# ABAP Code

## Global Declaration

TYPE-POOLS: abap, lvc, icon.  
  
" 4.1 Selection Screen Parameters  
SELECTION-SCREEN BEGIN OF BLOCK b1 WITH FRAME TITLE text-t01.  
 SELECT-OPTIONS: s\_matnr FOR mara-matnr,  
 s\_werks FOR ekpo-werks OBLIGATORY,  
 s\_lifnr FOR ekko-lifnr,  
 s\_mtart FOR mara-mtart.  
 PARAMETERS: p\_lout TYPE disvariant-variant.  
SELECTION-SCREEN END OF BLOCK b1.  
  
" Constants  
CONSTANTS: c\_ucomm\_close TYPE sy-ucomm VALUE 'ZCLOSE',  
 c\_ucomm\_refresh TYPE sy-ucomm VALUE 'ZREFR',  
 c\_pf\_status TYPE sypfkey VALUE 'ZS\_PO\_CLOSE',  
 c\_title TYPE sytitl VALUE 'Utility to Close Open POs',  
 c\_repid TYPE syrepid VALUE 'ZMM\_PO\_CLOSE'.  
  
" Range Types (optional helper ranges)  
TYPES: ty\_r\_matnr TYPE RANGE OF matnr,  
 ty\_r\_werks TYPE RANGE OF werks\_d,  
 ty\_r\_lifnr TYPE RANGE OF lifnr,  
 ty\_r\_mtart TYPE RANGE OF mtart.  
  
DATA: gr\_matnr TYPE ty\_r\_matnr,  
 gr\_werks TYPE ty\_r\_werks,  
 gr\_lifnr TYPE ty\_r\_lifnr,  
 gr\_mtart TYPE ty\_r\_mtart.  
  
" Output line type for ALV  
TYPES: BEGIN OF ty\_output,  
 sel TYPE abap\_bool, " user selection checkbox  
 ebeln TYPE ekpo-ebeln,  
 ebelp TYPE ekpo-ebelp,  
 bedat TYPE ekko-bedat,  
 lifnr TYPE ekko-lifnr,  
 name1 TYPE lfa1-name1,  
 matnr TYPE ekpo-matnr,  
 txz01 TYPE ekpo-txz01,  
 werks TYPE ekpo-werks,  
 mtart TYPE mara-mtart,  
 menge TYPE ekpo-menge,  
 meins TYPE ekpo-meins,  
 elikz TYPE ekpo-elikz,  
 loekz TYPE ekpo-loekz,  
 END OF ty\_output.  
  
TYPES: BEGIN OF ty\_item\_key,  
 ebeln TYPE ekpo-ebeln,  
 ebelp TYPE ekpo-ebelp,  
 END OF ty\_item\_key.  
  
" Data for ALV output  
DATA: gt\_out TYPE STANDARD TABLE OF ty\_output WITH DEFAULT KEY,  
 gs\_out TYPE ty\_output,  
 gt\_selected TYPE STANDARD TABLE OF ty\_item\_key WITH DEFAULT KEY.  
  
" ALV Grid Control objects  
DATA: go\_dock TYPE REF TO cl\_gui\_docking\_container,  
 go\_container TYPE REF TO cl\_gui\_custom\_container,  
 go\_grid TYPE REF TO cl\_gui\_alv\_grid.  
  
" ALV technical structures  
DATA: gt\_fcat TYPE lvc\_t\_fcat,  
 gs\_fcat TYPE lvc\_s\_fcat,  
 gs\_layo TYPE lvc\_s\_layo,  
 gt\_sort TYPE lvc\_t\_sort,  
 gs\_sort TYPE lvc\_s\_sort,  
 gt\_filt TYPE lvc\_t\_filt,  
 gs\_filt TYPE lvc\_s\_filt,  
 gs\_variant TYPE disvariant,  
 g\_save TYPE c VALUE 'A',  
 gv\_title TYPE string.  
  
