.  
        "No error destination for resource maintained  
        MESSAGE w377(/scwm/mfs) WITH ls\_rsrc-rsrc INTO lv\_msgtext.  
        co\_log->add\_message( ip\_row = 0 ).  
        RETURN.  
      ENDIF.  
  
      IF lv\_mfs\_err\_dest IS NOT INITIAL.  
        ls\_create\_hu-huident   = <ls\_huident>-huident.  
        ls\_create\_hu-procty    = ls\_tmfsplc-proctype\_stock\_trans.  
        ls\_create\_hu-creto     = wmegc\_creto\_no\_mfs\_trig.  
        ls\_create\_hu-nlpla     = lv\_mfs\_err\_dest.  
  
        APPEND ls\_create\_hu TO lt\_create\_hu.  
  
        "Move HU &1 to clarification bin &2  
        MESSAGE i364(/scwm/mfs) WITH <ls\_huident>-huident lv\_mfs\_err\_dest  
                            INTO lv\_msgtext.  
        co\_log->add\_message( ip\_row = 0 ).  
  
        IF iv\_no\_new\_task IS INITIAL.  
\*          gv\_wt\_created = ' '.  
\*  
\*          prevent trigger of /SCWM/MFS\_WT\_DET during WT creation  
\*          CALL FUNCTION '/SCWM/TO\_CREATE\_MOVE\_HU'  
\*            STARTING NEW TASK 'WT\_CREATE'  
\*            CALLING callback\_wt\_create ON END OF TASK  
\*            EXPORTING  
\*              iv\_lgnum       = iv\_lgnum  
\*              iv\_update\_task = ' '  
\*              iv\_commit\_work = 'X'  
\*              it\_create\_hu   = lt\_create\_hu.  
\*  
\*          WAIT UNTIL gv\_wt\_created = 'X'.  
\*        ELSE.  
          CLEAR: lt\_ordim\_o\_vb, lt\_bapiret, lv\_severity. "lt\_create\_hu.  
  
          "prevent trigger of /SCWM/MFS\_WT\_DET during WT creation  
          CALL FUNCTION '/SCWM/TO\_CREATE\_MOVE\_HU'  
            EXPORTING  
              iv\_lgnum       = iv\_lgnum  
              iv\_update\_task = ' '  
              iv\_commit\_work = 'X'  
              it\_create\_hu   = lt\_create\_hu  
            IMPORTING  
              et\_ltap\_vb     = lt\_ordim\_o\_vb  
              et\_bapiret     = lt\_bapiret  
              ev\_severity    = lv\_severity.  
  
\*         Get warehouse order of created WT  
          READ TABLE lt\_ordim\_o\_vb INTO ls\_ordim\_o\_vb  
                               WITH KEY tostat = wmegc\_to\_open.  
  
          IF sy-subrc = 0.  
            gv\_who = ls\_ordim\_o\_vb-who.  
          ENDIF.  
  
        ENDIF.  
  
        IF gv\_who IS INITIAL.  
          "WO for HU &1 to clarification bin &2 could not be created  
          MESSAGE i365(/scwm/mfs) WITH <ls\_huident>-huident ls\_rsrc-mfs\_err\_dest  
                             INTO lv\_msgtext.  
        ENDIF.  
      ENDIF.  
    ENDIF.  
  
    cv\_who = gv\_who.  
    IF cv\_who IS INITIAL.  
      RETURN.  
    ENDIF.  
  
    "Warehouse order &1 for rearrangement of HU &2 created  
    MESSAGE i366(/scwm/mfs) WITH cv\_who <ls\_huident>-huident INTO lv\_msgtext.  
    co\_log->add\_message( ip\_row = 0 ).  
  
  ENDMETHOD.

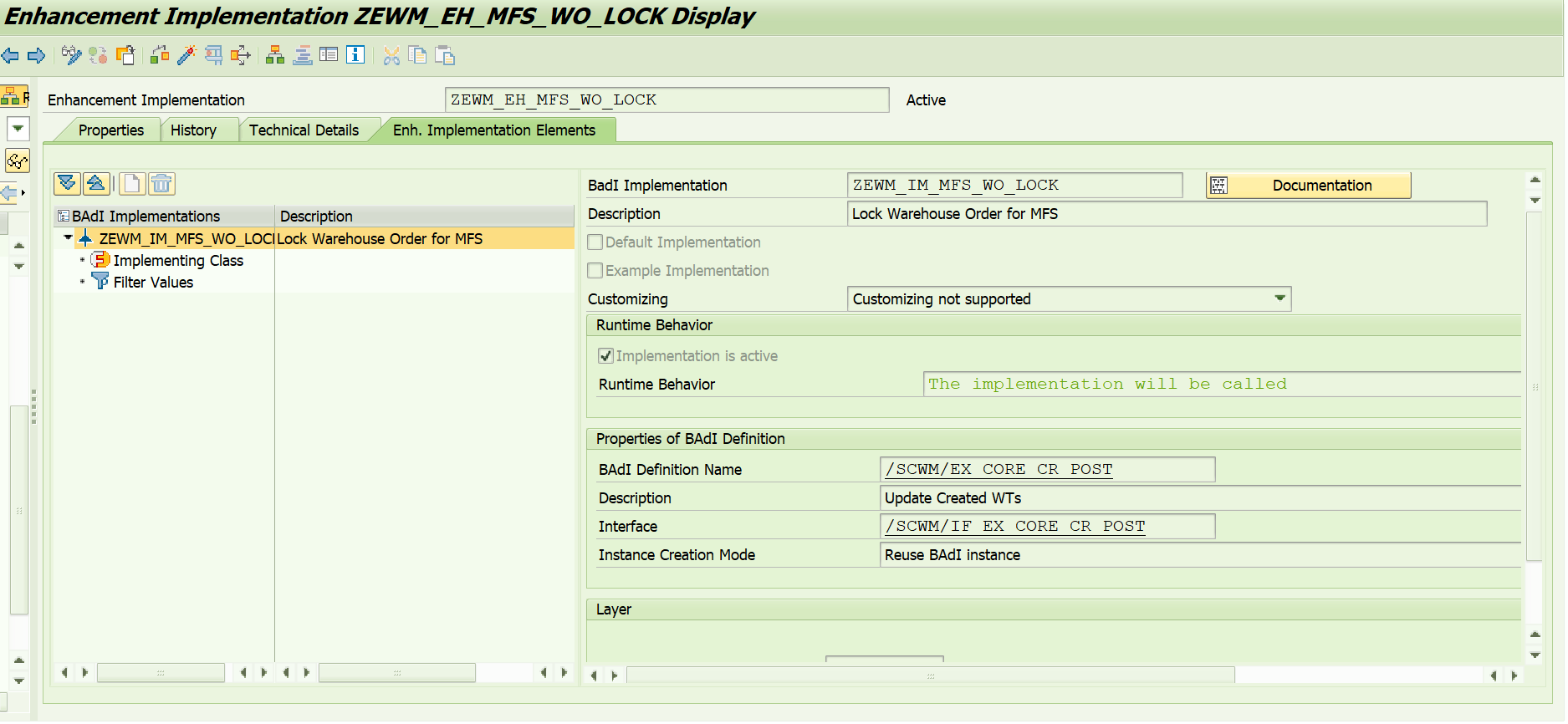
  METHOD callback\_wt\_create.  
  
    DATA: ls\_ordim\_o    TYPE /scwm/ltap.  
    DATA: lt\_ordim\_o    TYPE /scwm/tt\_ltap\_vb.  
  
    CLEAR gv\_who.  
  
    RECEIVE RESULTS FROM FUNCTION '/SCWM/TO\_CREA\_MOVE\_HU'  
       IMPORTING  
         et\_ltap\_vb     = lt\_ordim\_o.  
  
\*   Get warehouse order of created WT  
    READ TABLE lt\_ordim\_o INTO     ls\_ordim\_o  
                          WITH KEY tostat = wmegc\_to\_open.  
  
    IF sy-subrc = 0.  
      gv\_who = ls\_ordim\_o-who.  
  
    ENDIF.  
  
    gv\_wt\_created = 'X'.  
  
  
  ENDMETHOD.

  METHOD transfer\_hu1.  
    DATA: lt\_ordim\_o TYPE /scwm/tt\_ordim\_o.  
    DATA: lv\_msgtext TYPE string.  
    DATA: lv\_open\_task      TYPE i,  
          lv\_stn\_pallets    TYPE i,  
          lv\_total\_pallets  TYPE i,  
          lv\_rem\_pallets    TYPE i,  
          ls\_rsrc           TYPE /scwm/rsrc,  
          lv\_buffer\_pallets TYPE i.  
    DATA: lt\_stock\_bin TYPE /scwm/tt\_stock\_bin,  
          lt\_bapiret   TYPE bapiret2\_t.  
    DATA :gv\_msgtext      TYPE bapi\_msg,  
          lo\_log          TYPE REF TO /scwm/cl\_log,  
          lv\_tstamp\_start TYPE timestampl,  
          iv\_caller       TYPE /scwm/de\_seqnr,  
          iv\_check\_future TYPE xfeld,  
          lv\_who          TYPE /scwm/de\_who,  
          ls\_who          TYPE /scwm/who,  
          lv\_time         TYPE symsgv,  
          lv\_tstamp       TYPE timestampl,  
  
          ls\_who\_int      TYPE /scwm/s\_who\_int,  
          lv\_no\_new\_task  TYPE xfeld,  
          lt\_rearrange\_hu TYPE /scwm/tt\_huident,  
          ls\_rearrange\_hu LIKE LINE OF lt\_rearrange\_hu,  
          go\_log          TYPE REF TO /scwm/cl\_log.  
    DATA:cs\_who TYPE /scwm/who.  
    DATA: lv\_outer\_who TYPE /scwm/de\_who,  
          ls\_log       TYPE bal\_s\_log.  
\*    FIELD-SYMBOLS: ls\_ordim\_open     TYPE /scwm/ordim\_o.  
  
    DATA: lo\_badi\_rearr TYPE REF TO zewm\_cl\_mfs\_md\_rearr\_hu,  
          lv\_multideep  TYPE /scwm/de\_multideep.  
  
    DATA:lt\_132\_error TYPE TABLE OF z132\_error\_log,  
         ls\_132\_error LIKE LINE OF lt\_132\_error.  
    DATA:it\_ordim\_open   TYPE TABLE OF /scwm/ordim\_o,  
         ls\_ordim\_open   LIKE LINE OF it\_ordim\_open,  
         lt\_rewarehouse  TYPE TABLE OF zrewarehouse\_inf,  
         lv\_mfs\_err\_dest TYPE /scwm/de\_mfs\_err\_dest,  
         ls\_rewarehouse  LIKE LINE OF lt\_rewarehouse,  
         lt\_det2\_fg      TYPE TABLE OF zdet2\_fg\_log,  
         ls\_det2\_fg      LIKE LINE OF lt\_det2\_fg.  
    TYPES: BEGIN OF ty\_stype,  
             stype TYPE /scwm/lagp-lgtyp,  
           END OF ty\_stype.  
    DATA: ls\_prcty TYPE  rsdsselopt,  
          lt\_prcty TYPE rseloption.  
    CONSTANTS: wmegc\_apl\_object\_wme     TYPE balobj\_d  VALUE '/SCWM/WME'.  
  
  
\*    CALL FUNCTION '/SCWM/MFS\_GET\_RSRC\_BY\_QUEUE'  
\*      EXPORTING  
\*        iv\_lgnum = iv\_lgnum  
\*        iv\_queue = iv\_rsrc-queue  
\*      IMPORTING  
\*        es\_rsrc  = ls\_rsrc.  
\*    lv\_mfs\_err\_dest = ls\_rsrc-mfs\_err\_dest.  
  
    SELECT \* INTO TABLE it\_ordim\_open FROM /scwm/ordim\_o WHERE who = iv\_who.  
    IF sy-subrc = 0.  
      READ TABLE it\_ordim\_open INTO ls\_ordim\_open INDEX 1.  
    ENDIF.  
    lv\_multideep = /scwm/cl\_core\_multideep\_util=>get\_instance( )->multideep\_det(  
       EXPORTING  
         iv\_lgnum     = iv\_lgnum  
         iv\_lgtyp     = ls\_ordim\_open-vltyp ).  
  
  
    IF lv\_multideep = wmegc\_multideep\_logpos\_yes.  
\*         Do rearrangement only in multi-deep storage type  
      "Warehouse order & should be processed by MFS  
      MESSAGE i384(/scwm/mfs) WITH is\_whorder-who INTO gv\_msgtext.  
\*        go\_log->add\_message( ip\_row = 0 ).  
  
      GET TIME STAMP FIELD lv\_tstamp\_start. "Remember start time  
  
      CREATE OBJECT lo\_log.  
      MOVE: wmegc\_apl\_object\_wme TO ls\_log-object,  
zapl\_cl\_common\_constants=>gc\_det2 TO ls\_log-subobject.  
  
  
      CREATE OBJECT lo\_badi\_rearr.  
      SELECT SINGLE zvalue INTO @DATA(lv\_value) FROM zcon\_param\_val WHERE lgnum EQ @iv\_lgnum AND zreq\_name EQ 'CHECK\_HU\_REQ' AND zparam EQ 'CHECK\_HU\_P' AND zactive EQ @abap\_true.  
      IF sy-subrc = 0.  
        DATA:lv\_anzle        TYPE /scwm/lagp\_anzle.  
        SELECT SINGLE anzle INTO lv\_anzle FROM /scwm/lagp WHERE lgnum = iv\_lgnum  
                                                           AND   lgpla = ls\_ordim\_open-vlpla.  
        IF sy-subrc = 0 AND lv\_anzle = 2.  
          DATA: lt\_aqua TYPE TABLE OF /scwm/aqua,  
                ls\_aqua LIKE LINE OF lt\_aqua.  
          SELECT \* INTO TABLE lt\_aqua FROM /scwm/aqua WHERE lgnum EQ iv\_lgnum AND lgpla = ls\_ordim\_open-vlpla.  
          IF sy-subrc = 0.  
            SORT lt\_aqua BY logpos.  
            READ TABLE lt\_aqua INTO ls\_aqua WITH KEY logpos = '01' BINARY SEARCH.  
            IF sy-subrc = 0.  
              IF ls\_aqua-huident <> ls\_ordim\_open-vlenr.  
                ls\_det2\_fg-lgnum = iv\_lgnum.  
                ls\_det2\_fg-crane\_no = ls\_ordim\_open-vlpla+1(2).  
                ls\_det2\_fg-bin\_name = ls\_ordim\_open-vlpla.  
                ls\_det2\_fg-who = iv\_who.  
                ls\_det2\_fg-handling\_unit = ls\_aqua-huident.  
                ls\_det2\_fg-description = 'HU is present for Rewarehousing '.  
                ls\_det2\_fg-log\_date = sy-datum.  
                ls\_det2\_fg-log\_time = sy-uzeit.  
                APPEND ls\_det2\_fg TO lt\_det2\_fg.  
                MODIFY zdet2\_fg\_log FROM TABLE lt\_det2\_fg.  
                COMMIT WORK.  
  
                ls\_rewarehouse-lgnum = iv\_lgnum.  
                ls\_rewarehouse-handling\_unit = ls\_aqua-huident.  
                ls\_rewarehouse-bin\_name = ls\_ordim\_open-vlpla.  
                ls\_rewarehouse-description = 'Rewarhousing of HU Started'.  
                ls\_rewarehouse-log\_date = sy-datum.  
                ls\_rewarehouse-log\_time = sy-uzeit.  
                APPEND ls\_rewarehouse TO lt\_rewarehouse.  
                MODIFY zrewarehouse\_inf FROM TABLE lt\_rewarehouse.  
                COMMIT WORK.  
  
                SELECT \* INTO TABLE @DATA(lg\_prcty) FROM zcon\_param\_val WHERE lgnum = @iv\_lgnum AND zreq\_name = 'OUTER\_REW\_PROCTY\_REQ' AND zparam = 'OUTER\_REW\_PROCTY\_PARAM' AND zactive = @abap\_true.  
                IF sy-subrc EQ 0.  
                  LOOP AT lg\_prcty INTO DATA(lgs\_prcty).  
                    ls\_prcty-sign = 'I'.  
                    ls\_prcty-option = 'EQ'.  
                    ls\_prcty-low = lgs\_prcty-zvalue.  
                    APPEND ls\_prcty TO lt\_prcty.  
                  ENDLOOP.  
                ENDIF.  
  
                SELECT SINGLE who INTO lv\_outer\_who FROM /scwm/ordim\_o WHERE lgnum EQ iv\_lgnum AND  vlenr EQ ls\_aqua-huident  
                   AND procty IN lt\_prcty.  
                IF sy-subrc EQ 0.  
                  MOVE ls\_aqua-huident  TO ls\_log-extnumber.  
                  CALL METHOD lo\_log->create\_log  
                    EXPORTING  
                      is\_log = ls\_log.  
                  MESSAGE s023(zapl\_det) WITH   ls\_aqua-huident lv\_outer\_who  INTO lv\_msgtext.  
                  lo\_log->add\_message2log( ).  
\*          lo\_log->save\_applog2db( ).  
                  ev\_who = lv\_outer\_who.  
                  ls\_det2\_fg-lgnum = iv\_lgnum.  
                  ls\_det2\_fg-crane\_no = ls\_ordim\_open-vlpla+1(2).  
                  ls\_det2\_fg-bin\_name = ls\_ordim\_open-vlpla.  
                  ls\_det2\_fg-who = ev\_who.  
                  ls\_det2\_fg-handling\_unit = ls\_aqua-huident.  
                  ls\_det2\_fg-description = 'Task is present for outer HU '.  
                  ls\_det2\_fg-log\_date = sy-datum.  
                  ls\_det2\_fg-log\_time = sy-uzeit.  
                  APPEND ls\_det2\_fg TO lt\_det2\_fg.  
                  MODIFY zdet2\_fg\_log FROM TABLE lt\_det2\_fg.  
                  COMMIT WORK.  
                  RETURN.  
                ELSE.  
                  ls\_det2\_fg-lgnum = iv\_lgnum.  
                  ls\_det2\_fg-crane\_no = ls\_ordim\_open-vlpla+1(2).  
                  ls\_det2\_fg-bin\_name = ls\_ordim\_open-vlpla.  
                  ls\_det2\_fg-who = ev\_who.  
                  ls\_det2\_fg-handling\_unit = ls\_aqua-huident.  
                  ls\_det2\_fg-description = 'Task is not present for outer HU '.  
                  ls\_det2\_fg-log\_date = sy-datum.  
                  ls\_det2\_fg-log\_time = sy-uzeit.  
                  APPEND ls\_det2\_fg TO lt\_det2\_fg.  
                  MODIFY zdet2\_fg\_log FROM TABLE lt\_det2\_fg.  
                  COMMIT WORK.  
                  ls\_rearrange\_hu-lgnum = iv\_lgnum.  
                  ls\_rearrange\_hu-huident = ls\_aqua-huident.  
                  ls\_rearrange\_hu-vhi = ' '.  
                  APPEND ls\_rearrange\_hu TO lt\_rearrange\_hu.  
  
                  CALL METHOD lo\_badi\_rearr->/scwm/if\_ex\_mfs\_md\_rearr\_hu~transfer\_hu  
                    EXPORTING  
                      iv\_lgnum          = iv\_lgnum  
                      iv\_lgtyp          = ls\_ordim\_open-vltyp  
                      it\_huident        = lt\_rearrange\_hu  
                      iv\_queue          = is\_whorder-queue  
                      is\_rsrc           = iv\_rsrc  
                      iv\_orig\_wo        = iv\_who  
                      it\_orig\_wt        = it\_ordim\_o  
                      iv\_no\_new\_task    = lv\_no\_new\_task  
                      iv\_caller\_of\_badi = iv\_caller  
                    CHANGING  
                      co\_log            = lo\_log  
                      cv\_who            = lv\_who.  
  
                  ev\_who = lv\_who.  
                  ls\_det2\_fg-lgnum = iv\_lgnum.  
                  ls\_det2\_fg-crane\_no = ls\_ordim\_open-vlpla+1(2).  
                  ls\_det2\_fg-bin\_name = ls\_ordim\_open-vlpla.  
                  ls\_det2\_fg-who = ev\_who.  
                  ls\_det2\_fg-handling\_unit = ls\_aqua-huident.  
                  ls\_det2\_fg-description = 'Rewarehosing done for  outer HU '.  
                  ls\_det2\_fg-log\_date = sy-datum.  
                  ls\_det2\_fg-log\_time = sy-uzeit.  
                  APPEND ls\_det2\_fg TO lt\_det2\_fg.  
                  MODIFY zdet2\_fg\_log FROM TABLE lt\_det2\_fg.  
                  COMMIT WORK.  
                ENDIF.  
              ENDIF.  
            ENDIF.  
          ENDIF.  
        ELSE.  
          ls\_det2\_fg-lgnum = iv\_lgnum.  
          ls\_det2\_fg-crane\_no = ls\_ordim\_open-vlpla+1(2).  
          ls\_det2\_fg-bin\_name = ls\_ordim\_open-vlpla.  
          ls\_det2\_fg-who = iv\_who.  
          ls\_det2\_fg-description = 'No Hu in Bin for re-warehosuing'.  
          ls\_det2\_fg-log\_date = sy-datum.  
          ls\_det2\_fg-log\_time = sy-uzeit.  
          APPEND ls\_det2\_fg TO lt\_det2\_fg.  
          MODIFY zdet2\_fg\_log FROM TABLE lt\_det2\_fg.  
          COMMIT WORK.  
          MOVE ls\_ordim\_open-vlpla  TO ls\_log-extnumber.  
          CALL METHOD lo\_log->create\_log  
            EXPORTING  
              is\_log = ls\_log.  
          MESSAGE s021(zapl\_det) WITH   ls\_ordim\_open-vlpla INTO lv\_msgtext.  
          lo\_log->add\_message2log( ).  
          ls\_rewarehouse-lgnum = iv\_lgnum.  
          ls\_rewarehouse-bin\_name = ls\_ordim\_open-vlpla.  
          CONCATENATE 'NO HU for rewarehosuing in Bin ' ls\_ordim\_open-vlpla INTO ls\_rewarehouse-description .  
          ls\_rewarehouse-log\_date = sy-datum.  
          ls\_rewarehouse-log\_time = sy-uzeit.  
          APPEND ls\_rewarehouse TO lt\_rewarehouse.  
          MODIFY zrewarehouse\_inf FROM TABLE lt\_rewarehouse.  
          COMMIT WORK.  
        ENDIF.  
  
      ELSEIF sy-subrc <> 0.  
        IF lo\_badi\_rearr IS BOUND. "commented by vinayak  
          CALL METHOD lo\_badi\_rearr->/scwm/if\_ex\_mfs\_md\_rearr\_hu~check\_hu\_in\_bin  
            EXPORTING  
              iv\_lgnum          = iv\_lgnum  
              iv\_lgpla          = ls\_ordim\_open-vlpla  
              iv\_huident        = ls\_ordim\_open-vlenr  
              iv\_logpos         = ls\_ordim\_open-slogpos  
              iv\_orig\_wo        = iv\_who  
              it\_orig\_wt        = it\_ordim\_o  
              iv\_caller\_of\_badi = iv\_caller  
            CHANGING  
              co\_log            = lo\_log  
              ct\_rearrange\_hu   = lt\_rearrange\_hu.  
        ENDIF.  
        IF lt\_rearrange\_hu[] IS INITIAL.  
          ls\_rewarehouse-lgnum = iv\_lgnum.  
          ls\_rewarehouse-bin\_name = ls\_ordim\_open-vlpla.  
          ls\_rewarehouse-who = iv\_who.  
          CONCATENATE 'No hu for rewarehosuing in Bin ' ls\_ordim\_open-vlpla INTO ls\_rewarehouse-description .  
          ls\_rewarehouse-log\_date = sy-datum.  
          ls\_rewarehouse-log\_time = sy-uzeit.  
          APPEND ls\_rewarehouse TO lt\_rewarehouse.  
          MODIFY zrewarehouse\_inf FROM TABLE lt\_rewarehouse.  
          COMMIT WORK.  
        ENDIF.  
  
        IF lt\_rearrange\_hu[] IS NOT INITIAL.  
          READ TABLE lt\_rearrange\_hu INTO ls\_rearrange\_hu INDEX 1.  
          IF sy-subrc EQ 0.  
            ls\_rewarehouse-lgnum = iv\_lgnum.  
            ls\_rewarehouse-handling\_unit = ls\_rearrange\_hu-huident.  
            ls\_rewarehouse-bin\_name = ls\_ordim\_open-vlpla.  
            ls\_rewarehouse-who = iv\_who.  
            ls\_rewarehouse-description = 'HU is present for Rewarehousing'.  
            ls\_rewarehouse-log\_date = sy-datum.  
            ls\_rewarehouse-log\_time = sy-uzeit.  
            APPEND ls\_rewarehouse TO lt\_rewarehouse.  
            MODIFY zrewarehouse\_inf FROM TABLE lt\_rewarehouse.  
            COMMIT WORK.  
          ENDIF.  
          IF iv\_check\_future IS NOT INITIAL.  
            lv\_no\_new\_task = abap\_true.  
          ELSE.  
            lv\_no\_new\_task = abap\_false.  
          ENDIF.  
  
          SELECT \* INTO TABLE lg\_prcty FROM zcon\_param\_val WHERE lgnum = iv\_lgnum AND zreq\_name = 'OUTER\_REW\_PROCTY\_REQ' AND zparam = 'OUTER\_REW\_PROCTY\_PARAM' AND zactive = abap\_true.  
          IF sy-subrc EQ 0.  
            LOOP AT lg\_prcty INTO lgs\_prcty.  
              ls\_prcty-sign = 'I'.  
              ls\_prcty-option = 'EQ'.  
              ls\_prcty-low = lgs\_prcty-zvalue.  
              APPEND ls\_prcty TO lt\_prcty.  
            ENDLOOP.  
          ENDIF.  
  
          SELECT SINGLE who INTO lv\_outer\_who FROM /scwm/ordim\_o WHERE lgnum EQ iv\_lgnum AND  vlenr = ls\_rearrange\_hu-huident  
             AND nltyp EQ 'CPOT' AND flghuto EQ abap\_true.  
          IF sy-subrc EQ 0.  
            ev\_who = lv\_outer\_who.  
            ls\_rewarehouse-lgnum = iv\_lgnum.  
            ls\_rewarehouse-handling\_unit = ls\_rearrange\_hu-huident.  
            ls\_rewarehouse-bin\_name = ls\_ordim\_open-vlpla.  
            ls\_rewarehouse-who = ev\_who.  
            CONCATENATE 'Task is present for outer HU and warehouse order is' ev\_who INTO lv\_msgtext.  
            ls\_rewarehouse-description = lv\_msgtext.  
            ls\_rewarehouse-log\_date = sy-datum.  
            ls\_rewarehouse-log\_time = sy-uzeit.  
            APPEND ls\_rewarehouse TO lt\_rewarehouse.  
            MODIFY zrewarehouse\_inf FROM TABLE lt\_rewarehouse.  
            COMMIT WORK.  
            RETURN.  
  
          ELSE.  
  
            ls\_rewarehouse-lgnum = iv\_lgnum.  
            ls\_rewarehouse-handling\_unit = ls\_rearrange\_hu-huident.  
            ls\_rewarehouse-bin\_name = ls\_ordim\_open-vlpla.  
            ls\_rewarehouse-who = iv\_who.  
            ls\_rewarehouse-description = 'Task is not present for Outer HU'.  
            ls\_rewarehouse-log\_date = sy-datum.  
            ls\_rewarehouse-log\_time = sy-uzeit.  
            APPEND ls\_rewarehouse TO lt\_rewarehouse.  
            MODIFY zrewarehouse\_inf FROM TABLE lt\_rewarehouse.  
            COMMIT WORK.  
  
            CALL METHOD lo\_badi\_rearr->/scwm/if\_ex\_mfs\_md\_rearr\_hu~transfer\_hu  
              EXPORTING  
                iv\_lgnum          = iv\_lgnum  
                iv\_lgtyp          = ls\_ordim\_open-vltyp  
                it\_huident        = lt\_rearrange\_hu  
                iv\_queue          = is\_whorder-queue  
                is\_rsrc           = iv\_rsrc  
                iv\_orig\_wo        = iv\_who  
                it\_orig\_wt        = it\_ordim\_o  
                iv\_no\_new\_task    = lv\_no\_new\_task  
                iv\_caller\_of\_badi = iv\_caller  
              CHANGING  
                co\_log            = lo\_log  
                cv\_who            = lv\_who.  
  
            ev\_who = lv\_who.  
            ls\_rewarehouse-lgnum = iv\_lgnum.  
            ls\_rewarehouse-handling\_unit = ls\_rearrange\_hu-huident.  
            ls\_rewarehouse-bin\_name = ls\_ordim\_open-vlpla.  
            ls\_rewarehouse-who = ev\_who.  
            CONCATENATE 'Task is present for outer HU and warehouse order is' ev\_who INTO lv\_msgtext.  
            ls\_rewarehouse-description = lv\_msgtext.  
            ls\_rewarehouse-log\_date = sy-datum.  
            ls\_rewarehouse-log\_time = sy-uzeit.  
            APPEND ls\_rewarehouse TO lt\_rewarehouse.  
            MODIFY zrewarehouse\_inf FROM TABLE lt\_rewarehouse.  
            COMMIT WORK.  
  
          ENDIF.  
\*\*        lo\_log->save\_applog2db( ).  
\*          SET PARAMETER ID '/SCWM/LGN' FIELD iv\_lgnum.  
\*          WRITE lv\_tstamp\_start TO lv\_time USING EDIT MASK '==MFSTS'.  
\*          MESSAGE i381(/scwm/mfs) WITH lv\_time INTO gv\_msgtext. "Start of rearrangement at &1  
\*  
\*          IF lo\_log->mp\_protocol IS NOT INITIAL.  
\*  
\*          ENDIF.  
\*  
\*          GET TIME STAMP FIELD lv\_tstamp.  
\*  
\*          SET PARAMETER ID '/SCWM/LGN' FIELD iv\_lgnum.  
\*          WRITE lv\_tstamp TO lv\_time USING EDIT MASK '==MFSTS'.  
\*          MESSAGE i382(/scwm/mfs) WITH lv\_time INTO gv\_msgtext. "End of rearrangement at &1  
        ELSE.  
  
          IF lo\_log->mp\_protocol IS NOT INITIAL.  
          ENDIF.  
        ENDIF.  
      ENDIF.  
    ENDIF.  
  
  
  ENDMETHOD.

  METHOD WHO\_DET.  
  
    DATA: lt\_cancl  TYPE /scwm/tt\_cancl,  
          ls\_cancel LIKE LINE OF lt\_cancl.  
    DATA: lt\_create\_hu      TYPE /scwm/tt\_to\_crea\_hu,  
          ls\_create\_hu      TYPE /scwm/s\_to\_crea\_hu,  
          lt\_rewarehouse\_fg TYPE TABLE OF zrewareho\_fg\_log,  
          ls\_rewarehouse\_fg LIKE LINE OF lt\_rewarehouse\_fg,  
          lt\_wo\_info        TYPE TABLE OF /scwm/ordim\_o,  
          ls\_wo\_info        LIKE LINE OF lt\_wo\_info,  
          ls\_log            TYPE bal\_s\_log,  
  
          lv\_severity       TYPE bapi\_mtype.  
  
    CONSTANTS:  
              wmegc\_severity\_eax(3)   TYPE c          VALUE 'EAX'.  
\*  
    DATA: lt\_who                 TYPE TABLE OF /scwm/who,  
          lt\_dir\_procty          TYPE TABLE OF /scwm/who,  
          lt\_rtv\_procty          TYPE TABLE OF /scwm/who,  
          lt\_repl\_procty         TYPE TABLE OF /scwm/who,  
          lt\_str\_procty          TYPE TABLE OF /scwm/who,  
          lt\_str\_who             TYPE /scwm/tt\_who,  
          ls\_str\_who             LIKE LINE OF lt\_str\_who,  
          lt\_rtv\_who             TYPE   /scwm/tt\_who,  
          lt\_rtv\_who1            TYPE   /scwm/tt\_who,  
          lt\_repl\_who            TYPE  /scwm/tt\_who,  
          ls\_repl\_who            LIKE LINE OF lt\_repl\_who,  
          ls\_rtv\_who             LIKE LINE OF lt\_rtv\_who,  
          lt\_dir\_who             TYPE /scwm/tt\_who,  
          lt\_param\_agc\_schd      TYPE zapl\_tt\_param,  
          lt\_param\_mfs\_tele\_send TYPE zapl\_tt\_param,  
          lt\_crane\_check         TYPE TABLE OF zcon\_param\_val,  
          ls\_crane\_check         LIKE LINE OF lt\_crane\_check.  
    DATA: cs\_who LIKE LINE OF ct\_who.  
    DATA : lv\_count\_who TYPE i.  
  
    DATA : ls\_param      TYPE zapl\_param,  
           ls\_hdr\_procty TYPE /scwm/who.  
\*           ls\_who        TYPE /scwm/s\_who\_int.  
  
    DATA : lv\_id1          TYPE zapl\_de\_param\_id1,  
           lv\_id2          TYPE zapl\_de\_param\_id2,  
           lv\_msgtext      TYPE bapi\_msg,  
           lv\_count        TYPE i,  
           lv\_direct       TYPE i,  
           lv\_storage      TYPE i,  
           lv\_retrieve     TYPE i,  
           lv\_max\_direct   TYPE i,  
           lv\_max\_storage  TYPE i,  
           lv\_max\_retrieve TYPE i,  
           iv\_wo\_count     TYPE i.  
    DATA:lt\_error TYPE TABLE OF zcrane\_dashboard,  
         ls\_error LIKE LINE OF lt\_error.  
  
    FIELD-SYMBOLS : <ls\_who> TYPE /scwm/who.  
    DATA: lv\_rewarehouse TYPE zcon\_param\_val-zvalue.  
    DATA: lt\_ordim\_o TYPE /scwm/tt\_ordim\_o.  
    DATA: lv\_open\_task      TYPE i,  
          lv\_stn\_pallets    TYPE i,  
          lv\_total\_pallets  TYPE i,  
          lv\_rem\_pallets    TYPE i,  
          lv\_buffer\_pallets TYPE i.  
    DATA: lt\_stock\_bin TYPE /scwm/tt\_stock\_bin,  
          lt\_bapiret   TYPE bapiret2\_t.  
    DATA: lt\_132\_error TYPE TABLE OF z132\_error\_log,  
          ls\_132\_error LIKE LINE OF lt\_132\_error.  
    DATA:iv\_who TYPE /scwm/ordim\_o-who.  
    DATA :gv\_msgtext      TYPE bapi\_msg,  
          lo\_log          TYPE REF TO /scwm/cl\_log,  
          lv\_tstamp\_start TYPE timestampl,  
          iv\_caller       TYPE /scwm/de\_seqnr,  
          iv\_check\_future TYPE xfeld,  
          lv\_who          TYPE /scwm/de\_who,  
          ls\_who          TYPE /scwm/who,  
          lv\_time         TYPE symsgv,  
          lv\_tstamp       TYPE timestampl,  
  
          ls\_who\_int      TYPE /scwm/s\_who\_int,  
          lv\_no\_new\_task  TYPE xfeld,  
          lt\_rearrange\_hu TYPE /scwm/tt\_huident,  
          go\_log          TYPE REF TO /scwm/cl\_log.  
    CONSTANTS:lv\_load\_in\_rack TYPE i VALUE '132'.  
\*    FIELD-SYMBOLS: ls\_ordim\_open     TYPE /scwm/ordim\_o.  
  
    DATA: lo\_badi\_rearr TYPE REF TO zewm\_cl\_mfs\_md\_rearr\_hu,  
          lt\_inter\_who  TYPE TABLE OF zinter\_leave,  
          ls\_inter\_who  LIKE LINE OF lt\_inter\_who,  
          lv\_multideep  TYPE /scwm/de\_multideep,  
          ls\_prcty      TYPE  rsdsselopt,  
          lt\_prcty      TYPE rseloption.  
    DATA:lt\_interleave TYPE TABLE OF zinter\_leave,  
         ls\_interleave LIKE LINE OF lt\_interleave.  
  
    DATA: lt\_wo\_list  TYPE TABLE OF /scwm/s\_wo\_det\_mon\_out,  
          ls\_wo\_list  TYPE /scwm/s\_wo\_det\_mon\_out,  
          cs\_who\_list LIKE LINE OF ct\_who,  
          lt\_who\_to   TYPE  /scwm/tt\_to\_det\_mon\_out,  
          ls\_who\_to   LIKE LINE OF  lt\_who\_to.  
    TYPES:BEGIN OF ty\_hu\_tanum,  
            vlenr TYPE /scwm/ordim\_o-vlenr,  
            tanum TYPE /scwm/ordim\_o-tanum,  
          END OF ty\_hu\_tanum.  
    DATA:lt\_hu\_tanum        TYPE TABLE OF ty\_hu\_tanum,  
         ls\_hu\_tanum        LIKE LINE OF lt\_hu\_tanum,  
         lt\_det2\_fg         TYPE TABLE OF zdet2\_fg\_log,  
         ls\_det2\_fg         LIKE LINE OF lt\_det2\_fg,  
         lt\_warehouse\_order TYPE TABLE OF zwarehouse\_order,  
         ls\_warehouse\_order LIKE LINE OF lt\_warehouse\_order.  
    DATA : lv\_missing\_case TYPE i.  
    DATA : lv\_active1 TYPE zcon\_param\_val-zactive.  
  
    lv\_missing\_case = 1.  
    CONSTANTS: lc\_ptwy TYPE /scwm/ordim\_o-procty VALUE '3090'.  
  
    DATA:lo\_badi\_transfer TYPE REF TO zewm\_cl\_mfs\_md\_rearr\_hu.  
    CONSTANTS: wmegc\_apl\_object\_wme     TYPE balobj\_d  VALUE '/SCWM/WME'.  
    CREATE OBJECT lo\_badi\_transfer.  
  
    DATA(lv\_queue) = iv\_queue+0(2).  
    DATA(lv\_crane\_no) = iv\_queue+6(2).  
  
    MOVE-CORRESPONDING ct\_who TO lt\_warehouse\_order.  
    MODIFY zwarehouse\_order FROM TABLE lt\_warehouse\_order.  
    COMMIT WORK.  
  
    SELECT \* INTO TABLE lt\_crane\_check FROM zcon\_param\_val  
                         WHERE lgnum =  iv\_lgnum  
                         AND zreq\_name = 'CRANE\_STATUS'.  
    IF sy-subrc EQ 0.  
  
    ENDIF.  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  START OF  LOAD IN rack 132  
  
    READ TABLE lt\_crane\_check INTO ls\_crane\_check WITH KEY zvalue = lv\_crane\_no.  
    IF sy-subrc EQ 0.  
      IF ls\_crane\_check-zactive EQ abap\_true.  
        REFRESH:ct\_who.  
        RETURN.  
      ENDIF.  
    ENDIF.  
  
  
    SELECT SINGLE zvalue INTO @DATA(lv\_wo\_value)  
                              FROM zcon\_param\_val  
                              WHERE lgnum = @iv\_lgnum  
                              AND zreq\_name = 'WO\_COUNT\_REQ'  
                              AND zparam = 'WO\_COUNT\_PARAM'  
                              AND zactive = @abap\_true.  
    IF sy-subrc EQ 0.  
      iv\_wo\_count = lv\_wo\_value.  
    ENDIF.  
  
  
    MOVE: wmegc\_apl\_object\_wme TO ls\_log-object,  
    zapl\_cl\_common\_constants=>gc\_det2 TO ls\_log-subobject.  
    CREATE OBJECT lo\_log.  
  
  
    IF lv\_queue EQ 'SW'.  
      IF ct\_who[] IS NOT INITIAL.  
  
        SELECT SINGLE zvalue INTO @DATA(lv\_det2) FROM zcon\_param\_val  
                             WHERE lgnum = @iv\_lgnum  
                             AND zreq\_name = 'DET2\_NEW\_132'  
                             AND zparam = 'DET2\_NEW\_132\_P'  
                             AND zactive EQ @abap\_true.  
        IF sy-subrc = 0.  
          lv\_id1 = zapl\_cl\_common\_constants=>gc\_agc\_scheduler.  
          CALL METHOD zapl\_cl\_common\_constants=>get\_param\_value  
            EXPORTING  
              iv\_lgnum = iv\_lgnum  
              iv\_id1   = lv\_id1  
            IMPORTING  
              et\_param = lt\_param\_agc\_schd.  
          LOOP AT lt\_param\_agc\_schd INTO ls\_param WHERE id2 = zapl\_cl\_common\_constants=>gc\_storage\_interleave.  
            ls\_hdr\_procty-hdr\_procty = ls\_param-value.  
            APPEND ls\_hdr\_procty TO lt\_str\_procty.  
          ENDLOOP.  
  
          SELECT \* FROM /scwm/who INTO TABLE lt\_str\_who UP TO iv\_wo\_count ROWS  
          FOR ALL ENTRIES IN lt\_str\_procty  
            WHERE lgnum = iv\_lgnum  
              AND hdr\_procty = lt\_str\_procty-hdr\_procty  
              AND status = wmegc\_wo\_open  
              AND queue = iv\_queue.  
          IF sy-subrc EQ 0.  
            READ TABLE lt\_str\_who INTO ls\_str\_who WITH KEY hdr\_procty = '5010'. " DE task  
  
            IF sy-subrc EQ 0 .  
              INSERT ls\_str\_who INTO ct\_who INDEX 1.  
              ls\_det2\_fg-lgnum = iv\_lgnum.  
              ls\_det2\_fg-crane\_no = lv\_crane\_no.  
              ls\_det2\_fg-who = ls\_str\_who-who.  
              ls\_det2\_fg-description = '5010 process type task sent to subsystem '.  
              ls\_det2\_fg-log\_date = sy-datum.  
              ls\_det2\_fg-log\_time = sy-uzeit.  
              APPEND ls\_det2\_fg TO lt\_det2\_fg.  
              APPEND ls\_det2\_fg TO lt\_det2\_fg.  
              MODIFY zdet2\_fg\_log FROM TABLE lt\_det2\_fg.  
              RETURN.  
              lv\_missing\_case = 2.  
            ELSE.  
              READ TABLE lt\_str\_who INTO ls\_str\_who WITH KEY hdr\_procty = '5020'. " ADHOC task  
              IF sy-subrc EQ 0.  
                INSERT ls\_str\_who INTO ct\_who INDEX 1.  
                ls\_det2\_fg-lgnum = iv\_lgnum.  
                ls\_det2\_fg-crane\_no = lv\_crane\_no.  
                ls\_det2\_fg-who = ls\_str\_who-who.  
                ls\_det2\_fg-description = '5020 process type task sent to subsystem '.  
                ls\_det2\_fg-log\_date = sy-datum.  
                ls\_det2\_fg-log\_time = sy-uzeit.  
                APPEND ls\_det2\_fg TO lt\_det2\_fg.  
                APPEND ls\_det2\_fg TO lt\_det2\_fg.  
                MODIFY zdet2\_fg\_log FROM TABLE lt\_det2\_fg.  
                RETURN.  
                lv\_missing\_case = 2.  
              ENDIF.  
            ENDIF.  
          ENDIF.  
  
  
          SELECT SINGLE zvalue INTO lv\_rewarehouse FROM zcon\_param\_val  
                               WHERE lgnum EQ iv\_lgnum  
                               AND zreq\_name EQ 'REWAREHOUSE\_NEW\_ENABLE'  
                               AND zparam EQ 'REWAREHOUSE\_NEW\_PARAM'  
                               AND zactive EQ abap\_true.  
          IF sy-subrc = 0.  
            lv\_id1 = zapl\_cl\_common\_constants=>gc\_agc\_scheduler.  
            CALL METHOD zapl\_cl\_common\_constants=>get\_param\_value  
              EXPORTING  
                iv\_lgnum = iv\_lgnum  
                iv\_id1   = lv\_id1  
              IMPORTING  
                et\_param = lt\_param\_agc\_schd.  
            LOOP AT lt\_param\_agc\_schd INTO ls\_param WHERE id2 = zapl\_cl\_common\_constants=>gc\_retrieval\_interleave.  
              ls\_hdr\_procty-hdr\_procty = ls\_param-value.  
              APPEND ls\_hdr\_procty TO lt\_rtv\_procty.  
              CLEAR ls\_hdr\_procty.  
            ENDLOOP.  
  
            SELECT \* FROM /scwm/who INTO TABLE lt\_rtv\_who UP TO iv\_wo\_count ROWS  
            FOR ALL ENTRIES IN lt\_rtv\_procty  
            WHERE lgnum = iv\_lgnum  
              AND hdr\_procty = lt\_rtv\_procty-hdr\_procty  
              AND status = wmegc\_wo\_open  
              AND queue = iv\_queue.  
  
            IF sy-subrc NE 0.  
  
            ENDIF.  
  
            LOOP AT lt\_param\_agc\_schd INTO ls\_param WHERE id2 = zapl\_cl\_common\_constants=>gc\_storage\_interleave.  
              ls\_hdr\_procty-hdr\_procty = ls\_param-value.  
              APPEND ls\_hdr\_procty TO lt\_str\_procty.  
              CLEAR ls\_hdr\_procty.  
            ENDLOOP.  
  
            SELECT \* FROM /scwm/who INTO TABLE lt\_str\_who UP TO iv\_wo\_count ROWS  
            FOR ALL ENTRIES IN lt\_str\_procty  
            WHERE lgnum = iv\_lgnum  
              AND hdr\_procty = lt\_str\_procty-hdr\_procty  
              AND status = wmegc\_wo\_open  
              AND queue = iv\_queue.  
            IF sy-subrc NE 0.  
  
