```
FUNCTION /scwm/rf_pick_piblmt_pbo.
*"----------------------------------------------------------------------
*"*"Local Interface:
*"  CHANGING
*"     REFERENCE(ORDIM_CONFIRM) TYPE  /SCWM/S_RF_ORDIM_CONFIRM
*"     REFERENCE(TT_ORDIM_CONFIRM) TYPE  /SCWM/TT_RF_ORDIM_CONFIRM
*"     REFERENCE(TT_NESTED_HU) TYPE  /SCWM/TT_RF_NESTED_HU
*"     REFERENCE(WME_VERIF) TYPE  /SCWM/S_WME_VERIF
*"     REFERENCE(RESOURCE) TYPE  /SCWM/S_RSRC
*"     REFERENCE(T_RF_PICK_HUS) TYPE  /SCWM/TT_RF_PICK_HUS
*"     REFERENCE(WHO) TYPE  /SCWM/S_WHO_INT
*"----------------------------------------------------------------------

  DATA: ls_mat_global       TYPE /scwm/s_material_global,
        lt_mat_uom          TYPE /scwm/tt_material_uom,
        ls_mat_uom          TYPE /scwm/s_material_uom,
        ls_ordim_confirm    TYPE /scwm/s_rf_ordim_confirm,
        ls_ordim_o          TYPE /scwm/ordim_o,
        lv_dbatch_ind       TYPE /scwm/de_dbatchind,
        mtto_stock_fields   TYPE REF TO /scwm/cl_ui_stock_fields,
        lv_field            TYPE ddobjname,
        ls_dfies            TYPE dfies,
        lt_text             TYPE tdtab_c132,
        lt_valid_prf        TYPE /scwm/tt_valid_prf_ext,
        ls_valid_prf_pickhu TYPE /scwm/s_valid_prf_ext,
        ls_valid_prf_logpos TYPE /scwm/s_valid_prf_ext,
        lv_step             TYPE /scwm/de_step,
        lv_state            TYPE /scwm/de_state,
        lv_line             TYPE i,
        lv_fcode            TYPE /scwm/de_fcode,
        lv_pickhu_verif_req TYPE xfeld VALUE IS INITIAL,
        lv_pickhu_verif     TYPE xfeld VALUE IS INITIAL,
        lv_logpos_verif     TYPE xfeld VALUE IS INITIAL,
        ls_rf_pick_hus      TYPE /scwm/s_rf_pick_hus,
        lv_char40           TYPE /scwm/de_rf_text.
  DATA: lv_applic   TYPE /scwm/de_applic,
        lv_pres_prf TYPE /scwm/de_pres_prf,
        lv_ltrans   TYPE /scwm/de_ltrans,
        lv_pickhu   TYPE /scwm/de_rf_pickhu,
        ls_huhdr_x  TYPE /scwm/huhdr,
        ls_who      TYPE /scwm/who,
        lt_ordim_o  TYPE /scwm/tt_ordim_o.
  DATA: lo_badi2 TYPE REF TO /scwm/ex_rf_pick_pickhu_det.
  DATA: cv_fcode  TYPE /scwm/de_fcode,
        lv_fcode1 TYPE /scwm/de_fcode.
  DATA: ls_pickhu TYPE /scwm/s_huident,
        lt_pickhu TYPE /scwm/tt_huident.

  DATA: lv_vlenr_initial TYPE xfeld,
        lv_stock_docno   TYPE /scwm/de_ui_stock_docno.

  DATA: ls_mat_lgnum TYPE /scwm/s_material_lgnum,
        lv_pref_uom  TYPE /scwm/de_puom.

  DATA: lv_data_entry TYPE /scwm/de_data_entry,
        lv_stock      TYPE /scwm/de_ser_stock.

  DATA: lv_is_combined TYPE boolean,
        lv_tanum       TYPE /scwm/de_rf_tanum.

  DATA: lt_combined_lead TYPE /scwm/tt_rf_tanum,
        lv_lead_to       TYPE i.

  DATA: ls_lagp_prev  TYPE /scwm/lagp.

  BREAK-POINT ID /scwm/rf_picking.

  /scwm/cl_rf_bll_srvc=>set_prmod(
                     /scwm/cl_rf_bll_srvc=>c_prmod_foreground ).
* need to prepare data for current step.

* Set execution step for exception
  gv_exec_step = wmegc_execstep_05.

* Check if we work with a PbV device
  lv_data_entry = /scwm/cl_rf_bll_srvc=>get_data_entry( ).

  IF lv_data_entry = wmegc_data_entry_voice.
    gv_exec_step = wmegc_execstep_p3.
  ENDIF.

* Initiate screen parameter
  /scwm/cl_rf_bll_srvc=>init_screen_param( ).

* Set screen parameter
  /scwm/cl_rf_bll_srvc=>set_screen_param('TT_ORDIM_CONFIRM').

* Get actual fcode
  lv_fcode = /scwm/cl_rf_bll_srvc=>get_fcode( ).

* get current line.
  lv_line = /scwm/cl_rf_bll_srvc=>get_line( ).