" Source tables buffers (optional)  
DATA: gt\_ekko TYPE STANDARD TABLE OF ekko WITH DEFAULT KEY,  
 gt\_ekpo TYPE STANDARD TABLE OF ekpo WITH DEFAULT KEY,  
 gt\_mara TYPE STANDARD TABLE OF mara WITH DEFAULT KEY,  
 gt\_lfa1 TYPE STANDARD TABLE OF lfa1 WITH DEFAULT KEY.  
  
" BAPI structures for mass closing (Delivery Completed indicator)  
DATA: gt\_bapiitem TYPE STANDARD TABLE OF bapimepoitem WITH DEFAULT KEY,  
 gt\_bapiitemx TYPE STANDARD TABLE OF bapimepoitemx WITH DEFAULT KEY,  
 gt\_return TYPE STANDARD TABLE OF bapiret2 WITH DEFAULT KEY.  
  
" Event handler local class definition  
CLASS lcl\_event\_handler DEFINITION.  
 PUBLIC SECTION.  
 METHODS:  
 on\_toolbar FOR EVENT toolbar OF cl\_gui\_alv\_grid  
 IMPORTING e\_object e\_interactive,  
 on\_user\_command FOR EVENT user\_command OF cl\_gui\_alv\_grid  
 IMPORTING e\_ucomm,  
 on\_after\_user\_command FOR EVENT after\_user\_command OF cl\_gui\_alv\_grid  
 IMPORTING e\_ucomm,  
 on\_data\_changed FOR EVENT data\_changed OF cl\_gui\_alv\_grid  
 IMPORTING er\_data\_changed,  
 on\_hotspot\_click FOR EVENT hotspot\_click OF cl\_gui\_alv\_grid  
 IMPORTING e\_row\_id e\_column\_id es\_row\_no.  
ENDCLASS.  
  
DATA: go\_handler TYPE REF TO lcl\_event\_handler.

## Selection Screen

SELECTION-SCREEN BEGIN OF BLOCK b1 WITH FRAME TITLE text-t01.  
 SELECT-OPTIONS: s\_matnr FOR mara-matnr,  
 s\_werks FOR ekpo-werks OBLIGATORY,  
 s\_lifnr FOR ekko-lifnr,  
 s\_mtart FOR mara-mtart.  
 PARAMETERS: p\_lout TYPE disvariant-variant.  
SELECTION-SCREEN END OF BLOCK b1.

## Processing Logic

INITIALIZATION.  
 gv\_title = c\_title.  
 gs\_variant-report = c\_repid.  
 gs\_variant-variant = p\_lout.  
  
AT SELECTION-SCREEN.  
 IF s\_werks[] IS INITIAL.  
 MESSAGE e398(00) WITH 'Plant (S\_WERKS) is mandatory.'.  
 ENDIF.  
  
START-OF-SELECTION.  
 PERFORM get\_data.  
 PERFORM display\_alv.  
  
CLASS lcl\_event\_handler IMPLEMENTATION.  
 METHOD on\_toolbar.  
 DATA: ls\_button TYPE stb\_button.  
  
 CLEAR ls\_button.  
 ls\_button-function = c\_ucomm\_close.  
 ls\_button-icon = icon\_okay.  
 ls\_button-quickinfo = 'Close Selected'.  
 ls\_button-text = 'Close'.  
 ls\_button-butn\_type = 0.  
 APPEND ls\_button TO e\_object->mt\_toolbar.  
  
 CLEAR ls\_button.  
 ls\_button-function = c\_ucomm\_refresh.  
 ls\_button-icon = icon\_refresh.  
 ls\_button-quickinfo = 'Refresh'.  
 ls\_button-text = 'Refresh'.  
 ls\_button-butn\_type = 0.  
 APPEND ls\_button TO e\_object->mt\_toolbar.  
 ENDMETHOD.  
  
 METHOD on\_user\_command.  
 CASE e\_ucomm.  
 WHEN c\_ucomm\_close.  
 PERFORM close\_selected.  
 WHEN c\_ucomm\_refresh.  
 PERFORM refresh\_data.  
 ENDCASE.  
 ENDMETHOD.  
  
