```
*----------------------------------------------------------------------*
***INCLUDE LZFG_ODO_ASSIGN_TUI01.
*----------------------------------------------------------------------*
*&---------------------------------------------------------------------*
*&      Module  USER_COMMAND_0100  INPUT
*&---------------------------------------------------------------------*
*       text
*----------------------------------------------------------------------*
MODULE user_command_0100 INPUT.

  CASE sy-ucomm.
    WHEN 'EXT' OR 'CANCEL'.
      LEAVE TO SCREEN 0.
    WHEN 'ENTER'.

      DATA o_bom          TYPE REF TO /scwm/cl_sr_bom.
      DATA o_bo_tu        TYPE REF TO /scwm/cl_sr_bo_tu.

      DATA ls_tu_sr_act_num TYPE /scwm/s_tu_sr_act_num.
      DATA ls_asp_tu_whr    TYPE /scwm/s_asp_tu_whr.
      DATA ls_bo_tu_dlv     TYPE /scwm/s_bo_tu_dlv.
      DATA lt_bo_tu_dlv     TYPE  /scwm/tt_bo_tu_dlv.
      DATA lt_bo_tu_dlv_unassign     TYPE  /scwm/tt_bo_tu_dlv.
      DATA: ls_tu_sr_act TYPE /scwm/s_tu_sr_act_num,
            lv_message   TYPE bapi_msg,
            lv_docno     TYPE /scdl/dl_docno_int.


      DATA lv_plnstrtfrm  TYPE timestamp.
      DATA lv_plnstrtto   TYPE timestamp.
      DATA lv_plnendfrm   TYPE timestamp.
      DATA lv_plnendto    TYPE timestamp.

      CONSTANTS: lc_completed TYPE /scwm/sp_stm_goods_issue VALUE 'COMPLETED',
                 lc_lgnum     TYPE /scwm/lgnum VALUE '6002'.
      CLEAR: lv_message, gt_return.

      LOOP AT copy_dlv_tu INTO copy_wa.

        DATA: lv_seq_no TYPE /scmb/de_seq_num.
        lv_seq_no = sy-tabix.
        lv_docno = copy_wa-docno.
        SHIFT lv_docno LEFT DELETING LEADING '0'.
        READ TABLE it_dlv_tu INTO wa WITH KEY docno = lv_docno.

        IF copy_wa-tu NE wa-tu.
          " Get TU Details
          TRY.
              IF wa-status_gi EQ lc_completed.              "goods issue completed
                " Get TU Data and get TU and Activity Number
                SELECT SINGLE tu_num
                  FROM /scwm/tunit
                  INTO @DATA(lv_tu_num)
                  WHERE tu_num_ext = @wa-tu.
                IF sy-subrc <> 0.
                  CLEAR lv_message .
                  SHIFT wa-docno LEFT DELETING LEADING '0'.
                  CONCATENATE wa-docno TEXT-001 INTO lv_message .
                  PERFORM fill_return USING 'E' lv_message.
                ENDIF.
```

```
SELECT SINGLE tu_num, tu_sr_act_num
  FROM /scwm/tu_sr_act
  WHERE tu_num = @lv_tu_num AND end_actual IS INITIAL
  INTO @ls_tu_sr_act.

IF sy-subrc NE 0.
  CLEAR lv_message .
  SHIFT wa-docno LEFT DELETING LEADING '0'.
  CONCATENATE wa-docno TEXT-003 INTO lv_message .
  PERFORM fill_return USING 'E' lv_message.
  CONTINUE.
ENDIF.

SELECT SINGLE act_dir FROM /scwm/tu_sr_act INTO @DATA(lv_dir)
   WHERE tu_sr_act_num EQ @ls_tu_sr_act-tu_sr_act_num.
IF sy-subrc EQ 0.
  IF lv_dir NE '2'.
    CLEAR lv_message .
    SHIFT wa-docno LEFT DELETING LEADING '0'.
