

FUNCTION /SCWM/RF_PICK_P IPLHU_PBO.

```
* "-----
* " * "Local Interface:
* "   CHANGING
* "       REFERENCE(RESOURCE) TYPE   /SCWM/S_RSRC
* "       REFERENCE(ORDIM_CONFIRM) TYPE /SCWM/S_RF_ORDIM_CONFIRM
* "       REFERENCE(TT_ORDIM_CONFIRM) TYPE /SCWM/TT_RF_ORDIM_CONFIRM
* "       REFERENCE(WME_VERIF) TYPE   /SCWM/S_WME_VERIF
* "-----
```

```
DATA: lv_line          TYPE i,
      lv_tabline       TYPE i,
      lv_fcode         TYPE /scwm/de_fcode,
      lv_fcode1        TYPE /scwm/de_fcode,
      ls_huhdr         TYPE /scwm/s_huhdr_int,
      ls_ordim_confirm TYPE /scwm/s_rf_ordim_confirm,
      lv_field         TYPE ddoobjname,
      ls_dfies         TYPE dfies,
      lv_place_inv     TYPE /scwm/de_place_inv,
      lv_unique_dest   TYPE xfeld,
      lv_dest          TYPE /scwm/de_lgpla.
```

```
DATA: ls_docid_tab     TYPE /scwm/dlv_docid_item_str.
DATA: lt_docid_tab     TYPE /scwm/dlv_docid_item_tab.
DATA: ls_text          TYPE /scwm/s_rf_text.
DATA: lt_text          TYPE /scwm/tt_rf_text.
```

```
DATA: ls_t331 TYPE /scwm/t331,
      ls_lagp TYPE /scwm/lagp.
```

```
DATA: ls_lagp_prev TYPE /scwm/lagp.
```

```
DATA: lv_data_entry TYPE /scwm/de_data_entry,
      lv_return      TYPE xfeld.
```

```
DATA: lv_asynchronous_processing TYPE boole_d.
```

```
DATA: lt_rf_text TYPE tdtab_c132.
```

```
STATICS: hu_oref TYPE REF TO /scwm/cl_wm_packing.
```

```
FIELD-SYMBOLS:
  <ls_ordim_confirm> LIKE LINE OF tt_ordim_confirm.
```

```
BREAK-POINT ID /scwm/rf_picking.
```

```
CLEAR lt_rf_text.
CALL METHOD /scwm/cl_rf_bll_srvc=>set_rf_text
  EXPORTING
    it_text = lt_rf_text.
```

```
* need to prepare data for current step.
```

```
* Set execution step for exception
gv_exec_step = wmegc_execstep_04.
```

```
* 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_p2.
```

```

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( ).

* Set the processing mode to foreground
/scwm/cl_rf_bll_srvc=>set_prmod(
    /scwm/cl_rf_bll_srvc=>c_prmod_foreground ).

* 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.

* Switch to fcode of scan destination for one time if destinaitons are same
IF lv_data_entry NE wmegc_data_entry_voice.
    lv_fcode1 = /scwm/cl_rf_set_fcode_pckhu=>set_fcode(
        EXPORTING resource = resource
        CHANGING ordim_confirm = ordim_confirm
        tt_ordim_confirm = tt_ordim_confirm ).

    IF lv_fcode1 IS NOT INITIAL AND
        lv_fcode1 <> lv_fcode.
* Set Fcode
/scwm/cl_rf_bll_srvc=>set_fcode( lv_fcode1 ).
/scwm/cl_rf_bll_srvc=>set_prmod(
    /scwm/cl_rf_bll_srvc=>c_prmod_background ).

    EXIT.
ENDIF.
ENDIF.

READ TABLE tt_ordim_confirm INDEX lv_line INTO ordim_confirm.

IF /scwm/cl_rf_settings=>get_instance( )->is_async_pick_hu_wt_create( iv_lgnum = ordim_confirm-lg
    "the asynchronous creation of the pick HU warehouse task might have failed, e.g. due to locking
    "if this is the case we restart the queue here as there is a good chance that it works now
    "and that it has finished until the user confirms the warehouse task
    /scwm/cl_rf_confirm_picking=>get_instance( )->restart_pick_hu_wt_queue( ordim_confirm-vlenr ).

ENDIF.

CALL FUNCTION '/SCWM/RSRC_EXCEPTION_LIST_FILL'
    EXPORTING
        iv_lgnum = ordim_confirm-lgnum
        iv_exec_step = gv_exec_step
        iv_procty = ordim_confirm-procty.

* Save original TO data before displaying screen
ordim_confirm-vlpla_o = ordim_confirm-vlpla.
ordim_confirm-vlenr_o = ordim_confirm-vlenr.

ordim_confirm-nlpla_o = ordim_confirm-nlpla.

* Set bin display.
IF ordim_confirm-nlpla IS INITIAL.