            ENDIF.  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* When we have only retrival tasks  
            IF lt\_str\_who[] IS INITIAL AND lt\_rtv\_who[] IS NOT INITIAL.  
              READ TABLE lt\_rtv\_who INTO ls\_rtv\_who WITH KEY hdr\_procty = '3010'.  
              IF sy-subrc = 0.  
                INSERT ls\_rtv\_who INTO ct\_who INDEX 1.  
                READ TABLE ct\_who INTO cs\_who INDEX 1.  
                iv\_who = cs\_who-who.  
                ls\_det2\_fg-lgnum = iv\_lgnum.  
                ls\_det2\_fg-crane\_no = lv\_crane\_no.  
                ls\_det2\_fg-who = iv\_who.  
                ls\_det2\_fg-description = 'checking for picking task need rewarehousing '.  
                ls\_det2\_fg-log\_date = sy-datum.  
                ls\_det2\_fg-log\_time = sy-uzeit.  
                APPEND ls\_det2\_fg TO lt\_det2\_fg.  
                MODIFY zdet2\_fg\_log FROM TABLE lt\_det2\_fg.  
                COMMIT WORK.  
                lv\_missing\_case = 2.  
                CALL METHOD lo\_badi\_transfer->transfer\_hu1  
                  EXPORTING  
                    iv\_lgnum   = iv\_lgnum  
                    is\_whorder = is\_who  
                    iv\_who     = iv\_who  
                    iv\_rsrc    = is\_rsrc  
                    it\_ordim\_o = ct\_ordim\_o  
                  CHANGING  
                    ev\_who     = lv\_who.  
  
                CLEAR ls\_who.  
                IF lv\_who+0(1) NE 0.  
                  IF iv\_who NE lv\_who.  
                    "if current WO already contains WT for HU in front of HU to be picked => do nothing  
                    ls\_who-who = lv\_who.  
                    INSERT ls\_who INTO ct\_who INDEX 1.  
                    ls\_det2\_fg-lgnum = iv\_lgnum.  
                    ls\_det2\_fg-crane\_no = lv\_crane\_no.  
                    ls\_det2\_fg-who = lv\_who.  
                    ls\_det2\_fg-description = 'Rewarehousing task sent to subsystem Picking task'.  
                    ls\_det2\_fg-log\_date = sy-datum.  
                    ls\_det2\_fg-log\_time = sy-uzeit.  
                    APPEND ls\_det2\_fg TO lt\_det2\_fg.  
                    MODIFY zdet2\_fg\_log FROM TABLE lt\_det2\_fg.  
                    COMMIT WORK.  
  
                    UPDATE zinter\_leave SET active = abap\_true  
                                            who\_no = ls\_who-who  
                                        WHERE lgnum EQ iv\_lgnum AND resource\_name = iv\_queue.  
                    COMMIT WORK.  
                  ENDIF.  
                  cv\_rearr\_done = abap\_true.  
                  RETURN.  
                ELSE.  
                  SELECT \* INTO TABLE lt\_inter\_who FROM zinter\_leave  
                                                    WHERE lgnum = iv\_lgnum  
                                                    AND resource\_name = iv\_queue .  
                  IF sy-subrc = 0.  
                    LOOP AT lt\_inter\_who INTO ls\_inter\_who.  
                      READ TABLE ct\_who INTO cs\_who WITH KEY who = ls\_inter\_who-who\_no.  
                      IF sy-subrc = 0.  
                        "do nothing  
                      ELSE.  
                        UPDATE zinter\_leave SET active = abap\_false  
                                                who\_no = ls\_rtv\_who-who  
                                            WHERE lgnum EQ iv\_lgnum  
                                            AND resource\_name = iv\_queue.  
                        COMMIT WORK.  
                      ENDIF.  
                    ENDLOOP.  
                  ENDIF.  
                  RETURN.  
                ENDIF.  
  
              ENDIF."sy-subrc = 0 end  
  
              READ TABLE lt\_rtv\_who INTO ls\_rtv\_who WITH KEY hdr\_procty = '2010'.  
              IF sy-subrc = 0.  
                INSERT ls\_rtv\_who INTO ct\_who INDEX 1.  
                READ TABLE ct\_who INTO cs\_who INDEX 1.  
                iv\_who = cs\_who-who.  
                ls\_det2\_fg-lgnum = iv\_lgnum.  
                ls\_det2\_fg-crane\_no = lv\_crane\_no.  
                ls\_det2\_fg-who = iv\_who.  
                ls\_det2\_fg-description = 'checking for picking task need rewarehousing '.  
                ls\_det2\_fg-log\_date = sy-datum.  
                ls\_det2\_fg-log\_time = sy-uzeit.  
                APPEND ls\_det2\_fg TO lt\_det2\_fg.  
                MODIFY zdet2\_fg\_log FROM TABLE lt\_det2\_fg.  
  
                COMMIT WORK.  
                lv\_missing\_case = 2.  
                CALL METHOD lo\_badi\_transfer->transfer\_hu1  
                  EXPORTING  
                    iv\_lgnum   = iv\_lgnum  
                    is\_whorder = is\_who  
                    iv\_who     = iv\_who  
                    iv\_rsrc    = is\_rsrc  
                    it\_ordim\_o = ct\_ordim\_o  
                  CHANGING  
                    ev\_who     = lv\_who.  
  
                CLEAR ls\_who.  
                IF lv\_who+0(1) NE 0.  
                  IF iv\_who NE lv\_who.  
                    "if current WO already contains WT for HU in front of HU to be picked => do nothing  
                    ls\_who-who = lv\_who.  
                    INSERT ls\_who INTO ct\_who INDEX 1.  
                    ls\_det2\_fg-lgnum = iv\_lgnum.  
                    ls\_det2\_fg-crane\_no = lv\_crane\_no.  
                    ls\_det2\_fg-who = lv\_who.  
                    ls\_det2\_fg-description = 'Rewarehousing task sent to subsystem Picking task'.  
                    ls\_det2\_fg-log\_date = sy-datum.  
                    ls\_det2\_fg-log\_time = sy-uzeit.  
                    APPEND ls\_det2\_fg TO lt\_det2\_fg.  
                    MODIFY zdet2\_fg\_log FROM TABLE lt\_det2\_fg.  
                    COMMIT WORK.  
  
                    UPDATE zinter\_leave SET active = abap\_true  
                                            who\_no = ls\_who-who  
                                       WHERE lgnum EQ iv\_lgnum  
                                       AND resource\_name = iv\_queue.  
                    COMMIT WORK.  
                  ENDIF.  
                  cv\_rearr\_done = abap\_true.  
                  RETURN.  
                ELSE.  
                  SELECT \*  INTO TABLE lt\_inter\_who FROM zinter\_leave  
                                                    WHERE lgnum = iv\_lgnum  
                                                    AND resource\_name = iv\_queue .  
                  IF sy-subrc = 0.  
                    LOOP AT lt\_inter\_who INTO ls\_inter\_who.  
                      READ TABLE ct\_who INTO cs\_who WITH KEY who = ls\_inter\_who-who\_no.  
                      IF sy-subrc = 0.  
                        "do nothing  
                      ELSE.  
                        UPDATE zinter\_leave SET active = abap\_false  
                                                who\_no = ls\_rtv\_who-who  
                                            WHERE lgnum EQ iv\_lgnum  
                                            AND resource\_name = iv\_queue.  
                        COMMIT WORK.  
                      ENDIF.  
                    ENDLOOP.  
                  ENDIF.  
                  RETURN.  
                ENDIF.  
              ENDIF."sy-subrc = 0 end  
\*              ENDIF.  
            ENDIF." lt\_str\_who is initial and lt\_str\_who is not intial  
\*\*\*\*\*End of changes when we have only retrival tasks  
  
            READ TABLE lt\_rtv\_who INTO ls\_rtv\_who WITH KEY hdr\_procty = '3010'.  
            IF sy-subrc = 0.  
              SELECT \* INTO TABLE lt\_interleave FROM zinter\_leave  
                                                WHERE lgnum = iv\_lgnum  
                                                AND resource\_name = iv\_queue  
                                                AND active EQ abap\_true.  
              IF sy-subrc = 0.  
                READ TABLE lt\_rtv\_who INTO ls\_rtv\_who WITH KEY hdr\_procty = '3010'.  
                IF sy-subrc = 0.  
                  ls\_rewarehouse\_fg-lgnum = iv\_lgnum.  
                  ls\_rewarehouse\_fg-warehouse\_order = ls\_rtv\_who-who.  
                  SELECT \* INTO TABLE lt\_wo\_info FROM /scwm/ordim\_o  
                                                 WHERE lgnum = iv\_lgnum  
                                                 AND who = ls\_rtv\_who-who.  
                  IF sy-subrc = 0.  
                    READ TABLE lt\_wo\_info INTO ls\_wo\_info INDEX 1.  
                    ls\_rewarehouse\_fg-bin\_name = ls\_wo\_info-vlpla.  
                    ls\_rewarehouse\_fg-handling\_unit = ls\_wo\_info-vlenr.  
                    ls\_rewarehouse\_fg-crane\_number = ls\_wo\_info-vlpla+1(2).  
                    ls\_rewarehouse\_fg-log\_date = sy-datum.  
                    ls\_rewarehouse\_fg-log\_time = sy-uzeit.  
                    APPEND ls\_rewarehouse\_fg TO lt\_rewarehouse\_fg.  
                    MODIFY zrewareho\_fg\_log FROM TABLE lt\_rewarehouse\_fg.  
                    CLEAR:lt\_rewarehouse\_fg, lt\_wo\_info.  
                  ENDIF.  
                  INSERT ls\_rtv\_who INTO ct\_who INDEX 1.  
                  READ TABLE ct\_who INTO cs\_who INDEX 1.  
                  iv\_who = cs\_who-who.  
                  ls\_det2\_fg-lgnum = iv\_lgnum.  
                  ls\_det2\_fg-crane\_no = lv\_crane\_no.  
\*                    ls\_det2\_fg-error\_no = ls\_final\_error-error.  
                  ls\_det2\_fg-who = iv\_who.  
                  ls\_det2\_fg-description = 'checking for picking task need rewarehousing repl '.  
                  ls\_det2\_fg-log\_date = sy-datum.  
                  ls\_det2\_fg-log\_time = sy-uzeit.  
                  APPEND ls\_det2\_fg TO lt\_det2\_fg.  
                  MODIFY zdet2\_fg\_log FROM TABLE lt\_det2\_fg.  
                  COMMIT WORK.  
                  lv\_missing\_case = 2.  
                  CALL METHOD lo\_badi\_transfer->transfer\_hu1  
                    EXPORTING  
                      iv\_lgnum   = iv\_lgnum  
                      is\_whorder = is\_who  
                      iv\_who     = iv\_who  
                      iv\_rsrc    = is\_rsrc  
                      it\_ordim\_o = ct\_ordim\_o  
                    CHANGING  
                      ev\_who     = lv\_who.  
  
                  CLEAR ls\_who.  
                  IF lv\_who+0(1) NE 0.  
                    IF iv\_who NE lv\_who.  
                      "if current WO already contains WT for HU in front of HU to be picked => do nothing  
                      ls\_who-who = lv\_who.  
                      INSERT ls\_who INTO ct\_who INDEX 1.  
                      ls\_det2\_fg-lgnum = iv\_lgnum.  
                      ls\_det2\_fg-crane\_no = lv\_crane\_no.  
                      ls\_det2\_fg-tanum = ls\_str\_who-who.  
                      ls\_det2\_fg-description = 'Rewarehousing task sent to subsystem Picking task repl'.  
                      ls\_det2\_fg-log\_date = sy-datum.  
                      ls\_det2\_fg-log\_time = sy-uzeit.  
                      APPEND ls\_det2\_fg TO lt\_det2\_fg.  
                      MODIFY zdet2\_fg\_log FROM TABLE lt\_det2\_fg.  
                      COMMIT WORK.  
  
                      UPDATE zinter\_leave SET active = abap\_true  
                                              who\_no = ls\_who-who  
                                          WHERE lgnum EQ iv\_lgnum  
                                          AND resource\_name = iv\_queue.  
                      COMMIT WORK.  
                    ENDIF.  
                    cv\_rearr\_done = abap\_true.  
                    RETURN.  
                  ELSE.  
                    SELECT \*  INTO TABLE lt\_inter\_who FROM zinter\_leave  
                                                      WHERE lgnum = iv\_lgnum  
                                                      AND resource\_name = iv\_queue .  
                    IF sy-subrc = 0.  
                      LOOP AT lt\_inter\_who INTO ls\_inter\_who.  
                        READ TABLE ct\_who INTO cs\_who WITH KEY who = ls\_inter\_who-who\_no.  
                        IF sy-subrc = 0.  
                          "do nothing  
                        ELSE.  
                          UPDATE zinter\_leave SET active = abap\_false  
                                                  who\_no = ls\_rtv\_who-who  
                                              WHERE lgnum EQ iv\_lgnum  
                                              AND resource\_name = iv\_queue.  
                          COMMIT WORK.  
                        ENDIF.  
                      ENDLOOP.  
                    ENDIF.  
                    RETURN.  
                  ENDIF.  
  
                ENDIF."sy-subrc = 0 end  
  
              ENDIF.  
            ENDIF.  
  
            READ TABLE lt\_rtv\_who INTO ls\_rtv\_who WITH KEY hdr\_procty = '2010'.  
            IF sy-subrc = 0.  
              SELECT \* INTO TABLE lt\_interleave FROM zinter\_leave  
                                                WHERE lgnum = iv\_lgnum  
                                                AND resource\_name = iv\_queue  
                                                AND active EQ abap\_true.  
              IF sy-subrc = 0.  
                READ TABLE lt\_rtv\_who INTO ls\_rtv\_who WITH KEY hdr\_procty = '2010'.  
                IF sy-subrc = 0.  
                  INSERT ls\_rtv\_who INTO ct\_who INDEX 1.  
                  READ TABLE ct\_who INTO cs\_who INDEX 1.  
                  iv\_who = cs\_who-who.  
                  ls\_det2\_fg-lgnum = iv\_lgnum.  
                  ls\_det2\_fg-crane\_no = lv\_crane\_no.  
\*                    ls\_det2\_fg-error\_no = ls\_final\_error-error.  
                  ls\_det2\_fg-who = iv\_who.  
                  ls\_det2\_fg-description = 'checking for picking task need rewarehousing '.  
                  ls\_det2\_fg-log\_date = sy-datum.  
                  ls\_det2\_fg-log\_time = sy-uzeit.  
                  APPEND ls\_det2\_fg TO lt\_det2\_fg.  
                  MODIFY zdet2\_fg\_log FROM TABLE lt\_det2\_fg.  
                  COMMIT WORK.  
                  lv\_missing\_case = 2.  
                  CALL METHOD lo\_badi\_transfer->transfer\_hu1  
                    EXPORTING  
                      iv\_lgnum   = iv\_lgnum  
                      is\_whorder = is\_who  
                      iv\_who     = iv\_who  
                      iv\_rsrc    = is\_rsrc  
                      it\_ordim\_o = ct\_ordim\_o  
                    CHANGING  
                      ev\_who     = lv\_who.  
  
                  CLEAR ls\_who.  
                  IF lv\_who+0(1) NE 0.  
                    IF iv\_who NE lv\_who.  
                      "if current WO already contains WT for HU in front of HU to be picked => do nothing  
                      ls\_who-who = lv\_who.  
                      INSERT ls\_who INTO ct\_who INDEX 1.  
                      ls\_det2\_fg-lgnum = iv\_lgnum.  
                      ls\_det2\_fg-crane\_no = lv\_crane\_no.  
                      ls\_det2\_fg-who = lv\_who.  
                      ls\_det2\_fg-description = 'Rewarehousing task sent to subsystem Picking task'.  
                      ls\_det2\_fg-log\_date = sy-datum.  
                      ls\_det2\_fg-log\_time = sy-uzeit.  
                      APPEND ls\_det2\_fg TO lt\_det2\_fg.  
                      MODIFY zdet2\_fg\_log FROM TABLE lt\_det2\_fg.  
                      COMMIT WORK.  
  
                      UPDATE zinter\_leave SET active = abap\_true  
                                              who\_no = ls\_who-who  
                                          WHERE lgnum EQ iv\_lgnum  
                                          AND resource\_name = iv\_queue.  
                      COMMIT WORK.  
                    ENDIF.  
                    cv\_rearr\_done = abap\_true.  
                    RETURN.  
                  ELSE.  
                    SELECT \*  INTO TABLE lt\_inter\_who FROM zinter\_leave  
                                                      WHERE lgnum = iv\_lgnum  
                                                      AND resource\_name = iv\_queue .  
                    IF sy-subrc = 0.  
                      LOOP AT lt\_inter\_who INTO ls\_inter\_who.  
                        READ TABLE ct\_who INTO cs\_who WITH KEY who = ls\_inter\_who-who\_no.  
                        IF sy-subrc = 0.  
                          "do nothing  
                        ELSE.  
                          UPDATE zinter\_leave SET active = abap\_false  
                                                  who\_no = ls\_rtv\_who-who  
                                              WHERE lgnum EQ iv\_lgnum  
                                              AND resource\_name = iv\_queue.  
                          COMMIT WORK.  
                        ENDIF.  
                      ENDLOOP.  
                    ENDIF.  
                    RETURN.  
                  ENDIF.  
                ENDIF."sy-subrc = 0 end  
  
              ENDIF.  
            ENDIF.  
  
\*            ENDIF.  
          ENDIF.  
        ENDIF."sy-subrc  EQ 0  
  
        SELECT \* INTO TABLE @DATA(lt\_dprcty) FROM zcon\_param\_val  
                            WHERE lgnum = @iv\_lgnum  
                            AND zreq\_name = 'DET2\_PRCTY'  
                            AND zparam = 'DET2\_PRCTY\_PARAM'  
                            AND zactive = @abap\_true.  
        IF sy-subrc EQ 0.  
          LOOP AT lt\_dprcty INTO DATA(ls\_dprcty).  
            ls\_prcty-sign = 'I'.  
            ls\_prcty-option = 'EQ'.  
            ls\_prcty-low = ls\_dprcty-zvalue.  
            APPEND ls\_prcty TO lt\_prcty.  
          ENDLOOP.  
        ENDIF.  
        SELECT vlenr tanum INTO TABLE lt\_hu\_tanum FROM /scwm/ordim\_o  
                                                  FOR ALL ENTRIES IN ct\_who  
                                                  WHERE lgnum = iv\_lgnum  
                                                  AND queue = ct\_who-queue  
                                                  AND  who = ct\_who-who  
                                                  AND procty IN lt\_prcty.  
        IF sy-subrc EQ 0.  
          CLEAR:ls\_who.  
        ENDIF.  
      ENDIF."ct\_who[] is not initial  
  
      IF lt\_hu\_tanum[] IS NOT INITIAL.  
        LOOP AT lt\_hu\_tanum ASSIGNING FIELD-SYMBOL(<fs\_hu\_tanum>).  
          DATA(lv\_hu) = <fs\_hu\_tanum>-vlenr.  
          SHIFT lv\_hu LEFT DELETING LEADING '0'.    "to delete leading zero from handling unit  
          <fs\_hu\_tanum>-vlenr = lv\_hu.  
        ENDLOOP.  
        SELECT SINGLE zvalue INTO @DATA(lv\_seconds)  
                             FROM zcon\_param\_val  
                             WHERE lgnum = @iv\_lgnum  
                             AND zreq\_name = 'TASK\_DELAY\_REQ'  
                             AND zparam = 'TASK\_DELAY\_PARAM'  
                             AND zactive = @abap\_true.  
        IF sy-subrc EQ 0.  
          SELECT \* INTO TABLE lt\_error  FROM zcrane\_dashboard  
                        FOR ALL ENTRIES IN lt\_hu\_tanum  
                        WHERE lgnum = iv\_lgnum  
                        AND crane\_no  = lv\_crane\_no  
                        AND huident = lt\_hu\_tanum-vlenr  
                        AND error = lv\_load\_in\_rack .  
          IF sy-subrc EQ 0.  
            WAIT UP TO lv\_seconds SECONDS.  
          ENDIF.  
        ENDIF.  
      ENDIF.  
  
      LOOP AT lt\_hu\_tanum INTO ls\_hu\_tanum.  
        READ TABLE lt\_error INTO ls\_error WITH KEY huident = ls\_hu\_tanum-vlenr.  
        IF sy-subrc EQ 0.  
          ls\_det2\_fg-lgnum = iv\_lgnum.  
          ls\_det2\_fg-crane\_no = lv\_crane\_no.  
          ls\_det2\_fg-handling\_unit = ls\_hu\_tanum-vlenr.  
          ls\_det2\_fg-error\_no = ls\_error-error.  
          ls\_det2\_fg-tanum = ls\_hu\_tanum-tanum.  
          ls\_det2\_fg-description = '132 LOAD RACK OCCURED'.  
          ls\_det2\_fg-log\_date = sy-datum.  
          ls\_det2\_fg-log\_time = sy-uzeit.  
          APPEND ls\_det2\_fg TO lt\_det2\_fg.  
          MODIFY zdet2\_fg\_log FROM TABLE lt\_det2\_fg.  
          COMMIT WORK.  
  
          MOVE-CORRESPONDING ls\_who\_to TO ls\_who.  
          INSERT ls\_who INTO ct\_who INDEX 1.  
  
          CLEAR: ls\_132\_error,lv\_msgtext,lt\_132\_error.  
          ls\_det2\_fg-lgnum = iv\_lgnum.  
          ls\_det2\_fg-crane\_no = lv\_crane\_no.  
          ls\_det2\_fg-handling\_unit = ls\_hu\_tanum-vlenr.  
          ls\_det2\_fg-error\_no = ls\_error-error.  
          ls\_det2\_fg-tanum = ls\_who-who.  
          ls\_det2\_fg-description = 'Det2 132 task sent to subsystem'.  
          ls\_det2\_fg-log\_date = sy-datum.  
          ls\_det2\_fg-log\_time = sy-uzeit.  
          APPEND ls\_det2\_fg TO lt\_det2\_fg.  
          MODIFY zdet2\_fg\_log FROM TABLE lt\_det2\_fg.  
          COMMIT WORK.  
          CONCATENATE ls\_who-who 'DET2 132 trigger task sent' INTO lv\_msgtext.  
          ls\_132\_error-lgnum = iv\_lgnum.  
          ls\_132\_error-crane\_no = ls\_who-queue+7(2).  
          ls\_132\_error-log\_date = sy-datum.  
          ls\_132\_error-log\_time = sy-uzeit.  
          ls\_132\_error-description = lv\_msgtext.  
          APPEND ls\_132\_error TO lt\_132\_error.  
          MODIFY z132\_error\_log FROM TABLE lt\_132\_error.  
          COMMIT WORK.  
\*          lo\_log->save\_applog2db( ).  
          RETURN.  
        ENDIF.  
      ENDLOOP.  
  
    ENDIF.  
  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*END OF LOAD IN RACK 132  
  
  
    IF lv\_queue EQ 'SW'. "ALLOWED INTERLEAVING FOR Swislog only  
      IF lt\_hu\_tanum[] IS NOT INITIAL.  
        SELECT \* INTO TABLE @DATA(lt\_final\_error) FROM zcrane\_dashboard  
                            FOR ALL ENTRIES IN @lt\_hu\_tanum  
                            WHERE lgnum = @iv\_lgnum  
                            AND crane\_no = @lv\_crane\_no  
                            AND huident = @lt\_hu\_tanum-vlenr.  
        IF lt\_final\_error[] IS INITIAL.  
          LOOP AT lt\_final\_error INTO DATA(ls\_final\_error).  
            READ TABLE lt\_hu\_tanum INTO ls\_hu\_tanum WITH KEY vlenr = ls\_final\_error-huident.  
            IF sy-subrc NE 0.  
              MOVE ls\_hu\_tanum-vlenr  TO ls\_log-extnumber.  
\*              CALL METHOD lo\_log->create\_log  
\*                EXPORTING  
\*                  is\_log = ls\_log.  
\*              MESSAGE s032(zapl\_det) WITH ls\_final\_error-huident  INTO lv\_msgtext.  
\*              lo\_log->add\_message2log( ).  
              ls\_det2\_fg-lgnum = iv\_lgnum.  
              ls\_det2\_fg-crane\_no = lv\_crane\_no.  
              ls\_det2\_fg-handling\_unit = ls\_final\_error-huident.  
              ls\_det2\_fg-error\_no = ls\_final\_error-error.  
              ls\_det2\_fg-tanum = ls\_final\_error-open\_task.  
              ls\_det2\_fg-description = 'HU has no error'.  
              APPEND ls\_det2\_fg TO lt\_det2\_fg.  
              MODIFY zdet2\_fg\_log FROM TABLE lt\_det2\_fg.  
              COMMIT WORK.  
            ENDIF.  
          ENDLOOP.  
  
          SELECT SINGLE zvalue INTO @DATA(ls\_zvalue)  
                      FROM zcon\_param\_val  
                      WHERE lgnum = @iv\_lgnum  
                      AND zreq\_name = 'INTERLEAVING'  
                      AND zparam = 'INTERLEAVE\_PARAM'  
                      AND zactive = @abap\_true.  
          IF sy-subrc EQ 0.  
\*    REFRESH: ct\_who[].  
\*\*-- MFS Whse process type constants  
            lv\_id1 = zapl\_cl\_common\_constants=>gc\_mfs\_tele\_send.  
            CALL METHOD zapl\_cl\_common\_constants=>get\_param\_value  
              EXPORTING  
                iv\_lgnum = iv\_lgnum  
                iv\_id1   = lv\_id1  
              IMPORTING  
                et\_param = lt\_param\_mfs\_tele\_send.  
  
\*-- Scheduler constants  
            lv\_id1 = zapl\_cl\_common\_constants=>gc\_agc\_scheduler.  
            CALL METHOD zapl\_cl\_common\_constants=>get\_param\_value  
              EXPORTING  
                iv\_lgnum = iv\_lgnum  
                iv\_id1   = lv\_id1  
              IMPORTING  
                et\_param = lt\_param\_agc\_schd.  
  
            CLEAR ls\_param.  
            READ TABLE lt\_param\_agc\_schd INTO ls\_param WITH KEY id2 = zapl\_cl\_common\_constants=>gc\_max\_direct\_command.  
            IF sy-subrc = 0.  
              MOVE ls\_param-value TO lv\_max\_direct .  
            ENDIF.  
  
            CLEAR ls\_param.  
            READ TABLE lt\_param\_agc\_schd INTO ls\_param WITH KEY id2 = zapl\_cl\_common\_constants=>gc\_max\_storage\_command.  
            IF sy-subrc = 0.  
              MOVE ls\_param-value TO lv\_max\_storage .  
            ENDIF.  
  
            CLEAR ls\_param.  
            READ TABLE lt\_param\_agc\_schd INTO ls\_param WITH KEY id2 = zapl\_cl\_common\_constants=>gc\_max\_retrieve\_command.  
            IF sy-subrc = 0.  
              MOVE ls\_param-value TO lv\_max\_retrieve .  
            ENDIF.  
  
\*-- Read Whse procty for Storage, retrieval and Direct transfer operations  
            CLEAR : ls\_param.  
            READ TABLE lt\_param\_agc\_schd INTO ls\_param WITH KEY id2 = zapl\_cl\_common\_constants=>gc\_direct\_transfer.  
            IF sy-subrc = 0.  
              ls\_hdr\_procty-hdr\_procty = ls\_param-value .  
              APPEND ls\_hdr\_procty TO lt\_dir\_procty.  
            ENDIF.  
  
            CLEAR : ls\_param.  
            READ TABLE lt\_param\_agc\_schd INTO ls\_param WITH KEY id2 = zapl\_cl\_common\_constants=>gc\_retrieval\_interleave.  
            IF sy-subrc = 0.  
              ls\_hdr\_procty-hdr\_procty = ls\_param-value.  
              APPEND ls\_hdr\_procty TO lt\_rtv\_procty.  
            ENDIF.  
            READ TABLE lt\_param\_agc\_schd INTO ls\_param WITH KEY id2 = zapl\_cl\_common\_constants=>gc\_repl\_interleave.  
            IF sy-subrc = 0.  
              ls\_hdr\_procty-hdr\_procty = ls\_param-value.  
              APPEND ls\_hdr\_procty TO lt\_repl\_procty.  
            ENDIF.  
  
            CLEAR : ls\_param.  
            LOOP AT lt\_param\_agc\_schd INTO ls\_param WHERE id2 = zapl\_cl\_common\_constants=>gc\_storage\_interleave.  
              ls\_hdr\_procty-hdr\_procty = ls\_param-value.  
              APPEND ls\_hdr\_procty TO lt\_str\_procty.  
            ENDLOOP.  
  
\*    CLEAR :  lt\_who[] , ct\_who[], lv\_count, lv\_retrieve, lv\_storage, lv\_direct.  
  
\*-- get all 'in process' WO, to check current WO count for each operation.Based on current count  
\*   and maximum allowed count, next set of WOs shall be decided and sent  
            SELECT \* FROM /scwm/who INTO TABLE lt\_who  
                      WHERE lgnum = iv\_lgnum  
                        AND status = wmegc\_wo\_open.  
  
            IF NOT lt\_who[] IS INITIAL.  
  
              LOOP AT lt\_who ASSIGNING <ls\_who>.  
  
  
\*-- Retrieval  
                READ TABLE lt\_rtv\_procty TRANSPORTING NO FIELDS WITH KEY hdr\_procty = <ls\_who>-hdr\_procty.  
                IF sy-subrc = 0.  
                  lv\_retrieve = lv\_retrieve + 1.  
                ENDIF.  
\*\*\*                Replineshment  
                READ TABLE lt\_repl\_procty TRANSPORTING NO FIELDS WITH KEY hdr\_procty = <ls\_who>-hdr\_procty.  
                IF sy-subrc = 0.  
                  lv\_retrieve = lv\_retrieve + 1.  
                ENDIF.  
  
\*-- Storage  
                READ TABLE lt\_str\_procty TRANSPORTING NO FIELDS WITH KEY hdr\_procty = <ls\_who>-hdr\_procty.  
                IF sy-subrc = 0.  
                  lv\_storage = lv\_storage + 1.  
                ENDIF.  
              ENDLOOP.  
            ENDIF.  
  
            IF lv\_max\_retrieve = lv\_retrieve.  
              CLEAR lv\_msgtext.  
              MESSAGE w432(zapl\_asrs) WITH  lv\_retrieve 'Retrieve' INTO lv\_msgtext.  
              lo\_log->add\_message( ).  
            ENDIF.  
  
            IF lv\_max\_storage = lv\_storage.  
              CLEAR lv\_msgtext.  
              MESSAGE w432(zapl\_asrs) WITH  lv\_storage 'Storage' INTO lv\_msgtext.  
              lo\_log->add\_message( ).  
            ENDIF.  
  
  
  
\*-- Validate maximum WO limit and select WO  acordingly  
  
            lv\_retrieve = lv\_max\_retrieve - lv\_retrieve.  
            lv\_storage = lv\_max\_storage - lv\_storage.  
            lv\_direct = lv\_max\_direct - lv\_direct.  
  
\*            IF lv\_storage > 0 .  
  
            SELECT \* FROM /scwm/who INTO TABLE lt\_str\_who UP TO iv\_wo\_count ROWS  
            FOR ALL ENTRIES IN lt\_str\_procty  
              WHERE lgnum = iv\_lgnum  
                AND hdr\_procty = lt\_str\_procty-hdr\_procty  
                AND status = wmegc\_wo\_open  
                AND queue = iv\_queue.  
  
\*            ENDIF.  
  
\*            IF lv\_retrieve > 0.  
            SELECT \* FROM /scwm/who INTO  TABLE  lt\_rtv\_who UP TO iv\_wo\_count ROWS  
              FOR ALL ENTRIES IN lt\_rtv\_procty  
                WHERE lgnum = iv\_lgnum  
                  AND hdr\_procty = lt\_rtv\_procty-hdr\_procty  
                  AND status = wmegc\_wo\_open  
                  AND queue = iv\_queue.  
\*\*\*\*\*\*\*\*                REplenishment  
            SELECT \* FROM /scwm/who INTO  TABLE lt\_repl\_who UP TO iv\_wo\_count ROWS  
               FOR ALL ENTRIES IN lt\_repl\_procty  
              WHERE lgnum = iv\_lgnum  
              AND hdr\_procty = lt\_repl\_procty-hdr\_procty  
              AND status = wmegc\_wo\_open  
              AND queue = iv\_queue.  
\*            ENDIF .  
  
            SELECT SINGLE zvalue INTO @DATA(lv\_repl)  
                    FROM zcon\_param\_val  
                    WHERE lgnum = @iv\_lgnum  
                    AND zreq\_name = 'REPL\_DET2\_REQ'  
                    AND zparam = 'REPL\_DET2\_PARAM'  
                    AND zactive = @abap\_true.  
  
            IF sy-subrc = 0.  
              IF lt\_repl\_who[] IS NOT INITIAL.  
                lt\_rtv\_who1[] = lt\_rtv\_who[].  
                lt\_rtv\_who[] = lt\_repl\_who[].  
                APPEND LINES OF lt\_rtv\_who1 TO lt\_rtv\_who.  
              ENDIF.  
            ENDIF.  
            DATA(lv\_str\_cnt) = lines( lt\_str\_who ) .  
            DATA(lv\_rtv\_cnt) = lines( lt\_rtv\_who ).  
  
            MOVE iv\_lgnum  TO ls\_log-extnumber.  
            CALL METHOD lo\_log->create\_log  
              EXPORTING  
                is\_log = ls\_log.  
  
            MESSAGE s033(zapl\_det) WITH 'Interleaving block started'  INTO lv\_msgtext.  
            lo\_log->add\_message2log( ).  
  
\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*When we have only putaway tasks  
            IF lt\_str\_who[] IS  NOT INITIAL AND lt\_rtv\_who[] IS INITIAL.  
              READ TABLE lt\_str\_who INTO ls\_str\_who WITH KEY hdr\_procty = '3090'.  
              IF sy-subrc = 0.  
                SELECT \* INTO TABLE lt\_interleave FROM zinter\_leave  
                         WHERE lgnum = iv\_lgnum  
                         AND resource\_name = iv\_queue  
                         AND active EQ abap\_true.  
                IF sy-subrc NE 0.  
                  INSERT ls\_str\_who INTO ct\_who INDEX 1.  
  
                  SELECT \*  INTO TABLE lt\_inter\_who  
                             FROM zinter\_leave  
                             WHERE lgnum = iv\_lgnum  
                             AND resource\_name = iv\_queue .  
                  IF sy-subrc = 0.  
                    READ TABLE lt\_inter\_who INTO ls\_inter\_who INDEX 1.  
                    IF ls\_str\_who-who NE ls\_inter\_who-who\_no.  
                      UPDATE zinter\_leave SET active = abap\_false  
                                              who\_no = ls\_str\_who-who  
                                             WHERE lgnum EQ iv\_lgnum AND resource\_name = iv\_queue.  
                      COMMIT WORK.  
                      lv\_missing\_case = 2.  
                    ENDIF.  
                  ENDIF.  
                  RETURN.  
                ENDIF.  
  
              ENDIF.  
  
            ENDIF.  
\*\*\*\*\*\*\*\*\*\* end of when we have only putaway tasks  
  
\*\*\*\*\*\*\*\*\*\*\* When we have only retrival tasks  
  
  
            SELECT SINGLE zactive INTO lv\_active1 FROM zcon\_param\_val  
                              WHERE lgnum = iv\_lgnum AND zreq\_name = 'ZINTER' AND zactive EQ abap\_true.  
            IF lv\_active1 EQ abap\_true.  
              IF lt\_str\_who[] IS INITIAL AND lt\_rtv\_who[] IS NOT INITIAL.  
                READ TABLE lt\_rtv\_who INTO ls\_rtv\_who WITH KEY hdr\_procty = '3010'.  
                IF sy-subrc = 0.  
  
  
                  INSERT ls\_rtv\_who INTO ct\_who INDEX 1.  
                  READ TABLE ct\_who INTO cs\_who INDEX 1.  
                  iv\_who = cs\_who-who.  
                  ls\_det2\_fg-lgnum = iv\_lgnum.  
                  ls\_det2\_fg-crane\_no = lv\_crane\_no.  
                  ls\_det2\_fg-error\_no = ls\_final\_error-error.  
                  ls\_det2\_fg-who = iv\_who.  
                  ls\_det2\_fg-description = 'checking for picking task need rewarehousing repl '.  
                  ls\_det2\_fg-log\_date = sy-datum.  
                  ls\_det2\_fg-log\_time = sy-uzeit.  
                  APPEND ls\_det2\_fg TO lt\_det2\_fg.  
                  MODIFY zdet2\_fg\_log FROM TABLE lt\_det2\_fg.  
                  COMMIT WORK.  
  
                  lv\_missing\_case = 2.  
                  CALL METHOD lo\_badi\_transfer->transfer\_hu1  
                    EXPORTING  
                      iv\_lgnum   = iv\_lgnum  
                      is\_whorder = is\_who  
                      iv\_who     = iv\_who  
                      iv\_rsrc    = is\_rsrc  
                      it\_ordim\_o = ct\_ordim\_o  
                    CHANGING  
                      ev\_who     = lv\_who.  
  
                  CLEAR ls\_who.  
                  IF lv\_who+0(1) NE 0.  
                    IF iv\_who NE lv\_who.  
                      "if current WO already contains WT for HU in front of HU to be picked => do nothing  
                      ls\_who-who = lv\_who.  
                      INSERT ls\_who INTO ct\_who INDEX 1.  
                      ls\_det2\_fg-lgnum = iv\_lgnum.  
                      ls\_det2\_fg-crane\_no = lv\_crane\_no.  
                      ls\_det2\_fg-who = lv\_who.  
                      ls\_det2\_fg-description = 'Rewarehousing task sent to subsystem Picking task repl'.  
                      ls\_det2\_fg-log\_date = sy-datum.  
                      ls\_det2\_fg-log\_time = sy-uzeit.  
                      APPEND ls\_det2\_fg TO lt\_det2\_fg.  
                      MODIFY zdet2\_fg\_log FROM TABLE lt\_det2\_fg.  
                      COMMIT WORK.  
  
                      UPDATE zinter\_leave SET active = abap\_true  
                                              who\_no = ls\_who-who  
                                           WHERE lgnum EQ iv\_lgnum  
                                           AND resource\_name = iv\_queue.  
                      COMMIT WORK.  
                    ENDIF.  
                    cv\_rearr\_done = abap\_true.  
                    RETURN.  
                  ELSE.  
                    SELECT \*  INTO TABLE lt\_inter\_who FROM zinter\_leave  
                               WHERE lgnum = iv\_lgnum  
                               AND resource\_name = iv\_queue .  
                    IF sy-subrc = 0.  
                      LOOP AT lt\_inter\_who INTO ls\_inter\_who.  
                        READ TABLE ct\_who INTO cs\_who WITH KEY who = ls\_inter\_who-who\_no.  
                        IF sy-subrc = 0.  
                          "do nothing  
                        ELSE.  
                          UPDATE zinter\_leave SET active = abap\_false  
                                                  who\_no = ls\_rtv\_who-who  
                                              WHERE lgnum EQ iv\_lgnum  
                                              AND resource\_name = iv\_queue.  
                          COMMIT WORK.  
                        ENDIF.  
                      ENDLOOP.  
                    ENDIF.  
                    RETURN.  
                  ENDIF.  
  
                  SELECT \*  INTO TABLE lt\_inter\_who FROM zinter\_leave  
                            WHERE lgnum = iv\_lgnum  
                           AND resource\_name = iv\_queue .  
                  IF sy-subrc = 0.  
                    LOOP AT lt\_inter\_who INTO ls\_inter\_who.  
                      READ TABLE ct\_who INTO cs\_who WITH KEY who = ls\_inter\_who-who\_no.  
                      IF sy-subrc = 0.  
                        "do nothing  
                      ELSE.  
                        UPDATE zinter\_leave SET active = abap\_true  
                                                who\_no = ls\_rtv\_who-who  
                                            WHERE lgnum EQ iv\_lgnum  
                                            AND resource\_name = iv\_queue.  
                        COMMIT WORK.  
                      ENDIF.  
                    ENDLOOP.  
                  ENDIF.  
                  RETURN.  
                ENDIF.  
  
                READ TABLE lt\_rtv\_who INTO ls\_rtv\_who WITH KEY hdr\_procty = '2010'.  
                IF sy-subrc = 0.  
  
                  INSERT ls\_rtv\_who INTO ct\_who INDEX 1.  
                  READ TABLE ct\_who INTO cs\_who INDEX 1.  
                  iv\_who = cs\_who-who.  
                  ls\_det2\_fg-lgnum = iv\_lgnum.  
                  ls\_det2\_fg-crane\_no = lv\_crane\_no.  
                  ls\_det2\_fg-error\_no = ls\_final\_error-error.  
                  ls\_det2\_fg-who = iv\_who.  
                  ls\_det2\_fg-description = 'checking for picking task need rewarehousing '.  
                  ls\_det2\_fg-log\_date = sy-datum.  
                  ls\_det2\_fg-log\_time = sy-uzeit.  
                  APPEND ls\_det2\_fg TO lt\_det2\_fg.  
                  MODIFY zdet2\_fg\_log FROM TABLE lt\_det2\_fg.  
  
                  COMMIT WORK.  
                  lv\_missing\_case = 2.  
                  CALL METHOD lo\_badi\_transfer->transfer\_hu1  
                    EXPORTING  
                      iv\_lgnum   = iv\_lgnum  
                      is\_whorder = is\_who  
                      iv\_who     = iv\_who  
                      iv\_rsrc    = is\_rsrc  
                      it\_ordim\_o = ct\_ordim\_o  
                    CHANGING  
                      ev\_who     = lv\_who.  
  
                  CLEAR ls\_who.  
                  IF lv\_who+0(1) NE 0.  
                    IF iv\_who NE lv\_who.  
                      "if current WO already contains WT for HU in front of HU to be picked => do nothing  
                      ls\_who-who = lv\_who.  
                      INSERT ls\_who INTO ct\_who INDEX 1.  
                      ls\_det2\_fg-lgnum = iv\_lgnum.  
                      ls\_det2\_fg-crane\_no = lv\_crane\_no.  
                      ls\_det2\_fg-who = lv\_who.  
                      ls\_det2\_fg-description = 'Rewarehousing task sent to subsystem Picking task'.  
                      ls\_det2\_fg-log\_date = sy-datum.  
                      ls\_det2\_fg-log\_time = sy-uzeit.  
                      APPEND ls\_det2\_fg TO lt\_det2\_fg.  
                      MODIFY zdet2\_fg\_log FROM TABLE lt\_det2\_fg.  
                      COMMIT WORK.  
  
                      UPDATE zinter\_leave SET active = abap\_true  
                                              who\_no = ls\_who-who  
                                          WHERE lgnum EQ iv\_lgnum AND resource\_name = iv\_queue.  
                      COMMIT WORK.  
                    ENDIF.  
                    cv\_rearr\_done = abap\_true.  
                    RETURN.  
                  ELSE.  
                    SELECT \*  INTO TABLE lt\_inter\_who FROM zinter\_leave  
                             WHERE lgnum = iv\_lgnum AND resource\_name = iv\_queue .  
                    IF sy-subrc = 0.  
                      LOOP AT lt\_inter\_who INTO ls\_inter\_who.  
                        READ TABLE ct\_who INTO cs\_who WITH KEY who = ls\_inter\_who-who\_no.  
                        IF sy-subrc = 0.  
                          "do nothing  
                        ELSE.  
                          UPDATE zinter\_leave SET active = abap\_false  
                                                  who\_no = ls\_rtv\_who-who  
                                              WHERE lgnum EQ iv\_lgnum  
                                              AND resource\_name = iv\_queue.  
                          COMMIT WORK.  
                        ENDIF.  
                      ENDLOOP.  
                    ENDIF.  
                    RETURN.  
                  ENDIF.  
  
                  SELECT \*  INTO TABLE lt\_inter\_who FROM zinter\_leave  
                            WHERE lgnum = iv\_lgnum  
                            AND resource\_name = iv\_queue .  
                  IF sy-subrc = 0.  
                    LOOP AT lt\_inter\_who INTO ls\_inter\_who.  
                      READ TABLE ct\_who INTO cs\_who WITH KEY who = ls\_inter\_who-who\_no.  
                      IF sy-subrc = 0.  
                        "do nothing  
                      ELSE.  
                        UPDATE zinter\_leave SET active = abap\_true  
                                                who\_no = ls\_rtv\_who-who  
                                            WHERE lgnum EQ iv\_lgnum AND resource\_name = iv\_queue.  
                        COMMIT WORK.  
                      ENDIF.  
                    ENDLOOP.  
                  ENDIF.  
                  RETURN.  
                ENDIF.  
  
              ENDIF.  
  
            ELSE.  
              IF lt\_str\_who[] IS INITIAL AND lt\_rtv\_who[] IS NOT INITIAL.  
                READ TABLE lt\_rtv\_who INTO ls\_rtv\_who WITH KEY hdr\_procty = '3010'.  
                IF sy-subrc = 0.  
                  SELECT \* INTO TABLE lt\_interleave FROM zinter\_leave  
                    WHERE lgnum = iv\_lgnum AND resource\_name = iv\_queue AND active EQ abap\_true.  
                  IF sy-subrc = 0.  
  
