

```

FUNCTION zfm_odo_assign_tu.
*"-----
*" "Local Interface:
*"  IMPORTING
*"      REFERENCE(IV_LGNUM) TYPE  /SCWM/LGNUM
*"      REFERENCE(IT_DATA) TYPE  STANDARD TABLE
*"-----

*****
*                                WIPRO CONSUMERS                                *
*****
*  OBJECT INFORMATION                                                         *
*****
*  Module                          : EWM                                         *
*  Developer                       : Adam Siraj                                 *
*  Requestor                       : Rishabh Verma/Praveen K                     *
*  Date Of Creation                : 25.09.2023                                 *
*  Transport Request               : H1DK900429                                 *
*  Development Object              : ZFM_ODO_ASSIGN_TU                         *
*  Description                     : Assign ODO to TU after PGI Monitor rep.     *
*=====
*  MODIFICATION HISTORY                                                         *
*-----
*  Mod. No.   Date           Name                Transport Number      Change Desc.      *
*-----
*   1         <25/10/21>    <Developer's Name>    <Mod. TR number>      <short description> *
*-----
*-----

DATA(la_include) = VALUE /scwm/dlv_query_incl_str_prd( head_status = abap_true
head_refdoc = abap_true
item_refdoc = abap_true
head_partyloc = abap_true
item_partyloc = abap_true ).

DATA: lt_docid TYPE /scwm/dlv_docid_item_tab.
DATA(lo_dlv) = NEW /scwm/cl_dlv_management_prd( ).

CONSTANTS: lc_printer          TYPE fpmedium VALUE 'PRINTER',
lc_refdoccat_erpdel TYPE /scdl/dl_refdoccat VALUE 'ERP',
lc_refdoccat_so      TYPE /scdl/dl_refdoccat VALUE 'SO',
lc_refdoccat_po      TYPE /scdl/dl_refdoccat VALUE 'PO',
lv_form_name         TYPE fpname VALUE 'ZAF_LOADING_SHEET',
lv_lgnum             TYPE /scwm/lgnum VALUE '6002',
lv_gi_status_type    TYPE /scdl/dl_status_type VALUE 'DGI'.

CLEAR: it_dlv_tu, wa, gt_return, copy_dlv_tu, copy_wa.

IF it_data IS INITIAL.
    MESSAGE s040(zewm) DISPLAY LIKE 'E'.
    RETURN.
ENDIF.

FIELD-SYMBOLS <fs_data> TYPE ANY TABLE.
ASSIGN it_data TO <fs_data>.

IF <fs_data> IS ASSIGNED.
    LOOP AT <fs_data> ASSIGNING FIELD-SYMBOL(<fs_wa>).
        ASSIGN COMPONENT 'DOCID' OF STRUCTURE <fs_wa> TO FIELD-SYMBOL(<fs_docid>).
        ASSIGN COMPONENT 'SHIP_TO_PARTY' OF STRUCTURE <fs_wa> TO FIELD-SYMBOL(<fs_ship_to_party>).
        APPEND VALUE #( docid = <fs_docid> doccat = wmegc_doccat_pdo ) TO lt_docid.
    ENDLOOP.

```

ENDIF.

TRY.

```
lo_dlv->query(  
  EXPORTING  
    it_docid      = lt_docid  
    iv_who        = lv_lgnum  
    is_include_data = la_include  
  IMPORTING  
    et_headers    = DATA(lt_headers)  
    et_items      = DATA(lt_items)  
    et_hu_headers = DATA(lt_hu_headers)  
    et_hu_items   = DATA(lt_hu_items) ).
```

CATCH /scdl/cx\_delivery.

ENDTRY.

LOOP AT lt\_headers INTO DATA(lwa\_headers).

READ TABLE lwa\_headers-partyloc INTO DATA(ls\_partyloc) WITH KEY party\_role = 'STPRT'.

SELECT SINGLE customer FROM i\_businesspartnercustomer INTO @DATA(lv\_customer) WHERE businesspartner = lv\_customer.  
IF sy-subrc EQ 0.

SELECT SINGLE kunnr, name1, ort01 FROM kna1 INTO @DATA(ls\_kna1) WHERE kunnr EQ @lv\_customer.  
ENDIF.

READ TABLE lwa\_headers-status INTO DATA(ls\_gi\_status) WITH KEY status\_type = lv\_gi\_status\_type.

READ TABLE lt\_items INTO DATA(lwa\_items) WITH KEY docno = lwa\_headers-docno.

READ TABLE lwa\_items-refdoc INTO DATA(ls\_refdoc\_erpdel) WITH KEY refdoccat = lc\_refdoccat\_erpdel.

READ TABLE lwa\_items-refdoc INTO DATA(ls\_refdoc\_so) WITH KEY refdoccat = lc\_refdoccat\_so BINARY.

READ TABLE lwa\_items-refdoc INTO DATA(ls\_refdoc\_po) WITH KEY refdoccat = lc\_refdoccat\_po BINARY.

\* SELECT SINGLE tu\_num FROM /scwm/tu\_dlv INTO @DATA(lv\_tu) WHERE docno EQ @lwa\_headers-docno.  
FIELD-SYMBOLS <fs\_data1> TYPE ANY TABLE.  
ASSIGN it\_data TO <fs\_data1>.  
DATA: lv\_tu TYPE /scwm/de\_tu\_num.

IF <fs\_data1> IS ASSIGNED.

LOOP AT <fs\_data1> ASSIGNING FIELD-SYMBOL(<fs\_wa1>).

ASSIGN COMPONENT 'DOCID' OF STRUCTURE <fs\_wa1> TO FIELD-SYMBOL(<fs\_docid1>).

IF <fs\_docid1> EQ lwa\_headers-docid.

ASSIGN COMPONENT 'TU' OF STRUCTURE <fs\_wa1> TO FIELD-SYMBOL(<fs\_tu>).

lv\_tu = <fs\_tu>.

IF lv\_tu IS NOT INITIAL.

wa-unassign = 'X'.

ENDIF.

ENDIF.

ENDLOOP.

ENDIF.

\* SHIFT lwa\_headers-docno LEFT DELETING LEADING '0'.

wa-docno = lwa\_headers-docno.

wa-refdocno = ls\_refdoc\_erpdel-refdocno.

\* wa-refdocno\_so = ls\_refdoc\_so-refdocno.

IF ls\_refdoc\_po IS NOT INITIAL.

wa-refdocno\_so = ls\_refdoc\_po-refdocno.

ELSE.

wa-refdocno\_so = ls\_refdoc\_so-refdocno.

ENDIF.

wa-ship\_to\_party = ls\_partyloc-partyno.

wa-name1 = ls\_kna1-name1.

wa-ort01 = ls\_kna1-ort01.

```
wa-status_gi = SWITCH #( ls_gi_status-status_value
                        WHEN '0' THEN TEXT-008
                        WHEN '1' THEN TEXT-009
                        WHEN '2' THEN TEXT-010
                        WHEN '9' THEN TEXT-011 ).

wa-tu = lv_tu.
APPEND wa TO it_dlv_tu.
CLEAR: lv_tu, wa.
ENDLOOP.

APPEND LINES OF it_dlv_tu TO copy_dlv_tu.
LOOP AT it_dlv_tu INTO wa.
    SHIFT wa-docno LEFT DELETING LEADING '0'.
    MODIFY it_dlv_tu FROM wa INDEX sy-tabix.
ENDLOOP.
CALL SCREEN 0100 STARTING AT 15 05 ENDING AT 135 25.

ENDFUNCTION.
```