```

```

CALL FUNCTION '/SCWM/HU_FOLLOW_NLPLA'
  EXPORTING
    iv_lgnum      = ordim_confirm-lgnum
    iv_matid      = ordim_confirm-matid
    iv_doccat     = ordim_confirm-doccat
    iv_docid      = ordim_confirm-rdocid
    iv_itemid     = ordim_confirm-ritmid
    iv_huident    = ordim_confirm-nlenr
    iv_nltyp      = ordim_confirm-nltyp
    iv_nlber      = ordim_confirm-nlber
    iv_nlpla      = ordim_confirm-nlpla
    iv_who        = ordim_confirm-who
  IMPORTING
    ev_nlpla      = ordim_confirm-nlpla
  EXCEPTIONS
    internal_err  = 1
    dlv_read_err  = 2
    OTHERS        = 3.
IF sy-subrc <> 0.
  MESSAGE ID sy-msgid TYPE sy-msgty NUMBER sy-msgno
    WITH sy-msgv1 sy-msgv2 sy-msgv3 sy-msgv4.
ELSE.
*   update the relevant field for the new bin
  CALL FUNCTION '/SCWM/LAGP_READ_SINGLE'
    EXPORTING
      iv_lgnum      = ordim_confirm-lgnum
      iv_lgpla      = ordim_confirm-nlpla
    IMPORTING
      es_lagp       = ls_lagp
    EXCEPTIONS
      wrong_input   = 1
      not_found     = 2
      enqueue_error = 3
      OTHERS        = 4.
  IF sy-subrc = 0.
    ordim_confirm-nltyp = ls_lagp-lgtyp.
    ordim_confirm-nlber = ls_lagp-lgber.
  ENDIF.
ENDIF.
ENDIF.

"With PbV we don't support generic WT
IF lv_data_entry = wmegc_data_entry_voice AND
  ordim_confirm-nlpla IS INITIAL.
  MESSAGE e800.
ENDIF.

* If NLPLA is still initial -> turn input on and verification off
IF lv_data_entry <> wmegc_data_entry_voice.
  IF ordim_confirm-nlpla IS INITIAL.
    /scwm/cl_rf_bll_srvc=>set_screlm_input_off(
      gc_scr_elmnt_nlpla_vrf ).
    /scwm/cl_rf_bll_srvc=>set_screlm_invisible_on(
      gc_scr_elmnt_nlpla_vrf ).
    /scwm/cl_rf_bll_srvc=>set_screlm_input_on(
      gc_scr_elmnt_nlpla ).
  CALL METHOD /scwm/cl_rf_bll_srvc=>set_field
    EXPORTING
      iv_field = '/SCWM/S_RF_ORDIM_CONFIRM-NLPLA'.
  /scwm/cl_rf_bll_srvc=>fldinit_switch( abap_true ).

