

```

*&-----*
*& Report ZCONSUMPTION_POSTING_REPORT
*&-----*
*&
*&-----*
*****
*                               Asian Paints Khandala Phase 2                               *
*****
*  OBJECT INFORMATION                                                    *
*****
*  Module                      : EWM                                         *
*  Developer                   : Sai Vandana                                *
*  Requestor                   : Shubham Shete                               *
*  Date Of Creation            : 06.11.2023                                  *
*  Transport Request           : KSDK901173                                  *
*  Development Object          : ZCONSUMPTION_POSTING_REPORT                *
*  Description                  : Consumption Posting Report                 *
*
*=====
*  MODIFICATION HISTORY
*-----
*  Mod. No.   Date      Name                Transport Number      Change Desc.
*-----
*
*-----

```

REPORT zconsumption_posting_report LINE-SIZE 250.

***Table

TABLES :

zposting_log,/scdl/db_refdoc,/scdl/db_proci_p,zpsa_return,zcross_log.

***Types

TYPES : BEGIN OF ty_result,

```

    pmr                TYPE /scdl/dl_docno,
    docid              TYPE /scdl/dl_docid,
    itemid             TYPE /scdl/dl_itemid,
    itemno             TYPE /scdl/dl_itemno,
    po                 TYPE /scwm/de_prod_order,
    po_item            TYPE /scdl/dl_productno,
    po_qty             TYPE p DECIMALS 2,
    psa_qty            TYPE p DECIMALS 2,
    return_qty         TYPE p DECIMALS 2, "/scdl/dl_quantity,
    261_pass           TYPE p DECIMALS 2, "/scdl/dl_quantity,
    qty_bal_psa        TYPE p DECIMALS 2, "/scdl/dl_quantity,
    gr_post            TYPE p DECIMALS 2, "/scdl/dl_quantity,
    retry_261          TYPE p DECIMALS 2, "/scdl/dl_quantity,
    po_transf          TYPE p DECIMALS 2,
    scrap_posted       TYPE p DECIMALS 2,
    scrap_req          TYPE p DECIMALS 2, "/scdl/dl_quantity,
    potransqty_frompo  TYPE p DECIMALS 2,
    potransqty_topo    TYPE p DECIMALS 2,
    manual_gr          TYPE p DECIMALS 2,
    auto_gr            TYPE p DECIMALS 2,
    cross_post         TYPE p DECIMALS 2, "Added by Shubham on 07-Feb-2023

```

END OF ty_result.

TYPES : BEGIN OF ty_fg_gr,

```

    process_order TYPE /scdl/dl_refdocno,
END OF ty_fg_gr.

```

***Internal Table

DATA :

```
lt_refdoc      TYPE TABLE OF /scdl/db_refdoc,
lt_psa         TYPE TABLE OF /scdl/db_proci_p,
lt_po          TYPE TABLE OF zposting_log,
lt_fg_gr       TYPE TABLE OF ty_fg_gr, "zfg_gr_log,
lt_return      TYPE TABLE OF zpsa_return,
lt_result      TYPE TABLE OF ty_result,
lt_marm        TYPE TABLE OF /sapapo/marm,
lt_stock       TYPE /scwm/tt_stock_mon,
lt_posting     TYPE TABLE OF zpo_post,
lt_po_trans_src TYPE TABLE OF zpo_transfer,
lt_po_trans_dest TYPE TABLE OF zpo_transfer,
lt_auto_gr     TYPE TABLE OF zfg_gr_log,
lt_gr_rev      TYPE TABLE OF zfg_gr_log,
lt_manual_gr   TYPE TABLE OF zfg_gr_log,
lt_cross_post  TYPE TABLE OF zcross_log.    "Added by Shubham on 07-Feb-2023
```

DATA :

```
lr_po          TYPE rseloption,
lt_selection_pdi TYPE /scwm/dlv_selection_tab,
lt_head_pdi     TYPE /scwm/dlv_header_out_prd_tab,
lt_item_pdi     TYPE /scwm/dlv_item_out_prd_tab.
```

DATA : lt_lgtyp_r TYPE /scwm/tt_lgtyp_r.