 METHOD on\_after\_user\_command.  
 IF go\_grid IS BOUND.  
 TRY.  
 go\_grid->refresh\_table\_display( ).  
 CATCH cx\_root.  
 ENDTRY.  
 ENDIF.  
 ENDMETHOD.  
  
 METHOD on\_data\_changed.  
 " No special handling required; keep default.  
 ENDMETHOD.  
  
 METHOD on\_hotspot\_click.  
 DATA(ls\_row) = VALUE ty\_output( ).  
 READ TABLE gt\_out INTO ls\_row INDEX e\_row\_id-index.  
 IF sy-subrc = 0 AND e\_column\_id-fieldname = 'EBELN'.  
 SET PARAMETER ID 'BES' FIELD ls\_row-ebeln.  
 CALL TRANSACTION 'ME23N' AND SKIP FIRST SCREEN.  
 ENDIF.  
 ENDMETHOD.  
ENDCLASS.  
  
FORM get\_data.  
 DATA: lt\_r\_matnr\_mtart TYPE RANGE OF matnr,  
 ls\_r LIKE LINE OF lt\_r\_matnr\_mtart,  
 lt\_r\_ebeln TYPE RANGE OF ebeln,  
 ls\_r\_ebeln LIKE LINE OF lt\_r\_ebeln.  
  
 CLEAR: gt\_out, gt\_ekko, gt\_ekpo, gt\_mara, gt\_lfa1.  
  
 " Build material list from MTART if provided  
 IF s\_mtart[] IS NOT INITIAL.  
 SELECT matnr, mtart  
 FROM mara  
 INTO TABLE @gt\_mara  
 WHERE mtart IN @s\_mtart.  
 IF sy-subrc = 0.  
 SORT gt\_mara BY matnr.  
 DELETE ADJACENT DUPLICATES FROM gt\_mara COMPARING matnr.  
 LOOP AT gt\_mara ASSIGNING FIELD-SYMBOL(<m>).  
 CLEAR ls\_r.  
 ls\_r-sign = 'I'.  
 ls\_r-option = 'EQ'.  
 ls\_r-low = <m>-matnr.  
 APPEND ls\_r TO lt\_r\_matnr\_mtart.  
 ENDLOOP.  
 ENDIF.  
 ENDIF.  
  
 " If vendor provided, preselect EKKO list  
 IF s\_lifnr[] IS NOT INITIAL.  
 SELECT ebeln lifnr bedat  
 FROM ekko  
 INTO TABLE @gt\_ekko  
 WHERE lifnr IN @s\_lifnr.  
 IF sy-subrc = 0.  
 SORT gt\_ekko BY ebeln.  
 LOOP AT gt\_ekko ASSIGNING FIELD-SYMBOL(<k>).  
 CLEAR ls\_r\_ebeln.  
 ls\_r\_ebeln-sign = 'I'.  
 ls\_r\_ebeln-option = 'EQ'.  
 ls\_r\_ebeln-low = <k>-ebeln.  
 APPEND ls\_r\_ebeln TO lt\_r\_ebeln.  
 ENDLOOP.  
 ENDIF.  
 ENDIF.  
  
 " Select open PO items (EKPO)  
 SELECT ebeln ebelp matnr txz01 werks menge meins elikz loekz  
 FROM ekpo  
 INTO TABLE @gt\_ekpo  
 WHERE werks IN @s\_werks  
 AND elikz = @space  
 AND loekz <> 'L'  
 AND ( @s\_matnr[] IS INITIAL OR matnr IN @s\_matnr )  
 AND ( @lt\_r\_matnr\_mtart[] IS INITIAL OR matnr IN @lt\_r\_matnr\_mtart )  
 AND ( @lt\_r\_ebeln[] IS INITIAL OR ebeln IN @lt\_r\_ebeln ).  
  
 IF gt\_ekpo IS INITIAL.  
 RETURN.  
 ENDIF.  
  
