

```

*-----*
***INCLUDE LZFG_AF_PICK_LISTI01.
*-----*
*&-----*
*&      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 sfppjoboutput,
      is_doc_params    TYPE sfpdocparams,
      ls_header        TYPE zstruc_header_picklist,
      lt_item          TYPE ztt_item_picklist,
      ls_item          TYPE zstruc_item_picklist,
      lv_message       TYPE bapi_msg.

```

```

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_PICK_LIST',
            lv_lgnum         TYPE /scwm/lgnum VALUE '6002',
            lc_party_role    TYPE /scdl/dl_party_role VALUE 'STPRT'.

```

```

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 ).

```

```
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_who        = lv_lgnum
```

```
      is_include_data = la_include
```

```
    IMPORTING
```

```
      et_headers    = DATA(lt_headers)
```

```
      et_items      = DATA(lt_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 = lc_party_role
```

```
  SELECT SINGLE customer FROM i_businesspartnercustomer INTO @DATA(lv_customer) WHERE business
```

```
  IF sy-subrc EQ 0.
```

```

SELECT SINGLE kunnr, name1, ort01 FROM knal INTO @DATA(ls_knal) WHERE kunnr EQ @lv_custom
ENDIF.

```

```

IF lt_items IS NOT INITIAL.

```

```

    LOOP AT lt_items INTO DATA(lwa_items) WHERE docno EQ lwa_headers-docno.

```

```

        IF lwa_items-qty-qty IS INITIAL.

```

```

            CONTINUE.

```

```

        ENDIF.

```

```

        ls_item-itemno = lwa_items-itemno.

```

```

        ls_item-productno = lwa_items-product-productno.

```

```

        ls_item-product_text = lwa_items-product-product_text.

```

```

        ls_item-batchno = lwa_items-product-batchno.

```

```

        ls_item-qty = lwa_items-qty-qty.

```

```

        ls_item-uom = lwa_items-qty-uom.

```

```

        APPEND ls_item TO lt_item.

```

```

    ENDLLOOP.

```

```

IF lt_item IS INITIAL.

```

```

    CLEAR lv_message .

```

```

    SHIFT lwa_headers-docno LEFT DELETING LEADING '0'.

```

```

    CONCATENATE lwa_headers-docno TEXT-002 INTO lv_message .

```

```

    PERFORM fill_return USING 'E' lv_message.

```

```

    CONTINUE.

```

```

ENDIF.

```

```

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_items-docno.

```

```

ls_header-refdocno = ls_refdoc_erpdel-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_knal-kunnr.

```

```

ls_header-name1 = ls_knal-name1.

```

```

ls_header-ort01 = ls_knal-ort01.

```

```

*
```

```

*--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/SM00000008'  
    EXPORTING  
        /1bcdwb/docparams = is_doc_params  
        gt_item           = lt_item  
        gs_header         = ls_header  
    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 TEXT-003 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 TEXT-004 INTO lv_message .  
    PERFORM fill_return USING 'S' lv_message.  
ENDIF .
```

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

```
ELSE.  
    CLEAR lv_message .  
    SHIFT lwa_headers-docno LEFT DELETING LEADING '0'.  
    CONCATENATE lwa_headers-docno TEXT-005 INTO lv_message .  
    PERFORM fill_return USING 'S' lv_message.  
ENDIF.
```

```
CLEAR: ls_item, lt_item, ls_header.  
ENDLOOP.  
PERFORM display_log.  
LEAVE TO SCREEN 0.
```

```
ENDCASE.
```

ENDMODULE .