  IF lv_line = 0.
    lv_line = 1.
    /scwm/cl_rf_bll_srvc=>set_line( lv_line ).
  ENDIF.

  READ TABLE tt_ordim_confirm INDEX lv_line INTO ordim_confirm.


  IF ordim_confirm-flghuto = abap_false.
    lv_fcode1 = /scwm/cl_rf_comb_pick=>set_fcode_picpmt(
                         EXPORTING resource  = resource
                                   who       = who
                         CHANGING ordim_confirm = ordim_confirm
                                   tt_ordim_confirm = tt_ordim_confirm ).
    IF lv_fcode1 IS NOT INITIAL AND ordim_confirm-uncomb = abap_false.
      /scwm/cl_rf_bll_srvc=>set_prmod(
                            /scwm/cl_rf_bll_srvc=>c_prmod_background ).
      /scwm/cl_rf_bll_srvc=>set_fcode( fcode_go_to_pick_blcp ).
      RETURN.
    ELSE.
      CALL METHOD /scwm/cl_rf_comb_pick=>get_is_combined
        IMPORTING
          ev_is_combined = lv_is_combined.
      IF lv_is_combined = abap_true AND
        ordim_confirm-uncomb = abap_false.

* Check if following WT already showed in combine group before, disable PGDN.
        CALL METHOD /scwm/cl_rf_comb_pick=>get_lead
          IMPORTING
            et_combined_lead = lt_combined_lead.
        DESCRIBE TABLE lt_combined_lead LINES lv_lead_to.
        READ TABLE lt_combined_lead INTO lv_tanum INDEX lv_lead_to.
        IF sy-subrc = 0.
          IF ordim_confirm-tanum = lv_tanum.
            /scwm/cl_rf_bll_srvc=>set_screlm_invisible_on( '/SCWM/S_RF_SCRELM-PGDN').
          ENDIF.
        ENDIF.

* If there is combine group in WO, MTOs needs to be reset.
        CALL METHOD /scwm/cl_rf_comb_pick=>get_next_tanum
          EXPORTING
            ordim_confirm    = ordim_confirm
            tt_ordim_confirm = tt_ordim_confirm
            is_previous      = abap_false
          IMPORTING
            ev_next_tanum    = lv_tanum.

        IF lv_tanum <> ordim_confirm-tanum.
          READ TABLE tt_ordim_confirm INTO ls_ordim_confirm WITH KEY tanum = lv_tanum.
          IF sy-subrc = 0.
            IF ordim_confirm-vlpla = ls_ordim_confirm-vlpla.
              ordim_confirm-more = 'MTOs'.
            ELSE.
              ordim_confirm-more = ''.
            ENDIF.
          ENDIF.
        ELSE.
          ordim_confirm-more = ''.
        ENDIF.
      ENDIF.
    ENDIF.
  ENDIF.


* In Pick by Voice Catch-Weight and Serial numbers are not yet supported
  IF lv_data_entry = wmegc_data_entry_voice.
    IF ordim_confirm-cwrel = abap_true.
      MESSAGE e589. "Catch Weight is not supported with pick by voice
    ENDIF.
    CALL FUNCTION '/SCWM/RF_SN_CHECK'
      EXPORTING
        iv_lgnum    = ordim_confirm-lgnum
        iv_matid    = ordim_confirm-matid
        iv_entitled = ordim_confirm-entitled
      IMPORTING
        ev_stock    = lv_stock.
    IF lv_stock = 'C'.     "Serial number of inventory management level
      MESSAGE e590. "Serial numbers are not supported with pick by voice
    ENDIF.
  ENDIF.

  IF ordim_confirm-cwrel = abap_true.
    /scwm/cl_rf_bll_srvc=>set_screlm_required_off( '/SCWM/S_RF_ORDIM_CONFIRMD-NISTA_VERIF' ).
  ENDIF.
* If coming back from Catch-Weight - NISTA_VERIF field has to be gray and CW is not
* reachable any more
*  IF gv_cw_called = abap_true AND ordim_confirm-nista = ordim_confirm-vsola.
*    /scwm/cl_rf_bll_srvc=>set_screlm_input_off( '/SCWM/S_RF_ORDIM_CONFIRM-NISTA_VERIF' ).
*  ELSEIF gv_cw_called = abap_true AND ordim_confirm-nista <> ordim_confirm-vsola.
*    /scwm/cl_rf_bll_srvc=>set_screlm_input_on( '/SCWM/S_RF_ORDIM_CONFIRM-NISTA_VERIF' ).
*    gv_cw_called = abap_false.
*  ENDIF.

* Read original WT from database
  CALL FUNCTION '/SCWM/TO_READ_SINGLE'
    EXPORTING
      iv_lgnum        = ordim_confirm-lgnum
      iv_tanum        = ordim_confirm-tanum
      iv_flglock      = ' '
      iv_read_from_db = 'X'
    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.
    MESSAGE ID sy-msgid TYPE sy-msgty NUMBER sy-msgno
         WITH sy-msgv1 sy-msgv2 sy-msgv3 sy-msgv4.
  ELSE.