 SORT gt\_ekpo BY ebeln ebelp.  
  
 " Ensure we have headers for selected items  
 IF gt\_ekko IS INITIAL.  
 " No vendor filter provided; fetch headers for found EBELN  
 DATA(lt\_r\_ebeln\_all) = VALUE RANGE OF ebeln( ).  
 LOOP AT gt\_ekpo ASSIGNING FIELD-SYMBOL(<i1>).  
 CLEAR ls\_r\_ebeln.  
 ls\_r\_ebeln-sign = 'I'.  
 ls\_r\_ebeln-option = 'EQ'.  
 ls\_r\_ebeln-low = <i1>-ebeln.  
 APPEND ls\_r\_ebeln TO lt\_r\_ebeln\_all.  
 ENDLOOP.  
 SORT lt\_r\_ebeln\_all BY low.  
 DELETE ADJACENT DUPLICATES FROM lt\_r\_ebeln\_all COMPARING low.  
 IF lt\_r\_ebeln\_all IS NOT INITIAL.  
 SELECT ebeln lifnr bedat  
 FROM ekko  
 INTO TABLE @gt\_ekko  
 WHERE ebeln IN @lt\_r\_ebeln\_all.  
 SORT gt\_ekko BY ebeln.  
 ENDIF.  
 ENDIF.  
  
 " Get vendor names  
 IF gt\_ekko IS NOT INITIAL.  
 DATA(lt\_r\_lifnr) = VALUE RANGE OF lifnr( ).  
 DATA(ls\_r\_lifnr) = VALUE LINE OF lt\_r\_lifnr( ).  
 LOOP AT gt\_ekko ASSIGNING FIELD-SYMBOL(<h>).  
 CLEAR ls\_r\_lifnr.  
 ls\_r\_lifnr-sign = 'I'.  
 ls\_r\_lifnr-option = 'EQ'.  
 ls\_r\_lifnr-low = <h>-lifnr.  
 APPEND ls\_r\_lifnr TO lt\_r\_lifnr.  
 ENDLOOP.  
 SORT lt\_r\_lifnr BY low.  
 DELETE ADJACENT DUPLICATES FROM lt\_r\_lifnr COMPARING low.  
  
 IF lt\_r\_lifnr IS NOT INITIAL.  
 SELECT lifnr name1  
 FROM lfa1  
 INTO TABLE @gt\_lfa1  
 WHERE lifnr IN @lt\_r\_lifnr.  
 SORT gt\_lfa1 BY lifnr.  
 ENDIF.  
 ENDIF.  
  
 " Ensure we have material types for display  
 DATA(lt\_r\_matnr\_all) = VALUE RANGE OF matnr( ).  
 DATA(ls\_r\_matnr) = VALUE LINE OF lt\_r\_matnr\_all( ).  
 LOOP AT gt\_ekpo ASSIGNING FIELD-SYMBOL(<i2>).  
 CLEAR ls\_r\_matnr.  
 ls\_r\_matnr-sign = 'I'.  
 ls\_r\_matnr-option = 'EQ'.  
 ls\_r\_matnr-low = <i2>-matnr.  
 APPEND ls\_r\_matnr TO lt\_r\_matnr\_all.  
 ENDLOOP.  
 SORT lt\_r\_matnr\_all BY low.  
 DELETE ADJACENT DUPLICATES FROM lt\_r\_matnr\_all COMPARING low.  
  
 IF lt\_r\_matnr\_all IS NOT INITIAL.  
 SELECT matnr mtart  
 FROM mara  
 INTO TABLE @gt\_mara  
 WHERE matnr IN @lt\_r\_matnr\_all.  
 SORT gt\_mara BY matnr.  
 ENDIF.  
  