DATA :

```
ls_read_options TYPE /scwm/dlv_query_contr_str,
ls_include_data TYPE /scwm/dlv_query_incl_str_prd.
```

***Internal Structure

DATA :

```
ls_refdoc      TYPE /scdl/db_refdoc,
ls_psa         TYPE /scdl/db_proci_p,
ls_po          TYPE zposting_log,
ls_fg_gr       TYPE ty_fg_gr, "zfg_gr_log,
ls_return      TYPE zpsa_return,
ls_result      TYPE ty_result,
ls_marm        TYPE /sapapo/marm,
ls_cross_post  TYPE zcross_log.    "Added by Shubham on 07-Feb-2023
```

***Local Variable

DATA :

```
psa_c          TYPE /scwm/dl_psa,
lv_261         TYPE /scdl/dl_qty,
lv_qty_psa     TYPE /scdl/dl_qty,
lv_scrap_qty   TYPE /scdl/dl_qty,
lv_matnr       TYPE /sapapo/matnr.
```

*** INITIAL CHECK ENDED *****

```
DATA cl_po TYPE REF TO /scwm/cl_dlv_management_prd.
DATA lo_stock TYPE REF TO /scwm/cl_mon_stock.
DATA:wa_inc_data TYPE /scwm/dlv_query_incl_str_prd,
      is_read_opt TYPE /scwm/dlv_query_contr_str,
      it_select   TYPE /scwm/dlv_selection_tab,
      wa_select   LIKE LINE OF it_select.
```

```

DATA:it_head TYPE /scwm/dlv_header_out_prd_tab,
      wa_head LIKE LINE OF it_head,
      it_item TYPE /scwm/dlv_item_out_prd_tab,
      wa_item LIKE LINE OF it_item.
DATA : lt_docflow TYPE TABLE OF /scwm/dl_docflow_str,
      ls_docflow TYPE /scwm/dl_docflow_str.

DATA : lv_matidl6 TYPE /scwm/de_matid,
      ls_mat_global TYPE /scwm/s_material_global,
      lt_mat_uom TYPE /scwm/tt_material_uom.

CONSTANTS : lv_lgnum TYPE /scwm/lgnum VALUE 'KHPM',
            lc_fcode_back TYPE gui_code VALUE 'BACK',
            lc_fcode_exit TYPE gui_code VALUE 'EXIT'.
*****

***PARAMETERS

SELECTION-SCREEN BEGIN OF BLOCK b1 WITH FRAME.
  PARAMETERS : po TYPE /scwm/de_prod_order .
  PARAMETERS : psa TYPE /scwm/dl_psa .
  SELECT-OPTIONS : s_date FOR sy-datum NO-EXTENSION.
SELECTION-SCREEN END OF BLOCK b1.
*****

AT USER-COMMAND.
  CASE sy-ucomm.
  * WHEN .
  * WHEN .
  WHEN OTHERS.
  ENDCASE.

***START-OF-SELECTION.

START-OF-SELECTION.

  PERFORM refresh_variable.

  PERFORM data_validation.

  PERFORM fetch_data.

  PERFORM calc_data.

  PERFORM display_data.
*&-----*
*& Form refresh_variable
*&-----*
* text
*-----*
* --> p1 text
* <-- p2 text
*-----*
FORM refresh_variable .

  REFRESH : lt_po[],lt_psa[],lt_refdoc[],lr_po[],lt_selection_pdi[].
  REFRESH : lt_cross_post[].

ENDFORM. "refresh_variable
*&-----*