    IF ls_ordim_o-vlenr IS INITIAL.
      lv_vlenr_initial = 'X'.
    ELSE.
      CLEAR lv_vlenr_initial.
    ENDIF.
    lv_dbatch_ind = ls_ordim_o-dbind.
    ordim_confirm-nlenr_o = ls_ordim_o-nlenr.
  ENDIF.

* Read additional data (e.g. material master for description)
  TRY.
      CALL FUNCTION '/SCWM/MATERIAL_READ_SINGLE'
        EXPORTING
          iv_matid      = ordim_confirm-matid
          iv_langu      = sy-langu
          iv_entitled   = ordim_confirm-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    = lt_mat_uom
*         EV_APPLIC_PROC                   =
        .
    CATCH /scwm/cx_md_interface.
*       one/more parameter(s) missing
      MESSAGE e004 WITH ordim_confirm-matid.
    CATCH /scwm/cx_md_material_exist.
*       Material does not exist
      MESSAGE e004 WITH ordim_confirm-matid.
    CATCH /scwm/cx_md_lgnum_locid.
*       Warehouse number is not assigned to an APO Location
      MESSAGE e004 WITH ordim_confirm-matid.
    CATCH /scwm/cx_md.
      MESSAGE e004 WITH ordim_confirm-matid.
  ENDTRY.

*   Store original data before displaying in screen.
*   Actual data may changed after user input data.
  LOOP AT lt_mat_uom INTO ls_mat_uom
      WHERE meinh = ls_mat_global-meins.
    EXIT.
  ENDLOOP.

  ordim_confirm-p_brgew = ls_mat_uom-brgew.
  ordim_confirm-p_gewei = ls_mat_uom-gewei.
  ordim_confirm-p_volum = ls_mat_uom-volum.
  ordim_confirm-p_voleh = ls_mat_uom-voleh.
  ordim_confirm-p_laeng = ls_mat_uom-laeng.
  ordim_confirm-p_breit = ls_mat_uom-breit.
  ordim_confirm-p_hoehe = ls_mat_uom-hoehe.
  ordim_confirm-p_meabm = ls_mat_uom-meabm.
  ordim_confirm-maktx = ls_mat_global-maktx.
  ordim_confirm-matnr = ls_mat_global-matnr.
  ordim_confirm-vlpla_o = ordim_confirm-vlpla.
  ordim_confirm-vlenr_o = ordim_confirm-vlenr.
* If the user leaves the screen with F7 and return
*   VLENR could be filled and VLENR_O will have
*   not the original value
  IF lv_vlenr_initial IS NOT INITIAL.
    CLEAR ordim_confirm-vlenr_o.
  ENDIF.
  ordim_confirm-nlpla_o = ordim_confirm-nlpla.
  ordim_confirm-srsrc_o = ordim_confirm-srsrc.
  ordim_confirm-drsrc_o = ordim_confirm-drsrc.
* For display purpose.
  ordim_confirm-kquan_chr = ordim_confirm-kquan.
  ordim_confirm-vsola_chr = ordim_confirm-vsola.

  ordim_confirm-cwunit = ls_mat_global-cwunit.
  ordim_confirm-cwrel  = ls_mat_global-cwrel.

  CALL FUNCTION '/SCWM/RF_CW_IND_READ'
    EXPORTING
      iv_cwrel     = ordim_confirm-cwrel
    IMPORTING
      ev_cwrel_ind = ordim_confirm-cwrel_ind.

* If no remianing quantity for scraping turn display off.
  IF ordim_confirm-kquan = 0.
    /scwm/cl_rf_bll_srvc=>set_screlm_invisible_on(
                                   gc_scr_elmnt_kquan_chr ).
  ENDIF.
* If Material is not batch relevant turn display+verification off.
  IF ls_mat_global-batch_req IS INITIAL AND lv_dbatch_ind IS INITIAL.
    /scwm/cl_rf_bll_srvc=>set_screlm_invisible_on(
                                    gc_scr_elmnt_batch_vrf ).
    /scwm/cl_rf_bll_srvc=>set_screlm_invisible_on(
                                    gc_scr_elmnt_rfbatch ).
  ELSE.
    IF ordim_confirm-batch IS INITIAL.
      /scwm/cl_rf_bll_srvc=>set_screlm_input_on(
                                      gc_scr_elmnt_rfbatch ).
      /scwm/cl_rf_bll_srvc=>set_screlm_invisible_on(
                                      gc_scr_elmnt_batch_vrf ).
      IF NOT lv_dbatch_ind IS INITIAL.
```

```
        ordim_confirm-dbind = lv_dbatch_ind.
      ENDIF.
    ENDIF.
    IF NOT ls_mat_global-batch_req IS INITIAL.
      ordim_confirm-batch_req = gc_xfeld.

      IF NOT ordim_confirm-batchid IS INITIAL.
        /scwm/cl_rf_bll_srvc=>set_screlm_invisible_off(
                                    gc_scr_elmnt_batch_vrf ).
        /scwm/cl_rf_bll_srvc=>set_screlm_input_on(
                                      gc_scr_elmnt_batch_vrf ).