```

```

ELSE.
    /scwm/cl_rf_bll_srvc=>set_screlm_input_on(
        gc_scr_elmnt_nlpla_vrf ).
    /scwm/cl_rf_bll_srvc=>set_screlm_invisible_off(
        gc_scr_elmnt_nlpla_vrf ).
    /scwm/cl_rf_bll_srvc=>set_screlm_input_off(
        gc_scr_elmnt_nlpla ).
ENDIF.
ENDIF.

ordim_confirm-nlenr_o = ordim_confirm-nlenr.
* As default in HUTO source HU becomes destination HU.
IF ordim_confirm-nlenr IS INITIAL.
    ordim_confirm-pickhu = ordim_confirm-vlenr.
ELSE.
    ordim_confirm-pickhu = ordim_confirm-nlenr.
ENDIF.
ordim_confirm-srsrc_o = ordim_confirm-srsrc.
ordim_confirm-drsrsrc_o = ordim_confirm-drsrsrc.

* For display purpose.
ordim_confirm-kquan_chr = ordim_confirm-kquan.
ordim_confirm-vsola_chr = ordim_confirm-vsola.
* 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.

* Open destination HU if customized
DATA: lv_fld_ctrl_hu_dest TYPE /scwm/de_fld_ctrl_hu_dest_sty.

IF ordim_confirm-pickhu IS INITIAL.
    CALL FUNCTION '/SCWM/FLD_CTRL_HU_DEST_GET'
        EXPORTING
            iv_lgnum          = ordim_confirm-lgnum
            iv_lgtyp          = ordim_confirm-nltyp
            iv_entitled       = ordim_confirm-entitled
            iv_matid          = ordim_confirm-matid
            iv_matnr          = ordim_confirm-matnr
            iv_read_prd       = 'X'
        IMPORTING
            ev_fld_ctrl_hu_dest = lv_fld_ctrl_hu_dest.

* Verification field is always off
/scwm/cl_rf_bll_srvc=>set_screlm_input_off(
    '/SCWM/S_RF_ORDIM_CONFIRM-PICKHU_VERIF' ).
/scwm/cl_rf_bll_srvc=>set_screlm_invisible_on(
    '/SCWM/S_RF_ORDIM_CONFIRM-PICKHU_VERIF' ).

IF lv_fld_ctrl_hu_dest IS NOT INITIAL.    "HU_DEST open for input
    /scwm/cl_rf_bll_srvc=>set_screlm_input_on(
        '/SCWM/S_RF_ORDIM_CONFIRM-PICKHU' ).
ELSE.
    /scwm/cl_rf_bll_srvc=>set_screlm_input_off(
        '/SCWM/S_RF_ORDIM_CONFIRM-PICKHU' ).
ENDIF.
ENDIF.

"New logic 'SUMAHU' in Pick-by-Voice:
"If the user has only pick-HUs for one destination on the resource, the system tells the user

```

"to place all the HU's at once on the destination instead of one HU after the other
lv_unique_dest = abap_true.

"Count number of HUs

CLEAR ordim_confirm-sumahu.

LOOP AT tt_ordim_confirm INTO ls_ordim_confirm

WHERE srsrc = resource-rsrc AND

flghuto = gc_xfeld.

ordim_confirm-sumahu = ordim_confirm-sumahu + 1.

"Check if we have only one destination

IF lv_dest IS INITIAL.

lv_dest = ls_ordim_confirm-nlpla.

ENDIF.

IF lv_dest <> ls_ordim_confirm-nlpla.

CLEAR lv_unique_dest.

ENDIF.

ENDLOOP.

IF lv_data_entry = wmegc_data_entry_voice.

IF lv_unique_dest IS INITIAL.

ordim_confirm-sumahu = 1.

ENDIF.

"If we have more than one pick-HU, we close the HU validation and open the SUMAHU validation

IF ordim_confirm-sumahu > 1.

"save the original data also for the other warehouse tasks that are

"confirmed by the SUMAHU logic

LOOP AT tt_ordim_confirm ASSIGNING <ls_ordim_confirm>

WHERE srsrc = resource-rsrc AND

nlpla = ordim_confirm-nlpla AND

flghuto = abap_true.

<ls_ordim_confirm>-vlpla_o = <ls_ordim_confirm>-vlpla.

<ls_ordim_confirm>-vlenr_o = <ls_ordim_confirm>-vlenr.

<ls_ordim_confirm>-nlpla_o = <ls_ordim_confirm>-nlpla.

<ls_ordim_confirm>-nlenr_o = <ls_ordim_confirm>-nlenr.

<ls_ordim_confirm>-srsrc_o = <ls_ordim_confirm>-srsrc.

<ls_ordim_confirm>-drsrc_o = <ls_ordim_confirm>-drsrc.

ENDLOOP.

/scwm/cl_rf_bll_srvc=>set_screlm_input_off(

'/SCWM/S_RF_ORDIM_CONFIRM-PICKHU_VERIF').

/scwm/cl_rf_bll_srvc=>set_screlm_invisible_on(

'/SCWM/S_RF_ORDIM_CONFIRM-PICKHU_VERIF').

/scwm/cl_rf_bll_srvc=>set_screlm_input_on(

'/SCWM/S_RF_ORDIM_CONFIRM-SUMAHU_VERIF').