```

```

*&          Form  DATA_VALIDATION
*&-----*
*          text
*-----*
*  -->  p1          text
*  <--  p2          text
*-----*

FORM data_validation .

IF po IS INITIAL.
  IF psa IS INITIAL.
    IF s_date-low IS INITIAL.
      MESSAGE s445(zapl_asrs) WITH TEXT-001 DISPLAY LIKE 'E'.
      STOP.
    ENDIF.
  ENDIF.
ENDIF.

IF s_date-low = '*' .
  MESSAGE s445(zapl_asrs) WITH TEXT-002 DISPLAY LIKE 'E'.
  STOP.
ENDIF.

IF s_date-high IS NOT INITIAL.
  DATA(lv_diff) = s_date-high - s_date-low .

  IF lv_diff > 31.
    MESSAGE s445(zapl_asrs) WITH TEXT-003 DISPLAY LIKE 'E'.
    STOP.
  ENDIF.
ENDIF.
REFRESH : lt_refdoc[].
IF po IS NOT INITIAL.

  SELECT DISTINCT process_order FROM zposting_log
    INTO CORRESPONDING FIELDS OF TABLE lt_po
    WHERE process_order = po.
  IF sy-subrc = 0 .
    SELECT DISTINCT docid itemid refdocno FROM /scdl/db_refdoc
      INTO CORRESPONDING FIELDS OF TABLE lt_refdoc
      WHERE
        refdoccat = /scdl/if_dl_doc_c=>SC_DOCCAT_PPO AND
        doccat    = wmegc_doccat_pwr AND
        refdocno = po.
    IF sy-subrc = 0 .
      IF lt_refdoc[] IS INITIAL.
        MESSAGE s445(zapl_asrs) WITH TEXT-004 DISPLAY LIKE 'E'.
        STOP.
      ENDIF.
    ENDIF.
  ENDIF .
ENDIF .
IF lt_refdoc[] IS INITIAL.

  IF psa IS NOT INITIAL.
    CONCATENATE 'PS' psa+5(2) '%' INTO psa_c .
    SELECT DISTINCT a~docid refdocno FROM /scdl/db_refdoc AS a
      INNER JOIN /scdl/db_proci_p AS b
      ON a~docid = b~docid
      INTO CORRESPONDING FIELDS OF TABLE lt_refdoc

```

```

WHERE
a~refdoccat = /scdl/if_dl_doc_c=>SC_DOCCAT_PPO AND
b~/scwm/psa LIKE psa_c.
IF lt_refdoc[] IS INITIAL.
    MESSAGE s445(zapl_asrs) WITH TEXT-005 DISPLAY LIKE 'E'.
    STOP.
ENDIF.
ENDIF.

ENDIF.
IF lt_refdoc[] IS INITIAL.

    IF s_date-low IS NOT INITIAL.

        SELECT DISTINCT process_order FROM zposting_log
        INTO CORRESPONDING FIELDS OF TABLE lt_po
        WHERE log_date IN s_date.
        IF sy-subrc = 0 .
            SELECT DISTINCT process_order FROM zfg_gr_log
            INTO CORRESPONDING FIELDS OF TABLE lt_fg_gr
            WHERE
            log_date IN s_date.
            IF sy-subrc = 0 .
                SORT lt_fg_gr BY process_order.

                DELETE ADJACENT DUPLICATES FROM lt_fg_gr COMPARING process_order.
                IF lt_fg_gr[] IS NOT INITIAL.
                    SELECT DISTINCT docid itemid refdocno FROM /scdl/db_refdoc
                    INTO CORRESPONDING FIELDS OF TABLE lt_refdoc
                    FOR ALL ENTRIES IN lt_fg_gr
                    WHERE
                    refdoccat = /scdl/if_dl_doc_c=>SC_DOCCAT_PPO AND
                    doccat = wnegc_doccat_pwr AND "Added by Antony Richard A on 02-04-2019
                    refdocno = lt_fg_gr-process_order.
                    IF lt_refdoc[] IS INITIAL.
                        MESSAGE s445(zapl_asrs) WITH TEXT-006 DISPLAY LIKE 'E'.
                        STOP.
                    ENDIF.

                ENDIF.
            ENDIF.
        ENDIF .
    ENDIF .
ENDIF.
ENDFORM."DATA_VALIDATION

*&-----*
*&      Form  fetch_data
*&-----*
*      text
*-----*
*  -->  p1      text
*  <--  p2      text
*-----*
FORM fetch_data .

** PMR document read started ***
is_read_opt-data_retrival_only = abap_false.
wa_inc_data-item_docflow       = abap_true.