        /scwm/cl_rf_bll_srvc=>set_screlm_input_off(
                                        gc_scr_elmnt_rfbatch ).

        ordim_confirm-batchid_o = ordim_confirm-batchid.

        IF ls_ordim_o-batchid IS INITIAL.
          /scwm/cl_rf_bll_srvc=>set_screlm_invisible_on(
                                      gc_scr_elmnt_batch_vrf ).
          CLEAR ordim_confirm-batchid_o.
        ENDIF.

        IF mtto_stock_fields IS NOT BOUND.
          CREATE OBJECT mtto_stock_fields.
        ENDIF.
        mtto_stock_fields->get_batchno_by_id(
            EXPORTING
              iv_batchid = ordim_confirm-batchid
            RECEIVING
              ev_charg   = ordim_confirm-batch ).
        ordim_confirm-batch_o = ordim_confirm-batch.
      ENDIF.
    ENDIF.
  ENDIF.
  CLEAR ls_ordim_o.

* respect the pick_all functionality - change visibility of field
  IF ( ordim_confirm-pick_all = 2 ).
    /scwm/cl_rf_bll_srvc=>set_screlm_invisible_on(
                            gc_scr_elmnt_vsola_chr ).
  ENDIF.

* Clear Hazmat indicator text.
  CLEAR ordim_confirm-hazmat_ind.

  IF ordim_confirm-batch IS NOT INITIAL.
    ordim_confirm-rfbatch = ordim_confirm-batch.
  ENDIF.

* In case it is needed read the hazmat indicator text.
  IF NOT ls_mat_global-hazmat IS INITIAL.

    CALL FUNCTION '/SCWM/RF_HAZMAT_IND_READ'
      EXPORTING
        iv_hazmat     = ls_mat_global-hazmat
      IMPORTING
        ev_hazmat_ind = ordim_confirm-hazmat_ind.
  ENDIF.

* Clear Hazmat indicator text.
  CLEAR ordim_confirm-text_ind.
```

```
* Read texts (Delivery and Hazardous material)
  CALL FUNCTION '/SCWM/RF_TEXT_GET'
    EXPORTING
      iv_matid   = ordim_confirm-matid
      iv_rdoccat = ordim_confirm-rdoccat
      iv_rdocid  = ordim_confirm-rdocid
      iv_ritmid  = ordim_confirm-ritmid
      iv_lgnum   = ordim_confirm-lgnum
      iv_actty   = ordim_confirm-act_type
    IMPORTING
      et_text    = lt_text
    EXCEPTIONS
      error      = 1
      OTHERS     = 2.
  IF sy-subrc <> 0.
    MESSAGE ID sy-msgid TYPE sy-msgty NUMBER sy-msgno
          WITH sy-msgv1 sy-msgv2 sy-msgv3 sy-msgv4.
  ENDIF.

* If text exists
  IF lt_text IS NOT INITIAL.
*   adjust text to screen size
    CALL METHOD /scwm/cl_rf_bll_srvc=>set_rf_text
      EXPORTING
        it_text = lt_text.

*   Fill the text indicator with the text of data element
    CALL FUNCTION '/SCWM/RF_TEXT_IND_READ'
      EXPORTING
        iv_text     = gc_xfeld
      IMPORTING
        ev_text_ind = ordim_confirm-text_ind.
  ENDIF.

* Clear stock doccat indicator text
  CLEAR ordim_confirm-stock_doccat_ind.
* fill the relevant text for screen display
  IF NOT ordim_confirm-stock_doccat IS INITIAL.
    CALL FUNCTION '/SCWM/RF_DOCCAT_TXT_READ'
      EXPORTING
        iv_stock_doccat      = ordim_confirm-stock_doccat
      IMPORTING
        ev_stock_doccat_text = ordim_confirm-stock_doccat_ind.
  ENDIF.

* Clear pick_all text.
  CLEAR ordim_confirm-pick_all_ind.

* In case it is needed read the pick all text.
  IF NOT ordim_confirm-pick_all IS INITIAL.
    MOVE '/SCWM/DE_RF_PICK_ALL_IND' TO lv_field.
    CALL FUNCTION 'DDIF_FIELDINFO_GET'
      EXPORTING
        tabname        = lv_field
        langu          = sy-langu
        all_types      = 'X'
      IMPORTING
        dfies_wa       = ls_dfies
      EXCEPTIONS
        not_found      = 1
        internal_error = 2
        OTHERS         = 3.

*   Fill in the hazmat indicator text
    IF sy-subrc = 0.
*     Move short text to hazmat indicator
      MOVE ls_dfies-scrtext_s TO ordim_confirm-pick_all_ind.
    ENDIF.
  ENDIF.

  CALL FUNCTION '/SCWM/RF_SN_CHECK'
    EXPORTING
      iv_lgnum    = ordim_confirm-lgnum
      iv_matid    = ordim_confirm-matid
      iv_entitled = ordim_confirm-entitled
      iv_difty    = ' '
    IMPORTING
      ev_stock    = ordim_confirm-sn_type
*     ES_SERIAL   =
    .
* Set starting time and processor
  ordim_confirm-processor = sy-uname.