 " Build output  
 CLEAR gt\_out.  
 LOOP AT gt\_ekpo ASSIGNING FIELD-SYMBOL(<it>).  
 CLEAR gs\_out.  
 gs\_out-sel = abap\_false.  
 gs\_out-ebeln = <it>-ebeln.  
 gs\_out-ebelp = <it>-ebelp.  
 gs\_out-matnr = <it>-matnr.  
 gs\_out-txz01 = <it>-txz01.  
 gs\_out-werks = <it>-werks.  
 gs\_out-menge = <it>-menge.  
 gs\_out-meins = <it>-meins.  
 gs\_out-elikz = <it>-elikz.  
 gs\_out-loekz = <it>-loekz.  
  
 READ TABLE gt\_ekko ASSIGNING FIELD-SYMBOL(<hd>) WITH KEY ebeln = <it>-ebeln BINARY SEARCH.  
 IF sy-subrc = 0.  
 gs\_out-bedat = <hd>-bedat.  
 gs\_out-lifnr = <hd>-lifnr.  
 READ TABLE gt\_lfa1 ASSIGNING FIELD-SYMBOL(<vn>) WITH KEY lifnr = <hd>-lifnr BINARY SEARCH.  
 IF sy-subrc = 0.  
 gs\_out-name1 = <vn>-name1.  
 ENDIF.  
 ENDIF.  
  
 READ TABLE gt\_mara ASSIGNING FIELD-SYMBOL(<mm>) WITH KEY matnr = <it>-matnr BINARY SEARCH.  
 IF sy-subrc = 0.  
 gs\_out-mtart = <mm>-mtart.  
 ENDIF.  
  
 APPEND gs\_out TO gt\_out.  
 ENDLOOP.  
ENDFORM.  
  
FORM build\_fieldcatalog.  
 CLEAR gt\_fcat.  
  
 DEFINE add\_col.  
 CLEAR gs\_fcat.  
 gs\_fcat-fieldname = &1.  
 gs\_fcat-coltext = &2.  
 gs\_fcat-outputlen = &3.  
 gs\_fcat-key = &4.  
 gs\_fcat-edit = &5.  
 gs\_fcat-checkbox = &6.  
 gs\_fcat-hotspot = &7.  
 APPEND gs\_fcat TO gt\_fcat.  
 END-OF-DEFINITION.  
  
 add\_col 'SEL' 'Sel' 3 space 'X' 'X' space.  
 add\_col 'EBELN' 'PO' 10 'X' space space 'X'.  
 add\_col 'EBELP' 'Item' 5 'X' space space space.  
 add\_col 'BEDAT' 'Doc.Date' 10 space space space space.  
 add\_col 'LIFNR' 'Vendor' 10 space space space space.  
 add\_col 'NAME1' 'Name' 25 space space space space.  
 add\_col 'MATNR' 'Material' 18 space space space space.  
 add\_col 'TXZ01' 'Short Text' 40 space space space space.  
 add\_col 'WERKS' 'Plant' 4 space space space space.  
 add\_col 'MTART' 'Mat.Type' 4 space space space space.  
 add\_col 'MENGE' 'PO Qty' 13 space space space space.  
 add\_col 'MEINS' 'UoM' 3 space space space space.  
 add\_col 'ELIKZ' 'Del.Comp' 1 space space space space.  
 add\_col 'LOEKZ' 'Deleted' 1 space space space space.  
ENDFORM.  
  
FORM setup\_layout.  
 CLEAR gs\_layo.  
 gs\_layo-zebra = abap\_true.  
 gs\_layo-cwidth\_opt = abap\_true.  
 gs\_layo-grid\_title = gv\_title.  
ENDFORM.  
  
FORM display\_alv.  
 PERFORM build\_fieldcatalog.  
 PERFORM setup\_layout.  
  
 IF go\_grid IS INITIAL.  
 TRY.  
 CREATE OBJECT go\_dock  
 EXPORTING  
 repid = sy-repid  
 dynnr = sy-dynnr  
 side = cl\_gui\_docking\_container=>dock\_at\_left  
 extension = 3000.  
  
 CREATE OBJECT go\_grid  
 EXPORTING  
 i\_parent = go\_dock.  
  
 CREATE OBJECT go\_handler.  
  