```

```

wa_inc_data-item_product_ext    = abap_true.
REFRESH : it_select[].
IF lt_refdoc[] IS NOT INITIAL.
    REFRESH : lt_po[],lr_po[].
    REFRESH : lt_cross_post[].
    LOOP AT lt_refdoc INTO ls_refdoc.

        wa_select-fieldname = 'REFDOCNO_PPO_H'.
        wa_select-sign = wnegc_sign_inclusive.
        wa_select-option = wnegc_option_eq.
        wa_select-low = ls_refdoc-refdocno.
        APPEND wa_select TO it_select.
        ls_po-process_order = ls_refdoc-refdocno.
        APPEND ls_po TO lt_po.
        INSERT VALUE rsdsselopt( sign = wnegc_sign_inclusive option = wnegc_option_eq low = ls_refdocno )
        CLEAR : ls_refdoc,wa_select.
    ENDLOOP.

ENDIF.
IF it_select[] IS INITIAL.

    IF lt_po[] IS NOT INITIAL.
        LOOP AT lt_po INTO ls_po.

            wa_select-fieldname = 'REFDOCNO_PPO_H'.
            wa_select-sign = wnegc_sign_inclusive.
            wa_select-option = wnegc_option_eq.
            wa_select-low = ls_po-process_order.
            APPEND wa_select TO it_select.
            CLEAR : ls_po.
*            INSERT VALUE rsdsselopt( sign = wnegc_sign_inclusive option = wnegc_option_eq low = ls_po-process_order )
            ENDLOOP.
        ENDIF.

    ENDIF.

TRY.
    /scwm/cl_tm=>cleanup( ).
    CREATE OBJECT cl_po.
    cl_po->query(
        EXPORTING
            it_selection      = it_select
            iv_doccat         = /scdl/if_dl_doc_c=>SC_DOCCAT_PWR_PRD
            is_read_options   = is_read_opt
            is_include_data   = wa_inc_data
        IMPORTING
            et_headers        = it_head
            et_items          = it_item ).
    CATCH /scdl/cx_delivery.
ENDTRY.

*****

IF lr_po[] IS NOT INITIAL.
    SELECT *
    FROM zpsa_return
    WHERE
        lgnum EQ @lv_lgnum AND
        po IN @lr_po[]
    INTO CORRESPONDING FIELDS OF TABLE @lt_return.

```



```

message EQ 'REVERSAL'
GROUP BY process_order
INTO CORRESPONDING FIELDS OF TABLE @lt_gr_rev[].
IF sy-subrc = 0.
*To get all the cross posting qty
SELECT * FROM zcross_log           "#EC CI_ALL_FIELDS_NEEDED
WHERE
order_num      IN @lr_po
INTO TABLE @lt_cross_post.

ENDIF.
ENDIF.
ENDIF.
ENDIF.
ENDIF.
ENDIF.
ENDFORM.           "fetch_data
*&-----*
*&      Form  calc_data
*&-----*
*
*      text
*-----*
*  -->  p1      text
*  <--  p2      text
*-----*
FORM calc_data .
SORT it_item BY docid.
SORT lt_refdoc BY docid.

LOOP AT it_item INTO wa_item.
REFRESH : lt_docflow[].
MOVE-CORRESPONDING wa_item-docflow TO lt_docflow.

READ TABLE lt_refdoc INTO ls_refdoc
WITH KEY docid = wa_item-docid .
IF sy-subrc = 0.
CLEAR : lv_qty_psa,lv_261,ls_docflow.
LOOP AT lt_docflow INTO ls_docflow WHERE docidfr = wa_item-docid
AND itemidfr = wa_item-itemid AND doccatto = /scdl/if_dl_doc_c=>SC_DOCCAT_STA.
lv_qty_psa = ls_docflow-rel_qty + lv_qty_psa.
CLEAR :ls_docflow.
ENDLOOP.
CLEAR : lv_261,ls_docflow.
LOOP AT lt_docflow INTO ls_docflow WHERE docidfr = wa_item-docid
AND itemidfr = wa_item-itemid AND doccatto = /scdl/if_dl_doc_c=>SC_DOCCAT_GI.
lv_261 = ls_docflow-rel_qty + lv_261.
CLEAR :ls_docflow.
ENDLOOP.
LOOP AT lt_docflow INTO ls_docflow WHERE docidfr = wa_item-docid
AND itemidfr = wa_item-itemid AND doccatto = /scdl/if_dl_doc_c=>SC_DOCCAT_GIR.
lv_261 = ls_docflow-rel_qty + lv_261.
*      lv_261 = lv_261 - ls_docflow-rel_qty.
CLEAR :ls_docflow.
ENDLOOP.

CLEAR :ls_result,lv_scrap_qty.
ls_result-docid = wa_item-docid.
ls_result-itemid = wa_item-itemid.
ls_result-pmr = wa_item-docno.
ls_result-itemno = wa_item-itemno.