  IF ordim_confirm-started_at IS INITIAL.
    GET TIME STAMP FIELD ordim_confirm-started_at.
  ENDIF.

  MOVE-CORRESPONDING ordim_confirm TO ls_ordim_o.
* Getting Pick-HU system recommendation.
  CLEAR lt_pickhu.
  LOOP AT t_rf_pick_hus INTO ls_rf_pick_hus.
    ls_pickhu-lgnum = ordim_confirm-lgnum.
    ls_pickhu-huident = ls_rf_pick_hus-huident.
    APPEND ls_pickhu TO lt_pickhu.
  ENDLOOP.

  IF ordim_confirm-huent IS INITIAL.
* In case of overwrite huent determined in combined partial confirm.
    CALL FUNCTION '/SCWM/HUENT_DET'
      EXPORTING
        is_ordim_o   = ls_ordim_o
        iv_rsrc      = resource-rsrc
        it_pickhu    = lt_pickhu
        iv_desc_sort = 'X'
      IMPORTING
        ev_huent     = ordim_confirm-huent
        ev_nlenr     = ordim_confirm-pickhu
      EXCEPTIONS
        wrong_data   = 1
        OTHERS       = 2.
    IF sy-subrc <> 0.
      MESSAGE ID sy-msgid TYPE sy-msgty NUMBER sy-msgno
              WITH sy-msgv1 sy-msgv2 sy-msgv3 sy-msgv4.
    ENDIF.

  ENDIF.



* BADI to propose pick-hu
  TRY.
      GET BADI lo_badi2
```

```
        FILTERS
          lgnum = who-lgnum.

      lv_applic   = /scwm/cl_rf_bll_srvc=>get_applic( ).
      lv_pres_prf = /scwm/cl_rf_bll_srvc=>get_pres_prf( ).
      lv_ltrans   = /scwm/cl_rf_bll_srvc=>get_ltrans( ).
      lv_step = /scwm/cl_rf_bll_srvc=>get_step( ).
      lv_state = /scwm/cl_rf_bll_srvc=>get_state( ).
      MOVE-CORRESPONDING who TO ls_who.

      CLEAR ls_ordim_o.
      REFRESH lt_ordim_o.
      LOOP AT tt_ordim_confirm INTO ls_ordim_confirm.
        MOVE-CORRESPONDING ls_ordim_confirm TO ls_ordim_o.
        APPEND ls_ordim_o TO lt_ordim_o.
      ENDLOOP.

      lv_pickhu = ordim_confirm-pickhu.
      CALL BADI lo_badi2->propose
        EXPORTING
          iv_lgnum    = ls_who-lgnum
          iv_applic   = lv_applic
          iv_pres_prf = lv_pres_prf
          iv_ltrans   = lv_ltrans
          iv_step     = lv_step
          iv_fcode    = lv_fcode
          iv_state    = lv_state
          is_who      = ls_who
          it_ordim_o  = lt_ordim_o
          is_huhdr    = ls_huhdr_x
          it_pick_hus = t_rf_pick_hus
          iv_tanum    = ordim_confirm-tanum
          iv_pickhu   = lv_pickhu
        IMPORTING
          ev_pickhu   = lv_pickhu.

      ordim_confirm-pickhu = lv_pickhu.

    CATCH cx_badi.
  ENDTRY.

* PARTI Indicator: Confirmation of split quantity
* X=TO remain open; blank=End confirmation;
* Only filled by exception.
***  ordim_confirm-parti = ' '.  "Always initial is not always right!

* if pickhu is set we set the logical position
  IF ordim_confirm-pickhu IS NOT INITIAL.
    IF NOT t_rf_pick_hus[] IS INITIAL.
      LOOP AT t_rf_pick_hus INTO ls_rf_pick_hus
        WHERE huident = ordim_confirm-pickhu.
        ordim_confirm-hupos = ls_rf_pick_hus-logpos.
        EXIT.
      ENDLOOP.
    ENDIF.
  ENDIF.

  IF gv_lgpla_prev IS NOT INITIAL.
    CALL FUNCTION '/SCWM/LAGP_READ_SINGLE'
      EXPORTING
        iv_lgnum      = ordim_confirm-lgnum
        iv_lgpla      = gv_lgpla_prev
      IMPORTING
        es_lagp       = ls_lagp_prev
      EXCEPTIONS
        wrong_input   = 1
        not_found     = 2
        enqueue_error = 3
        OTHERS        = 4.
    IF sy-subrc <> 0.
      MESSAGE ID sy-msgid TYPE sy-msgty NUMBER sy-msgno
              WITH sy-msgv1 sy-msgv2 sy-msgv3 sy-msgv4.
    ENDIF.
  ENDIF.

* Check the storage bin delta between prev. and actual storage bin
  IF ordim_confirm-nltyp = ls_lagp_prev-lgtyp AND
     ordim_confirm-nlber = ls_lagp_prev-lgber AND
     ordim_confirm-nlpla = ls_lagp_prev-lgpla.
*   Switch off input for level/bin
    /scwm/cl_rf_bll_srvc=>set_screlm_input_off(
                      '/SCWM/S_RF_ORDIM_CONFIRM-VLPLA_VERIF' ).