    CONCATENATE wa-docno TEXT-002 INTO lv_message .
    PERFORM fill_return USING 'E' lv_message.
  ENDIF.
ENDIF.

SELECT status_type, status_value FROM /scwm/tu_status
  INTO TABLE @DATA(lt_status)
  WHERE tu_num EQ @lv_tu_num
  AND tu_sr_act_num EQ @ls_tu_sr_act-tu_sr_act_num
  AND status_type BETWEEN 'ISR19' AND 'ISR20'.

IF sy-subrc EQ 0.
  READ TABLE lt_status INTO DATA(ls_status19) WITH KEY status_type = 'ISR19'.
  IF ls_status19-status_value IS NOT INITIAL.
    READ TABLE lt_status INTO DATA(ls_status20) WITH KEY status_type = 'ISR20'.
    IF ls_status20-status_value IS NOT INITIAL.
      CLEAR lv_message .
      SHIFT wa-docno LEFT DELETING LEADING '0'.
      CONCATENATE wa-docno TEXT-003 INTO lv_message .
      PERFORM fill_return USING 'E' lv_message.
    ENDIF.
  ELSE.           "TU is not active
    CLEAR lv_message .
    SHIFT wa-docno LEFT DELETING LEADING '0'.
    CONCATENATE wa-docno TEXT-004 INTO lv_message .
    PERFORM fill_return USING 'E' lv_message.
  ENDIF.
ENDIF.

" Get TU Details
o_bom = /scwm/cl_sr_bom=>get_instance( io_log = NEW /scwm/cl_log( ) ).
o_bo_tu = o_bom->get_bo_tu_by_key( is_tu_sr_act_num = ls_tu_sr_act ).
o_bo_tu->get_data( IMPORTING es_bo_tu_data = DATA(ls_bo_tu_data) ).

" Fill the Delivery Details
ls_asp_tu_whr = CORRESPONDING #( ls_bo_tu_data ).
" Fill Timestamp
/scwm/cl_sr_my_service=>fill_timestamp(
  EXPORTING iv_tzone = 'INDIA'
  CHANGING cs_asp_tu_whr = ls_asp_tu_whr ).

SELECT SINGLE docid FROM /scdl/db_proch_o INTO @DATA(lv1_docid)
```

```
    WHERE docno EQ @copy_wa-docno.

            ls_bo_tu_dlv        = CORRESPONDING #( ls_tu_sr_act ).  " TU Details
            ls_bo_tu_dlv-seq_num  = sy-tabix.                             " Sequence Number
            ls_bo_tu_dlv-lgnum  = lc_lgnum.                             " Warehouse
            ls_bo_tu_dlv-docid  = lv1_docid.
            ls_bo_tu_dlv-docno  = wa-docno.                "|{ lwa_docno-low ALPHA = IN }|.
            ls_bo_tu_dlv-doccat = wmegc_doccat_pdo.                            " Document Categ
            APPEND ls_bo_tu_dlv TO lt_bo_tu_dlv.
            CLEAR ls_bo_tu_dlv.

            " Add Delivery to the TU
            o_bo_tu->add_tu_dlv( EXPORTING it_bo_tu_dlv = lt_bo_tu_dlv
                                 IMPORTING et_bo_tu_dlv_success  = DATA(lt_bo_tu_dlv_succ) ).

            IF lt_bo_tu_dlv_succ IS NOT INITIAL.
              CLEAR lv_message .
              SHIFT wa-docno LEFT DELETING LEADING '0'.
              CONCATENATE wa-docno TEXT-005 INTO lv_message .
              PERFORM fill_return USING 'S' lv_message.
            ELSE.
              CLEAR lv_message .
              SHIFT wa-docno LEFT DELETING LEADING '0'.
              CONCATENATE wa-docno TEXT-014 INTO lv_message .
              PERFORM fill_return USING 'S' lv_message.
            ENDIF.

            o_bom->save( ).
            COMMIT WORK AND WAIT.
          ELSE.