                    READ TABLE lt\_rtv\_who INTO ls\_rtv\_who WITH KEY hdr\_procty = '3010'.  
                    IF sy-subrc = 0.  
                      INSERT ls\_rtv\_who INTO ct\_who INDEX 1.  
                      READ TABLE ct\_who INTO cs\_who INDEX 1.  
                      iv\_who = cs\_who-who.  
                      ls\_det2\_fg-lgnum = iv\_lgnum.  
                      ls\_det2\_fg-crane\_no = lv\_crane\_no.  
                      ls\_det2\_fg-error\_no = ls\_final\_error-error.  
                      ls\_det2\_fg-who = iv\_who.  
                      ls\_det2\_fg-description = 'checking for picking task need rewarehousing repl '.  
                      ls\_det2\_fg-log\_date = sy-datum.  
                      ls\_det2\_fg-log\_time = sy-uzeit.  
                      APPEND ls\_det2\_fg TO lt\_det2\_fg.  
                      MODIFY zdet2\_fg\_log FROM TABLE lt\_det2\_fg.  
                      COMMIT WORK.  
  
                      lv\_missing\_case = 2.  
                      CALL METHOD lo\_badi\_transfer->transfer\_hu1  
                        EXPORTING  
                          iv\_lgnum   = iv\_lgnum  
                          is\_whorder = is\_who  
                          iv\_who     = iv\_who  
                          iv\_rsrc    = is\_rsrc  
                          it\_ordim\_o = ct\_ordim\_o  
                        CHANGING  
                          ev\_who     = lv\_who.  
  
                      CLEAR ls\_who.  
                      IF lv\_who+0(1) NE 0.  
                        IF iv\_who NE lv\_who.  
                          "if current WO already contains WT for HU in front of HU to be picked => do nothing  
                          ls\_who-who = lv\_who.  
                          INSERT ls\_who INTO ct\_who INDEX 1.  
                          ls\_det2\_fg-lgnum = iv\_lgnum.  
                          ls\_det2\_fg-crane\_no = lv\_crane\_no.  
                          ls\_det2\_fg-who = lv\_who.  
                          ls\_det2\_fg-description = 'Rewarehousing task sent to subsystem Picking task repl'.  
                          ls\_det2\_fg-log\_date = sy-datum.  
                          ls\_det2\_fg-log\_time = sy-uzeit.  
                          APPEND ls\_det2\_fg TO lt\_det2\_fg.  
                          MODIFY zdet2\_fg\_log FROM TABLE lt\_det2\_fg.  
                          COMMIT WORK.  
  
                          UPDATE zinter\_leave SET active = abap\_true  
                                                  who\_no = ls\_who-who  
                                              WHERE lgnum EQ iv\_lgnum  
                                              AND resource\_name = iv\_queue.  
                          COMMIT WORK.  
                        ENDIF.  
                        cv\_rearr\_done = abap\_true.  
                        RETURN.  
                      ELSE.  
                        SELECT \*  INTO TABLE lt\_inter\_who FROM zinter\_leave  
                                  WHERE lgnum = iv\_lgnum  
                                   AND resource\_name = iv\_queue .  
                        IF sy-subrc = 0.  
                          LOOP AT lt\_inter\_who INTO ls\_inter\_who.  
                            READ TABLE ct\_who INTO cs\_who WITH KEY who = ls\_inter\_who-who\_no.  
                            IF sy-subrc = 0.  
                              "do nothing  
                            ELSE.  
                              UPDATE zinter\_leave SET active = abap\_false  
                                                     who\_no = ls\_rtv\_who-who  
                                                  WHERE lgnum EQ iv\_lgnum AND resource\_name = iv\_queue.  
                              COMMIT WORK.  
                            ENDIF.  
                          ENDLOOP.  
                        ENDIF.  
                        RETURN.  
                      ENDIF.  
  
                      SELECT \*  INTO TABLE lt\_inter\_who FROM zinter\_leave  
                                 WHERE lgnum = iv\_lgnum AND resource\_name = iv\_queue .  
                      IF sy-subrc = 0.  
                        LOOP AT lt\_inter\_who INTO ls\_inter\_who.  
                          READ TABLE ct\_who INTO cs\_who WITH KEY who = ls\_inter\_who-who\_no.  
                          IF sy-subrc = 0.  
                            "do nothing  
                          ELSE.  
                            UPDATE zinter\_leave SET active = abap\_true  
                                                    who\_no = ls\_rtv\_who-who  
                                                WHERE lgnum EQ iv\_lgnum AND resource\_name = iv\_queue.  
                            COMMIT WORK.  
                          ENDIF.  
                        ENDLOOP.  
                      ENDIF.  
                      RETURN.  
                    ENDIF."sy-subrc = 0 end  
  
  
                  ENDIF.  
                ENDIF.  
  
                READ TABLE lt\_rtv\_who INTO ls\_rtv\_who WITH KEY hdr\_procty = '2010'.  
                IF sy-subrc = 0.  
                  SELECT \* INTO TABLE lt\_interleave FROM zinter\_leave WHERE  
                          lgnum = iv\_lgnum AND resource\_name = iv\_queue AND active EQ abap\_true.  
                  IF sy-subrc = 0.  
  
                    READ TABLE lt\_rtv\_who INTO ls\_rtv\_who WITH KEY hdr\_procty = '2010'.  
                    IF sy-subrc = 0.  
                      INSERT ls\_rtv\_who INTO ct\_who INDEX 1.  
                      READ TABLE ct\_who INTO cs\_who INDEX 1.  
                      iv\_who = cs\_who-who.  
                      ls\_det2\_fg-lgnum = iv\_lgnum.  
                      ls\_det2\_fg-crane\_no = lv\_crane\_no.  
                      ls\_det2\_fg-error\_no = ls\_final\_error-error.  
                      ls\_det2\_fg-who = iv\_who.  
                      ls\_det2\_fg-description = 'checking for picking task need rewarehousing '.  
                      ls\_det2\_fg-log\_date = sy-datum.  
                      ls\_det2\_fg-log\_time = sy-uzeit.  
                      APPEND ls\_det2\_fg TO lt\_det2\_fg.  
                      MODIFY zdet2\_fg\_log FROM TABLE lt\_det2\_fg.  
  
                      COMMIT WORK.  
                      lv\_missing\_case = 2.  
                      CALL METHOD lo\_badi\_transfer->transfer\_hu1  
                        EXPORTING  
                          iv\_lgnum   = iv\_lgnum  
                          is\_whorder = is\_who  
                          iv\_who     = iv\_who  
                          iv\_rsrc    = is\_rsrc  
                          it\_ordim\_o = ct\_ordim\_o  
                        CHANGING  
                          ev\_who     = lv\_who.  
  
                      CLEAR ls\_who.  
                      IF lv\_who+0(1) NE 0.  
                        IF iv\_who NE lv\_who.  
                          "if current WO already contains WT for HU in front of HU to be picked => do nothing  
                          ls\_who-who = lv\_who.  
                          INSERT ls\_who INTO ct\_who INDEX 1.  
                          ls\_det2\_fg-lgnum = iv\_lgnum.  
                          ls\_det2\_fg-crane\_no = lv\_crane\_no.  
                          ls\_det2\_fg-who = lv\_who.  
                          ls\_det2\_fg-description = 'Rewarehousing task sent to subsystem Picking task'.  
                          ls\_det2\_fg-log\_date = sy-datum.  
                          ls\_det2\_fg-log\_time = sy-uzeit.  
                          APPEND ls\_det2\_fg TO lt\_det2\_fg.  
                          MODIFY zdet2\_fg\_log FROM TABLE lt\_det2\_fg.  
                          COMMIT WORK.  
  
                          UPDATE zinter\_leave SET active = abap\_true  
                                                  who\_no = ls\_who-who  
                                                  WHERE lgnum EQ iv\_lgnum AND resource\_name = iv\_queue.  
                          COMMIT WORK.  
                        ENDIF.  
                        cv\_rearr\_done = abap\_true.  
                        RETURN.  
                      ELSE.  
                        SELECT \*  INTO TABLE lt\_inter\_who FROM zinter\_leave  
                                  WHERE lgnum = iv\_lgnum AND resource\_name = iv\_queue .  
                        IF sy-subrc = 0.  
                          LOOP AT lt\_inter\_who INTO ls\_inter\_who.  
                            READ TABLE ct\_who INTO cs\_who WITH KEY who = ls\_inter\_who-who\_no.  
                            IF sy-subrc = 0.  
                              "do nothing  
                            ELSE.  
                              UPDATE zinter\_leave SET active = abap\_false  
                                                      who\_no = ls\_rtv\_who-who  
                                                  WHERE lgnum EQ iv\_lgnum  
                                                  AND resource\_name = iv\_queue.  
                              COMMIT WORK.  
                            ENDIF.  
                          ENDLOOP.  
                        ENDIF.  
                        RETURN.  
                      ENDIF.  
  
                      SELECT \*  INTO TABLE lt\_inter\_who FROM zinter\_leave  
                                WHERE lgnum = iv\_lgnum AND resource\_name = iv\_queue .  
                      IF sy-subrc = 0.  
                        LOOP AT lt\_inter\_who INTO ls\_inter\_who.  
                          READ TABLE ct\_who INTO cs\_who WITH KEY who = ls\_inter\_who-who\_no.  
                          IF sy-subrc = 0.  
                            "do nothing  
                          ELSE.  
                            UPDATE zinter\_leave SET active = abap\_true  
                                                    who\_no = ls\_rtv\_who-who  
                                                WHERE lgnum EQ iv\_lgnum  
                                                AND resource\_name = iv\_queue.  
                            COMMIT WORK.  
                          ENDIF.  
                        ENDLOOP.  
                      ENDIF.  
                      RETURN.  
                    ENDIF."sy-subrc = 0 end  
  
                  ENDIF.  
                ENDIF.  
  
              ENDIF.  
  
            ENDIF.  
\*\*\*\*\*\*\*\*\*\* end of when we have retrival tasks  
  
  
            READ TABLE lt\_str\_who INTO ls\_str\_who WITH KEY hdr\_procty = '3090'.  
            IF sy-subrc = 0.  
              SELECT \* INTO TABLE lt\_interleave FROM zinter\_leave  
                     WHERE lgnum = iv\_lgnum AND resource\_name = iv\_queue AND active EQ abap\_true.  
              IF sy-subrc NE 0.  
                INSERT ls\_str\_who INTO ct\_who INDEX 1.  
  
                SELECT \*  INTO TABLE lt\_inter\_who FROM zinter\_leave  
                         WHERE lgnum = iv\_lgnum AND resource\_name = iv\_queue .  
                IF sy-subrc = 0.  
                  READ TABLE lt\_inter\_who INTO ls\_inter\_who INDEX 1.  
                  IF ls\_str\_who-who NE ls\_inter\_who-who\_no.  
                    UPDATE zinter\_leave SET active = abap\_true  
                                            who\_no = ls\_str\_who-who  
                                        WHERE lgnum EQ iv\_lgnum  
                                        AND resource\_name = iv\_queue.  
                    COMMIT WORK.  
                    lv\_missing\_case = 2.  
                  ENDIF.  
                ENDIF.  
  
                RETURN.  
              ELSEIF sy-subrc = 0.  
                READ TABLE lt\_rtv\_who INTO ls\_rtv\_who WITH KEY hdr\_procty = '3010'.  
                IF sy-subrc = 0.  
                  INSERT ls\_rtv\_who INTO ct\_who INDEX 1.  
                  READ TABLE ct\_who INTO cs\_who INDEX 1.  
                  iv\_who = cs\_who-who.  
                  ls\_det2\_fg-lgnum = iv\_lgnum.  
                  ls\_det2\_fg-crane\_no = lv\_crane\_no.  
                  ls\_det2\_fg-error\_no = ls\_final\_error-error.  
                  ls\_det2\_fg-who = iv\_who.  
                  ls\_det2\_fg-description = 'check for putaway task need rewarehousing when we have only putaway '.  
                  ls\_det2\_fg-log\_date = sy-datum.  
                  ls\_det2\_fg-log\_time = sy-uzeit.  
                  APPEND ls\_det2\_fg TO lt\_det2\_fg.  
                  MODIFY zdet2\_fg\_log FROM TABLE lt\_det2\_fg.  
                  COMMIT WORK.  
                  lv\_missing\_case = 2.  
                  CALL METHOD lo\_badi\_transfer->transfer\_hu1  
                    EXPORTING  
                      iv\_lgnum   = iv\_lgnum  
                      is\_whorder = is\_who  
                      iv\_who     = iv\_who  
                      iv\_rsrc    = is\_rsrc  
                      it\_ordim\_o = ct\_ordim\_o  
                    CHANGING  
                      ev\_who     = lv\_who.  
  
                  CLEAR ls\_who.  
                  IF lv\_who+0(1) NE 0.  
                    IF iv\_who NE lv\_who.  
                      "if current WO already contains WT for HU in front of HU to be picked => do nothing  
                      ls\_who-who = lv\_who.  
                      INSERT ls\_who INTO ct\_who INDEX 1.  
                      ls\_det2\_fg-lgnum = iv\_lgnum.  
                      ls\_det2\_fg-crane\_no = lv\_crane\_no.  
                      ls\_det2\_fg-who = lv\_who.  
                      ls\_det2\_fg-description = 'Rewarehouse task sent to subsystem Putaway tasks are present'.  
                      ls\_det2\_fg-log\_date = sy-datum.  
                      ls\_det2\_fg-log\_time = sy-uzeit.  
                      APPEND ls\_det2\_fg TO lt\_det2\_fg.  
                      MODIFY zdet2\_fg\_log FROM TABLE lt\_det2\_fg.  
                      COMMIT WORK.  
\*                      SELECT SINGLE who\_no INTO lv\_inter\_who FROM zinter\_leave WHERE lgnum = iv\_lgnum AND resource\_name = iv\_queue .  
\*                      IF sy-subrc = 0.  
\*                        IF lv\_inter\_who NE ls\_who-who.  
                      UPDATE zinter\_leave SET active = abap\_true  
                                              who\_no = ls\_who-who  
                                          WHERE lgnum EQ iv\_lgnum  
                                          AND resource\_name = iv\_queue.  
                      COMMIT WORK.  
                    ENDIF.  
                    cv\_rearr\_done = abap\_true.  
                    RETURN.  
                  ELSE.  
                    SELECT \*  INTO TABLE lt\_inter\_who FROM zinter\_leave  
                              WHERE lgnum = iv\_lgnum AND resource\_name = iv\_queue .  
                    IF sy-subrc = 0.  
                      LOOP AT lt\_inter\_who INTO ls\_inter\_who.  
                        READ TABLE ct\_who INTO cs\_who WITH KEY who = ls\_inter\_who-who\_no.  
                        IF sy-subrc = 0.  
                          "do nothing  
                        ELSE.  
                          UPDATE zinter\_leave SET active = abap\_false  
                                                  who\_no = ls\_rtv\_who-who  
                                              WHERE lgnum EQ iv\_lgnum  
                                              AND resource\_name = iv\_queue.  
                          COMMIT WORK.  
                        ENDIF.  
                      ENDLOOP.  
                    ENDIF.  
                    RETURN.  
                  ENDIF.  
  
                  SELECT \*  INTO TABLE lt\_inter\_who FROM zinter\_leave  
                             WHERE lgnum = iv\_lgnum AND resource\_name = iv\_queue .  
                  IF sy-subrc = 0.  
                    LOOP AT lt\_inter\_who INTO ls\_inter\_who.  
                      READ TABLE ct\_who INTO cs\_who WITH KEY who = ls\_inter\_who-who\_no.  
                      IF sy-subrc = 0.  
                        "do nothing  
                      ELSE.  
                        UPDATE zinter\_leave SET active = abap\_false  
                                                who\_no = ls\_rtv\_who-who  
                                            WHERE lgnum EQ iv\_lgnum AND resource\_name = iv\_queue.  
                        COMMIT WORK.  
                      ENDIF.  
                    ENDLOOP.  
                  ENDIF.  
                  RETURN.  
                ELSE."3010 task is not present  
                  READ TABLE lt\_rtv\_who INTO ls\_rtv\_who WITH KEY hdr\_procty = '2010'.  
                  IF sy-subrc = 0.  
                    INSERT ls\_rtv\_who INTO ct\_who INDEX 1.  
                    READ TABLE ct\_who INTO cs\_who INDEX 1.  
                    iv\_who = cs\_who-who.  
                    ls\_det2\_fg-lgnum = iv\_lgnum.  
                    ls\_det2\_fg-crane\_no = lv\_crane\_no.  
                    ls\_det2\_fg-error\_no = ls\_final\_error-error.  
                    ls\_det2\_fg-who = iv\_who.  
                    ls\_det2\_fg-description = 'checking for putaway task need rewarehousing when we have only putaway  '.  
                    ls\_det2\_fg-log\_date = sy-datum.  
                    ls\_det2\_fg-log\_time = sy-uzeit.  
                    APPEND ls\_det2\_fg TO lt\_det2\_fg.  
                    MODIFY zdet2\_fg\_log FROM TABLE lt\_det2\_fg.  
                    COMMIT WORK.  
                    lv\_missing\_case = 2.  
                    CALL METHOD lo\_badi\_transfer->transfer\_hu1  
                      EXPORTING  
                        iv\_lgnum   = iv\_lgnum  
                        is\_whorder = is\_who  
                        iv\_who     = iv\_who  
                        iv\_rsrc    = is\_rsrc  
                        it\_ordim\_o = ct\_ordim\_o  
                      CHANGING  
                        ev\_who     = lv\_who.  
  
                    CLEAR ls\_who.  
                    IF lv\_who+0(1) NE 0.  
                      IF iv\_who NE lv\_who.  
                        "if current WO already contains WT for HU in front of HU to be picked => do nothing  
                        ls\_who-who = lv\_who.  
                        INSERT ls\_who INTO ct\_who INDEX 1.  
                        ls\_det2\_fg-lgnum = iv\_lgnum.  
                        ls\_det2\_fg-crane\_no = lv\_crane\_no.  
                        ls\_det2\_fg-who = lv\_who.  
                        ls\_det2\_fg-description = 'Rewarehousing task sent to subsystem Putaway tasks are present'.  
                        ls\_det2\_fg-log\_date = sy-datum.  
                        ls\_det2\_fg-log\_time = sy-uzeit.  
                        APPEND ls\_det2\_fg TO lt\_det2\_fg.  
                        MODIFY zdet2\_fg\_log FROM TABLE lt\_det2\_fg.  
                        COMMIT WORK.  
  
                        UPDATE zinter\_leave SET active = abap\_true  
                                                who\_no = ls\_who-who  
                                            WHERE lgnum EQ iv\_lgnum  
                                            AND resource\_name = iv\_queue.  
                        COMMIT WORK.  
                      ENDIF.  
                      cv\_rearr\_done = abap\_true.  
                      RETURN.  
                    ELSE.  
                      SELECT \*  INTO TABLE lt\_inter\_who FROM zinter\_leave  
                                WHERE lgnum = iv\_lgnum AND resource\_name = iv\_queue .  
                      IF sy-subrc = 0.  
                        LOOP AT lt\_inter\_who INTO ls\_inter\_who.  
                          READ TABLE ct\_who INTO cs\_who WITH KEY who = ls\_inter\_who-who\_no.  
                          IF sy-subrc = 0.  
                            "do nothing  
                          ELSE.  
                            UPDATE zinter\_leave SET active = abap\_false  
                                                    who\_no = ls\_rtv\_who-who  
                                                WHERE lgnum EQ iv\_lgnum  
                                                AND resource\_name = iv\_queue.  
                            COMMIT WORK.  
                          ENDIF.  
                        ENDLOOP.  
                      ENDIF.  
                      RETURN.  
                    ENDIF.  
  
                    SELECT \*  INTO TABLE lt\_inter\_who FROM zinter\_leave  
                              WHERE lgnum = iv\_lgnum AND resource\_name = iv\_queue .  
                    IF sy-subrc = 0.  
                      LOOP AT lt\_inter\_who INTO ls\_inter\_who.  
                        READ TABLE ct\_who INTO cs\_who WITH KEY who = ls\_inter\_who-who\_no.  
                        IF sy-subrc = 0.  
                          "do nothing  
                        ELSE.  
                          UPDATE zinter\_leave SET active = abap\_false  
                                                  who\_no = ls\_rtv\_who-who  
                                              WHERE lgnum EQ iv\_lgnum  
                                              AND resource\_name = iv\_queue.  
                          COMMIT WORK.  
                        ENDIF.  
                      ENDLOOP.  
                    ENDIF.  
                    RETURN.  
                  ENDIF."sy-subrc = 0 end  
                ENDIF.  
              ENDIF.  
            ELSE." lt\_str\_who is initial  
              READ TABLE lt\_rtv\_who INTO ls\_rtv\_who WITH KEY hdr\_procty = '3010'.  
              IF sy-subrc = 0.  
                INSERT ls\_rtv\_who INTO ct\_who INDEX 1.  
                READ TABLE ct\_who INTO cs\_who INDEX 1.  
                iv\_who = cs\_who-who.  
                ls\_det2\_fg-lgnum = iv\_lgnum.  
                ls\_det2\_fg-crane\_no = lv\_crane\_no.  
                ls\_det2\_fg-error\_no = ls\_final\_error-error.  
                ls\_det2\_fg-who = iv\_who.  
                ls\_det2\_fg-description = 'check for putaway task need rewarehousing for picking  '.  
                ls\_det2\_fg-log\_date = sy-datum.  
                ls\_det2\_fg-log\_time = sy-uzeit.  
                APPEND ls\_det2\_fg TO lt\_det2\_fg.  
                MODIFY zdet2\_fg\_log FROM TABLE lt\_det2\_fg.  
                COMMIT WORK.  
                lv\_missing\_case = 2.  
                CALL METHOD lo\_badi\_transfer->transfer\_hu1  
                  EXPORTING  
                    iv\_lgnum   = iv\_lgnum  
                    is\_whorder = is\_who  
                    iv\_who     = iv\_who  
                    iv\_rsrc    = is\_rsrc  
                    it\_ordim\_o = ct\_ordim\_o  
                  CHANGING  
                    ev\_who     = lv\_who.  
  
                CLEAR ls\_who.  
                IF lv\_who+0(1) NE 0.  
                  IF iv\_who NE lv\_who.  
                    "if current WO already contains WT for HU in front of HU to be picked => do nothing  
                    ls\_who-who = lv\_who.  
                    INSERT ls\_who INTO ct\_who INDEX 1.  
                    ls\_det2\_fg-lgnum = iv\_lgnum.  
                    ls\_det2\_fg-crane\_no = lv\_crane\_no.  
                    ls\_det2\_fg-who = lv\_who.  
                    ls\_det2\_fg-description = 'Rewarehouse task sent to subsystem for picking'.  
                    ls\_det2\_fg-log\_date = sy-datum.  
                    ls\_det2\_fg-log\_time = sy-uzeit.  
                    APPEND ls\_det2\_fg TO lt\_det2\_fg.  
                    MODIFY zdet2\_fg\_log FROM TABLE lt\_det2\_fg.  
                    COMMIT WORK.  
  
                    UPDATE zinter\_leave SET active = abap\_true  
                                            who\_no = ls\_who-who  
                                        WHERE lgnum EQ iv\_lgnum  
                                        AND resource\_name = iv\_queue.  
                    COMMIT WORK.  
                  ENDIF.  
                  cv\_rearr\_done = abap\_true.  
                  RETURN.  
                ELSE.  
                  SELECT \*  INTO TABLE lt\_inter\_who FROM zinter\_leave  
                            WHERE lgnum = iv\_lgnum AND resource\_name = iv\_queue .  
                  IF sy-subrc = 0.  
                    LOOP AT lt\_inter\_who INTO ls\_inter\_who.  
                      READ TABLE ct\_who INTO cs\_who WITH KEY who = ls\_inter\_who-who\_no.  
                      IF sy-subrc = 0.  
                        "do nothing  
                      ELSE.  
                        UPDATE zinter\_leave SET active = abap\_false  
                                                who\_no = ls\_rtv\_who-who  
                                            WHERE lgnum EQ iv\_lgnum  
                                            AND resource\_name = iv\_queue.  
                        COMMIT WORK.  
                      ENDIF.  
                    ENDLOOP.  
                  ENDIF.  
                  RETURN.  
                ENDIF.  
  
                SELECT \*  INTO TABLE lt\_inter\_who FROM zinter\_leave  
                          WHERE lgnum = iv\_lgnum AND resource\_name = iv\_queue .  
                IF sy-subrc = 0.  
                  LOOP AT lt\_inter\_who INTO ls\_inter\_who.  
                    READ TABLE ct\_who INTO cs\_who WITH KEY who = ls\_inter\_who-who\_no.  
                    IF sy-subrc = 0.  
                      "do nothing  
                    ELSE.  
                      UPDATE zinter\_leave SET active = abap\_false  
                                              who\_no = ls\_rtv\_who-who  
                                          WHERE lgnum EQ iv\_lgnum  
                                          AND resource\_name = iv\_queue.  
                      COMMIT WORK.  
                    ENDIF.  
                  ENDLOOP.  
                ENDIF.  
                RETURN.  
              ELSE."3010 task is not present when lt\_str\_who is initial  
                READ TABLE lt\_rtv\_who INTO ls\_rtv\_who WITH KEY hdr\_procty = '2010'.  
                IF sy-subrc = 0.  
                  INSERT ls\_rtv\_who INTO ct\_who INDEX 1.  
                  READ TABLE ct\_who INTO cs\_who INDEX 1.  
                  iv\_who = cs\_who-who.  
                  ls\_det2\_fg-lgnum = iv\_lgnum.  
                  ls\_det2\_fg-crane\_no = lv\_crane\_no.  
                  ls\_det2\_fg-error\_no = ls\_final\_error-error.  
                  ls\_det2\_fg-who = iv\_who.  
                  ls\_det2\_fg-description = 'checking for putaway task need rewarehousing for picking  '.  
                  ls\_det2\_fg-log\_date = sy-datum.  
                  ls\_det2\_fg-log\_time = sy-uzeit.  
                  APPEND ls\_det2\_fg TO lt\_det2\_fg.  
                  MODIFY zdet2\_fg\_log FROM TABLE lt\_det2\_fg.  
                  COMMIT WORK.  
                  lv\_missing\_case = 2.  
                  CALL METHOD lo\_badi\_transfer->transfer\_hu1  
                    EXPORTING  
                      iv\_lgnum   = iv\_lgnum  
                      is\_whorder = is\_who  
                      iv\_who     = iv\_who  
                      iv\_rsrc    = is\_rsrc  
                      it\_ordim\_o = ct\_ordim\_o  
                    CHANGING  
                      ev\_who     = lv\_who.  
  
                  CLEAR ls\_who.  
                  IF lv\_who+0(1) NE 0.  
                    IF iv\_who NE lv\_who.  
                      "if current WO already contains WT for HU in front of HU to be picked => do nothing  
                      ls\_who-who = lv\_who.  
                      INSERT ls\_who INTO ct\_who INDEX 1.  
                      ls\_det2\_fg-lgnum = iv\_lgnum.  
                      ls\_det2\_fg-crane\_no = lv\_crane\_no.  
                      ls\_det2\_fg-who = lv\_who.  
                      ls\_det2\_fg-description = 'Rewarehousing task sent to subsystem Putaway tasks are present'.  
                      ls\_det2\_fg-log\_date = sy-datum.  
                      ls\_det2\_fg-log\_time = sy-uzeit.  
                      APPEND ls\_det2\_fg TO lt\_det2\_fg.  
                      MODIFY zdet2\_fg\_log FROM TABLE lt\_det2\_fg.  
                      COMMIT WORK.  
  
                      UPDATE zinter\_leave SET active = abap\_true  
                                              who\_no = ls\_who-who  
                                          WHERE lgnum EQ iv\_lgnum  
                                          AND resource\_name = iv\_queue.  
                      COMMIT WORK.  
  
                    ENDIF.  
                    cv\_rearr\_done = abap\_true.  
                    RETURN.  
                  ELSE.  
                    SELECT \*  INTO TABLE lt\_inter\_who FROM zinter\_leave  
                               WHERE lgnum = iv\_lgnum AND resource\_name = iv\_queue .  
                    IF sy-subrc = 0.  
                      LOOP AT lt\_inter\_who INTO ls\_inter\_who.  
                        READ TABLE ct\_who INTO cs\_who WITH KEY who = ls\_inter\_who-who\_no.  
                        IF sy-subrc = 0.  
                          "do nothing  
                        ELSE.  
                          UPDATE zinter\_leave SET active = abap\_false  
                                                  who\_no = ls\_rtv\_who-who  
                                              WHERE lgnum EQ iv\_lgnum  
                                              AND resource\_name = iv\_queue.  
                          COMMIT WORK.  
                        ENDIF.  
                      ENDLOOP.  
  
                    ENDIF.  
                    RETURN.  
                  ENDIF.  
  
                  SELECT \*  INTO TABLE lt\_inter\_who FROM zinter\_leave  
                            WHERE lgnum = iv\_lgnum AND resource\_name = iv\_queue .  
                  IF sy-subrc = 0.  
                    LOOP AT lt\_inter\_who INTO ls\_inter\_who.  
                      READ TABLE ct\_who INTO cs\_who WITH KEY who = ls\_inter\_who-who\_no.  
                      IF sy-subrc = 0.  
                        "do nothing  
                      ELSE.  
                        UPDATE zinter\_leave SET active = abap\_false  
                                                who\_no = ls\_rtv\_who-who  
                                            WHERE lgnum EQ iv\_lgnum  
                                            AND resource\_name = iv\_queue.  
                        COMMIT WORK.  
                      ENDIF.  
                    ENDLOOP.  
                  ENDIF.  
                  RETURN.  
                ENDIF."sy-subrc = 0  of 2010 read table end  
  
              ENDIF."3010 of read table end  
  
            ENDIF."3090 READ TABLE end  
  
          ENDIF." end of interleaving  
  
  
        ELSEIF lt\_final\_error[] IS NOT INITIAL. " if lt\_final\_error[] is not initial  
  
          LOOP AT lt\_final\_error INTO ls\_final\_error.  
            MOVE ls\_final\_error-huident  TO ls\_log-extnumber.  
            CALL METHOD lo\_log->create\_log  
              EXPORTING  
                is\_log = ls\_log.  
            MESSAGE s031(zapl\_det) WITH ls\_final\_error-error INTO lv\_msgtext.  
            lo\_log->add\_message2log( ).  
            IF ls\_final\_error-error IS INITIAL OR ls\_final\_error-error+0(2) EQ '00'.  
              SELECT SINGLE zvalue INTO ls\_zvalue FROM zcon\_param\_val  
                                   WHERE lgnum = iv\_lgnum  
                                   AND zreq\_name = 'INTERLEAVING'  
                                   AND zparam = 'INTERLEAVE\_PARAM'  
                                   AND zactive = abap\_true.  
              IF sy-subrc EQ 0.  
  
\*              refresh: ct\_who[].  
\*\*-- MFS Whse process type constants  
                lv\_id1 = zapl\_cl\_common\_constants=>gc\_mfs\_tele\_send.  
                CALL METHOD zapl\_cl\_common\_constants=>get\_param\_value  
                  EXPORTING  
                    iv\_lgnum = iv\_lgnum  
                    iv\_id1   = lv\_id1  
                  IMPORTING  
                    et\_param = lt\_param\_mfs\_tele\_send.  
  
\*-- Scheduler constants  
                lv\_id1 = zapl\_cl\_common\_constants=>gc\_agc\_scheduler.  
                CALL METHOD zapl\_cl\_common\_constants=>get\_param\_value  
                  EXPORTING  
                    iv\_lgnum = iv\_lgnum  
                    iv\_id1   = lv\_id1  
                  IMPORTING  
                    et\_param = lt\_param\_agc\_schd.  
  
                CLEAR ls\_param.  
                READ TABLE lt\_param\_agc\_schd INTO ls\_param WITH KEY id2 = zapl\_cl\_common\_constants=>gc\_max\_direct\_command.  
                IF sy-subrc = 0.  
                  MOVE ls\_param-value TO lv\_max\_direct .  
                ENDIF.  
  
                CLEAR ls\_param.  
                READ TABLE lt\_param\_agc\_schd INTO ls\_param WITH KEY id2 = zapl\_cl\_common\_constants=>gc\_max\_storage\_command.  
                IF sy-subrc = 0.  
                  MOVE ls\_param-value TO lv\_max\_storage .  
                ENDIF.  
  
                CLEAR ls\_param.  
                READ TABLE lt\_param\_agc\_schd INTO ls\_param WITH KEY id2 = zapl\_cl\_common\_constants=>gc\_max\_retrieve\_command.  
                IF sy-subrc = 0.  
                  MOVE ls\_param-value TO lv\_max\_retrieve .  
                ENDIF.  
  
\*-- Read Whse procty for Storage, retrieval and Direct transfer operations  
                CLEAR : ls\_param.  
                READ TABLE lt\_param\_agc\_schd INTO ls\_param WITH KEY id2 = zapl\_cl\_common\_constants=>gc\_direct\_transfer.  
                IF sy-subrc = 0.  
                  ls\_hdr\_procty-hdr\_procty = ls\_param-value .  
                  APPEND ls\_hdr\_procty TO lt\_dir\_procty.  
                ENDIF.  
  
                CLEAR : ls\_param.  
                READ TABLE lt\_param\_agc\_schd INTO ls\_param WITH KEY id2 = zapl\_cl\_common\_constants=>gc\_retrieval\_interleave.  
                IF sy-subrc = 0.  
                  ls\_hdr\_procty-hdr\_procty = ls\_param-value.  
                  APPEND ls\_hdr\_procty TO lt\_rtv\_procty.  
                ENDIF.  
                READ TABLE lt\_param\_agc\_schd INTO ls\_param WITH KEY id2 = zapl\_cl\_common\_constants=>gc\_repl\_interleave.  
                IF sy-subrc = 0.  
                  ls\_hdr\_procty-hdr\_procty = ls\_param-value.  
                  APPEND ls\_hdr\_procty TO lt\_repl\_procty.  
                ENDIF.  
  
                CLEAR : ls\_param.  
                LOOP AT lt\_param\_agc\_schd INTO ls\_param WHERE id2 = zapl\_cl\_common\_constants=>gc\_storage\_interleave.  
                  ls\_hdr\_procty-hdr\_procty = ls\_param-value.  
                  APPEND ls\_hdr\_procty TO lt\_str\_procty.  
                ENDLOOP.  
  
\*    CLEAR :  lt\_who[] , ct\_who[], lv\_count, lv\_retrieve, lv\_storage, lv\_direct.  
  
\*-- get all 'in process' WO, to check current WO count for each operation.Based on current count  
\*   and maximum allowed count, next set of WOs shall be decided and sent  
                SELECT \* FROM /scwm/who INTO TABLE lt\_who  
                          WHERE lgnum = iv\_lgnum  
                            AND status = wmegc\_wo\_open.  
  
                IF NOT lt\_who[] IS INITIAL.  
  
                  LOOP AT lt\_who ASSIGNING <ls\_who>.  
  
  
\*-- Retrieval  
                    READ TABLE lt\_rtv\_procty TRANSPORTING NO FIELDS WITH KEY hdr\_procty = <ls\_who>-hdr\_procty.  
                    IF sy-subrc = 0.  
                      lv\_retrieve = lv\_retrieve + 1.  
                    ENDIF.  
\*\*\*                Replineshment  
                    READ TABLE lt\_repl\_procty TRANSPORTING NO FIELDS WITH KEY hdr\_procty = <ls\_who>-hdr\_procty.  
                    IF sy-subrc = 0.  
                      lv\_retrieve = lv\_retrieve + 1.  
                    ENDIF.  
  
\*-- Storage  
                    READ TABLE lt\_str\_procty TRANSPORTING NO FIELDS WITH KEY hdr\_procty = <ls\_who>-hdr\_procty.  
                    IF sy-subrc = 0.  
                      lv\_storage = lv\_storage + 1.  
                    ENDIF.  
                  ENDLOOP.  
                ENDIF.  
  
                IF lv\_max\_retrieve = lv\_retrieve.  
                  CLEAR lv\_msgtext.  
                  MESSAGE w432(zapl\_asrs) WITH  lv\_retrieve 'Retrieve' INTO lv\_msgtext.  
                  lo\_log->add\_message( ).  
                ENDIF.  
  
                IF lv\_max\_storage = lv\_storage.  
                  CLEAR lv\_msgtext.  
                  MESSAGE w432(zapl\_asrs) WITH  lv\_storage 'Storage' INTO lv\_msgtext.  
                  lo\_log->add\_message( ).  
                ENDIF.  
  
  
  
\*-- Validate maximum WO limit and select WO  acordingly  
  
                lv\_retrieve = lv\_max\_retrieve - lv\_retrieve.  
                lv\_storage = lv\_max\_storage - lv\_storage.  
                lv\_direct = lv\_max\_direct - lv\_direct.  
  
\*            IF lv\_storage > 0 .  
  
                SELECT \* FROM /scwm/who INTO TABLE lt\_str\_who UP TO iv\_wo\_count ROWS  
                FOR ALL ENTRIES IN lt\_str\_procty  
                  WHERE lgnum = iv\_lgnum  
                    AND hdr\_procty = lt\_str\_procty-hdr\_procty  
                    AND status = wmegc\_wo\_open  
                    AND queue = iv\_queue.  
  
\*            ENDIF.  
  
\*            IF lv\_retrieve > 0.  
                SELECT \* FROM /scwm/who INTO  TABLE  lt\_rtv\_who UP TO iv\_wo\_count ROWS  
                  FOR ALL ENTRIES IN lt\_rtv\_procty  
                    WHERE lgnum = iv\_lgnum  
                      AND hdr\_procty = lt\_rtv\_procty-hdr\_procty  
                      AND status = wmegc\_wo\_open  
                      AND queue = iv\_queue.  
\*\*\*\*\*\*\*\*                REplenishment  
                SELECT \* FROM /scwm/who INTO  TABLE lt\_repl\_who UP TO iv\_wo\_count ROWS  
                   FOR ALL ENTRIES IN lt\_repl\_procty  
                  WHERE lgnum = iv\_lgnum  
                  AND hdr\_procty = lt\_repl\_procty-hdr\_procty  
                  AND status = wmegc\_wo\_open  
                  AND queue = iv\_queue.  
\*            ENDIF .  
  
                SELECT SINGLE zvalue INTO lv\_repl FROM zcon\_param\_val WHERE lgnum = iv\_lgnum AND zreq\_name = 'REPL\_DET2\_REQ' AND zparam = 'REPL\_DET2\_PARAM' AND zactive = abap\_true.  
                IF sy-subrc = 0.  
                  IF lt\_repl\_who[] IS NOT INITIAL.  
                    lt\_rtv\_who1[] = lt\_rtv\_who[].  
                    lt\_rtv\_who[] = lt\_repl\_who[].  
                    APPEND LINES OF lt\_rtv\_who1 TO lt\_rtv\_who.  
                  ENDIF.  
                ENDIF.  
                lv\_str\_cnt = lines( lt\_str\_who ) .  
                lv\_rtv\_cnt = lines( lt\_rtv\_who ).  
  
                MOVE iv\_lgnum  TO ls\_log-extnumber.  
                CALL METHOD lo\_log->create\_log  
                  EXPORTING  
                    is\_log = ls\_log.  
  
                MESSAGE s033(zapl\_det) WITH 'Interleaving block started'  INTO lv\_msgtext.  
                lo\_log->add\_message2log( ).  
  
\*\*\*\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*When we have only putaway tasks  
                IF lt\_str\_who[] IS  NOT INITIAL AND lt\_rtv\_who[] IS INITIAL.  
                  READ TABLE lt\_str\_who INTO ls\_str\_who WITH KEY hdr\_procty = '3090'.  
                  IF sy-subrc = 0.  
                    SELECT \* INTO TABLE lt\_interleave FROM zinter\_leave  
                             WHERE lgnum = iv\_lgnum AND resource\_name = iv\_queue AND active EQ abap\_true.  
                    IF sy-subrc NE 0.  
                      INSERT ls\_str\_who INTO ct\_who INDEX 1.  
  
                      SELECT \*  INTO TABLE lt\_inter\_who FROM zinter\_leave  
                                WHERE lgnum = iv\_lgnum AND resource\_name = iv\_queue .  
                      IF sy-subrc = 0.  
                        READ TABLE lt\_inter\_who INTO ls\_inter\_who INDEX 1.  
                        IF ls\_str\_who-who NE ls\_inter\_who-who\_no.  
                          UPDATE zinter\_leave SET active = abap\_false  
                                                  who\_no = ls\_str\_who-who  
                                              WHERE lgnum EQ iv\_lgnum  
                                              AND resource\_name = iv\_queue.  
                          COMMIT WORK.  
                          lv\_missing\_case = 2.  
                        ENDIF.  
                      ENDIF.  
                      RETURN.  
                    ENDIF.  
  
                  ENDIF.  
  
                ENDIF.  
\*\*\*\*\*\*\*\*\*\* end of when we have only putaway tasks  
  
\*\*\*\*\*\*\*\*\*\*\* When we have only retrival tasks  
  
                SELECT SINGLE zactive INTO lv\_active1 FROM zcon\_param\_val  
                                      WHERE lgnum = iv\_lgnum  
                                      AND zreq\_name = 'ZINTER'  
                                      AND zactive EQ abap\_true.  
                IF lv\_active1 EQ abap\_true.  
                  IF lt\_str\_who[] IS INITIAL AND lt\_rtv\_who[] IS NOT INITIAL.  
  
                    READ TABLE lt\_rtv\_who INTO ls\_rtv\_who WITH KEY hdr\_procty = '3010'.  
                    IF sy-subrc = 0.  
  
                      INSERT ls\_rtv\_who INTO ct\_who INDEX 1.  
                      READ TABLE ct\_who INTO cs\_who INDEX 1.  
                      iv\_who = cs\_who-who.  
                      ls\_det2\_fg-lgnum = iv\_lgnum.  
                      ls\_det2\_fg-crane\_no = lv\_crane\_no.  
                      ls\_det2\_fg-error\_no = ls\_final\_error-error.  
                      ls\_det2\_fg-who = iv\_who.  
                      ls\_det2\_fg-description = 'checking for picking task need rewarehousing repl '.  
                      ls\_det2\_fg-log\_date = sy-datum.  
                      ls\_det2\_fg-log\_time = sy-uzeit.  
                      APPEND ls\_det2\_fg TO lt\_det2\_fg.  
                      MODIFY zdet2\_fg\_log FROM TABLE lt\_det2\_fg.  
                      COMMIT WORK.  
                      lv\_missing\_case = 2.  
                      CALL METHOD lo\_badi\_transfer->transfer\_hu1  
                        EXPORTING  
                          iv\_lgnum   = iv\_lgnum  
                          is\_whorder = is\_who  
                          iv\_who     = iv\_who  
                          iv\_rsrc    = is\_rsrc  
                          it\_ordim\_o = ct\_ordim\_o  
                        CHANGING  
                          ev\_who     = lv\_who.  
  
                      CLEAR ls\_who.  
                      IF lv\_who+0(1) NE 0.  
                        IF iv\_who NE lv\_who.  
                          "if current WO already contains WT for HU in front of HU to be picked => do nothing  
                          ls\_who-who = lv\_who.  
                          INSERT ls\_who INTO ct\_who INDEX 1.  
                          ls\_det2\_fg-lgnum = iv\_lgnum.  
                          ls\_det2\_fg-crane\_no = lv\_crane\_no.  
                          ls\_det2\_fg-who = lv\_who.  
                          ls\_det2\_fg-description = 'Rewarehousing task sent to subsystem Picking task repl'.  
                          ls\_det2\_fg-log\_date = sy-datum.  
                          ls\_det2\_fg-log\_time = sy-uzeit.  
                          APPEND ls\_det2\_fg TO lt\_det2\_fg.  
                          MODIFY zdet2\_fg\_log FROM TABLE lt\_det2\_fg.  
                          COMMIT WORK.  
  
                          UPDATE zinter\_leave SET active = abap\_true  
                                                  who\_no = ls\_who-who  
                                              WHERE lgnum EQ iv\_lgnum  
                                              AND resource\_name = iv\_queue.  
                          COMMIT WORK.  
                        ENDIF.  
                        cv\_rearr\_done = abap\_true.  
                        RETURN.  
                      ELSE.  
                        SELECT \*  INTO TABLE lt\_inter\_who FROM zinter\_leave  
                                   WHERE lgnum = iv\_lgnum  
                                   AND resource\_name = iv\_queue .  
                        IF sy-subrc = 0.  
                          LOOP AT lt\_inter\_who INTO ls\_inter\_who.  
                            READ TABLE ct\_who INTO cs\_who WITH KEY who = ls\_inter\_who-who\_no.  
                            IF sy-subrc = 0.  
                              "do nothing  
                            ELSE.  
                              UPDATE zinter\_leave SET active = abap\_false  
                                                      who\_no = ls\_rtv\_who-who  
                                                  WHERE lgnum EQ iv\_lgnum  
                                                  AND resource\_name = iv\_queue.  
                              COMMIT WORK.  
                            ENDIF.  
                          ENDLOOP.  
                        ENDIF.  
                        RETURN.  
  
  
                      ENDIF.  
  
                      SELECT \*  INTO TABLE lt\_inter\_who FROM zinter\_leave  
                                WHERE lgnum = iv\_lgnum AND resource\_name = iv\_queue .  
                      IF sy-subrc = 0.  
                        LOOP AT lt\_inter\_who INTO ls\_inter\_who.  
                          READ TABLE ct\_who INTO cs\_who WITH KEY who = ls\_inter\_who-who\_no.  
                          IF sy-subrc = 0.  
                            "do nothing  
                          ELSE.  
                            UPDATE zinter\_leave SET active = abap\_true  
                                                    who\_no = ls\_rtv\_who-who  
                                                WHERE lgnum EQ iv\_lgnum  
                                                AND resource\_name = iv\_queue.  
                            COMMIT WORK.  
                          ENDIF.  
                        ENDLOOP.  
                      ENDIF.  
                      RETURN.  
                    ENDIF.  
  
                    READ TABLE lt\_rtv\_who INTO ls\_rtv\_who WITH KEY hdr\_procty = '2010'.  
                    IF sy-subrc = 0.  
  