```



```

ls_result-po          = ls_refdoc-refdocno.
*   CONDENSE lv_261.
ls_result-261_pass    = lv_261.
*   CONDENSE lv_qty_psa.
ls_result-psa_qty     = lv_qty_psa.
*   CONDENSE wa_item-qty-qty.
ls_result-po_qty      = wa_item-qty-qty.
ls_result-po_item     = wa_item-product-productno.
CLEAR : ls_result-return_qty.
LOOP AT lt_return INTO ls_return WHERE
po = ls_result-po AND psa_bin = wa_item-sapext-/scwm/gmbin .

*   CONDENSE ls_return-qty.
ls_result-return_qty = ls_return-qty + ls_result-return_qty.
CLEAR : ls_return-qty .
ENDLOOP.

LOOP AT lt_stock INTO DATA(ls_stock) WHERE qdocid = wa_item-docid AND gitmid = wa_item-itemid

IF ls_stock IS NOT INITIAL.
*   ls_result-qty_bal_psa = ( ls_result-psa_qty - ( lv_261 + ls_result-return_qty ) ).
ls_result-qty_bal_psa = ls_result-qty_bal_psa + ls_stock-quan.
ENDIF.
CLEAR ls_stock.
ENDLOOP.
*   CONDENSE ls_result-qty_bal_psa.
ENDIF.
LOOP AT lt_posting INTO DATA(ls_posting) WHERE po          EQ ls_refdoc-refdocno AND
matnr              EQ wa_item-product-productno AND
posting_type EQ 'SCRAP POST'.

lv_scrap_qty = lv_scrap_qty + ls_posting-quantity.
ENDLOOP.
ls_result-scrap_posted = lv_scrap_qty.
LOOP AT lt_cross_post INTO DATA(ls_cross_post) WHERE order_num EQ ls_refdoc-refdocno
AND product EQ wa_item-product-productno.
ls_result-cross_post = ls_result-cross_post + ls_cross_post-qty.
ENDLOOP.

LOOP AT lt_auto_gr INTO DATA(ls_auto_gr) WHERE process_order EQ ls_refdoc-refdocno.
ls_result-auto_gr = ls_result-auto_gr + ls_auto_gr-quan.
ENDLOOP.

LOOP AT lt_manual_gr INTO DATA(ls_manual_gr) WHERE process_order EQ ls_refdoc-refdocno.
ls_result-manual_gr = ls_result-manual_gr + ls_manual_gr-quan.
ENDLOOP.

ls_result-gr_post = ls_result-auto_gr + ls_result-manual_gr.

LOOP AT lt_gr_rev INTO DATA(ls_gr_rev) WHERE process_order EQ ls_refdoc-refdocno.
ls_result-gr_post = ls_result-gr_post - ls_gr_rev-quan.
ENDLOOP.

IF ls_auto_gr IS NOT INITIAL.
lv_matnr = ls_auto_gr-matnr.
ELSEIF ls_manual_gr IS NOT INITIAL.
lv_matnr = ls_manual_gr-matnr.
ENDIF.

IF lv_matnr IS NOT INITIAL.

CALL FUNCTION 'CONVERSION_EXIT_PRODU_INPUT'