            CLEAR lv_message .
            SHIFT wa-docno LEFT DELETING LEADING '0'.
            CONCATENATE wa-docno TEXT-006 INTO lv_message .
            PERFORM fill_return USING 'E' lv_message.
          ENDIF.
        CATCH /scwm/cx_sr_error .
      ENDTRY.
    ENDIF.

************************for unassigning TU from delivery************************
    IF copy_wa-unassign NE wa-unassign.
      " Get TU Details
      TRY.
          " Get TU Data and get TU and Activity Number
          SELECT SINGLE tu_num
            FROM /scwm/tunit
            INTO @lv_tu_num
            WHERE tu_num_ext = @wa-tu.
          IF sy-subrc <> 0.
            CLEAR lv_message .
            SHIFT wa-docno LEFT DELETING LEADING '0'.
            CONCATENATE wa-docno TEXT-001 INTO lv_message .
            PERFORM fill_return USING 'E' lv_message.
          ENDIF.

          SELECT SINGLE tu_num, tu_sr_act_num
            FROM /scwm/tu_sr_act
            WHERE tu_num = @lv_tu_num AND end_actual IS INITIAL
            INTO @ls_tu_sr_act.

          IF sy-subrc NE 0.
```

```
  CLEAR lv_message .
  SHIFT wa-docno LEFT DELETING LEADING '0'.
  CONCATENATE wa-docno TEXT-003 INTO lv_message .
  PERFORM fill_return USING 'E' lv_message.
  CONTINUE.
ENDIF.

SELECT status_type, status_value FROM /scwm/tu_status INTO TABLE @lt_status
  WHERE tu_num EQ @lv_tu_num
  AND tu_sr_act_num EQ @ls_tu_sr_act-tu_sr_act_num
  AND status_type BETWEEN 'ISR18' AND 'ISR20'.

IF sy-subrc EQ 0.
  READ TABLE lt_status INTO ls_status19 WITH KEY status_type = 'ISR19'.
  IF ls_status19-status_value IS NOT INITIAL.
    READ TABLE lt_status INTO ls_status20 WITH KEY status_type = 'ISR20'.
    IF ls_status20-status_value IS NOT INITIAL.
      CLEAR lv_message .
      SHIFT wa-docno LEFT DELETING LEADING '0'.
      CONCATENATE wa-docno TEXT-003 INTO lv_message .
      PERFORM fill_return USING 'E' lv_message.
    ENDIF.
  ELSE.           "TU is not active
    CLEAR lv_message .
    SHIFT wa-docno LEFT DELETING LEADING '0'.
    CONCATENATE wa-docno TEXT-004 INTO lv_message .
    PERFORM fill_return USING 'E' lv_message.
  ENDIF.
ENDIF.

SELECT * FROM /scwm/tu_status INTO TABLE @DATA(lt_gi_status) WHERE tu_num EQ @lv_tu_r
  AND tu_sr_act_num EQ @ls_tu_sr_act-tu_sr_act_num
  AND status_type EQ 'ISR18'.

LOOP AT lt_gi_status INTO DATA(wa_gi).
  wa_gi-status_value = ''.
  MODIFY lt_gi_status FROM wa_gi INDEX sy-tabix.
  MODIFY /scwm/tu_status FROM TABLE lt_gi_status.
ENDLOOP.

" Get TU Details
o_bom = /scwm/cl_sr_bom=>get_instance( io_log = NEW /scwm/cl_log( ) ).
o_bo_tu = o_bom->get_bo_tu_by_key( is_tu_sr_act_num = ls_tu_sr_act ).
o_bo_tu->get_data( IMPORTING es_bo_tu_data = ls_bo_tu_data ).

" Fill the Delivery Details
ls_asp_tu_whr = CORRESPONDING #( ls_bo_tu_data ).
" Fill Timestamp
/scwm/cl_sr_my_service=>fill_timestamp(
  EXPORTING iv_tzone = 'INDIA'
  CHANGING cs_asp_tu_whr = ls_asp_tu_whr ).