                      INSERT ls\_rtv\_who INTO ct\_who INDEX 1.  
                      READ TABLE ct\_who INTO cs\_who INDEX 1.  
                      iv\_who = cs\_who-who.  
                      ls\_det2\_fg-lgnum = iv\_lgnum.  
                      ls\_det2\_fg-crane\_no = lv\_crane\_no.  
                      ls\_det2\_fg-error\_no = ls\_final\_error-error.  
                      ls\_det2\_fg-who = iv\_who.  
                      ls\_det2\_fg-description = 'checking for picking task need rewarehousing '.  
                      ls\_det2\_fg-log\_date = sy-datum.  
                      ls\_det2\_fg-log\_time = sy-uzeit.  
                      APPEND ls\_det2\_fg TO lt\_det2\_fg.  
                      MODIFY zdet2\_fg\_log FROM TABLE lt\_det2\_fg.  
                      COMMIT WORK.  
                      lv\_missing\_case = 2.  
                      CALL METHOD lo\_badi\_transfer->transfer\_hu1  
                        EXPORTING  
                          iv\_lgnum   = iv\_lgnum  
                          is\_whorder = is\_who  
                          iv\_who     = iv\_who  
                          iv\_rsrc    = is\_rsrc  
                          it\_ordim\_o = ct\_ordim\_o  
                        CHANGING  
                          ev\_who     = lv\_who.  
  
                      CLEAR ls\_who.  
                      IF lv\_who+0(1) NE 0.  
                        IF iv\_who NE lv\_who.  
                          "if current WO already contains WT for HU in front of HU to be picked => do nothing  
                          ls\_who-who = lv\_who.  
                          INSERT ls\_who INTO ct\_who INDEX 1.  
                          ls\_det2\_fg-lgnum = iv\_lgnum.  
                          ls\_det2\_fg-crane\_no = lv\_crane\_no.  
                          ls\_det2\_fg-who = lv\_who.  
                          ls\_det2\_fg-description = 'Rewarehousing task sent to subsystem Picking task'.  
                          ls\_det2\_fg-log\_date = sy-datum.  
                          ls\_det2\_fg-log\_time = sy-uzeit.  
                          APPEND ls\_det2\_fg TO lt\_det2\_fg.  
                          MODIFY zdet2\_fg\_log FROM TABLE lt\_det2\_fg.  
                          COMMIT WORK.  
  
                          UPDATE zinter\_leave SET active = abap\_true  
                                                  who\_no = ls\_who-who  
                                              WHERE lgnum EQ iv\_lgnum  
                                              AND resource\_name = iv\_queue.  
                          COMMIT WORK.  
                        ENDIF.  
                        cv\_rearr\_done = abap\_true.  
                        RETURN.  
                      ELSE.  
                        SELECT \*  INTO TABLE lt\_inter\_who FROM zinter\_leave  
                                  WHERE lgnum = iv\_lgnum AND resource\_name = iv\_queue .  
                        IF sy-subrc = 0.  
                          LOOP AT lt\_inter\_who INTO ls\_inter\_who.  
                            READ TABLE ct\_who INTO cs\_who WITH KEY who = ls\_inter\_who-who\_no.  
                            IF sy-subrc = 0.  
                              "do nothing  
                            ELSE.  
                              UPDATE zinter\_leave SET active = abap\_false  
                                                      who\_no = ls\_rtv\_who-who  
                                                  WHERE lgnum EQ iv\_lgnum  
                                                  AND resource\_name = iv\_queue.  
                              COMMIT WORK.  
                            ENDIF.  
                          ENDLOOP.  
                        ENDIF.  
                        RETURN.  
                      ENDIF.  
  
                      SELECT \*  INTO TABLE lt\_inter\_who FROM zinter\_leave  
                                WHERE lgnum = iv\_lgnum AND resource\_name = iv\_queue .  
                      IF sy-subrc = 0.  
                        LOOP AT lt\_inter\_who INTO ls\_inter\_who.  
                          READ TABLE ct\_who INTO cs\_who WITH KEY who = ls\_inter\_who-who\_no.  
                          IF sy-subrc = 0.  
                            "do nothing  
                          ELSE.  
                            UPDATE zinter\_leave SET active = abap\_true  
                                                    who\_no = ls\_rtv\_who-who  
                                                WHERE lgnum EQ iv\_lgnum  
                                                AND resource\_name = iv\_queue.  
                            COMMIT WORK.  
                          ENDIF.  
                        ENDLOOP.  
                      ENDIF.  
                      RETURN.  
                    ENDIF.  
                  ENDIF.  
                ELSE.  
                  IF lt\_str\_who[] IS INITIAL AND lt\_rtv\_who[] IS NOT INITIAL.  
  
                    READ TABLE lt\_rtv\_who INTO ls\_rtv\_who WITH KEY hdr\_procty = '3010'.  
                    IF sy-subrc = 0.  
                      SELECT \* INTO TABLE lt\_interleave FROM zinter\_leave  
                               WHERE lgnum = iv\_lgnum  
                               AND resource\_name = iv\_queue  
                               AND active EQ abap\_true.  
                      IF sy-subrc = 0.  
  
                        READ TABLE lt\_rtv\_who INTO ls\_rtv\_who WITH KEY hdr\_procty = '3010'.  
                        IF sy-subrc = 0.  
                          INSERT ls\_rtv\_who INTO ct\_who INDEX 1.  
                          READ TABLE ct\_who INTO cs\_who INDEX 1.  
                          iv\_who = cs\_who-who.  
                          ls\_det2\_fg-lgnum = iv\_lgnum.  
                          ls\_det2\_fg-crane\_no = lv\_crane\_no.  
                          ls\_det2\_fg-error\_no = ls\_final\_error-error.  
                          ls\_det2\_fg-who = iv\_who.  
                          ls\_det2\_fg-description = 'checking for picking task need rewarehousing repl '.  
                          ls\_det2\_fg-log\_date = sy-datum.  
                          ls\_det2\_fg-log\_time = sy-uzeit.  
                          APPEND ls\_det2\_fg TO lt\_det2\_fg.  
                          MODIFY zdet2\_fg\_log FROM TABLE lt\_det2\_fg.  
                          COMMIT WORK.  
                          lv\_missing\_case = 2.  
                          CALL METHOD lo\_badi\_transfer->transfer\_hu1  
                            EXPORTING  
                              iv\_lgnum   = iv\_lgnum  
                              is\_whorder = is\_who  
                              iv\_who     = iv\_who  
                              iv\_rsrc    = is\_rsrc  
                              it\_ordim\_o = ct\_ordim\_o  
                            CHANGING  
                              ev\_who     = lv\_who.  
  
                          CLEAR ls\_who.  
                          IF lv\_who+0(1) NE 0.  
                            IF iv\_who NE lv\_who.  
                              "if current WO already contains WT for HU in front of HU to be picked => do nothing  
                              ls\_who-who = lv\_who.  
                              INSERT ls\_who INTO ct\_who INDEX 1.  
                              ls\_det2\_fg-lgnum = iv\_lgnum.  
                              ls\_det2\_fg-crane\_no = lv\_crane\_no.  
                              ls\_det2\_fg-who = lv\_who.  
                              ls\_det2\_fg-description = 'Rewarehousing task sent to subsystem Picking task repl'.  
                              ls\_det2\_fg-log\_date = sy-datum.  
                              ls\_det2\_fg-log\_time = sy-uzeit.  
                              APPEND ls\_det2\_fg TO lt\_det2\_fg.  
                              MODIFY zdet2\_fg\_log FROM TABLE lt\_det2\_fg.  
                              COMMIT WORK.  
  
                              UPDATE zinter\_leave SET active = abap\_true  
                                                      who\_no = ls\_who-who  
                                                  WHERE lgnum EQ iv\_lgnum  
                                                  AND resource\_name = iv\_queue.  
                              COMMIT WORK.  
                            ENDIF.  
                            cv\_rearr\_done = abap\_true.  
                            RETURN.  
                          ELSE.  
                            SELECT \*  INTO TABLE lt\_inter\_who FROM zinter\_leave  
                                      WHERE lgnum = iv\_lgnum AND resource\_name = iv\_queue .  
                            IF sy-subrc = 0.  
                              LOOP AT lt\_inter\_who INTO ls\_inter\_who.  
                                READ TABLE ct\_who INTO cs\_who WITH KEY who = ls\_inter\_who-who\_no.  
                                IF sy-subrc = 0.  
                                  "do nothing  
                                ELSE.  
                                  UPDATE zinter\_leave SET active = abap\_false  
                                                          who\_no = ls\_rtv\_who-who  
                                                      WHERE lgnum EQ iv\_lgnum AND resource\_name = iv\_queue.  
                                  COMMIT WORK.  
                                ENDIF.  
                              ENDLOOP.  
                            ENDIF.  
                            RETURN.  
  
  
                          ENDIF.  
  
                          SELECT \*  INTO TABLE lt\_inter\_who FROM zinter\_leave  
                                    WHERE lgnum = iv\_lgnum AND resource\_name = iv\_queue .  
                          IF sy-subrc = 0.  
                            LOOP AT lt\_inter\_who INTO ls\_inter\_who.  
                              READ TABLE ct\_who INTO cs\_who WITH KEY who = ls\_inter\_who-who\_no.  
                              IF sy-subrc = 0.  
                                "do nothing  
                              ELSE.  
                                UPDATE zinter\_leave SET active = abap\_true  
                                                        who\_no = ls\_rtv\_who-who  
                                                    WHERE lgnum EQ iv\_lgnum  
                                                    AND resource\_name = iv\_queue.  
                                COMMIT WORK.  
                              ENDIF.  
                            ENDLOOP.  
                          ENDIF.  
                          RETURN.  
                        ENDIF."sy-subrc = 0 end  
  
                      ENDIF.  
                    ENDIF.  
  
                    READ TABLE lt\_rtv\_who INTO ls\_rtv\_who WITH KEY hdr\_procty = '2010'.  
                    IF sy-subrc = 0.  
                      SELECT \* INTO TABLE lt\_interleave FROM zinter\_leave  
                               WHERE lgnum = iv\_lgnum AND resource\_name = iv\_queue AND active EQ abap\_true.  
                      IF sy-subrc = 0.  
  
                        READ TABLE lt\_rtv\_who INTO ls\_rtv\_who WITH KEY hdr\_procty = '2010'.  
                        IF sy-subrc = 0.  
                          INSERT ls\_rtv\_who INTO ct\_who INDEX 1.  
                          READ TABLE ct\_who INTO cs\_who INDEX 1.  
                          iv\_who = cs\_who-who.  
                          ls\_det2\_fg-lgnum = iv\_lgnum.  
                          ls\_det2\_fg-crane\_no = lv\_crane\_no.  
                          ls\_det2\_fg-error\_no = ls\_final\_error-error.  
                          ls\_det2\_fg-who = iv\_who.  
                          ls\_det2\_fg-description = 'checking for picking task need rewarehousing '.  
                          ls\_det2\_fg-log\_date = sy-datum.  
                          ls\_det2\_fg-log\_time = sy-uzeit.  
                          APPEND ls\_det2\_fg TO lt\_det2\_fg.  
                          MODIFY zdet2\_fg\_log FROM TABLE lt\_det2\_fg.  
                          COMMIT WORK.  
                          lv\_missing\_case = 2.  
                          CALL METHOD lo\_badi\_transfer->transfer\_hu1  
                            EXPORTING  
                              iv\_lgnum   = iv\_lgnum  
                              is\_whorder = is\_who  
                              iv\_who     = iv\_who  
                              iv\_rsrc    = is\_rsrc  
                              it\_ordim\_o = ct\_ordim\_o  
                            CHANGING  
                              ev\_who     = lv\_who.  
  
                          CLEAR ls\_who.  
                          IF lv\_who+0(1) NE 0.  
                            IF iv\_who NE lv\_who.  
                              "if current WO already contains WT for HU in front of HU to be picked => do nothing  
                              ls\_who-who = lv\_who.  
                              INSERT ls\_who INTO ct\_who INDEX 1.  
                              ls\_det2\_fg-lgnum = iv\_lgnum.  
                              ls\_det2\_fg-crane\_no = lv\_crane\_no.  
                              ls\_det2\_fg-who = lv\_who.  
                              ls\_det2\_fg-description = 'Rewarehousing task sent to subsystem Picking task'.  
                              ls\_det2\_fg-log\_date = sy-datum.  
                              ls\_det2\_fg-log\_time = sy-uzeit.  
                              APPEND ls\_det2\_fg TO lt\_det2\_fg.  
                              MODIFY zdet2\_fg\_log FROM TABLE lt\_det2\_fg.  
                              COMMIT WORK.  
  
                              UPDATE zinter\_leave SET active = abap\_true  
                                                      who\_no = ls\_who-who  
                                                  WHERE lgnum EQ iv\_lgnum  
                                                  AND resource\_name = iv\_queue.  
                              COMMIT WORK.  
                            ENDIF.  
                            cv\_rearr\_done = abap\_true.  
                            RETURN.  
                          ELSE.  
                            SELECT \*  INTO TABLE lt\_inter\_who FROM zinter\_leave  
                                      WHERE lgnum = iv\_lgnum AND resource\_name = iv\_queue .  
                            IF sy-subrc = 0.  
                              LOOP AT lt\_inter\_who INTO ls\_inter\_who.  
                                READ TABLE ct\_who INTO cs\_who WITH KEY who = ls\_inter\_who-who\_no.  
                                IF sy-subrc = 0.  
                                  "do nothing  
                                ELSE.  
                                  UPDATE zinter\_leave SET active = abap\_false  
                                                          who\_no = ls\_rtv\_who-who  
                                                      WHERE lgnum EQ iv\_lgnum  
                                                      AND resource\_name = iv\_queue.  
                                  COMMIT WORK.  
                                ENDIF.  
                              ENDLOOP.  
                            ENDIF.  
                            RETURN.  
                          ENDIF.  
  
                          SELECT \*  INTO TABLE lt\_inter\_who FROM zinter\_leave  
                                    WHERE lgnum = iv\_lgnum AND resource\_name = iv\_queue .  
                          IF sy-subrc = 0.  
                            LOOP AT lt\_inter\_who INTO ls\_inter\_who.  
                              READ TABLE ct\_who INTO cs\_who WITH KEY who = ls\_inter\_who-who\_no.  
                              IF sy-subrc = 0.  
                                "do nothing  
                              ELSE.  
                                UPDATE zinter\_leave SET active = abap\_true  
                                                        who\_no = ls\_rtv\_who-who  
                                                    WHERE lgnum EQ iv\_lgnum  
                                                    AND resource\_name = iv\_queue.  
                                COMMIT WORK.  
                              ENDIF.  
                            ENDLOOP.  
                          ENDIF.  
                          RETURN.  
                        ENDIF."sy-subrc = 0 end  
  
                      ENDIF.  
  
                    ENDIF.  
                  ENDIF.  
                ENDIF.  
\*\*\*\*\*\*\*\*\*\* end of when we have retrival tasks  
  
                READ TABLE lt\_str\_who INTO ls\_str\_who WITH KEY hdr\_procty = '3090'.  
                IF sy-subrc = 0.  
                  SELECT \* INTO TABLE lt\_interleave FROM zinter\_leave  
                           WHERE lgnum = iv\_lgnum  
                           AND resource\_name = iv\_queue  
                           AND active EQ abap\_true.  
                  IF sy-subrc NE 0.  
                    INSERT ls\_str\_who INTO ct\_who INDEX 1.  
  
                    SELECT \*  INTO TABLE lt\_inter\_who FROM zinter\_leave  
                              WHERE lgnum = iv\_lgnum AND resource\_name = iv\_queue .  
                    IF sy-subrc = 0.  
                      READ TABLE lt\_inter\_who INTO ls\_inter\_who INDEX 1.  
                      IF ls\_str\_who-who NE ls\_inter\_who-who\_no.  
                        UPDATE zinter\_leave SET active = abap\_true  
                                                who\_no = ls\_str\_who-who  
                                            WHERE lgnum EQ iv\_lgnum  
                                            AND resource\_name = iv\_queue.  
                        COMMIT WORK.  
                        lv\_missing\_case = 2.  
                      ENDIF.  
                    ENDIF.  
                    COMMIT WORK.  
                    RETURN.  
                  ELSEIF sy-subrc = 0.  
                    READ TABLE lt\_rtv\_who INTO ls\_rtv\_who WITH KEY hdr\_procty = '3010'.  
                    IF sy-subrc = 0.  
                      INSERT ls\_rtv\_who INTO ct\_who INDEX 1.  
                      READ TABLE ct\_who INTO cs\_who INDEX 1.  
                      iv\_who = cs\_who-who.  
                      ls\_det2\_fg-lgnum = iv\_lgnum.  
                      ls\_det2\_fg-crane\_no = lv\_crane\_no.  
                      ls\_det2\_fg-error\_no = ls\_final\_error-error.  
                      ls\_det2\_fg-who = iv\_who.  
                      ls\_det2\_fg-description = 'checking for putaway task need rewarehousing for picking repl '.  
                      ls\_det2\_fg-log\_date = sy-datum.  
                      ls\_det2\_fg-log\_time = sy-uzeit.  
                      APPEND ls\_det2\_fg TO lt\_det2\_fg.  
                      MODIFY zdet2\_fg\_log FROM TABLE lt\_det2\_fg.  
                      COMMIT WORK.  
                      lv\_missing\_case = 2.  
                      CALL METHOD lo\_badi\_transfer->transfer\_hu1  
                        EXPORTING  
                          iv\_lgnum   = iv\_lgnum  
                          is\_whorder = is\_who  
                          iv\_who     = iv\_who  
                          iv\_rsrc    = is\_rsrc  
                          it\_ordim\_o = ct\_ordim\_o  
                        CHANGING  
                          ev\_who     = lv\_who.  
  
                      CLEAR ls\_who.  
                      IF lv\_who+0(1) NE 0.  
                        IF iv\_who NE lv\_who.  
                          "if current WO already contains WT for HU in front of HU to be picked => do nothing  
                          ls\_who-who = lv\_who.  
                          INSERT ls\_who INTO ct\_who INDEX 1.  
                          ls\_det2\_fg-lgnum = iv\_lgnum.  
                          ls\_det2\_fg-crane\_no = lv\_crane\_no.  
                          ls\_det2\_fg-who = lv\_who.  
                          ls\_det2\_fg-description = 'Rewarehousing task sent to subsystem for picking repl'.  
                          ls\_det2\_fg-log\_date = sy-datum.  
                          ls\_det2\_fg-log\_time = sy-uzeit.  
                          APPEND ls\_det2\_fg TO lt\_det2\_fg.  
                          MODIFY zdet2\_fg\_log FROM TABLE lt\_det2\_fg.  
                          COMMIT WORK.  
                          UPDATE zinter\_leave SET active = abap\_true  
                                                  who\_no = ls\_who-who  
                                              WHERE lgnum EQ iv\_lgnum  
                                              AND resource\_name = iv\_queue.  
                          COMMIT WORK.  
                        ENDIF.  
                        cv\_rearr\_done = abap\_true.  
                        RETURN.  
                      ELSE.  
                        SELECT \*  INTO TABLE lt\_inter\_who FROM zinter\_leave  
                                  WHERE lgnum = iv\_lgnum AND resource\_name = iv\_queue .  
                        IF sy-subrc = 0.  
                          LOOP AT lt\_inter\_who INTO ls\_inter\_who.  
                            READ TABLE ct\_who INTO cs\_who WITH KEY who = ls\_inter\_who-who\_no.  
                            IF sy-subrc = 0.  
                              "do nothing  
                            ELSE.  
                              UPDATE zinter\_leave SET active = abap\_false  
                                                      who\_no = ls\_rtv\_who-who  
                                                  WHERE lgnum EQ iv\_lgnum  
                                                  AND resource\_name = iv\_queue.  
                              COMMIT WORK.  
                            ENDIF.  
                          ENDLOOP.  
                        ENDIF.  
                        RETURN.  
                      ENDIF.  
  
  
  
                      SELECT \*  INTO TABLE lt\_inter\_who FROM zinter\_leave  
                                WHERE lgnum = iv\_lgnum AND resource\_name = iv\_queue .  
                      IF sy-subrc = 0.  
                        LOOP AT lt\_inter\_who INTO ls\_inter\_who.  
                          READ TABLE ct\_who INTO cs\_who WITH KEY who = ls\_inter\_who-who\_no.  
                          IF sy-subrc = 0.  
                            "do nothing  
                          ELSE.  
                            UPDATE zinter\_leave SET active = abap\_false  
                                                    who\_no = ls\_rtv\_who-who  
                                                WHERE lgnum EQ iv\_lgnum  
                                                AND resource\_name = iv\_queue.  
                            COMMIT WORK.  
                          ENDIF.  
                        ENDLOOP.  
                      ENDIF.  
                      RETURN.  
                    ELSE."3010 task is not present  
                      READ TABLE lt\_rtv\_who INTO ls\_rtv\_who WITH KEY hdr\_procty = '2010'.  
                      IF sy-subrc = 0.  
                        INSERT ls\_rtv\_who INTO ct\_who INDEX 1.  
                        READ TABLE ct\_who INTO cs\_who INDEX 1.  
                        iv\_who = cs\_who-who.  
                        ls\_det2\_fg-lgnum = iv\_lgnum.  
                        ls\_det2\_fg-crane\_no = lv\_crane\_no.  
                        ls\_det2\_fg-error\_no = ls\_final\_error-error.  
                        ls\_det2\_fg-who = iv\_who.  
                        ls\_det2\_fg-description = 'checking for putaway task need rewarehousing for picking '.  
                        ls\_det2\_fg-log\_date = sy-datum.  
                        ls\_det2\_fg-log\_time = sy-uzeit.  
                        APPEND ls\_det2\_fg TO lt\_det2\_fg.  
                        MODIFY zdet2\_fg\_log FROM TABLE lt\_det2\_fg.  
                        COMMIT WORK.  
                        lv\_missing\_case = 2.  
                        CALL METHOD lo\_badi\_transfer->transfer\_hu1  
                          EXPORTING  
                            iv\_lgnum   = iv\_lgnum  
                            is\_whorder = is\_who  
                            iv\_who     = iv\_who  
                            iv\_rsrc    = is\_rsrc  
                            it\_ordim\_o = ct\_ordim\_o  
                          CHANGING  
                            ev\_who     = lv\_who.  
  
                        CLEAR ls\_who.  
                        IF lv\_who+0(1) NE 0.  
                          IF iv\_who NE lv\_who.  
                            "if current WO already contains WT for HU in front of HU to be picked => do nothing  
                            ls\_who-who = lv\_who.  
                            INSERT ls\_who INTO ct\_who INDEX 1.  
                            ls\_det2\_fg-lgnum = iv\_lgnum.  
                            ls\_det2\_fg-crane\_no = lv\_crane\_no.  
                            ls\_det2\_fg-who = lv\_who.  
                            ls\_det2\_fg-description = 'Rewarehousing task sent to subsystem for picking'.  
                            ls\_det2\_fg-log\_date = sy-datum.  
                            ls\_det2\_fg-log\_time = sy-uzeit.  
                            APPEND ls\_det2\_fg TO lt\_det2\_fg.  
                            MODIFY zdet2\_fg\_log FROM TABLE lt\_det2\_fg.  
                            COMMIT WORK.  
  
  
                            UPDATE zinter\_leave SET active = abap\_true  
                                                    who\_no = ls\_who-who  
                                                WHERE lgnum EQ iv\_lgnum  
                                                AND resource\_name = iv\_queue.  
                            COMMIT WORK.  
                          ENDIF.  
                          cv\_rearr\_done = abap\_true.  
                          RETURN.  
                        ELSE.  
                          SELECT \*  INTO TABLE lt\_inter\_who FROM zinter\_leave  
                                    WHERE lgnum = iv\_lgnum AND resource\_name = iv\_queue .  
                          IF sy-subrc = 0.  
                            LOOP AT lt\_inter\_who INTO ls\_inter\_who.  
                              READ TABLE ct\_who INTO cs\_who WITH KEY who = ls\_inter\_who-who\_no.  
                              IF sy-subrc = 0.  
                                "do nothing  
                              ELSE.  
                                UPDATE zinter\_leave SET active = abap\_false  
                                                        who\_no = ls\_rtv\_who-who  
                                                    WHERE lgnum EQ iv\_lgnum  
                                                    AND resource\_name = iv\_queue.  
                                COMMIT WORK.  
                              ENDIF.  
                            ENDLOOP.  
                          ENDIF.  
                          RETURN.  
                        ENDIF.  
  
  
                        SELECT \*  INTO TABLE lt\_inter\_who FROM zinter\_leave  
                                  WHERE lgnum = iv\_lgnum AND resource\_name = iv\_queue .  
                        IF sy-subrc = 0.  
                          LOOP AT lt\_inter\_who INTO ls\_inter\_who.  
                            READ TABLE ct\_who INTO cs\_who WITH KEY who = ls\_inter\_who-who\_no.  
                            IF sy-subrc = 0.  
                              "do nothing  
                            ELSE.  
                              UPDATE zinter\_leave SET active = abap\_false  
                                                      who\_no = ls\_rtv\_who-who  
                                                  WHERE lgnum EQ iv\_lgnum  
                                                  AND resource\_name = iv\_queue.  
                              COMMIT WORK.  
                            ENDIF.  
                          ENDLOOP.  
                        ENDIF.  
  
                        RETURN.  
                      ENDIF."sy-subrc = 0 end  
                    ENDIF.  
                  ENDIF.  
                ELSE." lt\_str\_who is initial  
                  READ TABLE lt\_rtv\_who INTO ls\_rtv\_who WITH KEY hdr\_procty = '3010'.  
                  IF sy-subrc = 0.  
                    INSERT ls\_rtv\_who INTO ct\_who INDEX 1.  
                    READ TABLE ct\_who INTO cs\_who INDEX 1.  
                    iv\_who = cs\_who-who.  
                    ls\_det2\_fg-lgnum = iv\_lgnum.  
                    ls\_det2\_fg-crane\_no = lv\_crane\_no.  
                    ls\_det2\_fg-error\_no = ls\_final\_error-error.  
                    ls\_det2\_fg-who = iv\_who.  
                    ls\_det2\_fg-description = 'checking for putaway task need rewarehousing for picking repl '.  
                    ls\_det2\_fg-log\_date = sy-datum.  
                    ls\_det2\_fg-log\_time = sy-uzeit.  
                    APPEND ls\_det2\_fg TO lt\_det2\_fg.  
                    MODIFY zdet2\_fg\_log FROM TABLE lt\_det2\_fg.  
                    COMMIT WORK.  
                    lv\_missing\_case = 2.  
                    CALL METHOD lo\_badi\_transfer->transfer\_hu1  
                      EXPORTING  
                        iv\_lgnum   = iv\_lgnum  
                        is\_whorder = is\_who  
                        iv\_who     = iv\_who  
                        iv\_rsrc    = is\_rsrc  
                        it\_ordim\_o = ct\_ordim\_o  
                      CHANGING  
                        ev\_who     = lv\_who.  
  
                    CLEAR ls\_who.  
                    IF lv\_who+0(1) NE 0.  
                      IF iv\_who NE lv\_who.  
                        "if current WO already contains WT for HU in front of HU to be picked => do nothing  
                        ls\_who-who = lv\_who.  
                        INSERT ls\_who INTO ct\_who INDEX 1.  
                        ls\_det2\_fg-lgnum = iv\_lgnum.  
                        ls\_det2\_fg-crane\_no = lv\_crane\_no.  
                        ls\_det2\_fg-who = lv\_who.  
                        ls\_det2\_fg-description = 'Rewarehousing task sent to subsystem for picking repl'.  
                        ls\_det2\_fg-log\_date = sy-datum.  
                        ls\_det2\_fg-log\_time = sy-uzeit.  
                        APPEND ls\_det2\_fg TO lt\_det2\_fg.  
                        MODIFY zdet2\_fg\_log FROM TABLE lt\_det2\_fg.  
                        COMMIT WORK.  
  
  
                        UPDATE zinter\_leave SET active = abap\_true  
                                                who\_no = ls\_who-who  
                                            WHERE lgnum EQ iv\_lgnum  
                                            AND resource\_name = iv\_queue.  
                        COMMIT WORK.  
  
  
                      ENDIF.  
                      cv\_rearr\_done = abap\_true.  
                      RETURN.  
                    ELSE.  
                      SELECT \*  INTO TABLE lt\_inter\_who FROM zinter\_leave  
                                WHERE lgnum = iv\_lgnum AND resource\_name = iv\_queue .  
                      IF sy-subrc = 0.  
                        LOOP AT lt\_inter\_who INTO ls\_inter\_who.  
                          READ TABLE ct\_who INTO cs\_who WITH KEY who = ls\_inter\_who-who\_no.  
                          IF sy-subrc = 0.  
                            "do nothing  
                          ELSE.  
                            UPDATE zinter\_leave SET active = abap\_false  
                                                    who\_no = ls\_rtv\_who-who  
                                                WHERE lgnum EQ iv\_lgnum  
                                                AND resource\_name = iv\_queue.  
                            COMMIT WORK.  
                          ENDIF.  
                        ENDLOOP.  
                      ENDIF.  
                      RETURN.  
                    ENDIF.  
  
                    SELECT \*  INTO TABLE lt\_inter\_who FROM zinter\_leave  
                              WHERE lgnum = iv\_lgnum AND resource\_name = iv\_queue .  
                    IF sy-subrc = 0.  
                      LOOP AT lt\_inter\_who INTO ls\_inter\_who.  
                        READ TABLE ct\_who INTO cs\_who WITH KEY who = ls\_inter\_who-who\_no.  
                        IF sy-subrc = 0.  
                          "do nothing  
                        ELSE.  
                          UPDATE zinter\_leave SET active = abap\_false  
                                                  who\_no = ls\_rtv\_who-who  
                                              WHERE lgnum EQ iv\_lgnum  
                                              AND resource\_name = iv\_queue.  
                          COMMIT WORK.  
                        ENDIF.  
                      ENDLOOP.  
                    ENDIF.  
  
                    RETURN.  
                  ELSE."3010 task is not present when lt\_str\_who is initial  
                    READ TABLE lt\_rtv\_who INTO ls\_rtv\_who WITH KEY hdr\_procty = '2010'.  
                    IF sy-subrc = 0.  
                      INSERT ls\_rtv\_who INTO ct\_who INDEX 1.  
                      READ TABLE ct\_who INTO cs\_who INDEX 1.  
                      iv\_who = cs\_who-who.  
                      ls\_det2\_fg-lgnum = iv\_lgnum.  
                      ls\_det2\_fg-crane\_no = lv\_crane\_no.  
                      ls\_det2\_fg-error\_no = ls\_final\_error-error.  
                      ls\_det2\_fg-who = iv\_who.  
                      ls\_det2\_fg-description = 'checking for putaway task need rewarehousing for picking  '.  
                      ls\_det2\_fg-log\_date = sy-datum.  
                      ls\_det2\_fg-log\_time = sy-uzeit.  
  
                      APPEND ls\_det2\_fg TO lt\_det2\_fg.  
                      MODIFY zdet2\_fg\_log FROM TABLE lt\_det2\_fg.  
                      COMMIT WORK.  
                      CALL METHOD lo\_badi\_transfer->transfer\_hu1  
                        EXPORTING  
                          iv\_lgnum   = iv\_lgnum  
                          is\_whorder = is\_who  
                          iv\_who     = iv\_who  
                          iv\_rsrc    = is\_rsrc  
                          it\_ordim\_o = ct\_ordim\_o  
                        CHANGING  
                          ev\_who     = lv\_who.  
  
                      CLEAR ls\_who.  
                      IF lv\_who+0(1) NE 0.  
                        IF iv\_who NE lv\_who.  
                          "if current WO already contains WT for HU in front of HU to be picked => do nothing  
                          ls\_who-who = lv\_who.  
                          INSERT ls\_who INTO ct\_who INDEX 1.  
                          ls\_det2\_fg-lgnum = iv\_lgnum.  
                          ls\_det2\_fg-crane\_no = lv\_crane\_no.  
                          ls\_det2\_fg-who = lv\_who.  
                          ls\_det2\_fg-description = 'Rewarehousing task sent to subsystemfor picking'.  
                          ls\_det2\_fg-log\_date = sy-datum.  
                          ls\_det2\_fg-log\_time = sy-uzeit.  
                          APPEND ls\_det2\_fg TO lt\_det2\_fg.  
                          MODIFY zdet2\_fg\_log FROM TABLE lt\_det2\_fg.  
                          COMMIT WORK.  
  
  
                          UPDATE zinter\_leave SET active = abap\_true  
                                                  who\_no = ls\_who-who  
                                              WHERE lgnum EQ iv\_lgnum  
                                              AND resource\_name = iv\_queue.  
                          COMMIT WORK.  
  
                        ENDIF.  
                        cv\_rearr\_done = abap\_true.  
                        RETURN.  
                      ELSE.  
                        SELECT \*  INTO TABLE lt\_inter\_who FROM zinter\_leave  
                                  WHERE lgnum = iv\_lgnum AND resource\_name = iv\_queue .  
                        IF sy-subrc = 0.  
                          LOOP AT lt\_inter\_who INTO ls\_inter\_who.  
                            READ TABLE ct\_who INTO cs\_who WITH KEY who = ls\_inter\_who-who\_no.  
                            IF sy-subrc = 0.  
                              "do nothing  
                            ELSE.  
                              UPDATE zinter\_leave SET active = abap\_false  
                                                      who\_no = ls\_rtv\_who-who  
                                                  WHERE lgnum EQ iv\_lgnum  
                                                  AND resource\_name = iv\_queue.  
                              COMMIT WORK.  
                            ENDIF.  
                          ENDLOOP.  
                        ENDIF.  
                        RETURN.  
                      ENDIF.  
  
                      SELECT \*  INTO TABLE lt\_inter\_who FROM zinter\_leave  
                                WHERE lgnum = iv\_lgnum AND resource\_name = iv\_queue .  
                      IF sy-subrc = 0.  
                        LOOP AT lt\_inter\_who INTO ls\_inter\_who.  
                          READ TABLE ct\_who INTO cs\_who WITH KEY who = ls\_inter\_who-who\_no.  
                          IF sy-subrc = 0.  
                            "do nothing  
                          ELSE.  
                            UPDATE zinter\_leave SET active = abap\_false  
                                                    who\_no = ls\_rtv\_who-who  
                                                WHERE lgnum EQ iv\_lgnum  
                                                AND resource\_name = iv\_queue.  
                            COMMIT WORK.  
                          ENDIF.  
                        ENDLOOP.  
                      ENDIF.  
                      RETURN.  
                    ENDIF."sy-subrc = 0  of 2010 read table end  
  
                  ENDIF."3010 of read table end  
  
                ENDIF."3090 READ TABLE end  
  
  
              ENDIF."end of sy-subrc interleave  
  
            ENDIF."ls\_final\_error end  
  
  
          ENDLOOP.  
        ENDIF."lt\_final\_error  
      ENDIF.  
\*      lo\_log->save\_applog2db( ).  
    ENDIF.  
  
    IF lv\_missing\_case = 1 AND lt\_rtv\_who IS NOT INITIAL.  
  
      ls\_det2\_fg-lgnum = iv\_lgnum.  
      ls\_det2\_fg-crane\_no = lv\_crane\_no.  
      ls\_det2\_fg-description = 'Missing case '.  
      ls\_det2\_fg-log\_date = sy-datum.  
      ls\_det2\_fg-log\_time = sy-uzeit.  
      APPEND ls\_det2\_fg TO lt\_det2\_fg.  
      MODIFY zdet2\_fg\_log FROM TABLE lt\_det2\_fg.  
      COMMIT WORK.  
    ENDIF.  
\*\*  
\*\*  
\*\*  
\*\*  
\*\*  
\*\*  
\*\*  
\*\*  
\*\*  
\*\*  
\*\*  
\*\*  
\*\*  
\*\*    DATA ls\_selcrit   TYPE /scwm/s\_to\_selcrit\_mon.  
\*\*    DATA ltr\_who      TYPE rseloption.  
\*\*    DATA lsr\_who      TYPE rsdsselopt.  
\*\*    DATA lt\_to        TYPE /scwm/tt\_to\_det\_mon.  
\*\*  
\*\*    lsr\_who-sign    = wmegc\_sign\_inclusive.  
\*\*    lsr\_who-option  = wmegc\_option\_eq.  
\*\*    lsr\_who-low     = iv\_current\_wo.  
\*\*    APPEND lsr\_who TO ltr\_who.  
\*\*  
\*\*    LOOP AT ct\_who ASSIGNING FIELD-SYMBOL(<fs\_who>).  
\*\*      lsr\_who-low = <fs\_who>-who.  
\*\*      APPEND lsr\_who TO ltr\_who.  
\*\*    ENDLOOP.  
\*\*  
\*\*    ls\_selcrit-r\_who = ltr\_who.  
\*\* Get all WT details for WHO in ASRS Queue  
\*\*    CALL FUNCTION '/SCWM/TO\_GET\_WIP'  
\*\*      EXPORTING  
\*\*        iv\_lgnum   = iv\_lgnum  
\*\*        iv\_open    = abap\_true  
\*\*        iv\_conf    = abap\_true  
\*\*        is\_selcrit = ls\_selcrit  
\*\*      IMPORTING  
\*\*        et\_to      = lt\_to.  
\*\*    IF sy-subrc = 0 AND lt\_to IS NOT INITIAL.  
\*\*  
\*\*      MOVE-CORRESPONDING lt\_to to ct\_ordim\_o.  
\*\*  
\*\*    ENDIF.  
\*\*  
\*\*  
  
  
  ENDMETHOD..



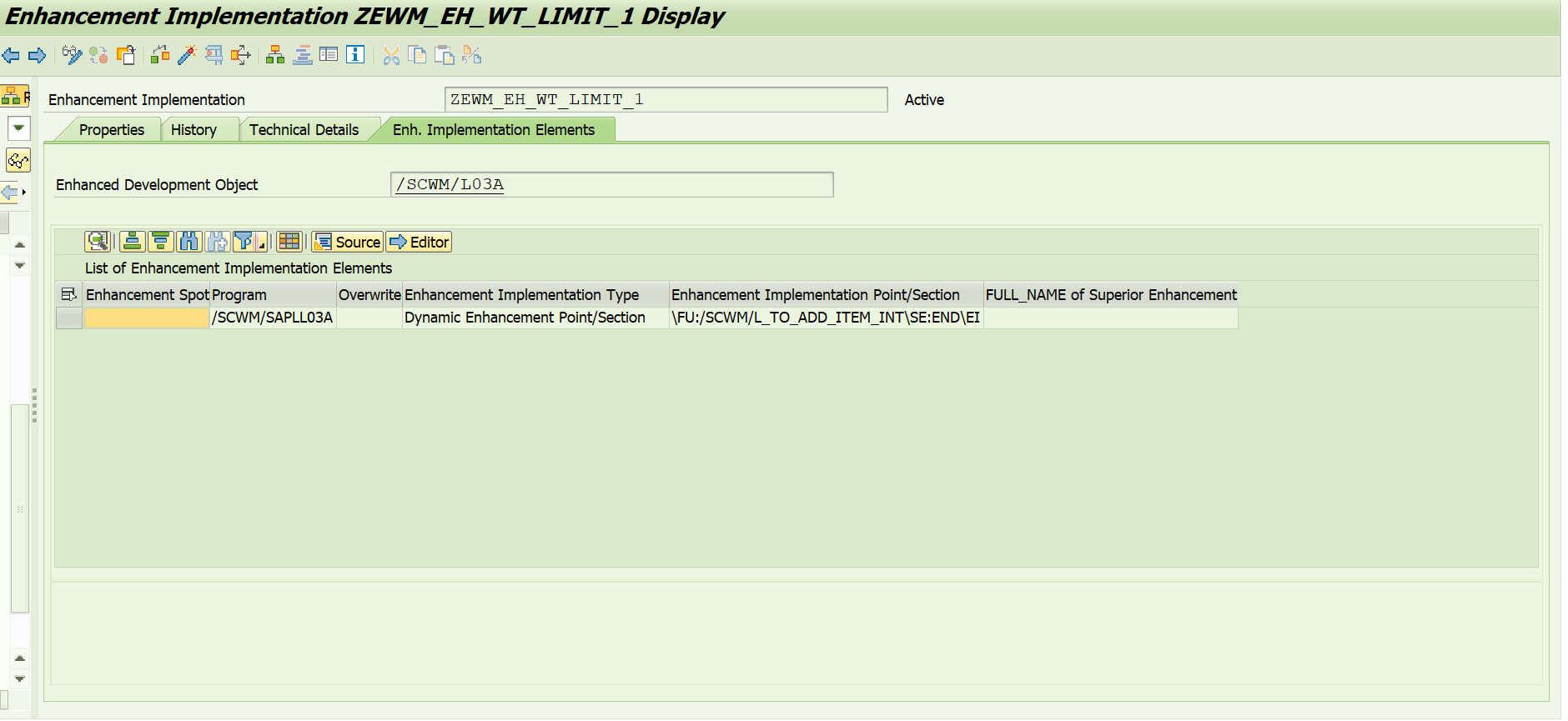
Same as above



A screenshot of a computer

Description automatically generated

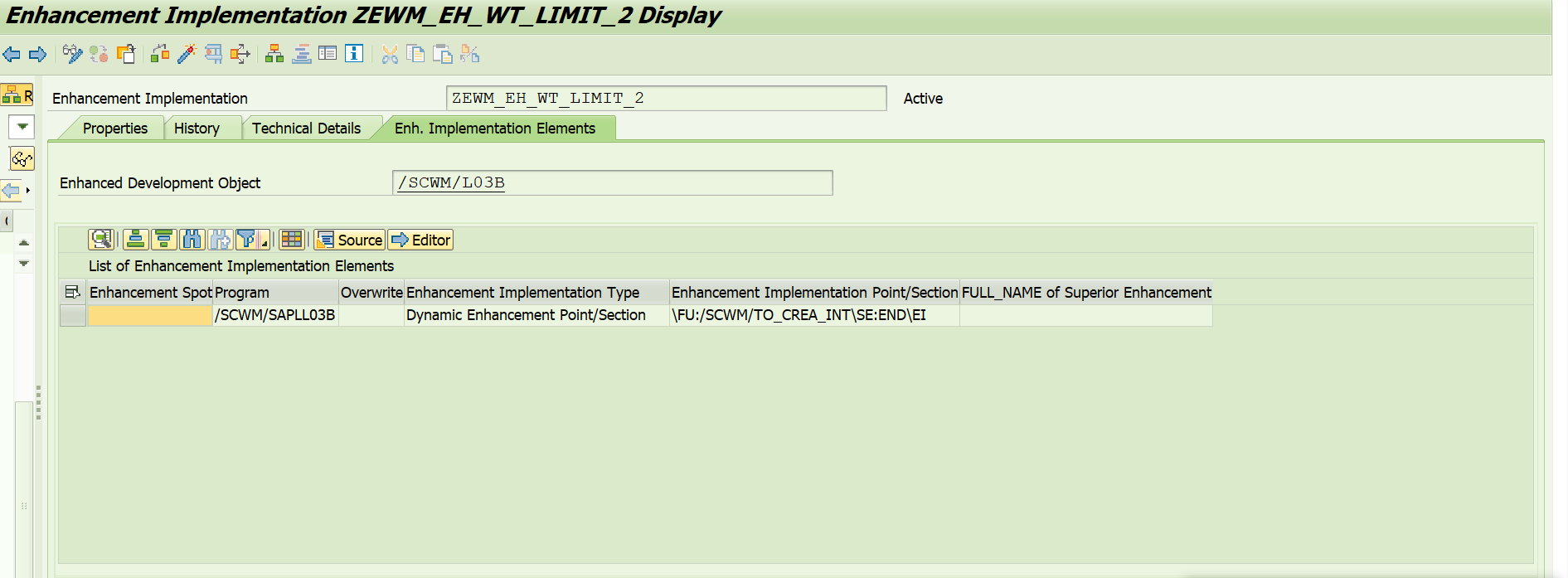
  METHOD /scwm/if\_ex\_core\_cr\_post~post.  
  
\*    DATA: lt\_whoid TYPE /scwm/tt\_whoid,  
\*          lv\_qname TYPE trfcqnam.  
\*  
\*    LOOP AT it\_ordim\_o ASSIGNING FIELD-SYMBOL(<lfs\_ordim\_o>) WHERE wcr = 'MFS'  
\*                                                             and act\_type = 'PICK'  
\*                                                             and vltyp = '3000'.  
\*  
\*      INSERT VALUE #( who = <lfs\_ordim\_o>-who ) INTO TABLE lt\_whoid.  
\*  
\*    ENDLOOP.  
\*  
\*  
\*    IF lt\_whoid IS NOT INITIAL.  
\*  
\*      DELETE ADJACENT DUPLICATES FROM lt\_whoid COMPARING who.  
\*  
\*      CALL FUNCTION 'ZEWM\_WHO\_HOLD\_UNHOLD'  
\*        IN BACKGROUND TASK  
\*        EXPORTING  
\*          iv\_lgnum = iv\_lgnum  
\*          iv\_hold  = abap\_true  
\*          it\_who   = lt\_whoid.  
\*    ENDIF.  
  
  ENDMETHOD.