```

```

EXPORTING
  input  = lv_matnr
IMPORTING
  output = lv_matnr.
CALL FUNCTION 'CONVERSION_EXIT_MDLPD_INPUT'
EXPORTING
  input  = lv_matnr
IMPORTING
  output = lv_matid16.
TRY .
  CALL FUNCTION '/SCWM/MATERIAL_READ_SINGLE'
    EXPORTING
      iv_matid      = lv_matid16
      IV_LANGU      = SY-LANGU
      iv_entitled   = wa_item-sapext-entitled
      IV_APPLIC     =
      iv_lgnum      = lv_lgnum
    IMPORTING
      es_mat_global = ls_mat_global
      et_mat_uom    = lt_mat_uom.
  CATCH /scwm/cx_md INTO DATA(lc_msg) .
ENDTRY .
READ TABLE lt_mat_uom INTO DATA(ls_mat_uom) WITH KEY meinh = 'KAR'.
IF sy-subrc EQ 0 .
  IF wa_item-product-productno+0(2) = 'P2'.
    ls_result-gr_post = ls_result-gr_post / ls_mat_uom-umrez.
    ls_result-auto_gr = ls_result-auto_gr / ls_mat_uom-umrez.
    ls_result-manual_gr = ls_result-manual_gr / ls_mat_uom-umrez.
  ENDIF.
ENDIF.
READ TABLE lt_mat_uom INTO ls_mat_uom WITH KEY meinh = 'CAR'.
IF sy-subrc EQ 0.
  IF wa_item-product-productno+0(2) = 'P2'.
    ls_result-gr_post = ls_result-gr_post / ls_mat_uom-umrez.
    ls_result-auto_gr = ls_result-auto_gr / ls_mat_uom-umrez.
    ls_result-manual_gr = ls_result-manual_gr / ls_mat_uom-umrez.
  ENDIF.
ENDIF.

LOOP AT lt_po_trans_src INTO DATA(ls_po_src) WHERE src_po EQ ls_refdoc-refdocno AND matnr EQ wa
  ls_result-potransqty_frompo = ls_result-potransqty_frompo + ls_po_src-quantity.
ENDLOOP.
LOOP AT lt_po_trans_dest INTO DATA(ls_po_dest) WHERE dest_po EQ ls_refdoc-refdocno AND matnr EQ
  ls_result-potransqty_topo = ls_result-potransqty_topo + ls_po_src-quantity.
ENDLOOP.

IF ls_result-gr_post GT 0.
  ls_result-retry_261 = ls_result-gr_post - ls_result-261_pass.
  ls_result-scrap_req = ls_result-qty_bal_psa - ( ls_result-retry_261 - ls_result-scrap_posted
ELSE.
  ls_result-scrap_req = ls_result-qty_bal_psa - lv_scrap_qty.
ENDIF.

APPEND ls_result TO lt_result.
CLEAR : wa_item,ls_result,lv_scrap_qty.
ENDLOOP.

ENDFORM.                                "calc_data
*&-----*
*&      Form  display_data

```

```

*&-----*
*          text
*-----*
* -->  p1          text
* <--  p2          text
*-----*

```

FORM display_data .

```

DATA : lt_fcat      TYPE slis_t_fieldcat_alv,
      lt_excluding TYPE slis_t_extab,
      ls_layout     TYPE slis_layout_alv.

```

REFRESH : lt_fcat,lt_excluding.

```

INSERT VALUE slis_fieldcat_alv( fieldname = 'PO'
                                col_pos   = 1
                                tabname   = 'LT_RESULT'
                                seltext_m = 'PO'
                                outputlen = 12
                                key       = abap_true
                                ) INTO TABLE lt_fcat.

```

```

INSERT VALUE slis_fieldcat_alv( fieldname = 'PO_ITEM'
                                col_pos   = 2
                                tabname   = 'LT_RESULT'
                                seltext_m = 'PO ITEM'
                                outputlen = 14
                                key       = abap_true
                                ) INTO TABLE lt_fcat.