  ELSE.
*   Switch on input for level/bin
    /scwm/cl_rf_bll_srvc=>set_screlm_input_on(
                      '/SCWM/S_RF_ORDIM_CONFIRM-VLPLA_VERIF' ).
  ENDIF.
*  CLEAR gv_lgpla_prev.

* call the Pick by Voice specific part
  PERFORM pbv_piblmt_pbo CHANGING ordim_confirm.

  MODIFY tt_ordim_confirm FROM ordim_confirm INDEX lv_line.

  PERFORM pbv_check_for_commands_prd USING ordim_confirm
                                           tt_ordim_confirm
                                           lt_mat_uom
                                           ls_mat_global-meins.

* Fill container for application specific verification
  CALL FUNCTION '/SCWM/RF_FILL_WME_VERIF'
    EXPORTING
      iv_lgnum     = ordim_confirm-lgnum
      iv_procty    = ordim_confirm-procty
      iv_trart     = ordim_confirm-trart
      iv_act_type  = ordim_confirm-act_type
      iv_aarea     = ordim_confirm-aarea
    IMPORTING
      es_wme_verif = wme_verif.

  lv_step = /scwm/cl_rf_bll_srvc=>get_step( ).
  lv_state = /scwm/cl_rf_bll_srvc=>get_state( ).

  CALL METHOD /scwm/cl_rf_bll_srvc=>get_valid_prf
    EXPORTING
      iv_step      = lv_step
      iv_state     = lv_state
    RECEIVING
      rt_valid_prf = lt_valid_prf.

* The VLNER input validation will check the HU and validate
* what is the correct qty for the possible partial confirmation.
* Thus we have to move this validation to the top of the list.
  READ TABLE lt_valid_prf INTO ls_valid_prf_pickhu
    WITH KEY valid_obj = 'VLENR'.
  IF sy-subrc IS INITIAL.
    DELETE lt_valid_prf INDEX sy-tabix.
    INSERT ls_valid_prf_pickhu INTO lt_valid_prf INDEX 1.
    CLEAR ls_valid_prf_pickhu.
    /scwm/cl_rf_bll_srvc=>set_valid_prf( lt_valid_prf ).
  ENDIF.

  CLEAR ls_valid_prf_pickhu.
  READ TABLE lt_valid_prf INTO ls_valid_prf_pickhu
       WITH KEY param_name = 'TT_ORDIM_CONFIRM'
                valid_obj = 'PICKHU'.

  IF NOT ls_valid_prf_pickhu IS INITIAL AND
     NOT ls_valid_prf_pickhu-flg_verif IS INITIAL.
    lv_pickhu_verif = /scmb/cl_c=>boole_true.
  ENDIF.

  CLEAR ls_valid_prf_logpos.
  READ TABLE lt_valid_prf INTO ls_valid_prf_logpos
       WITH KEY param_name = 'TT_ORDIM_CONFIRM'
                valid_obj = 'LOGPOS'.

  IF NOT ls_valid_prf_logpos IS INITIAL AND
     NOT ls_valid_prf_logpos-flg_verif IS INITIAL.
    lv_logpos_verif = /scmb/cl_c=>boole_true.
  ENDIF.

  CLEAR gv_hu_vrf_in_logpos_mode.

  IF lv_logpos_verif = /scmb/cl_c=>boole_true AND
     lv_pickhu_verif = /scmb/cl_c=>boole_true.

    IF gv_postn_mngmnt IS INITIAL.
*   Turn off Pick-HU posiotion verification => HU scanning
      /scwm/cl_rf_bll_srvc=>set_screlm_invisible_on(
                                      gc_scr_elmnt_hupos_vrf ).
      /scwm/cl_rf_bll_srvc=>set_screlm_invisible_on(
                                      gc_scr_elmnt_hupos ).
      IF ordim_confirm-pickhu_verif IS INITIAL.
        /scwm/cl_rf_bll_srvc=>set_screlm_invisible_off(
                                     gc_scr_elmnt_pickhu_vrf ).
        /scwm/cl_rf_bll_srvc=>set_screlm_input_on(
                                     gc_scr_elmnt_pickhu_vrf ).
      ENDIF.
*   Disable function code VRFPHU => Can switch by PB to
*   logical position verification.
      /scwm/cl_rf_bll_srvc=>set_fcode_off( fcode_vrfphu ).
      /scwm/cl_rf_bll_srvc=>set_fcode_on( fcode_vrflop ).
    ELSE.
*      Turn off Pick-HU verification => No HU scanning
      /scwm/cl_rf_bll_srvc=>set_screlm_invisible_on(
                                      gc_scr_elmnt_pickhu_vrf ).
      IF ordim_confirm-hupos_verif IS INITIAL.
        /scwm/cl_rf_bll_srvc=>set_screlm_invisible_off(
                                     gc_scr_elmnt_hupos_vrf ).
        /scwm/cl_rf_bll_srvc=>set_screlm_invisible_off(
                                     gc_scr_elmnt_hupos ).