A screenshot of a computer

Description automatically generated

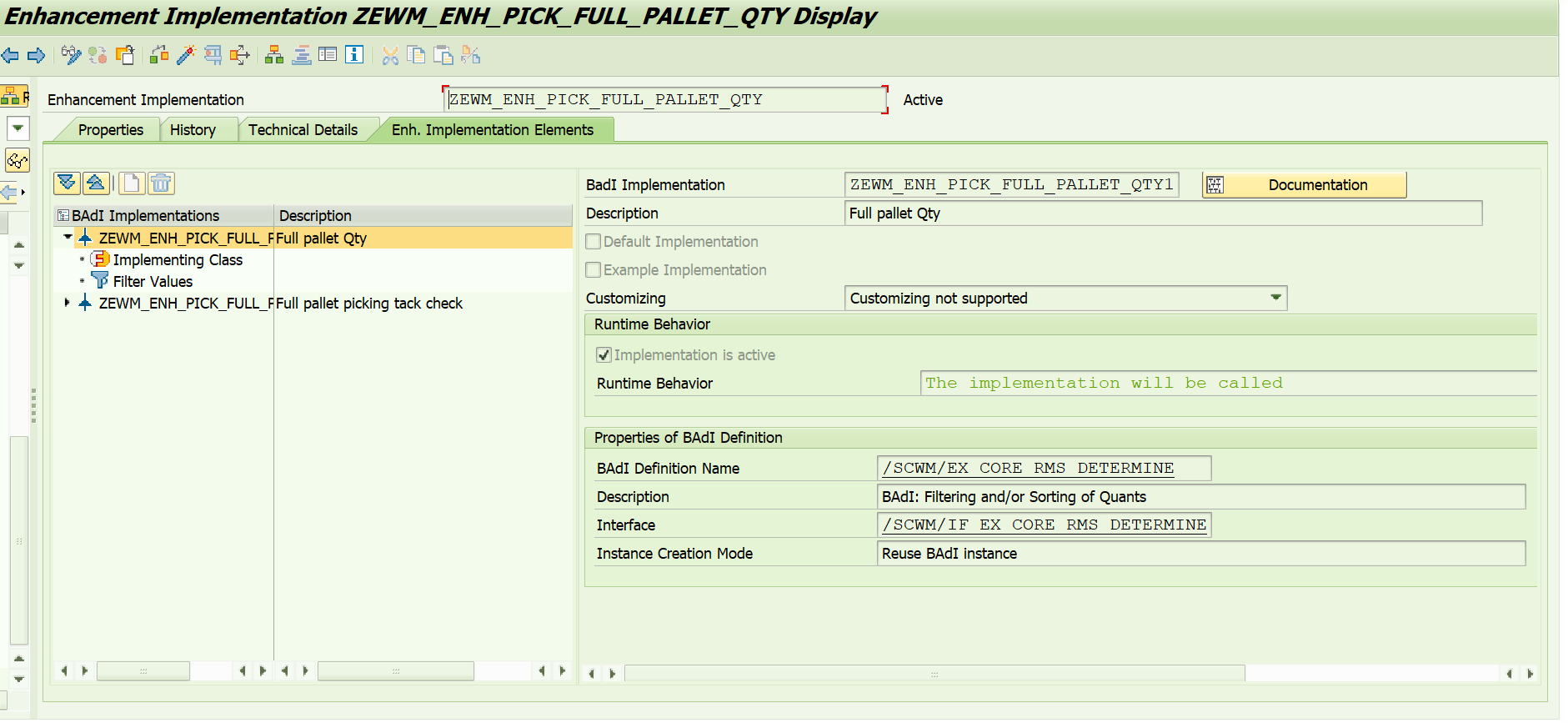
ENHANCEMENT 1  ZEWM\_EH\_WT\_LIMIT\_1.    "active version  
\* In current procees WT is being created for all request QTY for PDO but due to limit capacity of VMU, WT creation limit to be set  
\* This Enhancement is to limit the WT count to send the WT info to MFS. In this enhancement will send the WT count based in the WT items via WT-ID field  
\* will use the WT-ID in implecit enhancement ZEWM\_EH\_WT\_LIMIT\_2  
TYPES : BEGIN OF lty\_ordim\_o,  
          tanum TYPE /scwm/tanum,  
        END OF lty\_ordim\_o.  
DATA : lt\_con\_param TYPE TABLE OF zcon\_param\_val,  
       lt\_ordim\_o   TYPE TABLE OF lty\_ordim\_o.  
  
DATA : lt\_tap\_list TYPE /scwm/tt\_ltap\_vb,  
       lt\_docno    TYPE /scwm/dlv\_docno\_tab,  
       lt\_tanum    TYPE /scwm/tt\_wip\_wt4dlv,  
       ls\_docno    TYPE /scwm/dlv\_docno\_str,  
       lv\_lgnum    TYPE /scwm/lgnum,  
       lv\_docno    TYPE /scdl/dl\_docno\_int,  
       lv\_vltyp    TYPE /scwm/ltap\_vltyp,  
       lv\_mermory  TYPE i,  
       lv\_var1     TYPE char4,  
       lv\_var2     TYPE char46,  
       lv\_para     TYPE char50,  
       lv\_check    TYPE char1,  
       lv\_pass     TYPE char1.  
DATA : lc\_requirement\_name TYPE zde\_req\_name VALUE 'PDO\_PICK\_TASK\_LIMIT',  
       lc\_para\_name1       TYPE zde\_param    VALUE 'ENH\_ACTIVE',  
       lc\_para\_name2       TYPE zde\_param    VALUE 'VLTYP',  
       lc\_para\_name3       TYPE zde\_param    VALUE 'WT\_LIMIT'.  
  
  
  
CLEAR : lt\_con\_param, lv\_mermory, lv\_var1, lv\_var2, lv\_para, lv\_check,lv\_pass,lv\_vltyp,lv\_docno,lv\_lgnum.  
CLEAR : lt\_tap\_list, lt\_docno, lt\_tanum,lt\_ordim\_o, ls\_docno.  
  
\*get LGNUM  
LOOP AT tap.  
  lv\_lgnum = tap-lgnum.  
  EXIT.  
ENDLOOP.  
\*get all parametric values  
SELECT \* FROM zcon\_param\_val INTO TABLE lt\_con\_param  
         WHERE lgnum     = lv\_lgnum  
           AND zreq\_name = lc\_requirement\_name.  
\* condense variable in case any gap while maintaining  
LOOP AT lt\_con\_param ASSIGNING FIELD-SYMBOL(<lfs\_con\_param>).  
  CONDENSE <lfs\_con\_param>-zvalue NO-GAPS.  
ENDLOOP.  
  
  
\* check active parameter for enhancement  
READ TABLE lt\_con\_param ASSIGNING <lfs\_con\_param>  
                           WITH KEY zparam    = lc\_para\_name1  
                                    zvalue    = abap\_true  
                                    zactive   = abap\_true.  
CHECK sy-subrc = 0.  
  
MOVE-CORRESPONDING tap[] TO lt\_tap\_list[].  
SORT lt\_tap\_list BY rdoccat.  
DELETE lt\_tap\_list WHERE rdoccat <> wmegc\_doccat\_pdo.  
SORT lt\_tap\_list BY rdocid.  
DELETE ADJACENT DUPLICATES FROM lt\_tap\_list COMPARING rdocid.  
READ TABLE lt\_tap\_list ASSIGNING FIELD-SYMBOL(<lfs\_tap\_list>) INDEX 1.  
IF sy-subrc IS NOT INITIAL.  
  EXIT.  
ELSE.  
  IF <lfs\_tap\_list> IS ASSIGNED.  
    IF <lfs\_tap\_list>-trart    = wmegc\_trart\_pick AND   "2  
       <lfs\_tap\_list>-rdoccat  = wmegc\_doccat\_pdo AND   "PDO  
       <lfs\_tap\_list>-act\_type = wmegc\_iproc\_pick.      "PICK  
      lv\_check = abap\_true.   "memory ID need to export when WT is PDO relevant  
      lv\_var1 = <lfs\_tap\_list>-rdoccat.  
      lv\_var2 = <lfs\_tap\_list>-rdocid.  
      CONCATENATE lv\_var1 lv\_var2 INTO lv\_para.  
      CONDENSE lv\_para NO-GAPS.  
      IMPORT lv\_mermory TO lv\_mermory FROM MEMORY ID lv\_para.  
    ELSE.  
      EXIT.  
    ENDIF.  
  ENDIF.  
ENDIF.  
  
\* check for storage type validation  
LOOP AT et\_ltap\_vb ASSIGNING FIELD-SYMBOL(<lfs\_ltap\_temp>).  
  READ TABLE lt\_con\_param ASSIGNING <lfs\_con\_param>  
                                WITH KEY zparam    = lc\_para\_name2  
                                         zvalue    = <lfs\_ltap\_temp>-vltyp  
                                         zactive   = abap\_true.  
  IF sy-subrc = 0.  
    lv\_vltyp = <lfs\_ltap\_temp>-vltyp.  
    lv\_pass = abap\_true.  
    EXIT.  
  ENDIF.  
ENDLOOP.  
CLEAR: lv\_docno.  
IF lv\_check = abap\_true AND lv\_pass = abap\_true.  
  ADD 1 TO lv\_mermory.  
  EXPORT lv\_mermory TO MEMORY ID lv\_para. "Export memory ID for next use  
  
\* get the PDO document number against DOCID  
  SELECT docno INTO (lv\_docno)  
               FROM /scdl/db\_proch\_o  
               WHERE docid  = <lfs\_tap\_list>-rdocid  
                 AND doccat = <lfs\_tap\_list>-rdoccat.  
  ENDSELECT.  
\* checking for already existing open tasks for 3000 storage type  
  IF lv\_docno IS NOT INITIAL.  
    ls\_docno-docno  = lv\_docno.  
    ls\_docno-doccat = wmegc\_doccat\_pdo.  
    APPEND ls\_docno TO lt\_docno.  
  
    CALL FUNCTION '/SCWM/WIP\_GETWT4DLV'  
      EXPORTING  
        iv\_lgnum  = lv\_lgnum  
        iv\_doccat = /scdl/if\_dl\_doc\_c=>sc\_doccat\_out\_prd  
        it\_docno  = lt\_docno  
      IMPORTING  
        et\_wt4dlv = lt\_tanum.  
    IF lt\_tanum IS NOT INITIAL.  
\*checking already open task from 3000 storage type  
      SELECT tanum  
        FROM /scwm/ordim\_o  
        INTO TABLE lt\_ordim\_o  
        FOR ALL ENTRIES IN lt\_tanum  
        WHERE lgnum  = lv\_lgnum  
        AND tanum  = lt\_tanum-tanum  
        AND trart  = wmegc\_trart\_pick  
        AND tostat = abap\_false  
        AND vltyp  = lv\_vltyp.  
      IF lt\_ordim\_o IS NOT INITIAL.  
        DATA(lv\_count) = lines( lt\_ordim\_o ).  
        lv\_mermory = lv\_mermory + lv\_count. "Adding count from already created tasks  
      ENDIF.  
    ENDIF.  
  ENDIF.  
  READ TABLE lt\_con\_param ASSIGNING <lfs\_con\_param>  
                              WITH KEY zparam    = lc\_para\_name3  
                                       zactive   = abap\_true.  
  IF lv\_mermory GT <lfs\_con\_param>-zvalue.  
    LOOP AT et\_ltap\_vb ASSIGNING <lfs\_ltap\_temp>.  
      LOOP AT tap WHERE seqno   = <lfs\_ltap\_temp>-seqno  
                    AND guid\_to = <lfs\_ltap\_temp>-guid\_to.  
        TRY.  
            CALL FUNCTION '/SCWM/L\_TO\_DELETE\_ITEM\_INT'  
              EXPORTING  
                iv\_guid\_to = tap-guid\_to.  
          CATCH /scwm/cx\_core.  
            MESSAGE a100 INTO gv\_msgtext.  
            CALL METHOD go\_prot->add\_message(  
                ip\_row = ls\_ltap-seqno ).  
        ENDTRY.  
        MESSAGE e001(/scwm/delivery) WITH 'Due to WT limit WT creation cancelled from st. type/ st. bin'  
                                         <lfs\_ltap\_temp>-vltyp '/' <lfs\_ltap\_temp>-vlpla INTO gv\_msgtext.  
  
        CALL METHOD go\_prot->add\_message(  
            ip\_row = ls\_ltap-seqno ).  
      ENDLOOP.  
    ENDLOOP.  
    CLEAR et\_ltap\_vb.  
  ENDIF.  
ENDIF.  
  
ENDENHANCEMENT.



A screenshot of a computer

Description automatically generated

ENHANCEMENT 1  ZEWM\_EH\_WT\_LIMIT\_2.    "active version  
\* In current procees WT is being created for all request QTY for PDO but due to limit capacity of VMU, WT creation limit to be set  
\* This Enhancement is to limit the WT count to send the WT info to MFS. This enhancement linked to implecit enhancement ZEWM\_EH\_WT\_LIMIT\_1  
  
DATA : lt\_con\_param    TYPE TABLE OF zcon\_param\_val.  
DATA : lt\_ltap\_vp\_temp TYPE /scwm/tt\_ltap\_vb,  
       lt\_ltap\_vp\_del  TYPE /scwm/tt\_ltap\_vb.  
DATA : lv\_id      TYPE /scwm/ltap-id, "/lime/line\_item\_i,  
       lv\_mermory TYPE i,  
       lv\_var1    TYPE char4,  
       lv\_var2    TYPE char46,  
       lv\_para    TYPE char50.  
DATA : lc\_requirement\_name TYPE zde\_req\_name VALUE 'PDO\_PICK\_TASK\_LIMIT',  
       lc\_para\_name1       TYPE zde\_param    VALUE 'ENH\_ACTIVE',  
       lc\_para\_name2       TYPE zde\_param    VALUE 'WT\_LIMIT'.  
  
CLEAR : lt\_ltap\_vp\_temp,lt\_ltap\_vp\_del.  
CLEAR : lt\_con\_param, lv\_mermory, lv\_var1, lv\_var2, lv\_para.  
  
\*BREAK p00127339.  
  
READ TABLE et\_ltap\_vb ASSIGNING FIELD-SYMBOL(<lfs\_ltpa\_vb\_1>) INDEX 1.  
IF sy-subrc = 0 AND <lfs\_ltpa\_vb\_1> IS ASSIGNED.  
  
  SELECT \* FROM zcon\_param\_val INTO TABLE lt\_con\_param  
           WHERE lgnum       = <lfs\_ltpa\_vb\_1>-lgnum  
             AND zreq\_name   = lc\_requirement\_name.  
  IF sy-subrc = 0.  
    SORT lt\_con\_param BY zparam.  
\*check if enhancement parameter is active  
    READ TABLE lt\_con\_param ASSIGNING FIELD-SYMBOL(<lfs\_con\_param>)  
                                  WITH KEY zparam  = lc\_para\_name1  
                                           zvalue  = abap\_true  
                                           zactive = abap\_true.  
    IF sy-subrc = 0.  
\* only for PDO picking task  
      IF <lfs\_ltpa\_vb\_1> IS ASSIGNED AND  
         <lfs\_ltpa\_vb\_1>-trart      = 2 AND  
         ( <lfs\_ltpa\_vb\_1>-rdoccat  = 'PDO' OR  
        <lfs\_ltpa\_vb\_1>-qdoccat     = 'PDO' ) AND  
         <lfs\_ltpa\_vb\_1>-act\_type   = 'PICK'.  
  
        IF <lfs\_ltpa\_vb\_1>-rdoccat IS NOT INITIAL.  
          lv\_var1 = <lfs\_ltpa\_vb\_1>-rdoccat.  
          lv\_var2 = <lfs\_ltpa\_vb\_1>-rdocid.  
        ELSE.  
          lv\_var1 = <lfs\_ltpa\_vb\_1>-qdoccat.  
          lv\_var2 = <lfs\_ltpa\_vb\_1>-qdocid.  
        ENDIF.  
        CONCATENATE lv\_var1 lv\_var2 INTO lv\_para.  
        CONDENSE lv\_para NO-GAPS.  
        FREE MEMORY ID lv\_para.     "Memory ID from implecit enhancement ZEWM\_EH\_WT\_LIMIT\_1  
      ENDIF.  
    ENDIF.  
  ENDIF.  
ENDIF.  
ENDENHANCEMENT.

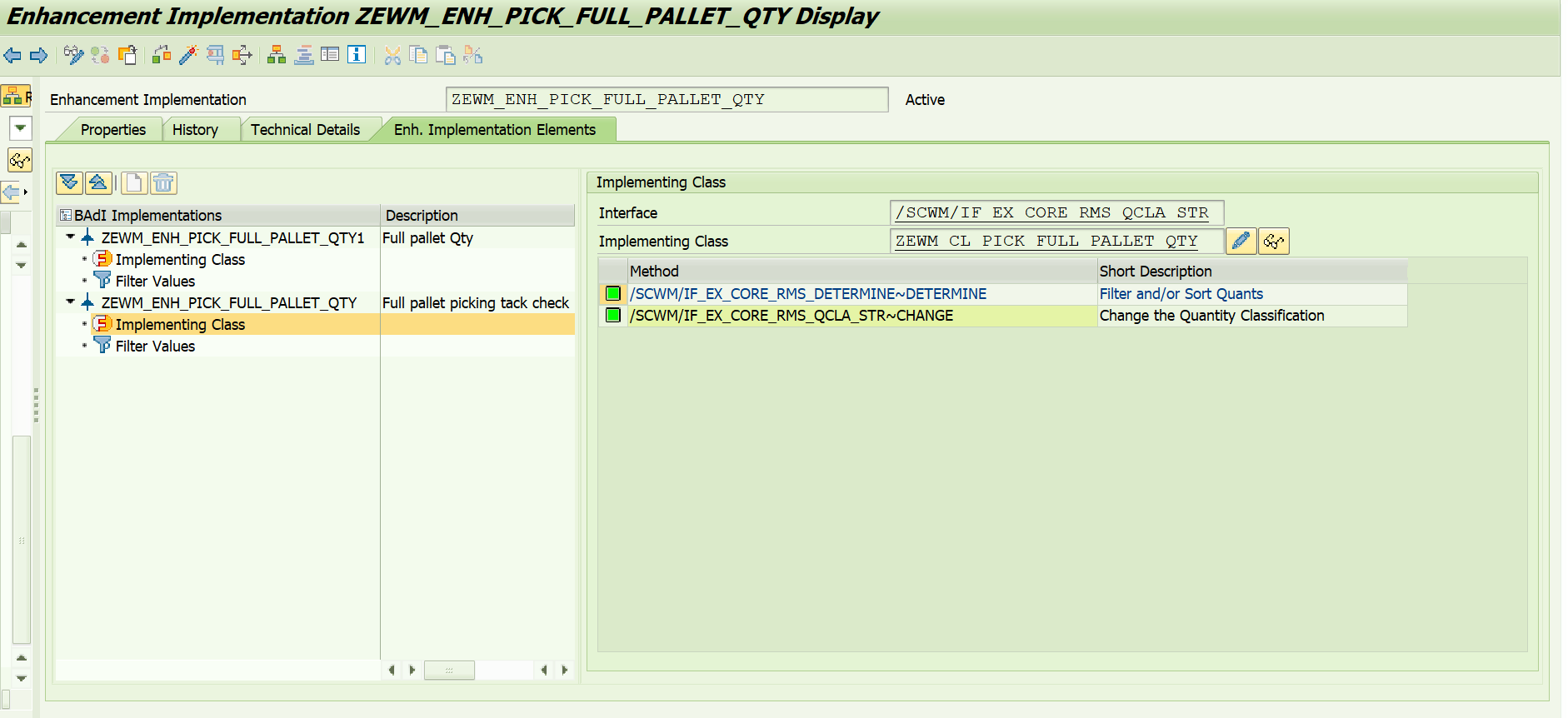


A screenshot of a computer

Description automatically generated

  METHOD /scwm/if\_ex\_core\_rms\_determine~determine.  
\*----------------------------------------------------------------------------------------------\*  
\* PROJECT     : Stellium                                                                       \*  
\* FUNCTIONAL  : Arthur                                                                         \*  
\* TECHNICAL   : Vinothkumar R                                                                  \*  
\* TR          : KSDK900438                                                                     \*  
\* DATA        : 18.02.2023                                                                     \*  
\* DESCRIPTION : Consider only full pallat stock on picking task create                         \*  
\*----------------------------------------------------------------------------------------------\*  
\*                                        MODIFICATION LOG                                      \*  
\*----------------------------------------------------------------------------------------------\*  
\*                                                                                              \*  
\*----------------------------------------------------------------------------------------------\*  
\*              GET this GV\_QUANCLA & GV\_QTY from BADI - /SCWM/EX\_CORE\_RMS\_QCLA\_STR             \*  
\*----------------------------------------------------------------------------------------------\*  
    DATA: ls\_stal       TYPE /scwm/s\_lgtyp\_aisle\_level,  
          lt\_stal       TYPE /scwm/tt\_lgtyp\_aisle\_level,  
          ls\_mfsrsrcais TYPE /scwm/mfsrsrcais,  
          lt\_mfsrsrcais TYPE /scwm/tt\_mfsrsrcais,  
          lt\_lgpla\_r    TYPE /scwm/tt\_lgpla\_r.  
  
    CLEAR ev\_set.  
    SELECT zparam,zvalue INTO TABLE @DATA(lt\_param)  
                             FROM zcon\_param\_val  
                             WHERE lgnum     EQ @is\_ltap-lgnum  
                             AND   zreq\_name EQ 'OUTBOUND\_FULL\_PALLET\_PICK'  
                             AND   zactive   EQ @abap\_true.  
    IF sy-subrc EQ 0.  
      SORT lt\_param BY zparam zvalue.  
      READ TABLE lt\_param INTO DATA(ls\_param) WITH KEY zparam = 'FULL\_PALLET\_PICK\_WPT'  
                                                       zvalue = is\_ltap-procty  
                                                       BINARY SEARCH.  
      IF sy-subrc NE 0.  
        DATA: lt\_blocked TYPE TABLE OF /scwm/mfsrsrcais,  
              lt\_mfsmbrk TYPE TABLE OF /scwm/rsrc.  
  
        TYPES: BEGIN OF ty\_wpt,  
                 procty TYPE /scwm/de\_procty,  
               END OF ty\_wpt.  
  
        DATA: lt\_wpt TYPE TABLE OF ty\_wpt,  
              ls\_wpt TYPE ty\_wpt.  
  
        ls\_wpt-procty = '3093'.  
        APPEND ls\_wpt TO lt\_wpt.  
  
        ls\_wpt-procty = 'ZADH'.  
        APPEND ls\_wpt TO lt\_wpt.  
  
        ls\_wpt-procty = '9999'.  
        APPEND ls\_wpt TO lt\_wpt.  
  
        SELECT \* FROM /scwm/rsrc INTO TABLE @lt\_mfsmbrk  
          WHERE lgnum = @is\_ltap-lgnum  
          AND exccode\_overall = 'MBRK'.  
  
        SELECT lgtyp FROM /scwm/mfsrsrcais INTO CORRESPONDING FIELDS OF TABLE @lt\_blocked  
          FOR ALL ENTRIES IN @lt\_mfsmbrk  
          WHERE lgnum EQ @is\_ltap-lgnum  
          AND rsrc = @lt\_mfsmbrk-rsrc.  
  
        IF line\_exists( lt\_blocked[ lgtyp = is\_ltap-nltyp ] ).  
          IF sy-tcode = '/SCWM/ADHU'.  
            IF line\_exists( lt\_blocked[ lgtyp = is\_ltap-nltyp ] ) OR line\_exists( lt\_wpt[ procty = is\_ltap-procty ] ).  
              MESSAGE 'Error' TYPE 'E'.  
            ENDIF.  
          ENDIF.  
        ENDIF.  
  
      ENDIF.  
  
      IF sy-subrc EQ 0.  
  
        DATA(lt\_qmat) = it\_qmat.  
        DELETE lt\_qmat WHERE lgpla IS INITIAL.  
  
        IF lt\_qmat IS NOT INITIAL.  
\* Get the Storage Bin Aisle from /scwm/lagp table  
          SELECT lgnum, lgpla, lgtyp, lgber, lptyp, aisle, lvl\_v  
              FROM /scwm/lagp  
              INTO TABLE @DATA(lt\_lagp)  
              FOR ALL ENTRIES IN @lt\_qmat  
              WHERE lgnum    = @is\_ltap-lgnum  
                AND lgpla    = @lt\_qmat-lgpla  
                AND lgtyp    = @is\_t331-lgtyp.  
  
        ENDIF.  
  
        DATA(lt\_lagp1) = lt\_lagp.  
        SORT:lt\_lagp1 BY aisle.  
        DELETE ADJACENT DUPLICATES FROM lt\_lagp1 COMPARING aisle.  
  
        IF lt\_lagp1 IS NOT INITIAL.  
\*  Get the MFS resource from /scwm/mfsrsrcais table  
          SELECT \* FROM /scwm/mfsrsrcais  
                     INTO TABLE lt\_mfsrsrcais  
                     FOR ALL ENTRIES IN lt\_lagp1  
                     WHERE lgnum = is\_ltap-lgnum  
                       AND lgtyp = lt\_lagp1-lgtyp  
                       AND aisle = lt\_lagp1-aisle.  
  
          IF lt\_mfsrsrcais IS NOT INITIAL.  
\* Fetch the Resources with Exception Code MBRK from /scwm/rsrc  
            SELECT \* FROM /scwm/rsrc INTO TABLE @DATA(lt\_rsrc)  
                               FOR ALL ENTRIES IN @lt\_mfsrsrcais  
                               WHERE lgnum = @is\_ltap-lgnum  
                               AND   rsrc = @lt\_mfsrsrcais-rsrc  
                               AND   exccode\_overall = 'MBRK'.  
  
          ENDIF.  
  
\* filter out storeage bins from Storage Bin Aisle for Exception Code MBRK  
          LOOP AT lt\_rsrc ASSIGNING FIELD-SYMBOL(<fs\_rsrc>).  
            READ TABLE lt\_mfsrsrcais ASSIGNING FIELD-SYMBOL(<fs\_mfsrsrcais>)  
                                           WITH KEY rsrc = <fs\_rsrc>-rsrc.  
            IF sy-subrc = 0.  
              LOOP AT lt\_lagp ASSIGNING FIELD-SYMBOL(<fs\_lagp>)  
                              WHERE  lgtyp = <fs\_mfsrsrcais>-lgtyp  
                              AND    aisle = <fs\_mfsrsrcais>-aisle.  
                INSERT VALUE #( sign   = wmegc\_sign\_inclusive  
                       option = wmegc\_option\_eq  
                       low    = <fs\_lagp>-lgpla ) INTO TABLE lt\_lgpla\_r.  
              ENDLOOP.  
            ENDIF.  
          ENDLOOP.  
          IF lt\_lgpla\_r IS NOT INITIAL.  
            DELETE lt\_qmat WHERE lgpla IN lt\_lgpla\_r.  
          ENDIF.  
  
        ENDIF.  
  
        et\_qmat\_cus = lt\_qmat.  
        ev\_set = abap\_true.  
        CLEAR ls\_param.  
        READ TABLE lt\_param INTO ls\_param WITH KEY zparam = 'FULL\_PALLET\_PICK\_TCODE'  
                                                   zvalue = sy-tcode  
                                                   BINARY SEARCH.  
        IF sy-subrc EQ 0 .  
  
          IF gv\_quancla IS NOT INITIAL.  
\*            et\_qmat\_cus = it\_qmat.  
            DELETE et\_qmat\_cus WHERE quan GT gv\_qty.  
\*            ev\_set = abap\_true.  
          ENDIF.  
        ENDIF.  
      ENDIF.  
    ENDIF.  
  
  ENDMETHOD.

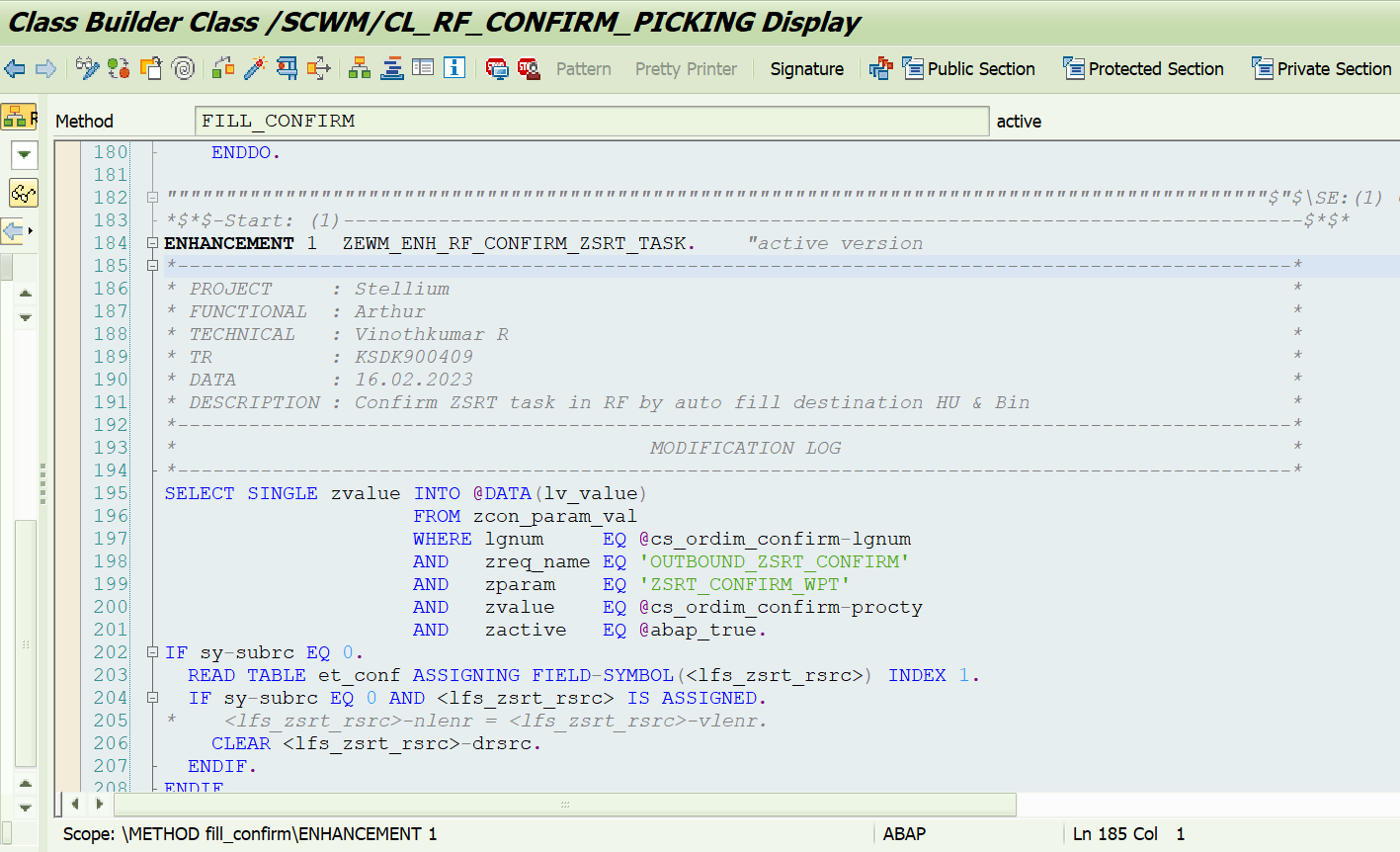
  METHOD /scwm/if\_ex\_core\_rms\_qcla\_str~change.  
\*----------------------------------------------------------------------------------------------\*  
\* PROJECT     : Stellium                                                                       \*  
\* FUNCTIONAL  : Arthur                                                                         \*  
\* TECHNICAL   : Vinothkumar R                                                                  \*  
\* TR          : KSDK900438                                                                     \*  
\* DATA        : 18.02.2023                                                                     \*  
\* DESCRIPTION : Consider only full pallat stock on picking task create                         \*  
\*----------------------------------------------------------------------------------------------\*  
\*                                        MODIFICATION LOG                                      \*  
\*----------------------------------------------------------------------------------------------\*  
\*                                                                                              \*  
\*----------------------------------------------------------------------------------------------\*  
    DATA : ltr\_lgtyp   TYPE RANGE OF /scwm/lgtyp,  
           lsr\_lgtyp   LIKE LINE OF ltr\_lgtyp,  
           ltr\_huident TYPE RANGE OF /scwm/de\_huident,  
           lsr\_huident LIKE LINE OF ltr\_huident,  
           lt\_ltap\_vb  TYPE /scwm/tt\_ltap\_vb.  
\*----------------------------------------------------------------------------------------------\*  
\*              If WPT & TCODE maintained in ZCON\_PARAM\_VAL only logic will work                \*  
\*----------------------------------------------------------------------------------------------\*  
    CLEAR : gv\_quancla,gv\_qty.  
    SELECT zparam,zvalue INTO TABLE @DATA(lt\_param)  
                         FROM zcon\_param\_val  
                         WHERE lgnum     EQ @is\_ltap-lgnum  
                         AND   zreq\_name EQ 'OUTBOUND\_FULL\_PALLET\_PICK'  
                         AND   zactive   EQ @abap\_true.  
    IF sy-subrc EQ 0.  
      SORT lt\_param BY zparam zvalue.  
      READ TABLE lt\_param INTO DATA(ls\_param) WITH KEY zparam = 'FULL\_PALLET\_PICK\_WPT'  
                                                       zvalue = is\_ltap-procty  
                                                       BINARY SEARCH.  
      IF sy-subrc EQ 0.  
        CLEAR ls\_param.  
        READ TABLE lt\_param INTO ls\_param WITH KEY zparam = 'FULL\_PALLET\_PICK\_TCODE'  
                                                   zvalue = sy-tcode  
                                                   BINARY SEARCH.  
        IF sy-subrc EQ 0.  
\*----------------------------------------------------------------------------------------------\*  
\*         Quantity calssification is initail then check full pallet exist or not               \*  
\*----------------------------------------------------------------------------------------------\*  
          IF cv\_quancla IS INITIAL.  
            SELECT quancla,rem\_sseq FROM /scwm/t334r  
                                    INTO TABLE @DATA(lt\_rem\_sseq)  
                                    WHERE lgnum  EQ @is\_ltap-lgnum  
                                    AND   procty EQ @is\_ltap-procty.  
            IF sy-subrc EQ 0.  
              DELETE lt\_rem\_sseq WHERE quancla IS INITIAL.  
              IF lt\_rem\_sseq IS NOT INITIAL.  
                SELECT lgnum,  
                       rem\_sseq,  
                       seqno,  
                       lgtyp FROM /scwm/t334rs  
                             INTO TABLE @DATA(lt\_t334rs)  
                             FOR ALL ENTRIES IN @lt\_rem\_sseq  
                             WHERE lgnum     EQ @is\_ltap-lgnum  
                             AND   rem\_sseq  EQ @lt\_rem\_sseq-rem\_sseq.  
                IF sy-subrc EQ 0.  
                  REFRESH ltr\_lgtyp. CLEAR lsr\_lgtyp.  
                  lsr\_lgtyp-sign   = 'I'.  
                  lsr\_lgtyp-option = 'EQ'.  
                  LOOP AT lt\_t334rs INTO DATA(ls\_t334rs).  
                    lsr\_lgtyp-low = ls\_t334rs-lgtyp.  
                    APPEND lsr\_lgtyp TO ltr\_lgtyp.  
                    CLEAR ls\_t334rs.  
                  ENDLOOP.  
                  CLEAR lsr\_lgtyp.  
                  IF ltr\_lgtyp IS NOT INITIAL.  
\*----------------------------------------------------------------------------------------------\*  
\*             Fetch the stock on full pallet location EX - 3000 search sequence                \*  
\*----------------------------------------------------------------------------------------------\*  
                    SELECT guid\_parent,  
                           guid\_stock,  
                           lgtyp,  
                           lgpla,  
                           huident,  
                           quan,  
                           unit  
                      FROM /scwm/aqua  
                      INTO TABLE @DATA(lt\_aqua)  
                      WHERE lgnum           =  @is\_ltap-lgnum  
                          AND matid         =  @is\_ltap-matid  
                          AND batchid       =  @is\_ltap-batchid  
                          AND cat           =  @is\_ltap-cat  
                          AND stock\_doccat  =  @is\_ltap-stock\_doccat  
                          AND stock\_docno   =  @is\_ltap-stock\_docno  
                          AND stock\_itmno   =  @is\_ltap-stock\_itmno  
                          AND doccat        =  @is\_ltap-doccat  
                          AND owner         =  @is\_ltap-owner  
                          AND owner\_role    =  @is\_ltap-owner\_role  
                          AND entitled      =  @is\_ltap-entitled  
                          AND entitled\_role =  @is\_ltap-entitled\_role  
                          AND unit         =  @iv\_unit  
                          AND lgtyp         IN @ltr\_lgtyp  
                          AND skzsi         =  @abap\_false.  
                    IF sy-subrc EQ 0.  
                      DATA(lt\_aqua\_1) = lt\_aqua.  
                      SORT lt\_aqua\_1 BY lgpla.  
                      DELETE ADJACENT DUPLICATES FROM lt\_aqua\_1 COMPARING lgpla.  
                      IF lt\_aqua\_1 IS NOT INITIAL.  
                        SELECT lgnum,  
                               lgpla,  
                               skzua FROM /scwm/lagp  
                                     INTO TABLE @DATA(lt\_bin)  
                                     FOR ALL ENTRIES IN @lt\_aqua\_1  
                                     WHERE lgnum EQ @is\_ltap-lgnum  
                                     AND   lgpla EQ @lt\_aqua\_1-lgpla.  
                        IF sy-subrc EQ 0.  
                          SORT lt\_bin BY lgpla.  
                          DELETE lt\_aqua WHERE quan GT iv\_quan.  
                          IF lt\_aqua IS NOT INITIAL.  
                            CALL FUNCTION '/SCWM/L\_TO\_GET\_INFO'  
                              IMPORTING  
                                et\_ltap\_vb = lt\_ltap\_vb.  
\*----------------------------------------------------------------------------------------------\*  
\*                                   Delete already used pallet                                 \*  
\*----------------------------------------------------------------------------------------------\*  
                            SORT lt\_ltap\_vb BY vlenr.  
                            REFRESH ltr\_huident. CLEAR lsr\_huident.  
                            lsr\_huident-sign   = 'I'.  
                            lsr\_huident-option = 'EQ'.  
                            LOOP AT lt\_aqua ASSIGNING FIELD-SYMBOL(<lfs\_aqua>).  
                              READ TABLE lt\_ltap\_vb INTO DATA(ls\_ltap\_vb) WITH KEY vlenr = <lfs\_aqua>-huident BINARY SEARCH.  
                              IF sy-subrc EQ 0.  
                                CLEAR : ls\_ltap\_vb,<lfs\_aqua>.  
                                CONTINUE.  
                              ENDIF.  
                              READ TABLE lt\_bin INTO DATA(ls\_bin) WITH KEY lgpla = <lfs\_aqua>-lgpla BINARY SEARCH.  
                              IF sy-subrc EQ 0 AND ls\_bin-skzua EQ abap\_true.  
                                CLEAR : ls\_ltap\_vb,ls\_bin,<lfs\_aqua>.  
                                CONTINUE.  
                              ENDIF.  
                              lsr\_huident-low = <lfs\_aqua>-huident.  
                              APPEND lsr\_huident TO ltr\_huident.  
                              CLEAR : ls\_ltap\_vb,ls\_bin.  
                            ENDLOOP.  
                            DELETE lt\_aqua WHERE huident IS INITIAL.  
\*----------------------------------------------------------------------------------------------\*  
\*               If still some more full pallet exit the set quantity calsification             \*  
\*----------------------------------------------------------------------------------------------\*  
                            IF lt\_aqua IS NOT INITIAL.  
                              READ TABLE lt\_rem\_sseq INTO DATA(ls\_rem\_sseq) INDEX 1.  
                              IF sy-subrc EQ 0.  
                                cv\_quancla = ls\_rem\_sseq-quancla.  
                                DESCRIBE TABLE gt\_pre\_aqua LINES DATA(lv\_pre\_aqua).  
                                DESCRIBE TABLE lt\_aqua LINES DATA(lv\_aqua).  
                                IF lv\_pre\_aqua EQ lv\_aqua.  
                                  DELETE gt\_pre\_aqua WHERE huident IN ltr\_huident.  
                                  IF gt\_pre\_aqua IS INITIAL.  
                                    CLEAR cv\_quancla.  
                                  ENDIF.  
                                ENDIF.  
                                REFRESH gt\_pre\_aqua.  
                                gt\_pre\_aqua = lt\_aqua.  
                                CLEAR : lv\_pre\_aqua,lv\_aqua. REFRESH : lt\_aqua,lt\_aqua\_1.  
                              ENDIF.  
                            ENDIF.  
                          ENDIF.  
                        ENDIF.  
                      ENDIF.  
                    ENDIF.  
                  ENDIF.  
                ENDIF.  
              ENDIF.  
            ENDIF.  
          ENDIF.  
\*----------------------------------------------------------------------------------------------\*  
\*              Set this GV\_QUANCLA & GV\_QTY for BADI /SCWM/EX\_CORE\_RMS\_DETERMINE               \*  
\*----------------------------------------------------------------------------------------------\*  
          IF cv\_quancla IS NOT INITIAL.  
            gv\_quancla = cv\_quancla.  
            gv\_qty     = iv\_quan.  
          ENDIF.  
        ENDIF.  
      ENDIF.  
    ENDIF.  
  
  ENDMETHOD.



Same as above

A screenshot of a computer

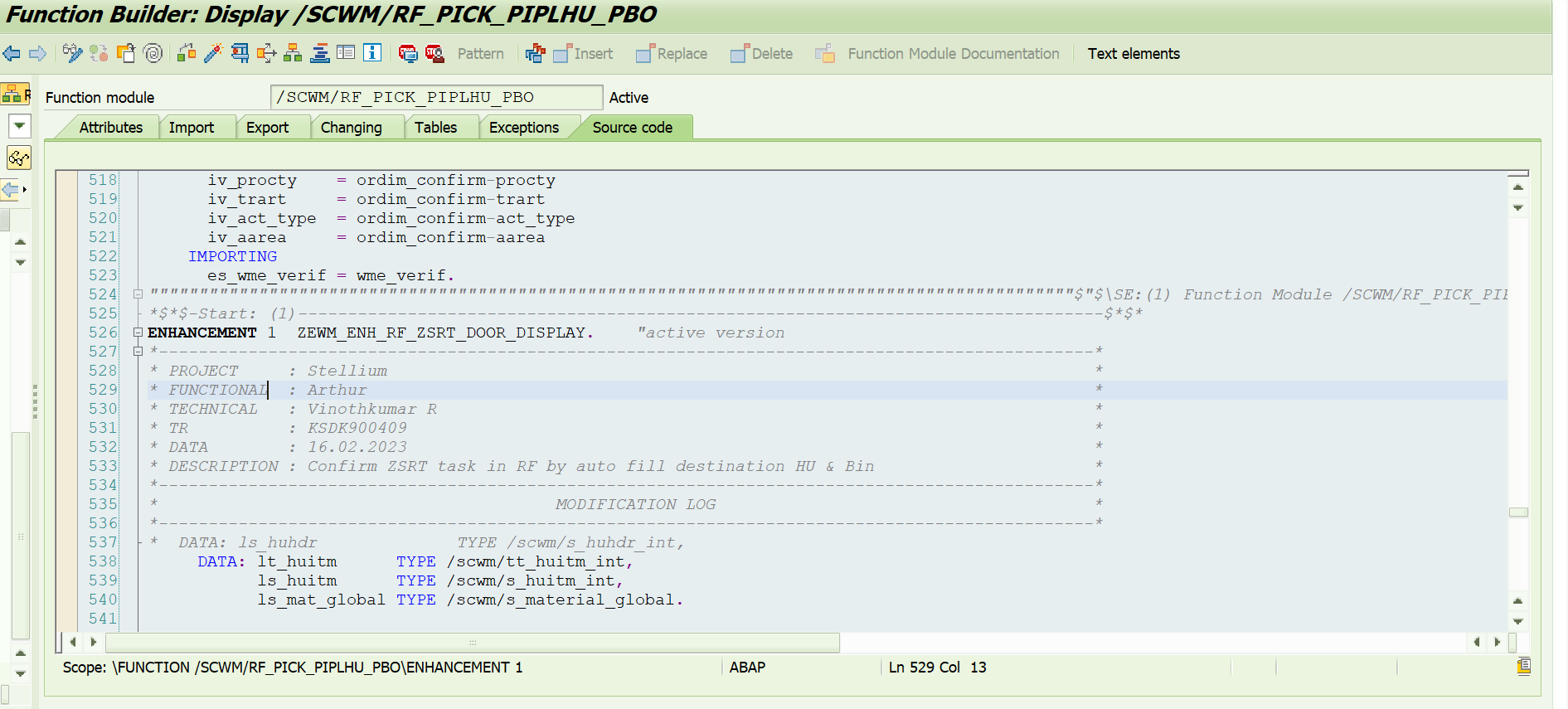
Description automatically generated



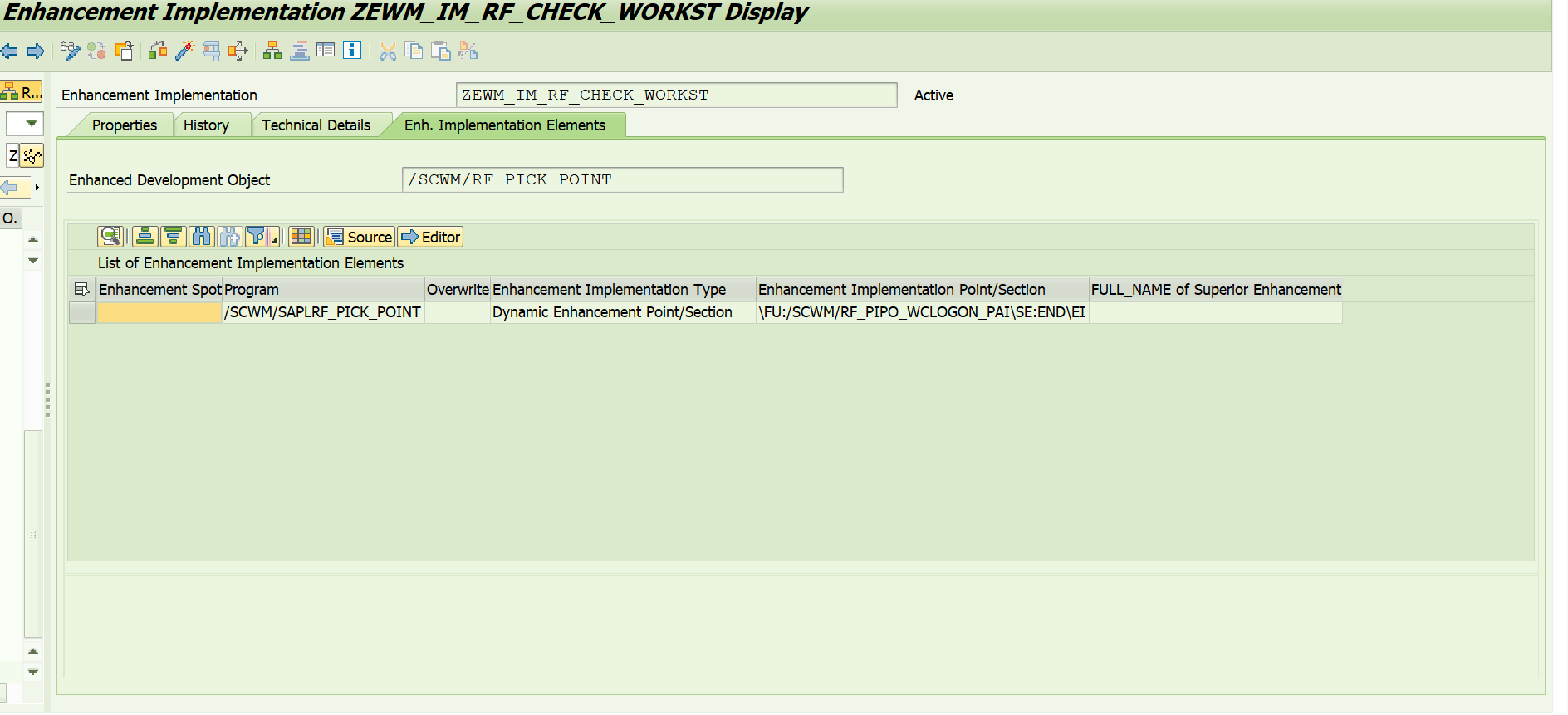
ENHANCEMENT 1  ZEWM\_ENH\_RF\_CONFIRM\_ZSRT\_TASK.    "active version  
\*----------------------------------------------------------------------------------------------\*  
\* PROJECT     : Stellium                                                                       \*  
\* FUNCTIONAL  : Arthur                                                                         \*  
\* TECHNICAL   : Vinothkumar R                                                                  \*  
\* TR          : KSDK900409                                                                     \*  
\* DATA        : 16.02.2023                                                                     \*  
\* DESCRIPTION : Confirm ZSRT task in RF by auto fill destination HU & Bin                      \*  
\*----------------------------------------------------------------------------------------------\*  
\*                                        MODIFICATION LOG                                      \*  
\*----------------------------------------------------------------------------------------------\*  
SELECT SINGLE zvalue INTO @DATA(lv\_value)  
                     FROM zcon\_param\_val  
                     WHERE lgnum     EQ @cs\_ordim\_confirm-lgnum  
                     AND   zreq\_name EQ 'OUTBOUND\_ZSRT\_CONFIRM'  
                     AND   zparam    EQ 'ZSRT\_CONFIRM\_WPT'  
                     AND   zvalue    EQ @cs\_ordim\_confirm-procty  
                     AND   zactive   EQ @abap\_true.  
IF sy-subrc EQ 0.  
  READ TABLE et\_conf ASSIGNING FIELD-SYMBOL(<lfs\_zsrt\_rsrc>) INDEX 1.  
  IF sy-subrc EQ 0 AND <lfs\_zsrt\_rsrc> IS ASSIGNED.  
\*    <lfs\_zsrt\_rsrc>-nlenr = <lfs\_zsrt\_rsrc>-vlenr.  
    CLEAR <lfs\_zsrt\_rsrc>-drsrc.  
  ENDIF.  
ENDIF.  
CLEAR lv\_value.  
  
