```
*----------------------------------------------------------------------*
***INCLUDE LZFG_AF_LOADING_SHEETI01.
*----------------------------------------------------------------------*
*&---------------------------------------------------------------------*
*&      Module  USER_COMMAND_9001  INPUT
*&---------------------------------------------------------------------*
*       text
*----------------------------------------------------------------------*
MODULE user_command_9001 INPUT.

  CASE sy-ucomm.
    WHEN 'CANCEL'.
      LEAVE TO SCREEN 0.
    WHEN 'ENTER'.
      DATA: lv_function_name TYPE funcname,
            es_form_output   TYPE fpformoutput,
            es_job_output    TYPE sfpjoboutput,
            is_doc_params    TYPE sfpdocparams,
            ls_header        TYPE zstruc_header_loading_sheet,
            lt_item          TYPE ztt_item_loading_sheet,
            ls_item          TYPE zstruc_item_loading_sheet,
            lv_matnr         TYPE /scwm/de_matnr,
            lv_maktx         TYPE /scwm/de_ui_maktx,
            lv_batchno       TYPE /scwm/de_charg,
            lv_message       TYPE bapi_msg,
            lv_total_ea      TYPE i,
            lv_total_cv      TYPE i,
            lv_frac          TYPE kbetr.

      DATA(lo_conv) = NEW /scwm/cl_ui_stock_fields( ).

      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'.

      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( ).


      IF  io_printer1 IS INITIAL.
        MESSAGE i041(zewm) DISPLAY LIKE 'E'.
        RETURN.
      ENDIF.

      TRY.
          lo_dlv->query(
           EXPORTING
             it_docid         = copy_lt_docid
             iv_whno          = 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 ibupacustomer INTO @DATA(lv_customer) WHERE businesspartner EQ
        SELECT SINGLE customer FROM i_businesspartnercustomer INTO @DATA(lv_customer) WHERE busines
        IF sy-subrc EQ 0.
          SELECT SINGLE kunnr, name1, ort01 FROM kna1 INTO @DATA(ls_kna1) WHERE kunnr EQ @lv_custom
        ENDIF.

        READ TABLE lwa_headers-status INTO DATA(ls_gi_status) WITH KEY status_type = lv_gi_status_t
        IF ls_gi_status-status_value NE '9'.
          CLEAR lv_message .
          SHIFT lwa_headers-docno LEFT DELETING LEADING '0'.
*          CONCATENATE lwa_headers-docno ':' ' ' 'Goods Issue Not Started' INTO lv_message .
          CONCATENATE lwa_headers-docno TEXT-001 INTO lv_message .
          PERFORM fill_return USING 'E' lv_message.
          CONTINUE.
        ELSE.

          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
          READ TABLE lwa_items-refdoc INTO DATA(ls_refdoc_so) WITH KEY refdoccat = lc_refdoccat_so
          READ TABLE lwa_items-refdoc INTO DATA(ls_refdoc_po) WITH KEY refdoccat = lc_refdoccat_po

          ls_header-datum = sy-datum.
          ls_header-uzeit = sy-uzeit.
          ls_header-uname = sy-uname.
          ls_header-docno = lwa_headers-docno.
          ls_header-refdocno = ls_refdoc_erpdel-refdocno.
*          ls_header-refdocno_so = ls_refdoc_so-refdocno.
          IF ls_refdoc_po IS NOT INITIAL.
            ls_header-refdocno_so = ls_refdoc_po-refdocno.
          ELSE.
            ls_header-refdocno_so = ls_refdoc_so-refdocno.
          ENDIF.
          ls_header-kunnr = ls_kna1-kunnr.
          ls_header-name1 = ls_kna1-name1.
          ls_header-ort01 = ls_kna1-ort01.

          LOOP AT lt_hu_items INTO DATA(ls_hu_items) WHERE qdocid EQ lwa_headers-docid.

            CALL METHOD lo_conv->get_matkey_by_id
              EXPORTING
                iv_matid = ls_hu_items-matid
              IMPORTING
                ev_matnr = lv_matnr
                ev_maktx = lv_maktx.

            CALL METHOD lo_conv->get_batchno_by_id
              EXPORTING
                iv_batchid = ls_hu_items-batchid
              RECEIVING
                ev_charg   = lv_batchno.

            READ TABLE lt_hu_headers INTO DATA(ls_headers)
            WITH KEY guid_hu = ls_hu_items-guid_parent BINARY SEARCH.

            ls_item-huident = ls_headers-huident.
            ls_item-productno = lv_matnr.
            ls_item-product_text = lv_maktx.
            ls_item-batchno = lv_batchno.
            "if alternative quantity is in decimals, we convert CV into EA.
            lv_frac = frac( ls_hu_items-quana ).