        /scwm/cl_rf_bll_srvc=>set_screlm_input_on(
                                     gc_scr_elmnt_hupos_vrf ).
      ENDIF.
*   Disable function code VRFLOP => Can switch by PB to
*   Pick-HU verification by scanning the HU number.
      /scwm/cl_rf_bll_srvc=>set_fcode_off( fcode_vrflop ).
      /scwm/cl_rf_bll_srvc=>set_fcode_on( fcode_vrfphu ).
    ENDIF.
  ELSEIF lv_logpos_verif = /scmb/cl_c=>boole_true AND
         lv_pickhu_verif = /scmb/cl_c=>boole_false.
* If LOGPOS is enabled , but PICKHU verification is off
    IF gv_postn_mngmnt IS INITIAL.
      /scwm/cl_rf_bll_srvc=>set_screlm_invisible_on(
                                    gc_scr_elmnt_hupos_vrf ).
      /scwm/cl_rf_bll_srvc=>set_screlm_invisible_on(
                                    gc_scr_elmnt_hupos ).
* Disable function code VRFPHU => Can switch by PB to
* logical position verification.
      /scwm/cl_rf_bll_srvc=>set_fcode_off( fcode_vrfphu ).
      /scwm/cl_rf_bll_srvc=>set_fcode_on( fcode_vrflop ).
    ENDIF.
  ELSE.
*   Disable both function codes VRFPHU and VRFLOP =>
*   PB is not displayed at all.
    /scwm/cl_rf_bll_srvc=>set_fcode_off( fcode_vrfphu ).
    /scwm/cl_rf_bll_srvc=>set_fcode_off( fcode_vrflop ).
  ENDIF.

* Special logic for pick by voice OR cloud
* If no pick-hu is proposed we close the pick-HU field
  IF ( ( lv_data_entry = wmegc_data_entry_voice OR
         CAST /scwm/if_tm_global_info( /scwm/cl_tm_factory=>get_service( /scwm/cl_tm_factory=>sc_gl
    AND
       ( ordim_confirm-pickhu IS INITIAL AND
         ordim_confirm-hupos IS INITIAL ) ).
    /scwm/cl_rf_bll_srvc=>set_screlm_invisible_on(
                                    gc_scr_elmnt_pickhu_vrf ).
  ENDIF.

  gv_last_step = lv_step.

* Fill the list for the valid Pick-HU's
  /scwm/cl_rf_bll_srvc=>init_listbox(
                      '/SCWM/S_RF_ORDIM_CONFIRM-PICKHU_VERIF' ).
  /scwm/cl_rf_bll_srvc=>init_listbox(
                      '/SCWM/S_RF_ORDIM_CONFIRM-HUPOS_VERIF' ).

  LOOP AT t_rf_pick_hus INTO ls_rf_pick_hus.
*   Only Pick-HU's with initial or same distribution group.
    IF ( ( ls_rf_pick_hus-dstgrp = ordim_confirm-dstgrp OR
           ls_rf_pick_hus-dstgrp IS INITIAL ) AND
         ( ls_rf_pick_hus-huident IS NOT INITIAL ) ).
      WRITE ls_rf_pick_hus-huident TO lv_char40.
      /scwm/cl_rf_bll_srvc=>insert_listbox(
        iv_fieldname = '/SCWM/S_RF_ORDIM_CONFIRM-PICKHU_VERIF'
        iv_value = lv_char40
        iv_text = lv_char40
        iv_add_dest_field = '/SCWM/S_RF_ORDIM_CONFIRM-PICKHU' ).

      IF ls_rf_pick_hus-logpos IS NOT INITIAL.
        CONCATENATE ls_rf_pick_hus-logpos lv_char40
          INTO lv_char40 SEPARATED BY space.
        /scwm/cl_rf_bll_srvc=>insert_listbox(
          iv_fieldname = '/SCWM/S_RF_ORDIM_CONFIRM-HUPOS_VERIF'
          iv_value = ls_rf_pick_hus-logpos
          iv_text = lv_char40
          iv_add_dest_field = '/SCWM/S_RF_ORDIM_CONFIRM-HUPOS' ).
      ENDIF.
    ENDIF.
  ENDLOOP.

  PERFORM switch_fields_for_dd USING who lv_line t_rf_pick_hus
                               CHANGING tt_ordim_confirm ordim_confirm.


* Switch off input for SUoM
  /scwm/cl_rf_bll_srvc=>set_screlm_input_off(
                          '/SCWM/S_RF_ORDIM_CONFIRM-ALTME' ).

* Fill SUoM F8 list
  IF ls_mat_lgnum-puom_wh IS NOT INITIAL.
    lv_pref_uom = ls_mat_lgnum-puom_wh.
  ELSEIF ls_mat_global-puom IS NOT INITIAL.
    lv_pref_uom = ls_mat_global-puom.
  ENDIF.

  CALL FUNCTION '/SCWM/RF_SUOM_F8_LIST'
    EXPORTING
      iv_field      = '/SCWM/S_RF_ORDIM_CONFIRM-ALTME'
      iv_lgnum      = ordim_confirm-lgnum
      iv_matid      = ordim_confirm-matid
      iv_pref_uom   = lv_pref_uom
      iv_buom       = ls_mat_global-meins
      it_mat_uom    = lt_mat_uom
      iv_data_entry = lv_data_entry.