ENDENHANCEMENT.

A screenshot of a computer

Description automatically generated



ENHANCEMENT 1  ZEWM\_ENH\_RF\_ZSRT\_DOOR\_DISPLAY.    "active version  
\*----------------------------------------------------------------------------------------------\*  
\* PROJECT     : Stellium                                                                       \*  
\* FUNCTIONAL  : Arthur                                                                         \*  
\* TECHNICAL   : Vinothkumar R                                                                  \*  
\* TR          : KSDK900409                                                                     \*  
\* DATA        : 16.02.2023                                                                     \*  
\* DESCRIPTION : Confirm ZSRT task in RF by auto fill destination HU & Bin                      \*  
\*----------------------------------------------------------------------------------------------\*  
\*                                        MODIFICATION LOG                                      \*  
\*----------------------------------------------------------------------------------------------\*  
\*  DATA: ls\_huhdr              TYPE /scwm/s\_huhdr\_int,  
     DATA: lt\_huitm      TYPE /scwm/tt\_huitm\_int,  
           ls\_huitm      TYPE /scwm/s\_huitm\_int,  
           ls\_mat\_global TYPE /scwm/s\_material\_global.  
  
  
     SELECT SINGLE zvalue INTO @DATA(lv\_value)  
                          FROM zcon\_param\_val  
                          WHERE lgnum     EQ @ordim\_confirm-lgnum  
                          AND   zreq\_name EQ 'OUTBOUND\_ZSRT\_CONFIRM'  
                          AND   zparam    EQ 'ZSRT\_CONFIRM\_WPT'  
                          AND   zvalue    EQ @ordim\_confirm-procty  
                          AND   zactive   EQ @abap\_true.  
     IF sy-subrc EQ 0.  
       READ TABLE tt\_ordim\_confirm INDEX lv\_line ASSIGNING FIELD-SYMBOL(<lfs\_door>).  
       IF sy-subrc = 0.  
         CALL METHOD hu\_oref->get\_hu  
           EXPORTING  
             iv\_huident = <lfs\_door>-vlenr  
\*            IV\_GUID\_HU =  
\*            iv\_lock    = gc\_lock  
           IMPORTING  
             et\_huitm   = lt\_huitm  
             es\_huhdr   = ls\_huhdr  
\*            ET\_HUHDR   =  
\*            ET\_HUTREE  =  
\*            et\_huref   = lt\_huref  
           EXCEPTIONS  
             not\_found  = 1  
             OTHERS     = 2.  
  
         IF NOT lt\_huitm[] IS INITIAL.  
\*   If HU contains items, loop at items to complete data for display  
           LOOP AT lt\_huitm INTO ls\_huitm.  
             lv\_line = sy-tabix.  
\*      CLEAR huitem.  
\*      MOVE-CORRESPONDING ls\_huitm TO huitem.  
             TRY.  
                 CALL FUNCTION '/SCWM/MATERIAL\_READ\_SINGLE'  
                   EXPORTING  
                     iv\_matid      = ls\_huitm-matid  
                     iv\_langu      = sy-langu  
                     iv\_entitled   = ls\_huitm-entitled  
\*                    IV\_APPLIC     =  
                     iv\_lgnum      = ordim\_confirm-lgnum  
\*                    IV\_LGTYP      =  
\*                    IT\_LGTYP      =  
                   IMPORTING  
                     es\_mat\_global = ls\_mat\_global  
\*                    ES\_MAT\_HAZARD =  
\*                    ES\_MAT\_LGNUM  =  
\*                    ES\_MAT\_LGTYP  =  
\*                    ET\_MAT\_LGTYP  =  
\*                    ET\_MAT\_UOM    =  
\*                    EV\_APPLIC\_PROC                   =  
                   .  
               CATCH /scwm/cx\_md\_interface.  
\*       one/more parameter(s) missing  
                 MESSAGE e004 WITH ls\_huitm-matid.  
               CATCH /scwm/cx\_md\_material\_exist.  
\*       Material does not exist  
                 MESSAGE e004 WITH ls\_huitm-matid.  
               CATCH /scwm/cx\_md\_lgnum\_locid.  
\*       Warehouse number is not assigned to an APO Location  
                 MESSAGE e004 WITH ls\_huitm-matid.  
               CATCH /scwm/cx\_md.  
                 MESSAGE e004 WITH ls\_huitm-matid.  
             ENDTRY.  
  
\*     Set MATNR and MAKTX for material  
             ordim\_confirm-maktx = ls\_mat\_global-maktx.  
             ordim\_confirm-matnr = ls\_mat\_global-matnr.  
\*           ordim\_confirm-vsolm = ls\_huitm-quan.  
             MODIFY tt\_ordim\_confirm INDEX lv\_line FROM ordim\_confirm.  
\*           huitem-quan\_chr = huitem-quan.  
  
           ENDLOOP.  
         ENDIF.  
  
         IF sy-subrc EQ 0 AND <lfs\_door> IS ASSIGNED.  
           IF ordim\_confirm-procty NE '3010'.  
             <lfs\_door>-nlpla\_verif = <lfs\_door>-nlpla.  
           ENDIF.  
           IF ordim\_confirm-qdocid IS NOT INITIAL.  
             DATA(lv\_docid) = ordim\_confirm-qdocid.  
             DATA(lv\_itmid) = ordim\_confirm-qitmid.  
           ELSEIF ordim\_confirm-rdocid IS NOT INITIAL.  
             CLEAR : lv\_docid,lv\_itmid.  
             lv\_docid       = ordim\_confirm-rdocid.  
             lv\_itmid       = ordim\_confirm-ritmid.  
           ENDIF.  
           IF lv\_docid IS NOT INITIAL AND lv\_itmid IS NOT INITIAL.  
             SELECT SINGLE docid,  
                           itemid,  
                           /scwm/door FROM /scdl/db\_proci\_o  
                                      INTO @DATA(ls\_door)  
                                      WHERE docid  EQ @lv\_docid  
                                      AND   itemid EQ @lv\_itmid.  
             IF sy-subrc EQ 0.  
               <lfs\_door>-text\_ind = ls\_door-/scwm/door.  
             ENDIF.  
           ENDIF.  
           IF ordim\_confirm-procty NE '3010'.  
             /scwm/cl\_rf\_bll\_srvc=>set\_screlm\_input\_off( gc\_scr\_elmnt\_nlpla\_vrf ).  
           ENDIF.  
         ENDIF.  
  
         CLEAR : lv\_value,ls\_door,lv\_docid,lv\_itmid.  
         IF <lfs\_door>-nlpla\_verif IS INITIAL.  
           /scwm/cl\_rf\_bll\_srvc=>set\_screlm\_invisible\_on( '/SCWM/S\_RF\_ORDIM\_CONFIRM-TEXT\_IND' ).  
         ENDIF.  
       ENDIF.  
     ENDIF.  
  
     IF ordim\_confirm-procty EQ '3010'.  
       /scwm/cl\_rf\_bll\_srvc=>set\_screlm\_invisible\_on( '/SCWM/S\_RF\_ORDIM\_CONFIRM-TEXT\_IND' ).  
     ENDIF.  
  
     IF ordim\_confirm-procty NE '3010'.  
       /scwm/cl\_rf\_bll\_srvc=>set\_screlm\_invisible\_on( '/SCWM/S\_RF\_ORDIM\_CONFIRM-MATNR' ).  
       /scwm/cl\_rf\_bll\_srvc=>set\_screlm\_invisible\_on( '/SCWM/S\_RF\_ORDIM\_CONFIRM-MAKTX' ).  
       /scwm/cl\_rf\_bll\_srvc=>set\_screlm\_invisible\_on( '/SCWM/S\_RF\_ORDIM\_CONFIRM-VSOLM' ).  
       /scwm/cl\_rf\_bll\_srvc=>set\_screlm\_invisible\_on( '/SCWM/S\_RF\_ORDIM\_CONFIRM-MEINS' ).  
     ENDIF.  
  
\*  
\*     /scwm/cl\_rf\_bll\_srvc=>set\_screen\_param( 'ORDIM\_CONFIRM' ).  
  
ENDENHANCEMENT.



A screenshot of a computer

Description automatically generated

ENHANCEMENT 1  ZEWM\_IM\_RF\_CHECK\_WORKST.    "active version  
DATA: lv\_ltrans  TYPE string,  
      lv\_station TYPE i.  
  
lv\_ltrans = /scwm/cl\_rf\_bll\_srvc=>get\_ltrans( ).  
  
IF lv\_ltrans EQ 'ZSHORT' OR lv\_ltrans EQ 'ZMERGE'.  
  lv\_station = cs\_pipo\_work-workstation+2(2).  
  
  IF lv\_station mod 2 EQ 0.  
    IF lv\_ltrans EQ 'ZSHORT'.  
      MESSAGE 'Please enter correct work station' TYPE 'E'.  
      RETURN.  
    ENDIF.  
  ENDIF.  
  
  IF lv\_station mod 2 NE 0.  
    IF lv\_ltrans EQ 'ZMERGE'.  
      MESSAGE 'Please enter correct work station' TYPE 'E'.  
      RETURN.  
    ENDIF.  
  ENDIF.  
  
  
ENDIF.  
  
  
  
  
  
  
ENDENHANCEMENT.

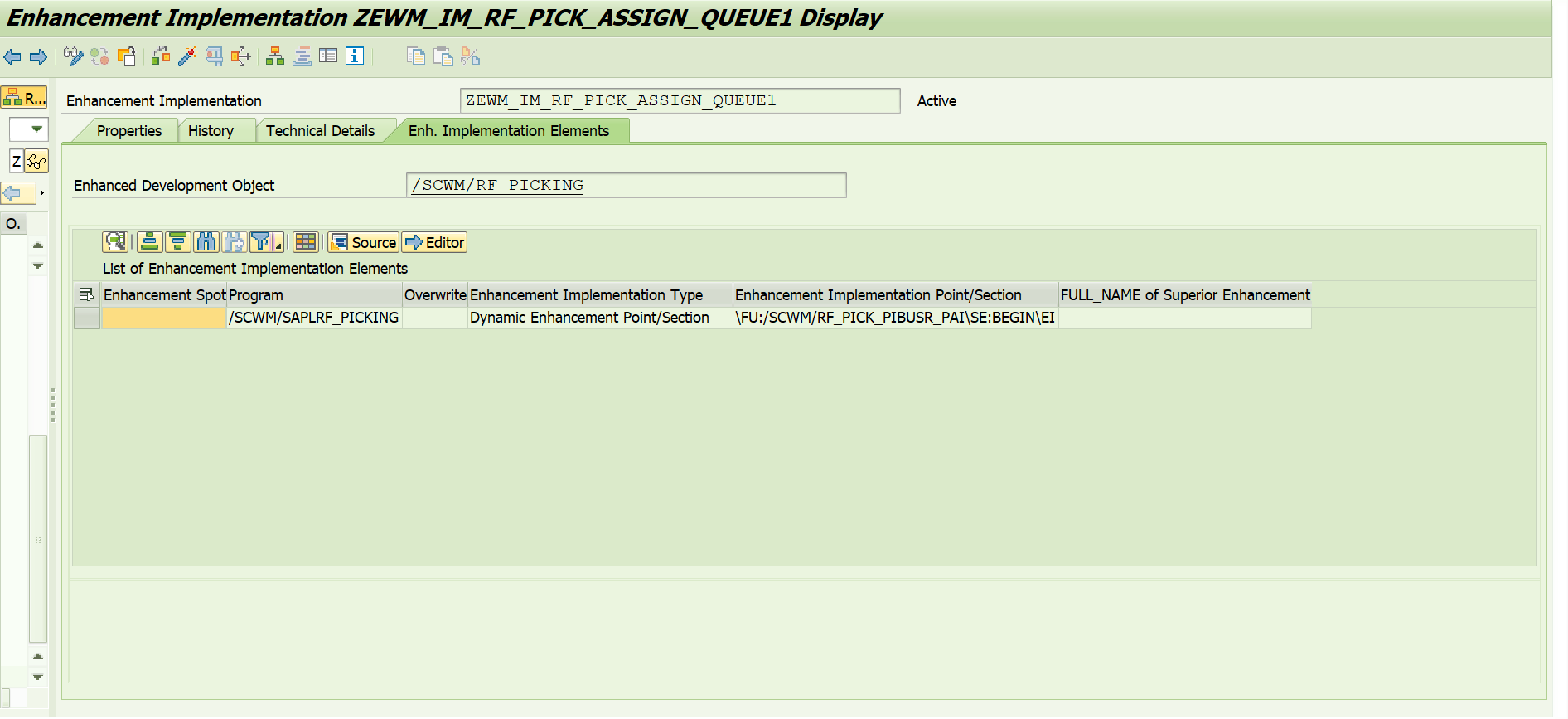
A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

ENHANCEMENT 1  ZEWM\_IM\_RF\_PICK\_ASSIGN\_QUEUE.    "active version  
  
DATA:lt\_src\_open\_to TYPE /scwm/tt\_ordim\_o,  
     ls\_who1        TYPE  /scwm/s\_who\_int,  
     lv\_queue1       TYPE  /scwm/de\_queue,  
     ls\_attributes  TYPE /scwm/s\_who\_att,  
     ls\_resource1   TYPE /scwm/rsrc.  
  
        selection-huident\_verif = selection-rfhu.  
  
\*               Get resource info  
        MOVE-CORRESPONDING resource TO ls\_resource1.  
        CALL FUNCTION '/SCWM/RSRC\_RESOURCE\_MEMORY'  
          EXPORTING  
            iv\_uname = sy-uname  
          CHANGING  
            cs\_rsrc  = ls\_resource1.  
\*                   Add leading zeros if needed.  
        CALL FUNCTION 'CONVERSION\_EXIT\_ALPHA\_INPUT'  
          EXPORTING  
            input  = selection-huident\_verif  
          IMPORTING  
            output = selection-huident.  
\*                   Get list of WO for HU  
\*                   Get list of open TO for HU as source HU  
        CALL FUNCTION '/SCWM/TO\_READ\_SRC'  
          EXPORTING  
            iv\_lgnum     = ls\_resource1-lgnum  
            iv\_huident   = selection-huident  
          IMPORTING  
            et\_ordim\_o   = lt\_src\_open\_to  
          EXCEPTIONS  
            wrong\_input  = 1  
            not\_found    = 2  
            foreign\_lock = 3  
            OTHERS       = 4.  
  
        IF NOT lt\_src\_open\_to[] IS INITIAL.  
          READ TABLE lt\_src\_open\_to INTO DATA(ls\_ordim\_o1) INDEX 1.  
          IF sy-subrc = 0 AND ls\_ordim\_o1-tostat IS INITIAL.  
            SELECT SINGLE \* INTO @DATA(ls\_param) FROM zcon\_param\_val  
                                     WHERE lgnum = @ls\_resource1-lgnum  
                                     AND zreq\_name = 'RF\_PICK'  
                                     AND zparam = @ls\_ordim\_o1-nltyp  
                                     AND zactive = @abap\_true.  
  
            IF sy-subrc = 0.  
              lv\_queue1 = ls\_param-zvalue.  
              CALL FUNCTION '/SCWM/WHO\_GET'  
                EXPORTING  
                  iv\_lgnum    = ls\_resource1-lgnum  
                  iv\_lock\_who = space  
                  iv\_lock\_to  = space  
                  iv\_whoid    = ls\_ordim\_o1-who  
                IMPORTING  
                  es\_who      = ls\_who1.  
  
  
              CALL FUNCTION 'ZSCWM\_RSRC\_WO\_QUEUE\_CHANGE'  
                EXPORTING  
                  iv\_lgnum = ls\_resource1-lgnum  
                  is\_who   = ls\_who1  
                  iv\_queue = lv\_queue1.  
  
  
            ENDIF.  
          ENDIF.  
        ENDIF.  
  
ENDENHANCEMENT.



ENHANCEMENT 1  ZEWM\_IM\_RF\_PICK\_ASSIGN\_QUEUE1.    "active version  
  
  
DATA:lt\_src\_open\_to TYPE /scwm/tt\_ordim\_o,  
     ls\_who1        TYPE  /scwm/s\_who\_int,  
     lv\_queue1      TYPE  /scwm/de\_queue,  
     ls\_attributes  TYPE /scwm/s\_who\_att,  
     ls\_resource1   TYPE /scwm/rsrc.  
  
selection-huident\_verif = selection-rfhu.  
  
\*               Get resource info  
MOVE-CORRESPONDING resource TO ls\_resource1.  
CALL FUNCTION '/SCWM/RSRC\_RESOURCE\_MEMORY'  
  EXPORTING  
    iv\_uname = sy-uname  
  CHANGING  
    cs\_rsrc  = ls\_resource1.  
\*                   Add leading zeros if needed.  
CALL FUNCTION 'CONVERSION\_EXIT\_ALPHA\_INPUT'  
  EXPORTING  
    input  = selection-huident\_verif  
  IMPORTING  
    output = selection-huident.  
\*                   Get list of WO for HU  
\*                   Get list of open TO for HU as source HU  
CALL FUNCTION '/SCWM/TO\_READ\_SRC'  
  EXPORTING  
    iv\_lgnum     = ls\_resource1-lgnum  
    iv\_huident   = selection-huident  
  IMPORTING  
    et\_ordim\_o   = lt\_src\_open\_to  
  EXCEPTIONS  
    wrong\_input  = 1  
    not\_found    = 2  
    foreign\_lock = 3  
    OTHERS       = 4.  
  
IF NOT lt\_src\_open\_to[] IS INITIAL.  
  READ TABLE lt\_src\_open\_to INTO DATA(ls\_ordim\_o1) INDEX 1.  
  IF sy-subrc = 0 AND ls\_ordim\_o1-tostat IS INITIAL.  
    SELECT SINGLE \* INTO @DATA(ls\_param) FROM zcon\_param\_val  
                             WHERE lgnum = @ls\_resource1-lgnum  
                             AND zreq\_name = 'RF\_PICK'  
                             AND zparam = @ls\_ordim\_o1-nltyp  
                             AND zactive = @abap\_true.  
  
    IF sy-subrc = 0.  
      lv\_queue1 = ls\_param-zvalue.  
      CALL FUNCTION '/SCWM/WHO\_GET'  
        EXPORTING  
          iv\_lgnum    = ls\_resource1-lgnum  
          iv\_lock\_who = space  
          iv\_lock\_to  = space  
          iv\_whoid    = ls\_ordim\_o1-who  
        IMPORTING  
          es\_who      = ls\_who1.  
  
  
      CALL FUNCTION 'ZSCWM\_RSRC\_WO\_QUEUE\_CHANGE'  
        EXPORTING  
          iv\_lgnum = ls\_resource1-lgnum  
          is\_who   = ls\_who1  
          iv\_queue = lv\_queue1.  
  
  
    ENDIF.  
  ENDIF.  
ENDIF.  
ENDENHANCEMENT.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

  METHOD /scwm/if\_ex\_core\_cr\_int\_cr~insert.  
  
  
\*-- Check the stock type is B6, If yes then error message will be displayed  
    IF sy-tcode = '/SCWM/ADHU'.  
\*-- Check WPT is active or not  
      SELECT \* FROM zcon\_param\_val  
              INTO TABLE @DATA(lt\_param)  
             WHERE lgnum = @is\_ltap-lgnum AND  
        zreq\_name = 'WPT\_TO\_STOP\_WT\_CREATION' AND  
        zparam = 'WPT' AND  
        zactive = @abap\_True.  
      IF sy-subrc = 0.  
        READ TABLE lt\_param TRANSPORTING NO FIELDS WITH KEY zvalue = is\_ltap-procty.  
        IF sy-subrc <> 0.  
          RETURN.  
        ENDIF.  
      ELSE.  
        RETURN.  
      ENDIF.  
  
\*-- For Blocked stock  
      IF is\_ltap-cat = 'B6'.  
        APPEND INITIAL LINE TO et\_bapiret ASSIGNING FIELD-SYMBOL(<fs\_msg>).  
        <fs\_msg>-id = 'ZEWM\_MSG'.  
        <fs\_msg>-number = '000'.  
        <fs\_msg>-type = 'E'.  
        <fs\_msg>-message\_v1 = is\_ltap-vlenr.  
        UNASSIGN <fs\_msg>.  
\*-- For Quality Stock  
      ELSEIF is\_ltap-cat = 'Q4'.  
        APPEND INITIAL LINE TO et\_bapiret ASSIGNING <fs\_msg>.  
        <fs\_msg>-id = 'ZEWM\_MSG'.  
        <fs\_msg>-number = '001'.  
        <fs\_msg>-type = 'E'.  
        <fs\_msg>-message\_v1 = is\_ltap-vlenr.  
        UNASSIGN <fs\_msg>.  
      ENDIF.  
\*-- Add Warning message  
      IF et\_bapiret IS NOT INITIAL.  
        APPEND INITIAL LINE TO et\_bapiret ASSIGNING <fs\_msg>.  
        <fs\_msg>-id = '/SCWM/L3'.  
        <fs\_msg>-number = '054'.  
        <fs\_msg>-type = 'W'.  
        <fs\_msg>-message\_v1 = is\_ltap-vlenr.  
        UNASSIGN <fs\_msg>.  
      ENDIF.  
    ENDIF.  
  
  ENDMETHOD.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

  METHOD /scwm/if\_ex\_core\_cr\_abort~abort.  
  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
\* Requirement : Do not allow to create WT if the stock is in blocked state B6  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
  
\*-- Check Stock type and stop creating WT  
    SELECT \* FROM zcon\_param\_val  
            INTO TABLE @DATA(lt\_param)  
           WHERE lgnum = @is\_t333-lgnum AND  
      zreq\_name = 'WPT\_TO\_STOP\_WT\_CREATION' AND  
      zparam = 'WPT' AND  
      zactive = @abap\_True.  
    IF sy-subrc = 0.  
      READ TABLE lt\_param TRANSPORTING NO FIELDS WITH KEY zvalue = is\_t333-procty.  
      IF sy-subrc <> 0.  
        RETURN.  
      ENDIF.  
    ELSE.  
      RETURN.  
    ENDIF.  
  
\*-- Error for Blcoked / Quality stock  
    IF ( is\_create-cat = 'B6' OR  is\_create-cat =  'Q4' ) AND sy-tcode = '/SCWM/ADPROD'.  
  
      IF is\_create-cat = 'B6'.  
        MESSAGE e000(zewm\_msg)  
           WITH is\_create-vlenr  
           INTO DATA(lv\_msgtext).  
        io\_prot->add\_message( ).  
      ELSEIF  is\_create-cat = 'Q4'.  
        MESSAGE e001(zewm\_msg)  
           WITH is\_create-vlenr  
           INTO lv\_msgtext.  
        io\_prot->add\_message( ).  
      ENDIF.  
      ev\_abort = abap\_true.  
      RETURN.  
    ENDIF.  
  
  ENDMETHOD.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

  METHOD /scwm/if\_ex\_mfs\_tele\_rcv~tele\_rcv.  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*FG TELE RECEIVE BADI\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
    DATA: lt\_status      TYPE TABLE OF zcrane\_dashboard,  
          ls\_ordim\_o     TYPE  /scwm/ordim\_o,  
          ls\_telegram    TYPE /scwm/s\_mfs\_teletotal,  
          lt\_ordim\_c     TYPE /scwm/tt\_ordim\_c,  
          ls\_ordim\_c     TYPE /scwm/ordim\_c,  
          lv\_tanum       TYPE /scwm/tanum,  
          lt\_crane\_log   TYPE TABLE OF zcrane\_error\_log,  
          lt\_crane\_mode  TYPE TABLE OF zcrane\_mode\_log,  
          ls\_status      LIKE LINE OF lt\_status,  
          lt\_ordim\_crane TYPE TABLE OF /scwm/ordim\_o,  
          ls\_ordim\_crane LIKE LINE OF lt\_ordim\_crane,  
          ls\_crane\_log   LIKE LINE OF lt\_crane\_log,  
          lv\_rsrc\_name   TYPE  /scwm/rsrc-rsrc,  
          lt\_log         TYPE TABLE OF zack\_telelog,  
          ls\_log         LIKE LINE OF lt\_log,  
          ls\_rsrc        TYPE /scwm/rsrc,  
          lt\_rsrc        TYPE TABLE OF /scwm/rsrc,  
          ls\_crane\_mode  LIKE LINE OF lt\_crane\_mode,  
          ps\_telegram    TYPE /scwm/s\_mfs\_teletotal.  
  
    DATA: lv\_already\_sent TYPE xfeld,  
          lr\_ex\_mfs       TYPE REF TO /scwm/cx\_mfs.  
  
    DATA: lv\_rsrc\_pattern(3) TYPE c.  
    CONSTANTS : lc\_crane\_mode\_2 TYPE char1 VALUE '2',  
                lc\_crane\_mode\_3 TYPE char1 VALUE '3'.  
  
    CASE iv\_plc.  
      WHEN zcl\_common\_constants=>gc\_plc\_swslg.  
  
        IF cs\_tele\_rcv-teletype EQ zcl\_common\_constants=>gc\_swslg\_tele\_csr OR cs\_tele\_rcv-teletype EQ zcl\_common\_constants=>gc\_swslg\_tele\_acp.  
  
          ls\_status-lgnum = iv\_lgnum.  
          ls\_crane\_log-lgnum    = iv\_lgnum.  
          ls\_status-crane\_no = cs\_tele\_rcv-crane\_number.  
          DATA(lv\_crane\_mode) = cs\_tele\_rcv-crane\_mode.  
          ls\_status-crane\_mode =  lv\_crane\_mode.  
          IF lv\_crane\_mode = 01.  
            ls\_status-crane\_m\_des = 'ONLINE'.  
          ELSEIF lv\_crane\_mode = 02.  
            ls\_status-crane\_m\_des = 'MANUAL'.  
          ELSEIF lv\_crane\_mode = 03.  
            ls\_status-crane\_m\_des = 'OFFLINE'.  
          ELSE.  
            ls\_status-crane\_m\_des = ''.  
          ENDIF.  
          ls\_crane\_log-crane\_no = cs\_tele\_rcv-crane\_number.  
  
          SELECT \* INTO TABLE lt\_ordim\_crane FROM /scwm/ordim\_o WHERE lgnum = iv\_lgnum AND tanum = cs\_tele\_rcv-assignment\_id.  
          IF sy-subrc = 0.  
            READ TABLE lt\_ordim\_crane INTO ls\_ordim\_crane WITH KEY tanum = cs\_tele\_rcv-assignment\_id.  
            IF sy-subrc = 0.  
              ls\_status-open\_task = ls\_ordim\_crane-tanum.  
              ls\_crane\_log-open\_task = ls\_ordim\_crane-tanum.  
              SHIFT ls\_ordim\_crane-vlenr LEFT DELETING LEADING '0'.  
              ls\_status-huident = ls\_ordim\_crane-vlenr.  
              ls\_crane\_log-huident = ls\_ordim\_crane-vlenr.  
              ls\_status-huident = ls\_ordim\_crane-vlenr.  
              ls\_status-source\_bin = ls\_ordim\_crane-vlpla.  
              ls\_crane\_log-source\_bin = ls\_ordim\_crane-vlpla.  
              ls\_status-dest\_bin  = ls\_ordim\_crane-nlpla.  
              ls\_crane\_log-dest\_bin = ls\_ordim\_crane-nlpla.  
              ls\_status-current\_pos = cs\_tele\_rcv-crane\_pos.  
              ls\_crane\_log-current\_pos = cs\_tele\_rcv-crane\_pos.  
              ls\_status-log\_date = sy-datum.  
              ls\_status-log\_time = sy-uzeit.  
            ENDIF.  
            IF cs\_tele\_rcv-return\_code IS NOT INITIAL.  
              DATA(lv\_error) = cs\_tele\_rcv-return\_code+0(2).  
              IF lv\_error NE '00'.  
                SELECT SINGLE zvalue INTO @DATA(lv\_delay) FROM zcon\_param\_val WHERE lgnum = @iv\_lgnum AND zreq\_name = 'DASHBOARD\_DELAY\_UPDATE' AND zparam = 'DASHBOARD\_DELAY\_PARAM' AND zactive = @abap\_true.  
                IF sy-subrc = 0.  
  
                ENDIF.  
                ls\_status-error = cs\_tele\_rcv-return\_code.  
                ls\_crane\_log-error = cs\_tele\_rcv-return\_code.  
                ls\_crane\_log-log\_date = sy-datum.  
                ls\_crane\_log-log\_time = sy-uzeit.  
                APPEND ls\_crane\_log TO lt\_crane\_log.  
                MODIFY zcrane\_error\_log FROM TABLE lt\_crane\_log.  
                COMMIT WORK.  
              ENDIF.  
  
              APPEND ls\_status TO lt\_status.  
  
              MODIFY zcrane\_dashboard FROM TABLE lt\_status.  
              COMMIT WORK.  
              UPDATE zcrane\_dashboard SET crane\_mode = cs\_tele\_rcv-crane\_mode WHERE lgnum = iv\_lgnum AND crane\_no = cs\_tele\_rcv-crane\_number.  
\*              IF cs\_tele\_rcv-crane\_mode NE '01'.  
\*                UPDATE zcon\_param\_val SET zactive = abap\_true  
\*                                           WHERE lgnum = iv\_lgnum  
\*                                           AND zreq\_name = 'CRANE\_STATUS'  
\*                                           AND  zvalue = cs\_tele\_rcv-crane\_number.  
\*              ELSE.  
\*                UPDATE zcon\_param\_val SET zactive = abap\_false  
\*                                      WHERE lgnum = iv\_lgnum  
\*                                      AND zreq\_name = 'CRANE\_STATUS'  
\*                                      AND  zvalue = cs\_tele\_rcv-crane\_number.  
\*  
\*              ENDIF.  
              COMMIT WORK.  
              WAIT UP TO lv\_delay SECONDS.  
            ENDIF.  
  
          ENDIF.  
          SELECT SINGLE zvalue INTO @DATA(lv\_update) FROM zcon\_param\_val WHERE lgnum EQ @iv\_lgnum AND zreq\_name EQ 'DASHBOARD\_UPDATE' AND zparam EQ 'DASHBOARD\_PARAM' AND zactive EQ @abap\_true.  
          IF sy-subrc = 0.  
            MODIFY zcrane\_dashboard FROM TABLE lt\_status.  
            COMMIT WORK.  
            UPDATE zcrane\_dashboard SET crane\_mode = cs\_tele\_rcv-crane\_mode WHERE lgnum = iv\_lgnum AND crane\_no = cs\_tele\_rcv-crane\_number.  
            COMMIT WORK.  
            WAIT UP TO lv\_delay SECONDS.  
          ENDIF.  
          SELECT SINGLE zactive FROM zcon\_param\_val INTO @DATA(lv\_zactive1) WHERE lgnum = @iv\_lgnum AND zreq\_name = 'ZMBRK'.  
          IF lv\_zactive1 EQ abap\_true.  
            ls\_crane\_mode-lgnum = iv\_lgnum.  
            ls\_crane\_mode-crane\_no = cs\_tele\_rcv-crane\_number.  
            ls\_crane\_mode-crane\_status = ls\_status-crane\_m\_des.  
            ls\_crane\_mode-log\_date = sy-datum.  
            ls\_crane\_mode-log\_time = sy-uzeit.  
            APPEND ls\_crane\_mode TO lt\_crane\_mode.  
            MODIFY zcrane\_mode\_log FROM TABLE lt\_crane\_mode.  
            COMMIT WORK.  
            IF cs\_tele\_rcv-crane\_mode EQ lc\_crane\_mode\_2 OR cs\_tele\_rcv-crane\_mode EQ lc\_crane\_mode\_3.  
              IF iv\_lgnum = 'KHFG'.  
                CONCATENATE 'SWLG30' cs\_tele\_rcv-crane\_number INTO lv\_rsrc\_name.  
              ELSEIF iv\_lgnum = 'KHPM'.  
                CONCATENATE 'SWISCRN' cs\_tele\_rcv-crane\_number INTO lv\_rsrc\_name.  
              ENDIF.  
              UPDATE /scwm/rsrc SET exccode\_overall = 'MBRK' WHERE lgnum = iv\_lgnum AND rsrc = lv\_rsrc\_name.  
              COMMIT WORK.  
            ENDIF.  
          ENDIF.  
        ENDIF.  
  
  
        IF  cs\_tele\_rcv-teletype = zcl\_common\_constants=>gc\_plc\_swslg\_dua  .  
  
          DELETE FROM zack\_telelog WHERE lgnum = iv\_lgnum.  
  
          ls\_log-lgnum = iv\_lgnum.  
          ls\_log-last\_time = sy-uzeit.  
          APPEND  ls\_log TO lt\_log.  
  
          MODIFY zack\_telelog FROM  ls\_log.  
          COMMIT WORK.  
        ENDIF.  
  
        cs\_tele\_rcv-handshake = 'AY'.  
        cs\_tele\_rcv-sender = iv\_plc.  
        cs\_tele\_rcv-receiver = 'MOVE01'.  
        cs\_tele\_rcv-startchar = '<'.  
        cs\_tele\_rcv-endchar = '>'.  
  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
\* ACP Telegram  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
        IF cs\_tele\_rcv-teletype EQ zcl\_common\_constants=>gc\_swslg\_tele\_acp.  
  
          lv\_tanum = cs\_tele\_rcv-assignment\_id.  
          CALL FUNCTION '/SCWM/TO\_READ\_SINGLE'  
            EXPORTING  
              iv\_lgnum     = iv\_lgnum  
              iv\_tanum     = lv\_tanum  
              iv\_flglock   = abap\_false  
            IMPORTING  
              es\_ordim\_o   = ls\_ordim\_o  
              et\_ordim\_c   = lt\_ordim\_c  
            EXCEPTIONS  
              wrong\_input  = 1  
              not\_found    = 2  
              foreign\_lock = 3  
              error        = 4  
              OTHERS       = 5.  
          IF sy-subrc EQ 0.  
            IF cs\_tele\_rcv-cnf\_flag EQ 1.  
              ls\_telegram-startchar = '<'.  
              ls\_telegram-cnf\_flag = '0'.  
              ls\_telegram-sequ\_no = 0000.  
              ls\_telegram-receiver = 'CRAN30'.  
              ls\_telegram-sender =  'MOVE01'.  
              ls\_telegram-teletype = 'ACK'.  
              ls\_telegram-fault\_indicator = '0'.  
              ls\_telegram-ack\_sequ\_no = cs\_tele\_rcv-sequ\_no+16(4).  
              ls\_telegram-crc = '00'.  
              ls\_telegram-endchar = '>'.  
              CALL FUNCTION 'ZFM\_SND\_ACK'  
                EXPORTING  
                  iv\_lgnum    = iv\_lgnum  
                  iv\_plc      = iv\_plc  
                  iv\_channel  = iv\_channel  
                  is\_telegram = ls\_telegram.  
  
  
            ENDIF.  
  
            IF ls\_ordim\_o IS NOT INITIAL.  
              cs\_tele\_rcv-rsrc    = ls\_ordim\_o-prsrc.  
              cs\_tele\_rcv-source  = ls\_ordim\_o-vlpla.  
              cs\_tele\_rcv-huident = ls\_ordim\_o-nlenr.  
              cs\_tele\_rcv-dest = ls\_ordim\_o-nlpla.  
            ENDIF.  
  
            SORT lt\_ordim\_c DESCENDING BY created\_at.  
            DELETE lt\_ordim\_c WHERE prsrc IS INITIAL.  
            DELETE lt\_ordim\_c WHERE prsrc CN 'SWSLG'.  
            CLEAR:ls\_ordim\_c.  
            READ TABLE lt\_ordim\_c INTO ls\_ordim\_c INDEX 1.  
            IF sy-subrc EQ 0.  
              cs\_tele\_rcv-rsrc    = ls\_ordim\_o-prsrc.  
              cs\_tele\_rcv-source  = ls\_ordim\_o-vlpla.  
              cs\_tele\_rcv-huident = ls\_ordim\_o-nlenr.  
              cs\_tele\_rcv-dest = ls\_ordim\_o-nlpla.  
            ENDIF.  
  
          ENDIF.  
        ENDIF.  
  
\*\*\*\*\*\*\*\*\*\*\*\*NCK Telegram  
        IF cs\_tele\_rcv-teletype  EQ zcl\_common\_constants=>gc\_swslg\_tele\_nck.  
  
          ls\_telegram-startchar = '<'.  
          ls\_telegram-cnf\_flag = '0'.  
          ls\_telegram-sequ\_no = '0000'.  
          ls\_telegram-receiver = 'CRAN30'.  
          ls\_telegram-sender =  'MOVE01'.  
          ls\_telegram-teletype = 'SYN'.  
          ls\_telegram-fault\_indicator = '0'.  
          ls\_telegram-ack\_sequ\_no = cs\_tele\_rcv-sequ\_no+16(4).  
          ls\_telegram-endchar = '>'.  
  
          CALL FUNCTION 'ZFM\_SND\_SYN'  
            EXPORTING  
              iv\_lgnum    = iv\_lgnum  
              iv\_plc      = iv\_plc  
              iv\_channel  = iv\_channel  
              is\_telegram = ls\_telegram.  
  
        ENDIF.  
  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
\* CSR Telegram  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
        IF cs\_tele\_rcv-teletype EQ zcl\_common\_constants=>gc\_swslg\_tele\_csr.  
          ls\_status-error = cs\_tele\_rcv-return\_code.  
          CONCATENATE '\*' cs\_tele\_rcv-crane\_number INTO lv\_rsrc\_pattern.  
  
\*          IF cs\_tele\_rcv-crane\_mode NE '01'.  
\*            UPDATE zcon\_param\_val SET zactive = abap\_true  
\*                                       WHERE lgnum = iv\_lgnum  
\*                                       AND zreq\_name = 'CRANE\_STATUS'  
\*                                       AND  zvalue = cs\_tele\_rcv-crane\_number.  
\*          ELSE.  
\*            UPDATE zcon\_param\_val SET zactive = abap\_false  
\*                                  WHERE lgnum = iv\_lgnum  
\*                                  AND zreq\_name = 'CRANE\_STATUS'  
\*                                  AND  zvalue = cs\_tele\_rcv-crane\_number.  
\*  
\*          ENDIF.  
\*          IF sy-subrc = 0.  
\*            COMMIT WORK.  
\*          ENDIF.  
  
          SELECT \* FROM /scwm/rsrc  
            INTO TABLE @lt\_rsrc  
            WHERE lgnum = @iv\_lgnum AND  
            rsrc\_type EQ @zcl\_common\_constants=>gc\_rsrc\_type\_swisslog.  
          IF sy-subrc EQ 0.  
            LOOP AT lt\_rsrc INTO ls\_rsrc.  
              IF ls\_rsrc-rsrc CP lv\_rsrc\_pattern.  
                cs\_tele\_rcv-rsrc = ls\_rsrc-rsrc.  
                EXIT.  
              ENDIF.  
            ENDLOOP.  
          ENDIF.  
  
          IF cs\_tele\_rcv-crane\_mode NE '1'.  
            cs\_tele\_rcv-mfs\_error = ''.  
          ENDIF.  
  
          IF cs\_tele\_rcv-cnf\_flag EQ 1.  
            ls\_telegram-startchar = '<'.  
            ls\_telegram-cnf\_flag = '0'.  
            ls\_telegram-receiver = 'CRAN30'.  
            ls\_telegram-sender =  'MOVE01'.  
            ls\_telegram-teletype = 'ACK'.  
            ls\_telegram-fault\_indicator = '0'.  
            ls\_telegram-ack\_sequ\_no = cs\_tele\_rcv-sequ\_no+16(4).  
            ls\_telegram-crc = '00'.  
            ls\_telegram-endchar = '>'.  
            CALL FUNCTION 'ZFM\_SND\_ACK'  
              EXPORTING  
                iv\_lgnum    = iv\_lgnum  
                iv\_plc      = iv\_plc  
                iv\_channel  = iv\_channel  
                is\_telegram = ls\_telegram.  
          ENDIF.  
  
        ENDIF.  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
\* DEC Telegram  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
        IF cs\_tele\_rcv-teletype EQ zcl\_common\_constants=>gc\_swslg\_tele\_dec.  
  
          lv\_tanum = cs\_tele\_rcv-assignment\_id.  
          CALL FUNCTION '/SCWM/TO\_READ\_SINGLE'  
            EXPORTING  
              iv\_lgnum     = iv\_lgnum  
              iv\_tanum     = lv\_tanum  
              iv\_flglock   = abap\_false  
            IMPORTING  
              es\_ordim\_o   = ls\_ordim\_o  
            EXCEPTIONS  
              wrong\_input  = 1  
              not\_found    = 2  
              foreign\_lock = 3  
              error        = 4  
              OTHERS       = 5.  
          IF sy-subrc EQ 0.  
            IF ls\_ordim\_o IS NOT INITIAL.  
  
              cs\_tele\_rcv-rsrc    = ls\_ordim\_o-prsrc.  
              cs\_tele\_rcv-source  = ls\_ordim\_o-vlpla.  
              cs\_tele\_rcv-huident = ls\_ordim\_o-nlenr.  
              cs\_tele\_rcv-dest = ls\_ordim\_o-nlpla.  
  
            ENDIF.  
  
          ENDIF.  
          IF cs\_tele\_rcv-cnf\_flag EQ 1.  
\*            DATA: ls\_telegram TYPE /scwm/s\_mfs\_teletotal.  
  
            ls\_telegram-startchar = '<'.  
            ls\_telegram-cnf\_flag = '0'.  
            ls\_telegram-sequ\_no = 0000.  
            ls\_telegram-receiver = 'CRAN30'.  
            ls\_telegram-sender =  'MOVE01'.  
            ls\_telegram-teletype = 'ACK'.  
            ls\_telegram-fault\_indicator = '0'.  
            ls\_telegram-ack\_sequ\_no = cs\_tele\_rcv-sequ\_no+16(4).  
            ls\_telegram-crc = '00'.  
            ls\_telegram-endchar = '>'.  
            CALL FUNCTION 'ZFM\_SND\_ACK'  
              EXPORTING  
                iv\_lgnum    = iv\_lgnum  
                iv\_plc      = iv\_plc  
                iv\_channel  = iv\_channel  
                is\_telegram = ls\_telegram.  
          ENDIF.  
        ENDIF.  
      WHEN zcl\_common\_constants=>gc\_plc\_daifuku.  
  
        cs\_tele\_rcv-sender = 'AGC'.  
        cs\_tele\_rcv-receiver = 'EWM'.  
        cs\_tele\_rcv-handshake = 'AY'.  
  
        cs\_tele\_rcv-huident = cs\_tele\_rcv-bc\_data.  
  