            IF lv_frac IS NOT INITIAL.
              ls_item-qty = ls_hu_items-quan.
              ls_item-uom = ls_hu_items-meins.
            ELSE.
              ls_item-qty = ls_hu_items-quana.
              ls_item-uom = ls_hu_items-altme.
            ENDIF.
            "summing up EA and CV separately for summary table at the end. Remember in coding the u
            "as CV in the form
            IF ls_item-uom EQ 'CS'.
              lv_total_cv = lv_total_cv + ls_item-qty.
            ELSE.
              lv_total_ea = lv_total_ea + ls_item-qty.
            ENDIF.
            APPEND ls_item TO lt_item.
          ENDLOOP.

          IF lt_item IS INITIAL.
            CLEAR lv_message.
            SHIFT lwa_headers-docno LEFT DELETING LEADING '0'.
            CONCATENATE lwa_headers-docno TEXT-004 INTO lv_message .
            PERFORM fill_return USING 'E' lv_message.
            CONTINUE.
          ENDIF.

*    *--No print preview needed
          DATA : xs_output_params TYPE sfpoutputparams.
          xs_output_params-nodialog = abap_true.
          xs_output_params-device = lc_printer.
          xs_output_params-dest = io_printer1.
          xs_output_params-reqimm = abap_true.
          xs_output_params-reqnew = abap_true.

          TRY.
              CALL FUNCTION 'FP_FUNCTION_MODULE_NAME'
                EXPORTING
                  i_name     = lv_form_name
                IMPORTING
                  e_funcname = lv_function_name.

            CATCH cx_fp_api_internal.
            CATCH cx_fp_api_repository.
            CATCH cx_fp_api_usage.
          ENDTRY.

          IF lv_function_name IS INITIAL.
            MESSAGE e012(zewm).
          ENDIF.

          CALL FUNCTION 'FP_JOB_OPEN'
            CHANGING
              ie_outputparams = xs_output_params
            EXCEPTIONS
```

```
              OTHERS          = 0.


*          CALL FUNCTION lv_function_name                         "'/1BCDWB/SM00000013'
*            EXPORTING
*              /1bcdwb/docparams  = is_doc_params
*              gs_header          = ls_header
*              gt_item            = lt_item
*            IMPORTING
*              /1bcdwb/formoutput = es_form_output
*            EXCEPTIONS
*              OTHERS             = 0.

          CALL FUNCTION lv_function_name                         "'/1BCDWB/SM00000013'
            EXPORTING
              /1bcdwb/docparams  = is_doc_params
              total_ea           = lv_total_ea
              total_cv           = lv_total_cv
              gs_header          = ls_header
              gt_item            = lt_item
            IMPORTING
              /1bcdwb/formoutput = es_form_output
            EXCEPTIONS
              OTHERS             = 0.

          IF es_form_output IS INITIAL.
            CLEAR lv_message.
            SHIFT lwa_headers-docno LEFT DELETING LEADING '0'.
*            CONCATENATE lwa_headers-docno ':' ' ' 'Not printed successfully' INTO lv_message .
            CONCATENATE lwa_headers-docno TEXT-002 INTO lv_message .
            PERFORM fill_return USING 'E' lv_message.
            EXIT.
          ENDIF.

          IF es_form_output IS NOT INITIAL .
            CLEAR lv_message .
            SHIFT lwa_headers-docno LEFT DELETING LEADING '0'.
*            CONCATENATE lwa_headers-docno ':' ' ' 'Printed Successfully' INTO lv_message .
            CONCATENATE lwa_headers-docno TEXT-003 INTO lv_message .
            PERFORM fill_return USING 'S' lv_message.
          ENDIF .

          IF sy-subrc <> 0.
            MESSAGE ID sy-msgid
                  TYPE sy-msgty
                NUMBER sy-msgno
                  WITH sy-msgv1 sy-msgv2 sy-msgv3 sy-msgv4.
          ENDIF.

          CALL FUNCTION 'FP_JOB_CLOSE'
            IMPORTING
              e_result = es_job_output
            EXCEPTIONS
              OTHERS   = 0.

          IF sy-subrc EQ 0.
          ENDIF .

          IF sy-subrc <> 0.
            MESSAGE ID sy-msgid
                  TYPE sy-msgty
                NUMBER sy-msgno
                  WITH sy-msgv1 sy-msgv2 sy-msgv3 sy-msgv4.

          ENDIF.

          CLEAR: ls_item, lt_item, ls_header, lv_customer, ls_kna1.
        ENDIF.
      ENDLOOP.

      PERFORM display_log.
      LEAVE TO SCREEN 0.
  ENDCASE.

ENDMODULE.
```