""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""$"$SE:
*$*$-Start: (1)----------------------------------------------------------------------------------$*$
ENHANCEMENT 1  ZENH_ADD_CUSTOMER_DETAILS_RF_2.    "active version

******************************************************************************************
*                                  WIPRO CONSUMERS                                       *
******************************************************************************************
* OBJECT INFORMATION                                                                     *
******************************************************************************************
* Module              : EWM                                                              *
* Developer           : Adam Siraj                                                       *
* Requestor           : Rishabh Verma/Praveen K                                          *
* Date Of Creation    : 14.09.2023                                                       *
* Transport Request   : H1DK900407                                                       *
* Development Object  : ZENH_ADD_CUSTOMER_DETAILS_RF_2                                   *
* Description         : Adding customer name and location to rf bulk picking screen      *
*========================================================================================*
* MODIFICATION HISTORY                                                                   *
*----------------------------------------------------------------------------------------*
* Mod. No.   Date          Name            Transport Number         Change Desc.         *
*----------------------------------------------------------------------------------------*
*                                                                                        *
*                                                                                        *
*--------------------------------------------------------------------------------------*

CONSTANTS: lv_custom_prf TYPE /scwm/de_pres_prf VALUE 'Z602',
           lv_reqname    TYPE zde_reqname VALUE 'OBD_RF',
           lv_paramname  TYPE zde_param_id VALUE 'WPT_CUS_DSPLY',
           lv_party_role TYPE /scdl/dl_party_role VALUE 'STPRT'.

CALL METHOD /scwm/cl_rf_bll_srvc=>get_pres_prf
  RECEIVING
    rv_pres_prf = lv_pres_prf.

IF lv_pres_prf EQ lv_custom_prf.

  SELECT param_value FROM zewmt_param INTO TABLE @DATA(lt_wpt) WHERE reqname EQ @lv_reqname
    AND param_name EQ @lv_paramname.

  IF line_exists( lt_wpt[ param_value = ordim_confirm-procty ] ).

    DATA(lwa_include) = VALUE /scwm/dlv_query_incl_str_prd( head_status = abap_true
                                                            head_partyloc = abap_true
                                                            item_partyloc = abap_true ).
    DATA: lt_docid TYPE /scwm/dlv_docid_item_tab.

    lt_docid = VALUE #( ( docid = ordim_confirm-rdocid ) ).

    DATA(lo_dlv) = NEW /scwm/cl_dlv_management_prd( ).
    TRY.
        lo_dlv->query(
         EXPORTING
           it_docid         = lt_docid
           iv_whno          = ordim_confirm-lgnum
           is_include_data  = lwa_include
          IMPORTING
            et_headers      = DATA(lt_headers)
            et_items        = DATA(lt_items) ).

      CATCH /scdl/cx_delivery.
    ENDTRY.

    READ TABLE lt_headers INTO DATA(lwa_headers) WITH KEY docid = ordim_confirm-rdocid.
    READ TABLE lwa_headers-partyloc INTO DATA(lwa_partyloc) WITH KEY party_role = lv_party_role.

    SELECT SINGLE customer FROM i_businesspartnercustomer INTO @DATA(lv_customer) WHERE businesspar
    IF sy-subrc EQ 0.
      SELECT SINGLE name1, ort01 FROM kna1 INTO @DATA(ls_populate) WHERE kunnr EQ @lv_customer.
    ENDIF.

    ordim_confirm-zzname1 = ls_populate-name1.
    ordim_confirm-zzort01 = ls_populate-ort01.

*    MODIFY tt_ordim_confirm FROM ordim_confirm INDEX 1.
    MODIFY tt_ordim_confirm FROM ordim_confirm TRANSPORTING zzname1 zzort01 WHERE tanum EQ ordim_co

    CALL METHOD /scwm/cl_rf_bll_srvc=>set_screen_param
      EXPORTING
        iv_param_name = 'ORDIM_CONFIRM'.

    CALL METHOD /scwm/cl_rf_bll_srvc=>set_screen_param
      EXPORTING
        iv_param_name = 'TT_ORDIM_CONFIRM'.

  ELSE.         "If WPT does not exist in param table, we ae setting Customer name and location fie

    CALL METHOD /scwm/cl_rf_bll_srvc=>set_screlm_invisible_on
      EXPORTING
```

```
        iv_screlm_name = '/SCWM/S_RF_ORDIM_CONFIRM-ZZNAME1'.

    CALL METHOD /scwm/cl_rf_bll_srvc=>set_screlm_invisible_on
      EXPORTING
        iv_screlm_name = '/SCWM/S_RF_ORDIM_CONFIRM-ZZORT01'.

  ENDIF.                "for WPT parameter check
ENDIF.                  "for display profile check

ENDENHANCEMENT.
*$*$-End:   (1)---------------------------------------------------------------------------------$*$
ENDFUNCTION.
```