/scwm/cl_rf_bll_srvc=>set_screlm_invisible_off(

'/SCWM/S_RF_ORDIM_CONFIRM-SUMAHU_VERIF').

ELSEIF ordim_confirm-sumahu = 1.

* If we have 1 pick-hu, we open the HU validation and
* close the SUMAHU validation

/scwm/cl_rf_bll_srvc=>set_screlm_input_off(

'/SCWM/S_RF_ORDIM_CONFIRM-SUMAHU_VERIF').

/scwm/cl_rf_bll_srvc=>set_screlm_invisible_on(

'/SCWM/S_RF_ORDIM_CONFIRM-SUMAHU_VERIF').

/scwm/cl_rf_bll_srvc=>set_screlm_input_on(

'/SCWM/S_RF_ORDIM_CONFIRM-PICKHU_VERIF').

/scwm/cl_rf_bll_srvc=>set_screlm_invisible_off(

'/SCWM/S_RF_ORDIM_CONFIRM-PICKHU_VERIF').

ENDIF.

ENDIF.

```

* Set previous storage bin and storage type, if storage bin is
* same for previous warehouse task, no need to scan the destination
* bin. The logic is only valid when you activate the single scan
IF ordim_confirm-sumchu = 1.
  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-NLPLA_VERIF' ).
  ENDIF.
  CLEAR gv_lgpla_prev.
ENDIF.

* Create HU instance reference if needed.
IF hu_oref IS NOT BOUND.
  CREATE OBJECT hu_oref.
  CALL METHOD hu_oref->init
    EXPORTING
      iv_lgnum = ordim_confirm-lgnum.
ENDIF.

* read all necessary data: HU Status
CALL METHOD hu_oref->get_hu
  EXPORTING
    iv_huident = ordim_confirm-vlenr
    IV_GUID_HU =
    iv_lock    = gc_lock
  IMPORTING
    ET_HUITM   =
    es_huhdr   = ls_huhdr
    ET_HUHDR   =
    ET_HUTREE  =
    et_huref   = lt_huref
  EXCEPTIONS
    not_found = 1
    OTHERS    = 2.

* Check if position management is active.
IF gc_manual_postn_mng IS NOT INITIAL OR gc_auto_postn_mng IS NOT INITIAL.
  ordim_confirm-hupos = ls_huhdr-logpos.
ENDIF.

```

```

* Clear Hazmat indicator text.
CLEAR ordim_confirm-hazmat_ind.

* In case it is needed read the hazmat indicator text.
IF NOT ls_huhdr-hzmt IS INITIAL.
    CALL FUNCTION '/SCWM/RF_HAZMAT_IND_READ'
        EXPORTING
            iv_hazmat      = ls_huhdr-hzmt
        IMPORTING
            ev_hazmat_ind = ordim_confirm-hazmat_ind.
ENDIF.

* 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.

* Checking if stock check at placement is required.
IF ordim_confirm-lowchk_inv IS INITIAL.
    TRY.
        CALL FUNCTION '/SCWM/TO_CONF_INV_RF'
            EXPORTING
                iv_lgnum      = ordim_confirm-lgnum
                iv_tanum      = ordim_confirm-tanum
            IMPORTING
                ev_place_inv = ordim_confirm-place_inv_recomm.
        CATCH /scwm/cx_core.
    ENDTRY.
ENDIF.

* In case inventory is required set text field.
IF ordim_confirm-place_inv_recomm = gc_place_inv_perform.
    MOVE '/SCWM/DE_RF_COUNT' 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 Place Inventory indicator text
IF sy-subrc = 0.
    * Move short text to Place Inventory indicator
    MOVE ls_dfies-scrtext_s TO ordim_confirm-place_inv_ind.
ENDIF.
ENDIF.

" Fill Delivery Text on Destination screen
" -----
CLEAR ordim_confirm-text_ind.
IF tt_ordim_confirm[] IS NOT INITIAL.
    LOOP AT tt_ordim_confirm INTO ls_ordim_confirm.
        IF ls_ordim_confirm-rtext IS NOT INITIAL.