```

```

INSERT VALUE slis_fieldcat_alv( fieldname = 'PO_QTY'
                                col_pos   = 3
                                tabname   = 'LT_RESULT'
                                seltext_m = 'PO QTY'
                                outputlen = 13
                                key       = abap_true
                                ) INTO TABLE lt_fcat.

```

```

INSERT VALUE slis_fieldcat_alv( fieldname = 'PSA_QTY'
                                col_pos   = 4
                                tabname   = 'LT_RESULT'
                                seltext_m = 'Delivered Qty'
                                outputlen = 13
                                emphasize = abap_true
                                ) INTO TABLE lt_fcat.

```

```

INSERT VALUE slis_fieldcat_alv( fieldname = 'RETURN_QTY'
                                col_pos   = 5
                                tabname   = 'LT_RESULT'
                                seltext_m = 'Return Qty'
                                outputlen = 13
                                ) INTO TABLE lt_fcat.

```

```

INSERT VALUE slis_fieldcat_alv( fieldname = '261_PASS'
                                col_pos   = 6
                                tabname   = 'LT_RESULT'
                                seltext_m = 'Consump.Posted'
                                outputlen = 13
                                ) INTO TABLE lt_fcat.

```

*Added by Shubham on 07-Feb-2023

```

INSERT VALUE slis_fieldcat_alv( fieldname = 'CROSS_POST'
                                col_pos   = 7

```



```

        seltext_m = 'PO2PO-Removed'
        outputlen = 13
        ) INTO TABLE lt_fcat.
ENDIF.
INSERT VALUE slis_fieldcat_alv( fieldname = 'SCRAP_REQ'
                                col_pos   = 13
                                tabname   = 'LT_RESULT'
                                seltext_m = 'Scrap Required.'
                                outputlen = 13
                                ) INTO TABLE lt_fcat.

INSERT VALUE slis_fieldcat_alv( fieldname = 'SCRAP_POSTED'
                                col_pos   = 14
                                tabname   = 'LT_RESULT'
                                seltext_m = 'Scrap Posted'
                                outputlen = 13
                                ) INTO TABLE lt_fcat.

CALL FUNCTION 'REUSE_ALV_GRID_DISPLAY'
EXPORTING
*      I_INTERFACE_CHECK           = ' '
*      I_BYPASSING_BUFFER         = ' '
*      I_BUFFER_ACTIVE            = ' '
      i_grid_title                 = 'Posting Details of Process Order'
      i_callback_program           = 'ZCONSUMPTION_POSTING_REPORT'
      i_callback_pf_status_set    = 'SET_PF_STATUS'
      i_callback_user_command     = 'UCOMM'
      it_fieldcat                 = lt_fcat
      it_excluding                = lt_excluding
TABLES
      t_outtab                   = lt_result
* EXCEPTIONS
*      PROGRAM_ERROR              = 1
*      OTHERS                     = 2
.
IF sy-subrc <> 0.
* Implement suitable error handling here
ENDIF.

ENDFORM.                                     "display_data

*&-----*
*&      Form  SET_PF_STATUS
*&-----*
*
*      text
*-----*
*      --> p1      text
*      <-- p2      text
*-----*
FORM set_pf_status USING lt_excluding.
      SET PF-STATUS 'STANDARD_FULLSCREEN' OF PROGRAM 'SAPLKKB1'. " 'ZCONSUMPTION_POSTING_REPORT'. " EXCL
ENDFORM.

*&-----*
*&      Form  UCOMM
*&-----*
*
*      text
*-----*
*      --> p1      text
*      <-- p2      text
*-----*

```

```
FORM ucomm USING zucomm TYPE sy-ucomm
              zsel_field TYPE slis_selfield..
CASE sy-ucomm.
  WHEN lc_fcode_back.
    LEAVE PROGRAM.
  WHEN lc_fcode_exit.
    LEAVE PROGRAM.
  WHEN OTHERS.
    ENDCASE.
ENDFORM.
```