 SET HANDLER go\_handler->on\_toolbar FOR go\_grid.  
 SET HANDLER go\_handler->on\_user\_command FOR go\_grid.  
 SET HANDLER go\_handler->on\_after\_user\_command FOR go\_grid.  
 SET HANDLER go\_handler->on\_data\_changed FOR go\_grid.  
 SET HANDLER go\_handler->on\_hotspot\_click FOR go\_grid.  
  
 CALL METHOD go\_grid->set\_table\_for\_first\_display  
 EXPORTING  
 is\_layout = gs\_layo  
 is\_variant = gs\_variant  
 i\_save = g\_save  
 CHANGING  
 it\_outtab = gt\_out  
 it\_fieldcatalog = gt\_fcat  
 it\_sort = gt\_sort  
 it\_filter = gt\_filt.  
  
 CALL METHOD go\_grid->register\_edit\_event  
 EXPORTING  
 i\_event\_id = cl\_gui\_alv\_grid=>mc\_evt\_modified.  
  
 SET PF-STATUS c\_pf\_status.  
 SET TITLEBAR c\_title.  
  
 CATCH cx\_root INTO DATA(lx).  
 MESSAGE lx->get\_text( ) TYPE 'E'.  
 ENDTRY.  
 ELSE.  
 TRY.  
 go\_grid->set\_frontend\_layout( gs\_layo ).  
 go\_grid->set\_variant( gs\_variant ).  
 go\_grid->refresh\_table\_display( ).  
 CATCH cx\_root.  
 ENDTRY.  
 ENDIF.  
ENDFORM.  
  
FORM refresh\_data.  
 PERFORM get\_data.  
 IF go\_grid IS BOUND.  
 TRY.  
 go\_grid->refresh\_table\_display( ).  
 CATCH cx\_root.  
 ENDTRY.  
 ENDIF.  
ENDFORM.  
  
FORM close\_selected.  
 DATA: lt\_sel TYPE STANDARD TABLE OF ty\_item\_key WITH DEFAULT KEY,  
 ls\_sel TYPE ty\_item\_key,  
 lv\_ebeln TYPE ekpo-ebeln,  
 lt\_items TYPE STANDARD TABLE OF bapimepoitem WITH DEFAULT KEY,  
 lt\_itemsx TYPE STANDARD TABLE OF bapimepoitemx WITH DEFAULT KEY,  
 ls\_item TYPE bapimepoitem,  
 ls\_itemx TYPE bapimepoitemx,  
 lt\_return TYPE STANDARD TABLE OF bapiret2 WITH DEFAULT KEY,  
 lv\_err TYPE abap\_bool,  
 lv\_ok\_cnt TYPE i VALUE 0,  
 lv\_er\_cnt TYPE i VALUE 0.  
  
 IF go\_grid IS BOUND.  
 TRY.  
 go\_grid->check\_changed\_data( ).  
 CATCH cx\_root.  
 ENDTRY.  
 ENDIF.  
  
 LOOP AT gt\_out ASSIGNING FIELD-SYMBOL(<o>) WHERE sel = abap\_true.  
 ls\_sel-ebeln = <o>-ebeln.  
 ls\_sel-ebelp = <o>-ebelp.  
 APPEND ls\_sel TO lt\_sel.  
 ENDLOOP.  
  
 IF lt\_sel IS INITIAL.  
 MESSAGE s398(00) WITH 'No items selected.'.  
 RETURN.  
 ENDIF.  
  
 SORT lt\_sel BY ebeln ebelp.  
  
 LOOP AT lt\_sel INTO ls\_sel.  
 AT NEW ebeln.  
 CLEAR: lv\_ebeln, lt\_items, lt\_itemsx, lt\_return, lv\_err.  
 lv\_ebeln = ls\_sel-ebeln.  
 ENDAT.  
  
 CLEAR: ls\_item, ls\_itemx.  
 ls\_item-po\_item = ls\_sel-ebelp.  
 ls\_item-no\_more\_gr = 'X'.  
 APPEND ls\_item TO lt\_items.  
  
 ls\_itemx-po\_item = ls\_sel-ebelp.  
 ls\_itemx-no\_more\_gr = 'X'.  
 APPEND ls\_itemx TO lt\_itemsx.  
  