SELECT SINGLE docid FROM /scdl/db_proch_o INTO @DATA(lv_docid)
  WHERE docno EQ @copy_wa-docno.

ls_bo_tu_dlv        = CORRESPONDING #( ls_tu_sr_act ).     " TU Details
ls_bo_tu_dlv-seq_num  = lv_seq_no.                         " Sequence Number
ls_bo_tu_dlv-lgnum  = lc_lgnum.                            " Warehouse
ls_bo_tu_dlv-docid  = lv_docid.
ls_bo_tu_dlv-docno  = wa-docno.                            "|{ lwa_docno-low ALPHA =
ls_bo_tu_dlv-doccat = wmegc_doccat_pdo.                    " Document Category
```

```abap
              IF lt_bo_tu_dlv_unassign IS NOT INITIAL.
                IF line_exists( lt_bo_tu_dlv_unassign[ tu_num = ls_bo_tu_dlv-tu_num ] ).
                  "do nothing
                ELSE.
                  CLEAR lv_message .
                  CONCATENATE '' TEXT-012 INTO lv_message .
                  PERFORM fill_return USING 'E' lv_message.
                  CONTINUE.
                ENDIF.
              ENDIF.

              APPEND ls_bo_tu_dlv TO lt_bo_tu_dlv_unassign.
              CLEAR ls_bo_tu_dlv.
            CATCH /scwm/cx_sr_error .
*              RETURN.
          ENDTRY.

        ENDIF.
************************************************************************************************
        CLEAR: lt_bo_tu_dlv, lt_bo_tu_dlv_succ.
        CLEAR: lt_gi_status, wa_gi, lv_seq_no, lv_docno.
      ENDLOOP.
*******************************************************************
      IF lt_bo_tu_dlv_unassign IS NOT INITIAL.
        TRY.
            CALL METHOD o_bo_tu->del_tu_dlv
              EXPORTING
                iv_check_only                = abap_false
                it_bo_tu_dlv                 = lt_bo_tu_dlv_unassign
                iv_skip_synch_check          = abap_false
                iv_skip_synch_complete       = abap_false
                iv_cross_hu_del              = abap_false
              IMPORTING
                et_bo_tu_dlv_success         = lt_bo_tu_dlv_succ
                et_bo_tu_dlv_no_sort_success = DATA(lt_bo_tu_dlv_no_sort_success).
          CATCH /scwm/cx_sr_error.
        ENDTRY.

        IF lt_bo_tu_dlv_succ IS NOT INITIAL.
          LOOP AT lt_bo_tu_dlv_succ INTO DATA(lwa).
            CLEAR lv_message .
            SHIFT lwa-docno LEFT DELETING LEADING '0'.
            CONCATENATE lwa-docno TEXT-007 INTO lv_message .
            PERFORM fill_return USING 'S' lv_message.
          ENDLOOP.
        ELSE.
          CLEAR lv_message .
          CONCATENATE '' TEXT-013 INTO lv_message .
          PERFORM fill_return USING 'E' lv_message.
        ENDIF.

        o_bom->save( ).
        COMMIT WORK AND WAIT.
      ENDIF.

*******************************************************************
      CLEAR: lt_bo_tu_dlv, lt_bo_tu_dlv_succ, lt_bo_tu_dlv_no_sort_success, lt_bo_tu_dlv_unassign.
      IF gt_return IS NOT INITIAL.
        PERFORM display_log.
      ENDIF.
      CLEAR: lv_message, gt_return.
      LEAVE TO SCREEN 0.
  ENDCASE.

ENDMODULE.

*&SPWIZARD: INPUT MODULE FOR TC 'TC1'. DO NOT CHANGE THIS LINE!
*&SPWIZARD: MODIFY TABLE
MODULE tc1_modify INPUT.
  MODIFY it_dlv_tu
    FROM wa
    INDEX tc1-current_line.
ENDMODULE.
```