        IF cs\_tele\_rcv-teletype = '26'.  
          cs\_tele\_rcv-cp = cs\_tele\_rcv-station\_no.  
        ENDIF.  
  
        IF cs\_tele\_rcv-teletype = '33'.  
          CLEAR lv\_tanum.  
          lv\_tanum = cs\_tele\_rcv-mc\_key.  
          CALL FUNCTION '/SCWM/TO\_READ\_SINGLE'  
            EXPORTING  
              iv\_lgnum     = iv\_lgnum  
              iv\_tanum     = lv\_tanum  
              iv\_flglock   = ' '  
            IMPORTING  
              es\_ordim\_o   = ls\_ordim\_o  
            EXCEPTIONS  
              wrong\_input  = 1  
              not\_found    = 2  
              foreign\_lock = 3  
              error        = 4  
              OTHERS       = 5.  
          IF sy-subrc = 0 AND ls\_ordim\_o-procty NE '2010'.  
            cs\_tele\_rcv-source  = ls\_ordim\_o-vlpla.  
\*          cs\_tele\_rcv-cp      = ls\_ordim\_o-nltyp.  
            cs\_tele\_rcv-dest    = ls\_ordim\_o-nltyp.  
            cs\_tele\_rcv-huident = ls\_ordim\_o-nlenr.  
          ELSE.  
            cs\_tele\_rcv-rsrc    = ls\_ordim\_o-prsrc.  
            cs\_tele\_rcv-source  = ls\_ordim\_o-vltyp.  
\*          cs\_tele\_rcv-cp      = ls\_ordim\_o-nltyp.  
            cs\_tele\_rcv-dest    = ls\_ordim\_o-nlpla+0(4).  
            cs\_tele\_rcv-huident = ls\_ordim\_o-nlenr.  
          ENDIF.  
  
          IF cs\_tele\_rcv-completion\_type = '1'.  
            cs\_tele\_rcv-mfs\_error = '01'."location full  
          ELSE.  
            IF cs\_tele\_rcv-completion\_type = '2'.  
              cs\_tele\_rcv-mfs\_error = '02'."location empty.  
            ENDIF.  
          ENDIF.  
        ENDIF.  
  
        IF cs\_tele\_rcv-teletype = '68'.  
          CLEAR: lv\_tanum, ls\_ordim\_o.  
          lv\_tanum = cs\_tele\_rcv-mc\_key.  
          CALL FUNCTION '/SCWM/TO\_READ\_SINGLE'  
            EXPORTING  
              iv\_lgnum     = iv\_lgnum  
              iv\_tanum     = lv\_tanum  
              iv\_flglock   = ' '  
            IMPORTING  
              es\_ordim\_o   = ls\_ordim\_o  
            EXCEPTIONS  
              wrong\_input  = 1  
              not\_found    = 2  
              foreign\_lock = 3  
              error        = 4  
              OTHERS       = 5.  
          IF sy-subrc <> 0.  
\* Implement suitable error handling here  
          ENDIF.  
          cs\_tele\_rcv-rsrc    = ls\_ordim\_o-prsrc.  
          cs\_tele\_rcv-source  = ls\_ordim\_o-vlpla.  
\*          cs\_tele\_rcv-cp      = ls\_ordim\_o-nltyp.  
          cs\_tele\_rcv-dest    = ls\_ordim\_o-nlpla.  
          cs\_tele\_rcv-huident = ls\_ordim\_o-nlenr.  
        ENDIF.  
  
  
        IF cs\_tele\_rcv-teletype = '23'.  
  
\*   stop communication channel  
          DATA: lv\_lines TYPE sytabix,  
                lv\_mtext TYPE string.  
          DATA: lo\_log             TYPE REF TO /scwm/cl\_log.  
          DATA: lx\_mfs             TYPE REF TO /scwm/cx\_mfs.  
          FIELD-SYMBOLS: <fs\_tele> TYPE /scwm/s\_mfscch\_mon.  
          IF cs\_tele\_rcv-response\_type = '00'.  
            CREATE OBJECT lo\_log.  
            TRY.  
                CALL FUNCTION '/SCWM/MFS\_STOP\_CCH'  
                  EXPORTING  
                    iv\_lgnum   = iv\_lgnum  
                    iv\_plc     = iv\_plc  
                    iv\_channel = iv\_channel.  
              CATCH /scwm/cx\_mfs INTO lx\_mfs.  
                MESSAGE ID lx\_mfs->if\_t100\_message~t100key-msgid  
                        TYPE lx\_mfs->mv\_msgty  
                        NUMBER lx\_mfs->if\_t100\_message~t100key-msgno  
                        WITH lx\_mfs->mv\_msgv1 lx\_mfs->mv\_msgv2  
                             lx\_mfs->mv\_msgv3 lx\_mfs->mv\_msgv4  
                        INTO lv\_mtext.  
                lo\_log->add\_message( ).  
            ENDTRY.  
  
          ENDIF.  
        ENDIF.  
  
        IF cs\_tele\_rcv-teletype = '61'..  
  
          CLEAR: ps\_telegram.  
          ps\_telegram-sequ\_no = '0000'.  
          ps\_telegram-teletype = '41'.  
          ps\_telegram-id\_section = '00'.  
          ps\_telegram-mc\_trans\_time = sy-uzeit.  
          ps\_telegram-agc\_trans\_time = '000000'.  
          ps\_telegram-station\_no = cs\_tele\_rcv-station\_no.  
          ps\_telegram-bcc = '00'.  
  
          IF cs\_tele\_rcv-station\_no = '1301' .  
            ps\_telegram-request\_rspnc = '00'.  
          ELSE.  
            ps\_telegram-request\_rspnc = '02'.  
          ENDIF.  
  
          CALL FUNCTION '/SCWM/MFS\_TELE\_MAP'  
            EXPORTING  
              iv\_lgnum    = iv\_lgnum  
              iv\_plc      = iv\_plc  
              iv\_plc2ewm  = abap\_false  
              is\_telegram = ps\_telegram  
            IMPORTING  
              es\_telegram = ps\_telegram.  
  
          TRY .  
              CALL FUNCTION '/SCWM/MFS\_SEND'  
                EXPORTING  
                  iv\_lgnum           = iv\_lgnum  
                  iv\_plc             = iv\_plc  
                  iv\_channel         = iv\_channel  
                  iv\_no\_change       = 'X'  
                  is\_telegram        = ps\_telegram  
                IMPORTING  
                  ev\_wt\_already\_sent = lv\_already\_sent.  
  
            CATCH /scwm/cx\_mfs INTO lr\_ex\_mfs.  
          ENDTRY.  
        ENDIF.  
  
  
        IF cs\_tele\_rcv-teletype = '22'.  
  
          CLEAR: ps\_telegram.  
          ps\_telegram-sequ\_no = '0000'.  
          ps\_telegram-teletype = '02'.  
          ps\_telegram-id\_section = '00'.  
          ps\_telegram-mc\_trans\_time = sy-uzeit.  
          ps\_telegram-agc\_trans\_time = '000000'.  
          ps\_telegram-date\_agc = sy-datum.  
          ps\_telegram-time = sy-uzeit.  
          ps\_telegram-bcc = '00'.  
  
          CALL FUNCTION '/SCWM/MFS\_TELE\_MAP'  
            EXPORTING  
              iv\_lgnum    = iv\_lgnum  
              iv\_plc      = iv\_plc  
              iv\_plc2ewm  = abap\_false  
              is\_telegram = ps\_telegram  
            IMPORTING  
              es\_telegram = ps\_telegram.  
  
          TRY .  
              CALL FUNCTION '/SCWM/MFS\_SEND'  
                EXPORTING  
                  iv\_lgnum           = iv\_lgnum  
                  iv\_plc             = iv\_plc  
                  iv\_channel         = iv\_channel  
                  iv\_no\_change       = 'X'  
                  is\_telegram        = ps\_telegram  
                IMPORTING  
                  ev\_wt\_already\_sent = lv\_already\_sent.  
  
            CATCH /scwm/cx\_mfs INTO lr\_ex\_mfs.  
          ENDTRY.  
        ENDIF.  
  
        IF cs\_tele\_rcv-teletype = '27'..  
  
          CLEAR: ps\_telegram.  
          ps\_telegram-sequ\_no = '0000'.  
          ps\_telegram-teletype = '08'.  
          ps\_telegram-id\_section = '00'.  
          ps\_telegram-mc\_trans\_time = sy-uzeit.  
          ps\_telegram-agc\_trans\_time = '000000'.  
          ps\_telegram-mc\_key = cs\_tele\_rcv-mc\_key.  
          ps\_telegram-command\_type = '1'.  
          ps\_telegram-location\_no = '000000000000'.  
          ps\_telegram-station\_no = cs\_tele\_rcv-station\_no.  
          ps\_telegram-bcc = '00'.  
  
          CALL FUNCTION '/SCWM/MFS\_TELE\_MAP'  
            EXPORTING  
              iv\_lgnum    = iv\_lgnum  
              iv\_plc      = iv\_plc  
              iv\_plc2ewm  = abap\_false  
              is\_telegram = ps\_telegram  
            IMPORTING  
              es\_telegram = ps\_telegram.  
  
          TRY .  
              CALL FUNCTION '/SCWM/MFS\_SEND'  
                EXPORTING  
                  iv\_lgnum           = iv\_lgnum  
                  iv\_plc             = iv\_plc  
                  iv\_channel         = iv\_channel  
                  iv\_no\_change       = 'X'  
                  is\_telegram        = ps\_telegram  
                IMPORTING  
                  ev\_wt\_already\_sent = lv\_already\_sent.  
  
            CATCH /scwm/cx\_mfs INTO lr\_ex\_mfs.  
          ENDTRY.  
        ENDIF.  
  
  
  
  
  
  
      WHEN OTHERS.  
    ENDCASE.  
  
  
  
  ENDMETHOD.

A screenshot of a computer

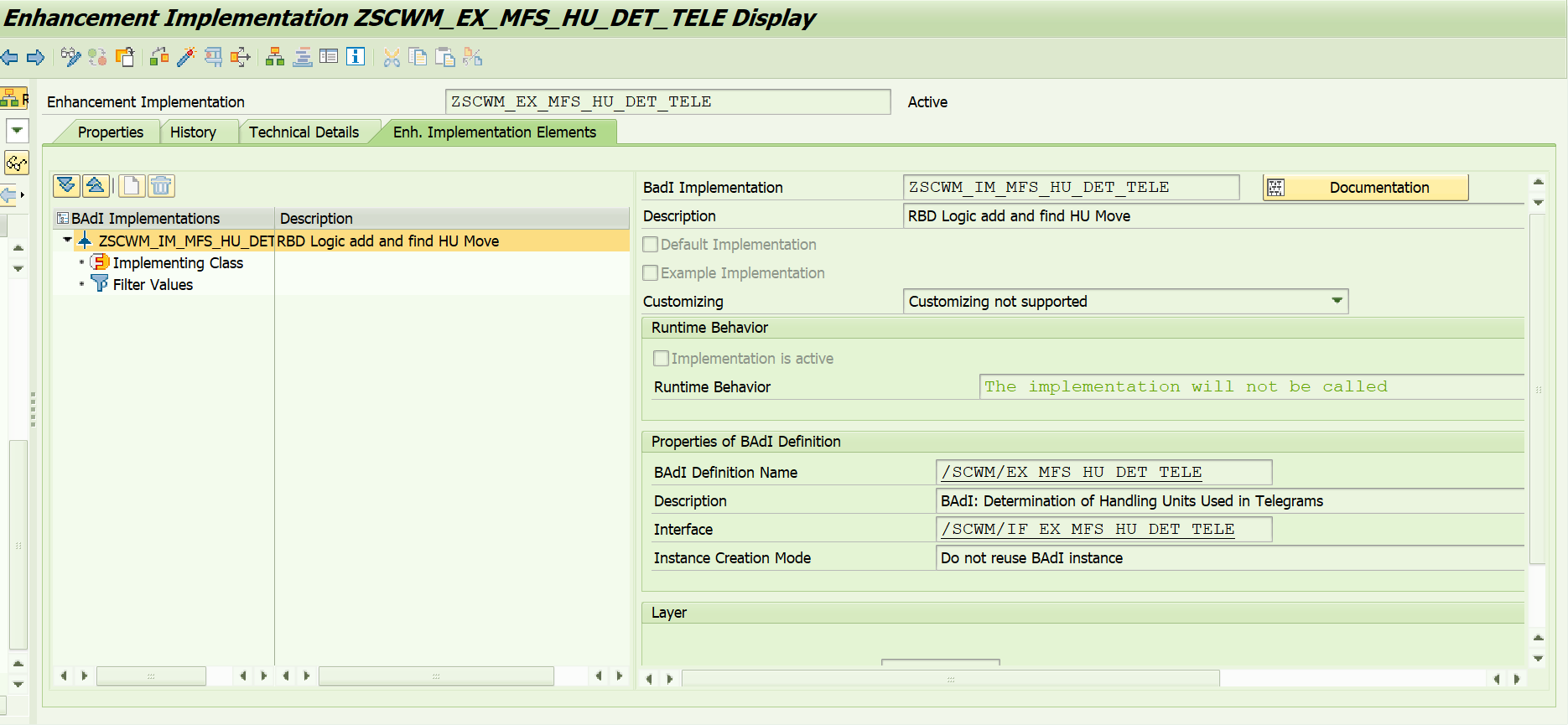
Description automatically generated

A screenshot of a computer

Description automatically generated

  METHOD /scwm/if\_ex\_mfs\_tele\_snd~tele\_snd.  
  
    DATA: lr\_teletype\_cnf\_req     TYPE RANGE OF /scwm/de\_mfsteletype,  
          lr\_teletype\_cnf\_not\_req TYPE RANGE OF /scwm/de\_mfsteletype,  
          ls\_telegram\_ack         TYPE /scwm/s\_mfs\_teletotal,  
          lv\_telegram\_raw         TYPE /scwm/de\_mfstele,  
          ls\_tmfstelto            TYPE /scwm/tmfstelto,  
          lv\_error                TYPE xfeld,  
          ls\_teletype\_cnf         LIKE LINE OF lr\_teletype\_cnf\_req.  
    DATA: lt\_ordim\_o   TYPE TABLE OF /scwm/ordim\_o,  
          ls\_ordim\_o   TYPE /scwm/ordim\_o,  
          lv\_agc\_aisle TYPE /scwm/de\_aisle,  
          ls\_lagp      TYPE /scwm/lagp,  
          ls\_con\_param TYPE  zcon\_param\_val,  
          ls\_tmfscch   TYPE /scwm/tmfscch.  
    DATA: lv\_tempdata TYPE c LENGTH 9.  
    DATA: lv\_tempdata\_1 TYPE c LENGTH 12.  
    DATA: lv\_mm(2)  TYPE c,  
          lv\_rrr(3) TYPE c,  
          lv\_sss(3) TYPE c,  
          lv\_hh(2)  TYPE c,  
          lv\_dd(2)  TYPE c,  
          lv\_aisle  TYPE i.  
  
    DATA: lv\_stx(1) TYPE c  ,  "ASCII character for STX - start of text  
          lv\_etx(1) TYPE c  .  "ASCII character for STX - start of text  
    DATA: lv\_unicode\_stx TYPE x VALUE 2,   "Unicode decimal value for STX is 02  
          lv\_unicode\_etx TYPE x VALUE 3.   "Unicode decimal value for STX is 03  
  
    CONSTANTS: lc\_de\_prcty    TYPE /scwm/ordim\_o-procty VALUE '5010',  
               lc\_req\_clear   TYPE zcon\_param\_val-zreq\_name VALUE 'CLEAR\_BARCODE\_REQ',  
               lc\_param\_clear TYPE zcon\_param\_val-zparam    VALUE 'CLEAR\_BARCODE\_PARAM'.  
  
\*Channel read  
    TRY.  
        CALL FUNCTION '/SCWM/TMFSCCH\_READ\_SINGLE'  
          EXPORTING  
            iv\_lgnum   = iv\_lgnum  
            iv\_plc     = iv\_plc  
            iv\_channel = iv\_channel  
\*           IV\_NOBUF   = ' '  
          IMPORTING  
            es\_tmfscch = ls\_tmfscch.  
      CATCH /scwm/cx\_core.  
      CATCH /scwm/cx\_mfs.  
    ENDTRY.  
  
  
\*    BREAK-POINT ID  zapl\_mfs\_tele.  
    CASE iv\_plc.  
      WHEN zcl\_common\_constants=>gc\_plc\_swslg.  
  
        cs\_tele\_snd-crc = '00'.  
  
        IF is\_ordim\_o-tanum+4(8) IS NOT INITIAL.  
          cs\_tele\_snd-assignment\_id = is\_ordim\_o-tanum+4(8).  
        ENDIF.  
        IF is\_ordim\_o-prsrc+6(2) IS NOT INITIAL.  
          cs\_tele\_snd-crane\_number = is\_ordim\_o-prsrc+6(2).  
        ENDIF.  
  
        SELECT \* INTO TABLE lt\_ordim\_o FROM /scwm/ordim\_o WHERE lgnum = iv\_lgnum AND tanum = cs\_tele\_snd-assignment\_id.  
        IF sy-subrc = 0.  
          READ TABLE lt\_ordim\_o INTO ls\_ordim\_o WITH KEY tanum  = cs\_tele\_snd-assignment\_id.  
          IF sy-subrc = 0.  
  
          ENDIF.  
        ENDIF.  
\* Confirmation Not required  
  
        ls\_teletype\_cnf-sign   = 'I'.  
        ls\_teletype\_cnf-option = 'EQ'.  
        DO 6 TIMES.  
          CLEAR ls\_teletype\_cnf-low.  
          CASE sy-index.  
            WHEN 1.  
              ls\_teletype\_cnf-low = zcl\_common\_constants=>gc\_swslg\_tele\_ack.  
            WHEN 2.  
              ls\_teletype\_cnf-low = zcl\_common\_constants=>gc\_swslg\_tele\_nck.  
            WHEN 3.  
              ls\_teletype\_cnf-low = zcl\_common\_constants=>gc\_swslg\_tele\_syn.  
            WHEN 4.  
              ls\_teletype\_cnf-low = zcl\_common\_constants=>gc\_swslg\_tele\_sta.  
            WHEN 5.  
              ls\_teletype\_cnf-low = zcl\_common\_constants=>gc\_swslg\_tele\_sto.  
            WHEN 6.  
              ls\_teletype\_cnf-low = zcl\_common\_constants=>gc\_swslg\_tele\_dum.  
          ENDCASE.  
          APPEND ls\_teletype\_cnf TO lr\_teletype\_cnf\_not\_req.  
        ENDDO.  
\* Confirmation required  
        DO 3 TIMES.  
          CLEAR ls\_teletype\_cnf-low.  
          CASE sy-index.  
            WHEN 1.  
              ls\_teletype\_cnf-low = zcl\_common\_constants=>gc\_swslg\_tele\_arq.  
            WHEN 2.  
              ls\_teletype\_cnf-low = zcl\_common\_constants=>gc\_swslg\_tele\_der.  
            WHEN 3.  
              ls\_teletype\_cnf-low = zcl\_common\_constants=>gc\_swslg\_tele\_crq.  
          ENDCASE.  
          APPEND ls\_teletype\_cnf TO lr\_teletype\_cnf\_req.  
        ENDDO.  
  
        IF cs\_tele\_snd-teletype IN lr\_teletype\_cnf\_req.  
  
          cs\_tele\_snd-cnf\_flag = zcl\_common\_constants=>gc\_swslg\_cnf\_flg\_req.  
  
        ELSEIF cs\_tele\_snd-teletype IN lr\_teletype\_cnf\_not\_req.  
  
          cs\_tele\_snd-cnf\_flag = zcl\_common\_constants=>gc\_swslg\_cnf\_flg\_not\_req.  
          cs\_tele\_snd-sequ\_no = '0000'.  
  
          IF cs\_tele\_snd-teletype = zcl\_common\_constants=>gc\_swslg\_tele\_dum.  
            cs\_tele\_snd-cnf\_flag = '0'.  
          ENDIF.  
  
        ENDIF.  
  
\*-- ARQ telegram  
        IF cs\_tele\_snd-teletype EQ zcl\_common\_constants=>gc\_swslg\_tele\_arq.  
  
          cs\_tele\_snd-assignment\_type = 'CC'.  
          cs\_tele\_snd-tu\_type = '02'.  
  
          IF is\_ordim\_o-vltyp EQ zcl\_common\_constants=>gc\_swslg\_lgtyp.  
  
            CLEAR: lv\_mm,lv\_rrr,lv\_sss,lv\_hh,lv\_dd.  
            TRY.  
                CALL FUNCTION '/SCWM/LAGP\_READ\_SINGLE'  
                  EXPORTING  
                    iv\_lgnum      = iv\_lgnum  
                    iv\_lgpla      = is\_ordim\_o-vlpla  
                  IMPORTING  
                    es\_lagp       = ls\_lagp  
                  EXCEPTIONS  
                    wrong\_input   = 1  
                    not\_found     = 2  
                    enqueue\_error = 3  
                    OTHERS        = 4.  
              CATCH /scwm/cx\_core.  
              CATCH /scwm/cx\_mfs.  
            ENDTRY.  
            IF sy-subrc EQ 0.  
  
              lv\_mm = '30'.  
  
              lv\_aisle = ls\_lagp-lgpla+1(2).  
              IF ls\_lagp-lgpla+3(1) EQ 'L'.  
\*                lv\_rrr = lv\_aisle  \* 2.  
                lv\_rrr =  ( lv\_aisle  \* 2 ) - 1.  
                CALL FUNCTION 'CONVERSION\_EXIT\_ALPHA\_INPUT'  
                  EXPORTING  
                    input  = lv\_rrr  
                  IMPORTING  
                    output = lv\_rrr.  
  
              ELSEIF ls\_lagp-lgpla+3(1) EQ 'R'.  
\*                lv\_rrr =  ( lv\_aisle  \* 2 ) - 1.  
                lv\_rrr = lv\_aisle  \* 2.  
                CALL FUNCTION 'CONVERSION\_EXIT\_ALPHA\_INPUT'  
                  EXPORTING  
                    input  = lv\_rrr  
                  IMPORTING  
                    output = lv\_rrr.  
              ENDIF.  
  
              lv\_sss = ls\_lagp-lgpla+6(2).  
              CALL FUNCTION 'CONVERSION\_EXIT\_ALPHA\_INPUT'  
                EXPORTING  
                  input  = lv\_sss  
                IMPORTING  
                  output = lv\_sss.  
  
              lv\_hh = ls\_lagp-lgpla+4(2).  
  
              IF ls\_ordim\_o-slogpos EQ '01'.  
                lv\_dd = zcl\_common\_constants=>gc\_outer\_log\_pos.  
              ELSEIF ls\_ordim\_o-slogpos EQ '02'.  
                lv\_dd = zcl\_common\_constants=>gc\_inner\_log\_pos.  
              ENDIF.  
  
              CONCATENATE lv\_mm lv\_rrr lv\_sss lv\_hh lv\_dd INTO cs\_tele\_snd-start\_position.  
            ENDIF.  
          ELSE.  
            cs\_tele\_snd-start\_position = cs\_tele\_snd-source.  
          ENDIF.  
  
          IF is\_ordim\_o-nltyp EQ zcl\_common\_constants=>gc\_swslg\_lgtyp.  
            CLEAR: lv\_mm,lv\_rrr,lv\_sss,lv\_hh,lv\_dd.  
            TRY.  
                CALL FUNCTION '/SCWM/LAGP\_READ\_SINGLE'  
                  EXPORTING  
                    iv\_lgnum      = iv\_lgnum  
                    iv\_lgpla      = is\_ordim\_o-nlpla  
                  IMPORTING  
                    es\_lagp       = ls\_lagp  
                  EXCEPTIONS  
                    wrong\_input   = 1  
                    not\_found     = 2  
                    enqueue\_error = 3  
                    OTHERS        = 4.  
              CATCH /scwm/cx\_core.  
              CATCH /scwm/cx\_mfs.  
            ENDTRY.  
            IF sy-subrc EQ 0.  
              lv\_mm = '30'.  
  
              lv\_aisle = ls\_lagp-lgpla+1(2).  
              IF ls\_lagp-lgpla+3(1) EQ 'L'.  
\*                lv\_rrr = lv\_aisle  \* 2.  
                lv\_rrr =  ( lv\_aisle  \* 2 ) - 1.  
                "CONCATENATE '00' lv\_rrr INTO lv\_rrr.  
                CALL FUNCTION 'CONVERSION\_EXIT\_ALPHA\_INPUT'  
                  EXPORTING  
                    input  = lv\_rrr  
                  IMPORTING  
                    output = lv\_rrr.  
  
              ELSEIF ls\_lagp-lgpla+3(1) EQ 'R'.  
                lv\_rrr = lv\_aisle  \* 2.  
\*                lv\_rrr =  ( lv\_aisle  \* 2 ) - 1.  
                CALL FUNCTION 'CONVERSION\_EXIT\_ALPHA\_INPUT'  
                  EXPORTING  
                    input  = lv\_rrr  
                  IMPORTING  
                    output = lv\_rrr.  
              ENDIF.  
  
              lv\_sss = ls\_lagp-lgpla+6(2).  
              TRY.  
                  CALL FUNCTION 'CONVERSION\_EXIT\_ALPHA\_INPUT'  
                    EXPORTING  
                      input  = lv\_sss  
                    IMPORTING  
                      output = lv\_sss.  
                CATCH /scwm/cx\_core.  
                CATCH /scwm/cx\_mfs.  
              ENDTRY.  
  
              lv\_hh = ls\_lagp-lgpla+4(2).  
  
              IF ls\_ordim\_o-dlogpos EQ '01'.  
                lv\_dd = zcl\_common\_constants=>gc\_outer\_log\_pos.  
              ELSEIF ls\_ordim\_o-dlogpos EQ '02'.  
                lv\_dd = zcl\_common\_constants=>gc\_inner\_log\_pos.  
              ENDIF.  
  
              CONCATENATE lv\_mm lv\_rrr lv\_sss lv\_hh lv\_dd INTO cs\_tele\_snd-dest\_position.  
            ENDIF.  
          ELSE.  
            cs\_tele\_snd-dest\_position = cs\_tele\_snd-dest.  
          ENDIF.  
  
          SELECT SINGLE tanum INTO @DATA(lv\_de\_tanum) FROM /scwm/ordim\_o WHERE lgnum = @iv\_lgnum  AND  tanum = @cs\_tele\_snd-assignment\_id  
                                                                               AND procty = @lc\_de\_prcty.  
          IF sy-subrc EQ 0.  
            cs\_tele\_snd-start\_position = '000000000000'.  
            cs\_tele\_snd-assignment\_type = 'DE'.  
          ENDIF.  
          cs\_tele\_snd-fork = 'RE'.  
          cs\_tele\_snd-speed = 'HI'.  
          cs\_tele\_snd-rear\_fork\_side = 'FU'.  
          cs\_tele\_snd-front\_fork\_side = 'FU'.  
  
          SELECT SINGLE \* FROM  zcon\_param\_val  
      INTO CORRESPONDING FIELDS OF ls\_con\_param  
      WHERE  lgnum = 'RHPM'  
         AND  zreq\_name = 'BIN\_BLOCK\_UNBLOCK'  
           AND  zparam = 'BIN\_BLOCK\_UNBLOCK' AND zactive = abap\_true.  
          IF sy-subrc = 0.  
            UPDATE /scwm/lagp SET skzue = abap\_true WHERE lgpla = ls\_lagp-lgpla AND lgnum = iv\_lgnum.  
            COMMIT WORK.  
  
          ENDIF.  
        ENDIF.  
  
\*-- DER telegram  
        IF cs\_tele\_snd-teletype EQ zcl\_common\_constants=>gc\_swslg\_tele\_der.  
        ENDIF.  
\*-- CRQ telegram  
        IF cs\_tele\_snd-teletype EQ zcl\_common\_constants=>gc\_swslg\_tele\_crq.  
          cs\_tele\_snd-crane\_number = '05'.  
        ENDIF.  
  
\*-- STA telegram  
        IF cs\_tele\_snd-teletype EQ zcl\_common\_constants=>gc\_swslg\_tele\_sta.  
        ENDIF.  
  
\*-- STO Telegram  
        IF cs\_tele\_snd-teletype EQ zcl\_common\_constants=>gc\_swslg\_tele\_sto.  
        ENDIF.  
  
      WHEN zcl\_common\_constants=>gc\_plc\_miifgin OR  
      zcl\_common\_constants=>gc\_plc\_miifgout OR  
      zcl\_common\_constants=>gc\_plc\_miifgshi.  
  
        CHECK cs\_tele\_snd-teletype EQ 'PLRO'. " To send the data to MII only for the PLRO telegram  
  
        DATA : lo\_proxy\_mii       TYPE REF TO zmii\_connectionco\_xacute\_wssoa,   "zmiisrv\_co\_xacute\_wssoap,  
               ls\_input\_params    TYPE zmii\_connectioninput\_params,             "zmiisrv\_input\_params,  
               ls\_aacute\_soap\_in  TYPE zmii\_connectionaacute\_soap\_in,           "zmiisrv\_aacute\_soap\_in,  
               ls\_aacute\_soap\_out TYPE zmii\_connectionaacute\_soap\_out.          "zmiisrv\_aacute\_soap\_out.  
        DATA: lv\_port       TYPE prx\_logical\_port\_name,  
              lv\_tag\_name   TYPE zde\_tag\_name,  
              lv\_tag\_value  TYPE zde\_tag\_value,  
              lv\_tag\_string TYPE string.  
        CONSTANTS: lc\_req          TYPE zcon\_param\_val-zreq\_name VALUE 'MII\_LOGIN\_REQ',  
                   lc\_req\_port     TYPE zcon\_param\_val-zreq\_name VALUE 'MII\_LOGIN\_PORT',  
                   lc\_req\_plro\_ack TYPE zcon\_param\_val-zreq\_name VALUE 'MII\_PLRO\_ACK',  
                   lc\_param\_ack    TYPE zcon\_param\_val-zparam VALUE 'MII\_PLRO\_ACK\_NEW'.  
  
        ls\_input\_params-cpname = cs\_tele\_snd-cp.  
        ls\_input\_params-plant  = iv\_lgnum.  
  
        SELECT \* FROM zcp\_tag\_map  
        INTO TABLE @DATA(lt\_cp\_tag\_map)  
        WHERE lgnum = @iv\_lgnum  
        AND plc = @iv\_plc  
        AND channel = @iv\_channel  
        AND cp = @cs\_tele\_snd-cp.  
        IF sy-subrc EQ 0.  
  
          LOOP AT lt\_cp\_tag\_map ASSIGNING FIELD-SYMBOL(<fs\_cp\_tag\_map>) WHERE cp = cs\_tele\_snd-cp AND outb\_tele\_field IS NOT INITIAL.  
  
            ASSIGN COMPONENT <fs\_cp\_tag\_map>-outb\_tele\_field OF STRUCTURE cs\_tele\_snd TO FIELD-SYMBOL(<fs\_tele\_field>).  
            IF <fs\_tele\_field> IS ASSIGNED.  
              SELECT SINGLE \* FROM zmii\_dest\_tag WHERE lgnum EQ @iv\_lgnum AND cp EQ @cs\_tele\_snd-cp AND dest EQ @cs\_tele\_snd-dest  
                INTO @DATA(ls\_mii\_dest).  
              lv\_tag\_name = <fs\_cp\_tag\_map>-tag\_name.  
              lv\_tag\_value = ls\_mii\_dest-dest\_tag\_value. "<fs\_tele\_field>.  
              IF lv\_tag\_string IS INITIAL.  
                CONCATENATE lv\_tag\_name ',' lv\_tag\_value INTO lv\_tag\_string.  
              ELSE.  
                CONCATENATE lv\_tag\_string ';' lv\_tag\_name ',' lv\_tag\_value INTO lv\_tag\_string.  
              ENDIF.  
            ENDIF.  
  
          ENDLOOP.  
          READ TABLE lt\_cp\_tag\_map ASSIGNING <fs\_cp\_tag\_map> WITH KEY cp = cs\_tele\_snd-cp tag\_type = 'AOR'.  
          IF sy-subrc EQ 0.  
            lv\_tag\_name = <fs\_cp\_tag\_map>-tag\_name.  
            lv\_tag\_value = '0'. "ls\_mii\_dest-dest\_tag\_value.  
            IF lv\_tag\_string IS INITIAL.  
              CONCATENATE lv\_tag\_name ',' lv\_tag\_value INTO lv\_tag\_string.  
            ELSE.  
              CONCATENATE lv\_tag\_string ';' lv\_tag\_name ',' lv\_tag\_value INTO lv\_tag\_string.  
            ENDIF.  
          ENDIF.  
  
          READ TABLE lt\_cp\_tag\_map ASSIGNING <fs\_cp\_tag\_map> WITH KEY cp = cs\_tele\_snd-cp tag\_type = 'BARCODE'.  
          IF sy-subrc EQ 0.  
            lv\_tag\_name = <fs\_cp\_tag\_map>-tag\_name.  
            SELECT SINGLE zactive INTO @DATA(lv\_active) FROM zcon\_param\_val WHERE lgnum = @iv\_lgnum AND zreq\_name = @lc\_req\_clear AND zparam = @lc\_param\_clear AND zactive = @abap\_true.  
            IF sy-subrc EQ 0.  
              lv\_tag\_value = 0. "ls\_mii\_dest-dest\_tag\_value.  
            ELSE.  
              lv\_tag\_value = ' '. "ls\_mii\_dest-dest\_tag\_value.  
            ENDIF.  
  
            IF lv\_tag\_string IS INITIAL.  
              CONCATENATE lv\_tag\_name ',' lv\_tag\_value INTO lv\_tag\_string.  
            ELSE.  
              CONCATENATE lv\_tag\_string ';' lv\_tag\_name ',' lv\_tag\_value INTO lv\_tag\_string.  
            ENDIF.  
          ENDIF.  
  
        ENDIF.  
  
        SELECT \* INTO TABLE @DATA(lt\_req\_mii) FROM zmii\_credentials WHERE lgnum = @iv\_lgnum AND zactive EQ @abap\_true.  
        IF sy-subrc EQ 0.  
          READ TABLE lt\_req\_mii INTO DATA(ls\_req\_mii) INDEX 1.  
        ENDIF.  
        ls\_input\_params-input\_tags       = lv\_tag\_string.  
        ls\_aacute\_soap\_in-input\_params   = ls\_input\_params.  
        ls\_aacute\_soap\_in-login\_name     = ls\_req\_mii-zuserid.  
        ls\_aacute\_soap\_in-login\_password = ls\_req\_mii-zpassword.  
  
        TRY.  
            SELECT \* INTO TABLE @DATA(lt\_req\_port) FROM zcon\_param\_val WHERE lgnum = @iv\_lgnum AND zreq\_name = @lc\_req\_port AND zactive EQ @abap\_true.  
            IF sy-subrc = 0.  
              READ TABLE lt\_req\_port INTO DATA(ls\_req\_port) INDEX 1.  
            ENDIF.  
  
            lv\_port = ls\_req\_port-zvalue.  
            CREATE OBJECT lo\_proxy\_mii  
              EXPORTING  
                logical\_port\_name = lv\_port.  
          CATCH cx\_ai\_system\_fault.  
        ENDTRY.  
  
        TRY.  
            CALL METHOD lo\_proxy\_mii->xacute  
              EXPORTING  
                input  = ls\_aacute\_soap\_in  
              IMPORTING  
                output = ls\_aacute\_soap\_out.  
  
\*          CATCH cx\_ai\_system\_fault.  
\*          CATCH cx\_ai\_application\_fault.  
\*        ENDTRY.  
  
        DATA(ls\_mii\_send) = VALUE zmii\_send(  
                                             mandt = sy-mandt  
                                             log\_date = sy-datum  
                                             log\_time = sy-uzeit  
                                             lgnum    = iv\_lgnum  
                                             plc      = iv\_plc  
                                             channel  = iv\_channel  
                                             cp       = cs\_tele\_snd-cp  
                                             huident  = cs\_tele\_snd-huident  
                                             tags     = lv\_tag\_string  
                                             user\_name = sy-uname ).  
        MODIFY zmii\_send FROM ls\_mii\_send.  
  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Trigger Acknowlagement fro PLRO  
        SELECT SINGLE zactive INTO @DATA(lv\_active\_ack) FROM zcon\_param\_val  
                                    WHERE lgnum = @iv\_lgnum AND zreq\_name = @lc\_req\_plro\_ack  
                                    AND zparam = @lc\_param\_ack AND zactive = @abap\_true.  
        IF sy-subrc EQ 0.  
  
          CALL FUNCTION 'ZMFS\_TELE\_ACK' IN BACKGROUND TASK  
            DESTINATION 'NONE'  
            EXPORTING  
              iv\_lgnum    = iv\_lgnum  
              iv\_plc      = iv\_plc  
              iv\_channel  = iv\_channel  
              is\_telegram = cs\_tele\_snd.  
\*                IMPORTING  
\*                  es\_telegram = ls\_telegram\_ack.  
  
\*              CALL FUNCTION '/SCWM/MFS\_TELE\_ACK'  
\*                EXPORTING  
\*                  iv\_lgnum    = iv\_lgnum  
\*                  iv\_plc      = iv\_plc  
\*                  iv\_channel  = iv\_channel  
\*\*                 IV\_MFS\_ERROR       =  
\*                  is\_telegram = cs\_tele\_snd  
\*                IMPORTING  
\*                  es\_telegram = ls\_telegram\_ack.  
        ENDIF.  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
  
          CATCH cx\_ai\_system\_fault.  
          CATCH cx\_ai\_application\_fault.  
        ENDTRY.  
  
      WHEN zcl\_common\_constants=>gc\_plc\_daifuku.  
  
        lv\_stx = lv\_unicode\_stx.  
        lv\_etx = lv\_unicode\_etx.  
  
  
        CONSTANTS : c\_0(1) TYPE c VALUE '0'.  
        cs\_tele\_snd-mc\_trans\_time = sy-uzeit.  
        CONCATENATE c\_0 c\_0 c\_0 c\_0 c\_0 c\_0 INTO cs\_tele\_snd-agc\_trans\_time.  
\*      cs\_tele\_snd-AGC\_TRANS\_TIME = '000000'.  
        cs\_tele\_snd-date\_agc = sy-datum.  
        cs\_tele\_snd-time = sy-uzeit.  
  
        cs\_tele\_snd-bcc = '00'.  
        cs\_tele\_snd-agc\_trans\_time = '000000'.  
        cs\_tele\_snd-mc\_trans\_time = sy-uzeit.  
        cs\_tele\_snd-id\_section = '00'.  
        cs\_tele\_snd-group\_no = '000'.        " group should be always 000  
        cs\_tele\_snd-load\_size\_info = '000'.   " always 000  
        cs\_tele\_snd-startchar = ls\_tmfscch-startchar.         "Start of Text  STX = dec 2  
        cs\_tele\_snd-endchar = ls\_tmfscch-endchar.           "End of Text  ETX  = dec 3  
  
  
        IF is\_ordim\_o-vltyp = '9001' OR  
                is\_ordim\_o-vltyp = '9002' OR  
                is\_ordim\_o-vltyp = '9003' OR  
                is\_ordim\_o-vltyp = '9004' OR  
                is\_ordim\_o-vltyp = '9005'.  
          cs\_tele\_snd-teletype = '12'.  
        ENDIF.  
  
  
        IF cs\_tele\_snd-huident+10(10) IS NOT INITIAL.  
          cs\_tele\_snd-bc\_data = cs\_tele\_snd-huident+10(10).  
        ENDIF.  
  
  
        "SHIFT cs\_tele\_snd-bc\_data RIGHT BY 20 PLACES.  
\*      cs\_tele\_snd-OPERATION\_NUMBER = ls\_ordim\_o-tanum+4(8).  
  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
\*        Telegram 05  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
  
        IF cs\_tele\_snd-teletype EQ '05'.  
  
          cs\_tele\_snd-mc\_key = is\_ordim\_o-tanum+4(8).  
  
  
          cs\_tele\_snd-transport\_type = '1'.  
          cs\_tele\_snd-dest  =  is\_ordim\_o-nltyp. " destination storage type  
          cs\_tele\_snd-source = is\_ordim\_o-vlpla .  
          cs\_tele\_snd-control\_info = '0'.  
          cs\_tele\_snd-setting\_type = '2'.  
\*       ,   DATA: lv\_tempdata TYPE c LENGTH 9.  
\*          DATA: lv\_tempdata\_1 TYPE c LENGTH 12.  
          CONCATENATE '1' is\_ordim\_o-nlpla INTO lv\_tempdata.  
          CONCATENATE lv\_tempdata '000' INTO lv\_tempdata\_1.  
          cs\_tele\_snd-location\_no = lv\_tempdata\_1.  
          IF is\_ordim\_o-procty = '3087'.  
            cs\_tele\_snd-teletype = '11'.  
            cs\_tele\_snd-command\_type = '01'.  
            DATA lv\_nlenr TYPE /scwm/ltap\_nlenr.  
            lv\_nlenr = cs\_tele\_snd-huident.  
            SELECT tanum, confirmed\_at INTO TABLE @DATA(lt\_tanum)  
              FROM /scwm/ordim\_c WHERE  
              lgnum = @iv\_lgnum AND  
              nlenr = @lv\_nlenr AND  
              tostat = 'A' AND  
              procty = '3010' OR procty = '3093'.  
            IF sy-subrc = 0.  
  
              SORT lt\_tanum BY confirmed\_at DESCENDING.  
              READ TABLE lt\_tanum INTO DATA(ls\_tanum) INDEX 1.  
              IF sy-subrc = 0.  
                DATA(lv\_tanum) = ls\_tanum-tanum.  
                cs\_tele\_snd-mc\_key = lv\_tanum+4(8).  
              ENDIF.  
            ENDIF.  
          ENDIF.  
  
  
          IF is\_ordim\_o-nltyp = 'CPOT'.  
            cs\_tele\_snd-transport\_type = '3'.  
            cs\_tele\_snd-location\_no = '000000000000'.  
            cs\_tele\_snd-dest = is\_ordim\_o-nlpla+0(4).  
            IF is\_ordim\_o-procty = '3081'.  
              cs\_tele\_snd-control\_info = '1'.  
            ELSEIF is\_ordim\_o-procty = '3082'.  
              cs\_tele\_snd-control\_info = '2'.  
            ELSEIF is\_ordim\_o-procty = '3083'.  
              cs\_tele\_snd-control\_info = '5'.  
            ENDIF.  
          ELSEIF is\_ordim\_o-vltyp = 'DEST'.  
            cs\_tele\_snd-transport\_type = '3'.  
            cs\_tele\_snd-location\_no = '000000000000'.  
            cs\_tele\_snd-dest = is\_ordim\_o-nlpla+0(4).  
          ENDIF.  
        ENDIF.  
  
  
\*      IMPORT final TO ls\_ordim\_m FROM MEMORY ID 'BIN\_FULL\_33'.  
\*      FREE MEMORY ID 'BIN\_FULL\_33'.  
\*\*        IF ls\_huhdr-mfserror = 'MBNO'.  
\*      IF ls\_ordim\_m-ordim\_dummy = 'F'.  
  
\*          IF ls\_ordim\_m-vlenr = ls\_ordim\_o-vlenr.  
\*        cs\_tele\_snd-teletype = '11'.  
\*        cs\_tele\_snd-command\_classification = '01'.  
\*        SELECT SINGLE \* FROM zcon\_param\_val  
\*            INTO  
\*            ls\_con\_param\_val  
\*            WHERE  
\*            lgnum = iv\_lgnum  
\*            AND zreq\_name = iv\_plc  
\*            AND zparam = 'MAINTANANCE\_STATION'.  
\*        IF ls\_ordim\_m-nlpla = ls\_con\_param\_val-zvalue.  
\*          cs\_tele\_snd-location\_no = '000000000000'.  
\*          SELECT SINGLE \* FROM zcon\_param\_val  
\*              INTO  
\*              ls\_con\_param\_val  
\*              WHERE  
\*              lgnum = iv\_lgnum  
\*              AND zreq\_name = iv\_plc  
\*              AND zparam = 'MAINTANANCE\_STATION\_NO'.  
\*          cs\_tele\_snd-station\_no = ls\_con\_param\_val-zvalue.  
\*          cs\_tele\_snd-command\_classification = '02'.  
\*        ELSE.  
\*          cs\_tele\_snd-location\_no = ls\_ordim\_m-nlpla+5(12).  
\*          cs\_tele\_snd-station\_no = '0000'.  
\*        ENDIF.  
\*\*          ENDIF.  
\*  
\*      ENDIF.  
\*  
\*      IMPORT final TO ls\_ordim\_m FROM MEMORY ID 'BIN\_EMPTY\_33'.  
\*      FREE MEMORY ID 'BIN\_EMPTY\_33'.  
\*      IF ls\_ordim\_m-ordim\_dummy = 'E'.  
\*        IF ls\_ordim\_m-nlpla = 'EMPTY'.  
\*          cs\_tele\_snd-teletype = '11'.  
\*          cs\_tele\_snd-command\_classification = '12'.  
\*          cs\_tele\_snd-location\_no = '000000000000'.  
\*          cs\_tele\_snd-station\_no = '0000'.  
\*\*            cs\_tele\_snd-mc\_key = '00000000'. " Incase of error  
\*\*            cs\_tele\_snd-bc\_data = '9999999999'. " Incase of error  
\*        ENDIF.  
\*  
\*      ENDIF.  
  