```

```

ls_docid_tab-doccat = ls_ordim_confirm-rdoccat.
ls_docid_tab-docid = ls_ordim_confirm-rdocid.
CLEAR: ls_docid_tab-itemid.
APPEND ls_docid_tab TO lt_docid_tab.
ENDIF.
ENDLOOP.
SORT lt_docid_tab BY doccat docid.
DELETE ADJACENT DUPLICATES FROM lt_docid_tab COMPARING doccat docid.
ELSE.
IF ordim_confirm-rtext IS NOT INITIAL.
ls_docid_tab-doccat = ordim_confirm-rdoccat.
ls_docid_tab-docid = ordim_confirm-rdocid.
CLEAR: ls_docid_tab-itemid.
APPEND ls_docid_tab TO lt_docid_tab.
ENDIF.
ENDIF.

IF lt_docid_tab[] IS NOT INITIAL.
*   Read texts (Delivery and Hazardous material)
CALL FUNCTION '/SCWM/RF_TEXT_GET_AND_SET'
EXPORTING
    iv_lgnum      = ordim_confirm-lgnum
    iv_actty      = ordim_confirm-act_type
    iv_rdoccat    = ordim_confirm-rdoccat
    iv_rdocid     = ordim_confirm-rdocid
*   iv_ritmid     = ordim_confirm-ritmid " only Header Selection Possible
*   iv_matid      = ordim_confirm-matid
    iv_rtext      = abap_true
    iv_huident    = ordim_confirm-vlenr
    iv_dlv_txt_head = abap_true
    it_docid_tab  = lt_docid_tab
    iv_filter_txt  = abap_true
IMPORTING
    ev_text_ind   = ordim_confirm-text_ind
EXCEPTIONS
    interface_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.
ELSE.
IF ordim_confirm-text_ind IS NOT INITIAL.
CLEAR: ordim_confirm-text_scr.
CALL METHOD /scwm/cl_rf_bll_srvc=>get_rf_text
RECEIVING
    rt_rf_text = lt_text.
LOOP AT lt_text INTO ls_text.
FIND ls_text-text IN ordim_confirm-text_scr.
IF sy-subrc = 0.
CONTINUE.
ENDIF.

IF ls_text-text IS NOT INITIAL.
IF ordim_confirm-text_scr IS INITIAL.
ordim_confirm-text_scr = ls_text-text.
ELSE.
CONCATENATE
    ordim_confirm-text_scr
    ls_text-text
    INTO ordim_confirm-text_scr SEPARATED BY space.
ENDIF.

```



```

        ENDIF.
    ENDLOOP.
ELSE.
    CLEAR: ordim_confirm-text_scr.
ENDIF.
ENDIF.
ENDIF.
IF ordim_confirm-text_scr IS INITIAL.
    " hide field.
    /scwm/cl_rf_bll_srvc=>set_screlm_invisible_on(
        '/SCWM/S_RF_ORDIM_CONFIRM-TEXT_SCR'
    ).
ELSE.
    " make visible field.
    /scwm/cl_rf_bll_srvc=>set_screlm_invisible_off(
        '/SCWM/S_RF_ORDIM_CONFIRM-TEXT_SCR'
    ).
ENDIF.

PERFORM pbv_piplhu_pbo CHANGING ordim_confirm.

MODIFY tt_ordim_confirm FROM ordim_confirm INDEX lv_line.

PERFORM pbv_check_for_commands USING tt_ordim_confirm.

* 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.
*****"$"$$SE:
*$$-Start: (1)-----$$-
ENHANCEMENT 1 ZENH_ADD_CUSTOMER_DETAILS_RF_1.    "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_1              *
* Description                    : Adding customer name and location to rf picking screen *
*=====
* MODIFICATION HISTORY                                                         *
*-----
* Mod. No.   Date           Name           Transport Number           Change Desc.           *
*-----
*
*
*-----

```

DATA: lv_pres_prf TYPE /scwm/de_pres_prf.

```

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.

    SELECT SINGLE docid FROM /scwm/huref INTO @DATA(lv_docid) WHERE guid_hu EQ @ordim_confirm-sguid
    IF sy-subrc EQ 0.
      lt_docid = VALUE #( ( docid = lv_docid ) ).
    ENDIF.

    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 = lv_docid.
    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 businesspartner EQ @lv_customer.
    IF sy-subrc EQ 0.
      SELECT SINGLE name1, ort01 FROM knal 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_confirm-tanum.

  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.

```