  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
\*        Telegram 12  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
  
        IF cs\_tele\_snd-teletype EQ '12'.  
  
          cs\_tele\_snd-mc\_key = is\_ordim\_o-tanum+4(8).  
          cs\_tele\_snd-transport\_type = '2'.  
          cs\_tele\_snd-category = '2'.  " Only planned retreival  
          cs\_tele\_snd-restoring\_flag = '0'.  
          cs\_tele\_snd-retrieval\_cmd\_detail = '1'.  
          cs\_tele\_snd-dest  =  is\_ordim\_o-nlpla+0(4). " destination storage bin  
          cs\_tele\_snd-source = is\_ordim\_o-vltyp .  
          CONCATENATE '1' is\_ordim\_o-vlpla INTO lv\_tempdata.  
          CONCATENATE lv\_tempdata '000' INTO lv\_tempdata\_1.  
          cs\_tele\_snd-location\_no = lv\_tempdata\_1.  
          cs\_tele\_snd-move\_location\_no = '000000000000'.  
  
        ENDIF.  
  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
\*         Telegram 04  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
        IF cs\_tele\_snd-teletype EQ '04'.  
  
          cs\_tele\_snd-mc\_key = is\_ordim\_o-tanum+4(8).  
          cs\_tele\_snd-move\_location\_no = '000000000000'.  
  
        ENDIF.  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
\*        Telegram 08  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
\*      IF cs\_tele\_snd-teletype EQ '08'.  
\*  
\*        IF sy-subrc EQ 0.  
\*          "cs\_tele\_snd-mc\_key = wa\_tele\_log-mc\_key.  
\*          cs\_tele\_snd-agc\_data = wa\_tele\_log-agc\_data.  
\*        ENDIF.  
\*  
\*        IF cs\_tele\_snd-mc\_key IS INITIAL.  
\*          cs\_tele\_snd-command\_classification = '2'.  
\*        ELSE.  
\*          cs\_tele\_snd-command\_classification = '1'.  
\*        ENDIF.  
\*      ENDIF.  
\*  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
\*\*        Telegram 11  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
\*  
\*      IF cs\_tele\_snd-teletype EQ '11'.  
        "cs\_tele\_snd-mc\_key = wa\_tele\_log-mc\_key.  
\*          cs\_tele\_snd-agc\_data = wa\_tele\_log-agc\_data.  
\*          cs\_tele\_snd-operation\_number = wa\_tele\_log-operation\_number.  
\*          cs\_tele\_snd-control\_info = wa\_tele\_log-control\_info.  
\*      ENDIF.  
\*  
\*  
      WHEN OTHERS.  
  
    ENDCASE.  
  
  ENDMETHOD.

  method /SCWM/IF\_EX\_MFS\_TELE\_SND~GET\_START\_END\_STRING.  
  
     IF iv\_plc = 'AGC'.  
      cv\_start\_unicode = '\02'."STX.  
      cv\_end\_unicode = '\03'."ETX  
  
    ENDIF.  
  
  endmethod.



A screenshot of a computer

Description automatically generated

  METHOD /scwm/if\_ex\_mfs\_hu\_det\_tele~hu\_det.  
  
\*   Build table with HUs for telegram processing  
\*    DATA: ls\_tele\_hu TYPE /scwm/s\_mfs\_tele\_hu.  
\*    DATA:ls\_humove         TYPE  /scwm/mfshumove,  
\*         lv\_case\_aisle\_det TYPE c,  
\*         lv\_no\_async       TYPE  /scwm/de\_no\_async\_fm\_call.  
\*  
\*    lv\_case\_aisle\_det = 'X'.  
\*    EXPORT lv\_case\_aisle\_det FROM lv\_case\_aisle\_det TO MEMORY ID 'MFSACT\_CASE\_AISLE\_DET'.   " EXPORT to ZSCWM\_CL\_CORE\_PUTSTRA->ZDET\_BIN\_FOR\_PRODHU  
\*  
\*    CALL FUNCTION 'ZSCWM\_MFSACT\_CASE\_AISLE\_DET'  
\*      EXPORTING  
\*        iv\_lgnum    = iv\_lgnum  
\*        iv\_plc      = iv\_plc  
\*        iv\_channel  = iv\_channel  
\*        is\_telegram = is\_telegram  
\*        io\_log      = io\_log  
\*      IMPORTING  
\*        ev\_no\_async = lv\_no\_async  
\*        es\_humove   = ls\_humove.  
\*  
\*    FREE MEMORY ID 'MFSACT\_CASE\_AISLE\_DET'.  
\*  
\*  
\*    ls\_tele\_hu-huident   = is\_telegram-huident.  
\*    ls\_tele\_hu-dest      = ls\_humove-nlpla.  
\*  
\*    APPEND ls\_tele\_hu TO ct\_tele\_hu.  
\*    CLEAR:ls\_tele\_hu.  
  
  ENDMETHOD.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

  METHOD if\_ex\_exec\_methodcall\_ppf~execute.  
    DATA:  
      lo\_dlv\_ppf     TYPE REF TO /scdl/cl\_dlv\_ppf,  
      ls\_docid       TYPE /scwm/dlv\_docid\_item\_str,  
      lo\_ppf         TYPE REF TO cl\_manager\_ppf,  
      " processing time  
      l\_trigger\_guid TYPE guid\_32,  
      lo\_trigger     TYPE REF TO cl\_trigger\_ppf,  
      l\_os\_guid      TYPE os\_guid,  
      lv\_dispatch    TYPE ppfddsptch,  
  
      lo\_af          TYPE REF TO /scdl/cl\_af\_management,  
      lo\_af\_tm       TYPE REF TO /scwm/cl\_adapter\_tm,  
      lo\_tm          TYPE REF TO /scwm/if\_tm,  
      lo\_tm\_trace    TYPE REF TO /scwm/if\_tm\_trace,  
      lv\_trace\_par   TYPE symsgv,  
  
      lv\_msg\_txt     TYPE string ##NEEDED.  
  
    rp\_status = '1'.  
  
    " No processing for 'Preview'  
    CHECK ip\_preview IS INITIAL.  
  
\* Access persistent application data  
    TRY.  
        lo\_dlv\_ppf ?= io\_appl\_object.  
      CATCH  cx\_sy\_move\_cast\_error cx\_os\_object\_not\_found.  
        ASSERT ID /scwm/ppf CONDITION 1 = 0.  
        MESSAGE e015(/scwm/delivery) INTO lv\_msg\_txt.  
        cl\_log\_ppf=>add\_message(  
            ip\_problemclass = sppf\_pclass\_1  
            ip\_handle       = ip\_application\_log ).  
        rp\_status = sppf\_status\_error.  
        RETURN.  
    ENDTRY.  
  
    BREAK-POINT ID /scwm/ppf.  
  
  
\* Determine processing time (immediately, when saving, report)  
    GET PARAMETER ID 'TRIGGERGUID' FIELD l\_trigger\_guid.  
  
    IF l\_trigger\_guid IS NOT INITIAL.  
      l\_os\_guid = l\_trigger\_guid.  
      TRY.  
          lo\_trigger ?=  
           ca\_trigger\_ppf=>agent->if\_os\_ca\_persistency~get\_persistent\_by\_oid(  
                                  l\_os\_guid ).  
        CATCH cx\_sy\_move\_cast\_error.  
\*       do nothing  
      ENDTRY.  
  
      IF NOT lo\_trigger IS INITIAL.  
        lv\_dispatch = lo\_trigger->get\_dispatch( ).  
      ENDIF.  
    ENDIF.  
  
  
\* Control save for dispatch time 'report'  
    IF lv\_dispatch = sppf\_process\_later.   " report  
      " set that PPF save BADI should be used to save instead of normal save.  
      /scdl/cl\_im\_trig\_exec=>sv\_call\_save\_after\_execution = abap\_true.  
      " set prohibited flag, as application needs to decide later if a  
      " real save should be done.  
      /scdl/cl\_im\_trig\_exec=>sv\_save\_prohibited           = abap\_true.  
    ENDIF.  
  
\* get doccat, docid  
    TRY.  
        ls\_docid-docid  = lo\_dlv\_ppf->get\_docid( ).  
        ls\_docid-doccat = lo\_dlv\_ppf->get\_doccat( ).  
      CATCH cx\_os\_object\_not\_found  
            cx\_sy\_move\_cast\_error.                      "#EC NO\_HANDLER  
        " Handled by the following IF statement  
        ASSERT ID /scwm/ppf CONDITION 1 = 0.  
    ENDTRY.  
  
    IF ls\_docid-docid  IS INITIAL OR  
       ls\_docid-doccat IS INITIAL.  
      MESSAGE e015(/scwm/delivery) INTO lv\_msg\_txt.  
      cl\_log\_ppf=>add\_message(  
          ip\_problemclass = sppf\_pclass\_1  
          ip\_handle       = ip\_application\_log ).  
      rp\_status = sppf\_status\_error.  
      RETURN.  
    ENDIF.  
  
  
\* set warehouse for LUW and connect to transaction trace  
    lo\_ppf  = cl\_manager\_ppf=>get\_instance( ).  
    lo\_tm ?= /scwm/cl\_tm\_factory=>get\_service(  
        /scwm/cl\_tm\_factory=>sc\_manager ).  
    lo\_tm\_trace ?= /scwm/cl\_tm\_factory=>get\_service(  
        /scwm/cl\_tm\_factory=>sc\_trace ).  
  
    IF lo\_ppf->execution\_mode = sppf\_execute\_report OR  
       lo\_ppf->execution\_mode = sppf\_execute\_arfc.  
  
      lo\_af = /scdl/cl\_af\_management=>get\_instance( ).  
      lo\_af\_tm ?= lo\_af->get\_service(  
         /scdl/if\_af\_management\_c=>sc\_trans\_manager ).  
      lo\_af\_tm->set\_lgnum(  
        EXPORTING  
          iv\_doccat = ls\_docid-doccat  
          iv\_docid  = ls\_docid-docid ).  
      lo\_tm->start\_transaction(  
          iv\_ppfaction = ip\_action ).  
  
    ELSE.  
      lv\_trace\_par = ip\_action.  
      lo\_tm\_trace->log\_event(  
          iv\_event     = /scwm/if\_tm\_c=>sc\_event\_ppf\_exec  
          iv\_param2    = lv\_trace\_par ).  
    ENDIF.  
  
    /scdl/cl\_ppf\_msg\_log\_map=>set\_log\_validity(  
        iv\_log  = ip\_application\_log  
        iv\_whno = /scwm/cl\_tm=>sv\_lgnum ).  
  
    CHECK flt\_val = 'ZSCWM\_CL\_ODO\_WT\_CR'.  
  
    DATA: lv\_lgnum           TYPE /scwm/lgnum.  
  
    DATA: ls\_create\_whr TYPE /scwm/s\_to\_prepare\_whr\_int,  
          lt\_create\_whr TYPE /scwm/tt\_to\_prep\_whr\_int.  
  
    DATA: lv\_qname   TYPE trfcqnam,  
          lv\_guid    TYPE guid,  
          lt\_ltap\_vb TYPE /scwm/tt\_ltap\_vb.  
  
\* Query  
    DATA: lt\_docid        TYPE /scwm/dlv\_docid\_item\_tab,  
          ls\_include\_data TYPE /scwm/dlv\_query\_incl\_str\_prd,  
          lt\_whr\_items    TYPE /scwm/dlv\_item\_out\_prd\_tab.  
  
    DATA: lv\_mve\_hu      TYPE /scwm/de\_whr\_mve\_hu,  
          lv\_mve\_mat     TYPE /scwm/de\_whr\_mve\_mat,  
          lv\_mve\_hu\_mult TYPE /scwm/de\_whr\_mve\_hu\_mult.  
  
    DATA: lo\_badi TYPE REF TO /scwm/ex\_dlv\_towhr\_crea\_ppf,  
          lo\_dlv  TYPE REF TO /scwm/cl\_dlv\_management\_prd.  
  
    DATA: lv\_docid           TYPE char100.  
  
    FIELD-SYMBOLS:  
      <item>        TYPE /scwm/dlv\_item\_out\_prd\_str.  
  
    APPEND ls\_docid TO lt\_docid.  
  
\* Check: Do not execute this PPF-action during 'SAVE' of  
\*        TO creation or TO confirmation.  
    CALL FUNCTION '/SCWM/L\_TO\_GET\_INFO'  
      IMPORTING  
        et\_ltap\_vb = lt\_ltap\_vb.  
    IF lt\_ltap\_vb IS NOT INITIAL.  
      rp\_status = sppf\_status\_processed.  
      RETURN.  
    ENDIF.  
  
    " Select warehouse request  
    TRY.  
        CALL FUNCTION '/SCWM/WHR\_QUERY'  
          EXPORTING  
            it\_docid        = lt\_docid  
            iv\_doccat\_whr   = ls\_docid-doccat  
            is\_include\_data = ls\_include\_data  
          IMPORTING  
            et\_whr\_items    = lt\_whr\_items.  
  
      CATCH /scdl/cx\_delivery.  
        " just for faulty calls  
        ASSERT 1 = 0.  
    ENDTRY.  
  
    READ TABLE lt\_whr\_items ASSIGNING <item> INDEX 1.  
    IF sy-subrc = 0.  
      lv\_lgnum = <item>-sapext-/scwm/whno.  
  
      IF lv\_lgnum IS INITIAL.  
        MESSAGE e015(/scwm/delivery) INTO lv\_msg\_txt.  
        cl\_log\_ppf=>add\_message(  
            ip\_problemclass = sppf\_pclass\_1  
            ip\_handle       = ip\_application\_log ).  
        rp\_status = sppf\_status\_error.  
        RETURN.  
      ENDIF.  
  
    ELSE.  
\*      rp\_status = set\_deleted( ip\_application\_log ).  
      RETURN.  
    ENDIF.  
  
  
    " Call BADI to reset Importing Parameter  
    lv\_mve\_hu      = 'X'.  
    lv\_mve\_mat     = 'X'.  
    lv\_mve\_hu\_mult = 'X'.  
  
    TRY.  
        " Get Badi instance and set filter  
        GET BADI lo\_badi  
          FILTERS  
            lgnum = lv\_lgnum.  
        CALL BADI lo\_badi->create\_wt\_ppf  
          EXPORTING  
            iv\_lgnum       = lv\_lgnum  
            is\_docid       = ls\_docid  
            iv\_mve\_hu      = lv\_mve\_hu  
            iv\_mve\_mat     = lv\_mve\_mat  
            iv\_mve\_hu\_mult = lv\_mve\_hu\_mult  
          IMPORTING  
            ev\_mve\_hu      = lv\_mve\_hu  
            ev\_mve\_mat     = lv\_mve\_mat  
            ev\_mve\_hu\_mult = lv\_mve\_hu\_mult.  
      CATCH cx\_badi.  
    ENDTRY.  
  
  
    ls\_create\_whr-rdocid  = ls\_docid-docid.  
    ls\_create\_whr-rdoccat = ls\_docid-doccat.  
    APPEND ls\_create\_whr TO lt\_create\_whr.  
  
  
\*    IF ip\_dispatch = sppf\_process\_at\_once.   " immediately  
\*      " do not execute 'save' in /SCWM/TO\_UPDATE\_WHR\_INT  
\*      /scdl/cl\_im\_trig\_exec=>sv\_save\_prohibited = abap\_true.  
\*    ENDIF.  
  
    " Call TO creation  
    lo\_dlv = /scwm/cl\_dlv\_management\_prd=>get\_instance( ).  
    lo\_dlv->get\_queue\_name(  
      EXPORTING  
        iv\_docid  = ls\_docid-docid  
        iv\_doccat = ls\_docid-doccat  
        iv\_prefix = 'WMDP'  
      RECEIVING  
        ev\_qname  = lv\_qname  
      EXCEPTIONS  
        OTHERS    = 0 ).  
  
    IF lv\_qname IS INITIAL.  
      lv\_docid = ls\_docid-docid.  
      CALL FUNCTION 'CONVERSION\_EXIT\_ALPHA\_OUTPUT'  
        EXPORTING  
          input  = lv\_docid  
        IMPORTING  
          output = lv\_docid.  
      CONCATENATE 'WMDP' ls\_docid-doccat lv\_docid INTO lv\_qname.  
    ENDIF.  
  
    TRY.  
        lv\_guid = cl\_system\_uuid=>create\_uuid\_x16\_static( ).  
      CATCH cx\_uuid\_error INTO DATA(lx\_uuid).  
        " nothing will work w/o guid creation  
        ASSERT FIELDS lx\_uuid->get\_text( ) CONDITION 1 = 0.  
    ENDTRY.  
  
    " Trigger WT Creation in Background Task from the below update task.  
    " This method ensures data consistency.  
    CALL FUNCTION '/SCWM/PPF\_TO\_CREATE\_UPD'  
      IN UPDATE TASK  
      EXPORTING  
        is\_docid       = ls\_docid  
        ip\_action      = ip\_action  
        it\_create\_whr  = lt\_create\_whr  
        iv\_mve\_hu      = lv\_mve\_hu  
        iv\_mve\_mat     = lv\_mve\_mat  
        iv\_mve\_hu\_mult = lv\_mve\_hu\_mult  
        iv\_qname       = lv\_qname  
        iv\_guid        = lv\_guid  
        iv\_lgnum       = lv\_lgnum.  
  
    " WT Creation was triggered  
    MESSAGE s017(/scwm/delivery) WITH lv\_qname INTO lv\_msg\_txt.  
  
    cl\_log\_ppf=>add\_message(  
        ip\_problemclass = sppf\_pclass\_5  
        ip\_handle       = ip\_application\_log ).  
  
    rp\_status = sppf\_status\_processed.  
  
  
  
  ENDMETHOD.

A screenshot of a computer

Description automatically generated

  METHOD if\_ex\_eval\_schedcond\_ppf~evaluate\_schedule\_condition.  
  
\*       Application data:  
    DATA: lo\_ao\_dlv           TYPE REF TO /scdl/cl\_dlv\_ppf,  
          ls\_bo               TYPE        /scdl/af\_ppf\_str,  
\*       Headers and Items  
          ls\_hdr              TYPE        ltys\_hdr,  
          ls\_itm              TYPE        ltys\_itm,  
          lv\_msg\_dummy        TYPE        string,  
\*       Exceptions:  
          lo\_ex\_cast\_error    TYPE REF TO cx\_sy\_move\_cast\_error,  
          lo\_ex\_obj\_not\_found TYPE REF TO cx\_os\_object\_not\_found,  
          lo\_ex\_bo            TYPE REF TO /scdl/cx\_bo.  
  
\* Query  
    DATA: lt\_docid        TYPE /scwm/dlv\_docid\_item\_tab,  
          ls\_include\_data TYPE /scwm/dlv\_query\_incl\_str\_prd,  
          lt\_whr\_items    TYPE /scwm/dlv\_item\_out\_prd\_tab.  
  
    DATA: lo\_header        TYPE REF TO /scdl/cl\_dl\_header,  
          lo\_item\_prd\_curr TYPE REF TO /scdl/if\_dl\_item\_prd\_readonly,  
          lo\_item\_prd\_old  TYPE REF TO /scdl/if\_dl\_item\_prd\_readonly,  
          lt\_item\_curr     TYPE /scdl/dl\_item\_tab,  
          ls\_item\_curr     TYPE /scdl/dl\_item\_str,  
          lt\_item\_old      TYPE /scdl/dl\_item\_tab,  
          ls\_item\_old      TYPE /scdl/dl\_item\_str,  
          lt\_partyloc      TYPE        /scdl/dl\_partyloc\_tab,  
          ls\_partyloc      TYPE        /scdl/dl\_partyloc\_str,  
          ls\_lgnumdata     TYPE        /scwm/s\_lgnumlocid,  
          ls\_itemext\_curr  TYPE /scdl/dl\_sap\_prd\_item\_o\_str,  
          ls\_itemext\_old   TYPE /scdl/dl\_sap\_prd\_item\_o\_str.  
  
    DATA: lt\_status       TYPE /scdl/dl\_status\_tab,  
          lv\_status\_found TYPE boole\_d,  
          lv\_itemtype     TYPE /scdl/dl\_itemtype,  
          ls\_status       TYPE /scdl/dl\_status\_str.  
  
    DATA: lt\_tvarvc TYPE TABLE OF tvarvc.  
    DATA: lt\_con\_param TYPE TABLE OF zcon\_param\_val,  
          ls\_con\_param TYPE zcon\_param\_val.  
  
\*    BREAK p00127339.  
    BREAK-POINT ID /scwm/ppf.  
  
    ep\_rc = 0.  
\* 0. Create info message in log  
    MESSAGE i024(/scwm/delivery) WITH ip\_ttype INTO lv\_msg\_dummy.  
    CALL METHOD cl\_log\_ppf=>add\_message  
      EXPORTING  
        ip\_problemclass = sppf\_pclass\_4  
        ip\_handle       = ip\_protocol.  
  
\* 1. first checks: need context and appl.object  
    IF NOT io\_context       IS BOUND           OR  
       NOT io\_context->appl IS BOUND.  
      ep\_rc = 99.  
      MESSAGE e020(/scwm/delivery) WITH flt\_val ip\_ttype io\_context->applctn flt\_val INTO lv\_msg\_dummy.  
  
      CALL METHOD cl\_log\_ppf=>add\_message  
        EXPORTING  
          ip\_problemclass = sppf\_pclass\_1  
          ip\_handle       = ip\_protocol.  
      EXIT.  
    ENDIF.  
    IF NOT io\_context->applctn = /scwm/if\_dl\_c=>sc\_ppf\_applctn\_dlv.  
\*   not a delivery application relevant object  
      ep\_rc = 99.  
      MESSAGE e020(/scwm/delivery) WITH flt\_val ip\_ttype io\_context->applctn flt\_val INTO lv\_msg\_dummy.  
  
      CALL METHOD cl\_log\_ppf=>add\_message  
        EXPORTING  
          ip\_problemclass = sppf\_pclass\_1  
          ip\_handle       = ip\_protocol.  
      EXIT.  
    ENDIF.  
  
    IF NOT io\_context->name = '/SCWM/PDO\_01'.  
\*   not a delivery application relevant object  
      ep\_rc = 99.  
      MESSAGE e020(/scwm/delivery) WITH flt\_val ip\_ttype io\_context->applctn flt\_val INTO lv\_msg\_dummy.  
  
      CALL METHOD cl\_log\_ppf=>add\_message  
        EXPORTING  
          ip\_problemclass = sppf\_pclass\_1  
          ip\_handle       = ip\_protocol.  
      EXIT.  
    ENDIF.  
  
\* 2. access application data  
\* Abstract objects  
  
\* Access appl. data  
    TRY.  
        lo\_ao\_dlv ?= io\_context->appl.  
        ls\_bo      = lo\_ao\_dlv->get\_mo\_bo( ).  
  
      CATCH  cx\_sy\_move\_cast\_error INTO lo\_ex\_cast\_error.  
        lv\_msg\_dummy =  lo\_ex\_cast\_error->get\_text( ).  
        ep\_rc = 98.  
        CALL METHOD cl\_log\_ppf=>add\_message  
          EXPORTING  
            ip\_problemclass = sppf\_pclass\_1  
            ip\_handle       = ip\_protocol.  
        EXIT.  
      CATCH  cx\_os\_object\_not\_found INTO lo\_ex\_obj\_not\_found.  
        lv\_msg\_dummy = lo\_ex\_obj\_not\_found->get\_text( ).  
        ep\_rc = 98.  
        CALL METHOD cl\_log\_ppf=>add\_message  
          EXPORTING  
            ip\_problemclass = sppf\_pclass\_1  
            ip\_handle       = ip\_protocol.  
        EXIT.  
    ENDTRY.  
  
    ls\_hdr-id     = ls\_bo-docid.  
    ls\_hdr-doccat = ls\_bo-doccat.  
  
    TRY.  
\* current delivery details from  header & item  
        ls\_hdr-ai     = ls\_bo-bo->get\_header( iv\_objectstate = /scdl/if\_dl\_c=>sc\_object\_state\_curr ).  
  
        ls\_bo-bo->get\_item\_tab( EXPORTING iv\_objectstate = /scdl/if\_dl\_c=>sc\_object\_state\_curr  
                                IMPORTING et\_item        = ls\_itm-ai\_tab ).  
\* old delivery details from  header & item  
        ls\_hdr-bi     = ls\_bo-bo->get\_header( iv\_objectstate = /scdl/if\_dl\_c=>sc\_object\_state\_db ).  
  
        ls\_bo-bo->get\_item\_tab( EXPORTING iv\_objectstate = /scdl/if\_dl\_c=>sc\_object\_state\_db  
                                IMPORTING et\_item        = ls\_itm-bi\_tab ).  
      CATCH /scdl/cx\_bo INTO lo\_ex\_bo.  
        ep\_rc = 98.  
        lv\_msg\_dummy = lo\_ex\_bo->get\_text( ).  
        CALL METHOD cl\_log\_ppf=>add\_message  
          EXPORTING  
            ip\_problemclass = sppf\_pclass\_1  
            ip\_handle       = ip\_protocol.  
        EXIT.  
    ENDTRY.  
  
    CLEAR: lt\_partyloc,lt\_status,ls\_partyloc,ls\_lgnumdata.  
\* get delivery header  
    lo\_header    = ls\_bo-bo->get\_header( ).  
    lt\_partyloc  = lo\_header->get\_partyloc( ).  
    lt\_status    = lo\_header->get\_status( ).  
  
    READ TABLE lt\_partyloc INTO ls\_partyloc  
                              WITH KEY party\_role = /scdl/if\_dl\_partyloc\_c=>sc\_party\_role\_wh.  
    IF sy-subrc = 0.  
      CALL FUNCTION '/SCWM/LGNUM\_LOCID\_READ'  
        EXPORTING  
          iv\_locid16     = ls\_partyloc-locationid  
        IMPORTING  
          es\_data        = ls\_lgnumdata  
        EXCEPTIONS  
          data\_not\_found = 1  
          OTHERS         = 2.  
      IF sy-subrc = 0.  
        IF ls\_lgnumdata-lgnum IS INITIAL. "If warehouse not found  
          ep\_rc = 99.  
          EXIT.  
        ENDIF.  
\* Check if auto WT PPF activated at warehouse level  
        CLEAR: ls\_con\_param.  
        SELECT \* FROM zcon\_param\_val INTO ls\_con\_param  
                 WHERE lgnum     = ls\_lgnumdata-lgnum  
                 AND   zreq\_name = lc\_con\_param\_req\_name  
                 AND   zparam    = lc\_con\_para\_at\_door  
                 AND   zvalue    = abap\_true  
                 AND   zactive   = abap\_true.  
        ENDSELECT.  
        IF sy-subrc <> 0. "If warehouse not maintained for auto WT creation  
          ep\_rc = 99.  
          EXIT.  
        ENDIF.  
      ELSE. "If warehouse not found  
        ep\_rc = 99.  
        EXIT.  
      ENDIF.  
    ENDIF.  
  
\* delivery block status  
    READ TABLE lt\_status ASSIGNING FIELD-SYMBOL(<lfs\_status>)  
                                            WITH KEY status\_type = /scdl/if\_dl\_status\_c=>sc\_t\_blocked\_overall.  
    IF sy-subrc = 0 AND <lfs\_status>-status\_value = /scdl/if\_dl\_status\_c=>sc\_v\_true.  
      ep\_rc = 99.  
      EXIT.  
    ENDIF.  
\* delivery inconsistency status  
    READ TABLE lt\_status ASSIGNING <lfs\_status>  
                              WITH KEY status\_type = /scdl/if\_dl\_status\_c=>sc\_t\_blocked\_dt\_incon\_item.  
    IF sy-subrc = 0 AND <lfs\_status>-status\_value = /scdl/if\_dl\_status\_c=>sc\_v\_true.  
      ep\_rc = 99.  
      EXIT.  
    ENDIF.  
\* deliery item planned picking status  
    READ TABLE lt\_status ASSIGNING <lfs\_status>  
                                WITH KEY status\_type = /scdl/if\_dl\_status\_c=>sc\_t\_planning\_picking.  
    IF sy-subrc = 0 AND (  
    <lfs\_status>-status\_value = /scdl/if\_dl\_status\_c=>sc\_v\_not\_relevant OR  
    <lfs\_status>-status\_value = /scdl/if\_dl\_status\_c=>sc\_v\_finished ).  
      ep\_rc = 99.  
      EXIT.  
    ENDIF.  
  
\* deliery item picking status  
    READ TABLE lt\_status ASSIGNING <lfs\_status>  
                              WITH KEY status\_type = /scdl/if\_dl\_status\_c=>sc\_t\_picking.  
    IF sy-subrc = 0 AND (  
    <lfs\_status>-status\_value = /scdl/if\_dl\_status\_c=>sc\_v\_not\_relevant OR  
    <lfs\_status>-status\_value = /scdl/if\_dl\_status\_c=>sc\_v\_finished ).  
      ep\_rc = 99.  
      EXIT.  
    ENDIF.  
  
  
\* get delivery item.  
    lt\_item\_curr = ls\_itm-ai\_tab.  
    lt\_item\_old  = ls\_itm-bi\_tab.  
  
    SORT lt\_item\_curr BY itemid.  
    SORT lt\_item\_old BY itemid.  
  
    LOOP AT lt\_item\_curr INTO ls\_item\_curr.  
      lo\_item\_prd\_curr  ?= ls\_item\_curr-item.  
  
\* only relevant, if item is standard item and it is for picking and not yet completely picked.  
      IF /scwm/cl\_dlv\_field\_det=>check\_item\_relevance( io\_item = lo\_item\_prd\_curr  
      iv\_is\_itemcat\_standard\_check    = abap\_true  
      iv\_wm\_move\_relevant\_check       = abap\_true  
      iv\_wm\_move\_not\_completed\_check  = abap\_true ) = abap\_false.  
        CONTINUE.  
      ENDIF.  
  
      lv\_itemtype = lo\_item\_prd\_curr->get\_itemtype( ).  
\*Item type validation  
      CLEAR: ls\_con\_param.  
      SELECT \* FROM zcon\_param\_val INTO ls\_con\_param  
               WHERE lgnum     = ls\_lgnumdata-lgnum  
               AND   zreq\_name = lc\_con\_param\_req\_name  
               AND   zparam    = lc\_con\_para\_itemtype  
               AND   zvalue    = lv\_itemtype  
               AND   zactive   = abap\_true.  
      ENDSELECT.  
      IF sy-subrc <> 0.  
        ep\_rc = 99.  
        EXIT.  
      ENDIF.  
  
      lo\_item\_prd\_curr->get\_sapext( IMPORTING es\_sapext\_o = ls\_itemext\_curr ).  
  
\*if door is blank then PPF need not to be trigger  
      IF ls\_itemext\_curr-/scwm/door IS INITIAL.  
        ep\_rc = 99.  
        EXIT.  
      ENDIF.  
\* check if dooor is dummy door  
      CLEAR: ls\_con\_param.  
      SELECT \* FROM zcon\_param\_val INTO ls\_con\_param  
               WHERE lgnum     = ls\_lgnumdata-lgnum  
               AND   zreq\_name = lc\_con\_param\_req\_name  
               AND   zparam    = lc\_con\_para\_dummy\_door  
               AND   zvalue    = ls\_itemext\_curr-/scwm/door  
               AND   zactive   = abap\_true.  
      ENDSELECT.  
      IF sy-subrc = 0.  
        ep\_rc = 99.  
        EXIT.  
      ENDIF.  
  
      READ TABLE lt\_item\_old INTO ls\_item\_old  
                                      WITH KEY itemid = ls\_item\_curr-itemid.  
      IF sy-subrc = 0.  
        lo\_item\_prd\_old  ?= ls\_item\_old-item.  
        lo\_item\_prd\_old->get\_sapext( IMPORTING es\_sapext\_o = ls\_itemext\_old ).  
  
\* now compare old door with new door  
        IF ls\_itemext\_old-/scwm/door = ls\_itemext\_curr-/scwm/door.  
          ep\_rc = 99.  
          EXIT.  
        ENDIF.  
      ENDIF.  
    ENDLOOP.  
  ENDMETHOD.

A screenshot of a computer

Description automatically generated

METHOD if\_ex\_eval\_schedcond\_ppf~evaluate\_schedule\_condition.  
  
\*       Application data:  
  DATA: lo\_document         TYPE        ltys\_document,  
        lo\_ao\_dlv           TYPE REF TO /scdl/cl\_dlv\_ppf,  
        ls\_bo               TYPE        /scdl/af\_ppf\_str,  
\*       Headers and Items  
        ls\_hdr              TYPE        ltys\_hdr,  
        ls\_itm              TYPE        ltys\_itm,  
        lv\_msg\_dummy        TYPE        string,  
\*       Exceptions:  
        lo\_ex\_cast\_error    TYPE REF TO cx\_sy\_move\_cast\_error,  
        lo\_ex\_obj\_not\_found TYPE REF TO cx\_os\_object\_not\_found,  
        lo\_ex\_bo            TYPE REF TO /scdl/cx\_bo.  
  
  
  
  
  
  BREAK-POINT ID /scwm/ppf.  
  
\* 0. Create info message in log  
  MESSAGE i024(/scwm/delivery) WITH ip\_ttype INTO lv\_msg\_dummy.  
  CALL METHOD cl\_log\_ppf=>add\_message  
    EXPORTING  
      ip\_problemclass = sppf\_pclass\_4  
      ip\_handle       = ip\_protocol.  
  
  
\* 1. first checks: need context and appl.object  
  IF NOT io\_context       IS BOUND           OR  
     NOT io\_context->appl IS BOUND.  
    ep\_rc = 99.  
    MESSAGE e020(/scwm/delivery) WITH flt\_val ip\_ttype io\_context->applctn flt\_val INTO lv\_msg\_dummy.  
  
    CALL METHOD cl\_log\_ppf=>add\_message  
      EXPORTING  
        ip\_problemclass = sppf\_pclass\_1  
        ip\_handle       = ip\_protocol.  
    EXIT.  
  ENDIF.  
  IF NOT io\_context->applctn = /scwm/if\_dl\_c=>sc\_ppf\_applctn\_dlv.  
\*   not a delivery application relevant object  
    ep\_rc = 99.  
    MESSAGE e020(/scwm/delivery) WITH flt\_val ip\_ttype io\_context->applctn flt\_val INTO lv\_msg\_dummy.  
  
    CALL METHOD cl\_log\_ppf=>add\_message  
      EXPORTING  
        ip\_problemclass = sppf\_pclass\_1  
        ip\_handle       = ip\_protocol.  
    EXIT.  
  ENDIF.  
  
  
  
\* 2. access application data  
\* Abstract objects  
  
\* Access appl. data  
  TRY.  
      lo\_ao\_dlv ?= io\_context->appl.  
      ls\_bo      = lo\_ao\_dlv->get\_mo\_bo( ).  
  
    CATCH  cx\_sy\_move\_cast\_error INTO lo\_ex\_cast\_error.  
      lv\_msg\_dummy =  lo\_ex\_cast\_error->get\_text( ).  
      ep\_rc = 98.  
      CALL METHOD cl\_log\_ppf=>add\_message  
        EXPORTING  
          ip\_problemclass = sppf\_pclass\_1  
          ip\_handle       = ip\_protocol.  
      EXIT.  
    CATCH  cx\_os\_object\_not\_found INTO lo\_ex\_obj\_not\_found.  
      lv\_msg\_dummy = lo\_ex\_obj\_not\_found->get\_text( ).  
      ep\_rc = 98.  
      CALL METHOD cl\_log\_ppf=>add\_message  
        EXPORTING  
          ip\_problemclass = sppf\_pclass\_1  
          ip\_handle       = ip\_protocol.  
      EXIT.  
  ENDTRY.  
  
  ls\_hdr-id     = ls\_bo-docid.  
  ls\_hdr-doccat = ls\_bo-doccat.  
  
  TRY.  
      ls\_hdr-ai     = ls\_bo-bo->get\_header( iv\_objectstate = /scdl/if\_dl\_c=>sc\_object\_state\_curr ).  
      ls\_bo-bo->get\_item\_tab(  
        EXPORTING iv\_objectstate = /scdl/if\_dl\_c=>sc\_object\_state\_curr  
        IMPORTING et\_item        = ls\_itm-ai\_tab ).  
\*      IF ls\_bo-bo->mv\_new\_bo EQ abap\_false.  
      ls\_hdr-bi     = ls\_bo-bo->get\_header( iv\_objectstate = /scdl/if\_dl\_c=>sc\_object\_state\_db ).  
      ls\_bo-bo->get\_item\_tab(  
        EXPORTING iv\_objectstate = /scdl/if\_dl\_c=>sc\_object\_state\_db  
        IMPORTING et\_item        = ls\_itm-bi\_tab ).  
\*      ENDIF.  
    CATCH /scdl/cx\_bo INTO lo\_ex\_bo.  
      ep\_rc = 98.  
      lv\_msg\_dummy = lo\_ex\_bo->get\_text( ).  
      CALL METHOD cl\_log\_ppf=>add\_message  
        EXPORTING  
          ip\_problemclass = sppf\_pclass\_1  
          ip\_handle       = ip\_protocol.  
      EXIT.  
  ENDTRY.  
  
  lo\_document-hdr = ls\_hdr.  
  lo\_document-itm = ls\_itm.  
  
  CASE flt\_val.  
  
    WHEN '/SCWM/WHR\_FULL\_PACKED'.  
      ep\_rc = me->whr\_full\_packed(  
                  io\_document  = lo\_document  
                  ip\_protocol  = ip\_protocol  
                  ip\_ttype     = ip\_ttype  
                  io\_ao\_dlv    = lo\_ao\_dlv ).  
  
    WHEN '/SCWM/WM\_ACTIVITY\_FINISHED'.  
      ep\_rc = me->wm\_activity\_finished(  
                  io\_document  = lo\_document  
                  ip\_protocol  = ip\_protocol  
                  ip\_ttype     = ip\_ttype  
                  io\_ao\_dlv    = lo\_ao\_dlv ).  
  
    WHEN '/SCWM/WT\_CREATION'.  
      ep\_rc = me->wt\_creation(  
                  io\_document  = lo\_document  
                  ip\_protocol  = ip\_protocol  
                  ip\_ttype     = ip\_ttype  
                  io\_ao\_dlv    = lo\_ao\_dlv ).  
  
    WHEN '/SCWM/WHR\_WAVE\_ASSIGN\_NEW'.  
      ep\_rc = me->whr\_wave\_assign(  
                  iv\_check\_on\_new = abap\_true  
                  io\_document = lo\_document  
                  io\_ao\_dlv    = lo\_ao\_dlv  
                  ip\_ttype     = ip\_ttype  
                  ip\_protocol = ip\_protocol ).  
  
    WHEN '/SCWM/WHR\_WAVE\_ASSIGN\_CHANGE'.  
      ep\_rc = me->whr\_wave\_assign(  
                  iv\_check\_on\_change = abap\_true  
                  io\_document = lo\_document  
                  io\_ao\_dlv    = lo\_ao\_dlv  
                  ip\_ttype     = ip\_ttype  
                  ip\_protocol = ip\_protocol ).  
  
    WHEN '/SCWM/GM\_POSTED'.  
      ep\_rc = me->gm\_posting\_finished(  
                  io\_document  = lo\_document  
                  ip\_protocol  = ip\_protocol  
                  ip\_ttype     = ip\_ttype  
                  io\_ao\_dlv    = lo\_ao\_dlv ).  
  
    WHEN '/SCWM/GI\_POSTING'.  
      ep\_rc = me->gi\_posting(  
                  io\_document  = lo\_document  
                  ip\_protocol  = ip\_protocol  
                  ip\_ttype     = ip\_ttype  
                  io\_ao\_dlv    = lo\_ao\_dlv ).  
  
    WHEN '/SCWM/GR\_POSTING' OR 'ZSCWM\_GR\_POSTING'.         " added by satish  
      ep\_rc = me->gr\_posting(  
                  io\_document  = lo\_document  
                  ip\_protocol  = ip\_protocol  
                  ip\_ttype     = ip\_ttype  
                  io\_ao\_dlv    = lo\_ao\_dlv ).  
  
    WHEN '/SCWM/WHR\_VEH\_CREATE\_NEW'.  
      ep\_rc = me->whr\_veh\_create\_new(  
                io\_document  = lo\_document  
                ip\_protocol  = ip\_protocol  
                ip\_ttype     = ip\_ttype  
                io\_ao\_dlv    = lo\_ao\_dlv ).  
  
    WHEN '/SCWM/WHR\_VEH\_CREATE\_CHANGE'.  
      ep\_rc = me->whr\_veh\_create\_change(  
                io\_document  = lo\_document  
                ip\_protocol  = ip\_protocol  
                ip\_ttype     = ip\_ttype  
                io\_ao\_dlv    = lo\_ao\_dlv ).  
  
    WHEN '/SCWM/NCTS'.  
      ep\_rc = /scwm/cl\_dlv\_ncts=>check\_ncts\_relevance(  
                is\_bo = ls\_bo  
                ip\_protocol = ip\_protocol ).  
      IF ep\_rc IS INITIAL.  
\*      -> call condition technique  
        /scwm/cl\_dlv\_ppf\_conf=>evaluate(  
          EXPORTING  
            iv\_ttype    = ip\_ttype  
            io\_dlv\_ppf  = lo\_ao\_dlv  
            iv\_protocol = ip\_protocol  
          RECEIVING  
            rv\_rc       = ep\_rc  
          EXCEPTIONS  
            wrong\_call  = 0  
            OTHERS      = 0 ).  
  
        IF ep\_rc = 0 OR  
           ep\_rc = 1.  
          ep\_rc = 0.  
        ENDIF.  
      ENDIF.  
    WHEN '/SCWM/OCTS'.  
      ep\_rc = /scwm/cl\_dlv\_ncts=>check\_ncts\_relevance(  
                is\_bo = ls\_bo  
                ip\_protocol = ip\_protocol  
                iv\_octs = abap\_true ).  
      IF ep\_rc IS INITIAL.  
\*      -> call condition technique  
        /scwm/cl\_dlv\_ppf\_conf=>evaluate(  
          EXPORTING  
            iv\_ttype    = ip\_ttype  
            io\_dlv\_ppf  = lo\_ao\_dlv  
            iv\_protocol = ip\_protocol  
          RECEIVING  
            rv\_rc       = ep\_rc  
          EXCEPTIONS  
            wrong\_call  = 0  
            OTHERS      = 0 ).  
  
        IF ep\_rc = 0 OR  
           ep\_rc = 1.  
          ep\_rc = 0.  
        ENDIF.  
      ENDIF.  
    WHEN '/SCWM/WHR\_IN\_YARD'.  
      ep\_rc = me->whr\_in\_yard(  
                io\_document  = lo\_document  
                ip\_protocol  = ip\_protocol  
                ip\_ttype     = ip\_ttype  
                io\_ao\_dlv    = lo\_ao\_dlv ).  
  
    WHEN '/SCWM/WHR\_VAS\_KTS'.  
      ep\_rc = me->whr\_vas\_kts(  
                io\_document = lo\_document  
                ip\_protocol = ip\_protocol  
                ip\_ttype     = ip\_ttype  
                io\_ao\_dlv    = lo\_ao\_dlv ).  
  
    WHEN OTHERS.  
      MESSAGE e020(/scwm/delivery) WITH flt\_val ip\_ttype io\_context->applctn flt\_val INTO lv\_msg\_dummy.  
      cl\_log\_ppf=>add\_message(  
          ip\_problemclass = sppf\_pclass\_1  
          ip\_handle       = ip\_protocol ).  
      ep\_rc = 99.  
  ENDCASE.  
  
  IF ( ep\_rc = 0 ) AND  
     ( ( flt\_val = '/SCWM/WHR\_VEH\_CREATE\_NEW' ) OR  
       ( flt\_val = '/SCWM/WHR\_VEH\_CREATE\_CHANGE' ) OR  
       ( flt\_val = '/SCWM/WHR\_IN\_YARD' ) OR  
       ( flt\_val = '/SCWM/GR\_POSTING' ) OR  
       ( flt\_val = '/SCWM/WT\_CREATION' ) OR  
       ( flt\_val = '/SCWM/GI\_POSTING' ) OR  
       ( flt\_val = '/SCWM/WHR\_VAS\_KTS' ) OR  
       ( flt\_val = '/SCWM/WHR\_WAVE\_ASSIGN\_NEW' ) OR  
       ( flt\_val = '/SCWM/WHR\_WAVE\_ASSIGN\_CHANGE' ) OR  
       ( flt\_val = '/SCWM/WHR\_FULL\_PACKED' ) OR  
       ( flt\_val = '/SCWM/WM\_ACTIVITY\_FINISHED' ) OR  
       ( flt\_val = '/SCWM/GM\_POSTED' ) OR  
       ( flt\_val = 'ZSCWM\_GR\_POSTING' ) ).             " added by satish  
\* check for already sheduled PPF actions  
    ep\_rc = /scdl/cl\_common\_ppf=>set\_sched\_cond\_lock(  
              io\_context = io\_context  
              ip\_ttype = ip\_ttype ).  
  ENDIF.  
  
  IF ep\_rc = 0.  
    me->log\_event\_sched(  
        iv\_action     = ip\_ttype  
        iv\_sched\_cond = flt\_val ).  
  ELSE.  
    me->log\_event\_sched(  
        iv\_action     = ip\_ttype  
        iv\_no\_res     = abap\_true  
        iv\_sched\_cond = flt\_val ).  
  ENDIF.  
  
ENDMETHOD.