 AT END OF ebeln.  
 CALL FUNCTION 'BAPI\_PO\_CHANGE'  
 EXPORTING  
 purchaseorder = lv\_ebeln  
 TABLES  
 poitem = lt\_items  
 poitemx = lt\_itemsx  
 return = lt\_return.  
  
 READ TABLE lt\_return WITH KEY type = 'E' TRANSPORTING NO FIELDS.  
 IF sy-subrc <> 0.  
 READ TABLE lt\_return WITH KEY type = 'A' TRANSPORTING NO FIELDS.  
 ENDIF.  
 IF sy-subrc = 0.  
 lv\_err = abap\_true.  
 ENDIF.  
  
 IF lv\_err = abap\_false.  
 CALL FUNCTION 'BAPI\_TRANSACTION\_COMMIT'  
 EXPORTING  
 wait = abap\_true.  
 " Update local table and counters  
 LOOP AT gt\_out ASSIGNING <o> WHERE ebeln = lv\_ebeln AND sel = abap\_true.  
 <o>-elikz = 'X'.  
 <o>-sel = abap\_false.  
 ENDLOOP.  
 ADD 1 TO lv\_ok\_cnt.  
 ELSE.  
 CALL FUNCTION 'BAPI\_TRANSACTION\_ROLLBACK'.  
 ADD 1 TO lv\_er\_cnt.  
 ENDIF.  
 ENDAT.  
 ENDLOOP.  
  
 " Remove closed items from view  
 DELETE gt\_out WHERE elikz = 'X'.  
  
 IF go\_grid IS BOUND.  
 TRY.  
 go\_grid->refresh\_table\_display( ).  
 CATCH cx\_root.  
 ENDTRY.  
 ENDIF.  
  
 DATA(lv\_msg) = |Closed: { lv\_ok\_cnt } PO(s), Errors: { lv\_er\_cnt }|.  
 MESSAGE s398(00) WITH lv\_msg.  
ENDFORM.

## Output Display

FORM build\_fieldcatalog.  
 CLEAR gt\_fcat.  
  
 DEFINE add\_col.  
 CLEAR gs\_fcat.  
 gs\_fcat-fieldname = &1.  
 gs\_fcat-coltext = &2.  
 gs\_fcat-outputlen = &3.  
 gs\_fcat-key = &4.  
 gs\_fcat-edit = &5.  
 gs\_fcat-checkbox = &6.  
 gs\_fcat-hotspot = &7.  
 APPEND gs\_fcat TO gt\_fcat.  
 END-OF-DEFINITION.  
  
 add\_col 'SEL' 'Sel' 3 space 'X' 'X' space.  
 add\_col 'EBELN' 'PO' 10 'X' space space 'X'.  
 add\_col 'EBELP' 'Item' 5 'X' space space space.  
 add\_col 'BEDAT' 'Doc.Date' 10 space space space space.  
 add\_col 'LIFNR' 'Vendor' 10 space space space space.  
 add\_col 'NAME1' 'Name' 25 space space space space.  
 add\_col 'MATNR' 'Material' 18 space space space space.  
 add\_col 'TXZ01' 'Short Text' 40 space space space space.  
 add\_col 'WERKS' 'Plant' 4 space space space space.  
 add\_col 'MTART' 'Mat.Type' 4 space space space space.  
 add\_col 'MENGE' 'PO Qty' 13 space space space space.  
 add\_col 'MEINS' 'UoM' 3 space space space space.  
 add\_col 'ELIKZ' 'Del.Comp' 1 space space space space.  
 add\_col 'LOEKZ' 'Deleted' 1 space space space space.  
  
 " Enable totals for quantity  
 READ TABLE gt\_fcat INTO gs\_fcat WITH KEY fieldname = 'MENGE'.  
 IF sy-subrc = 0.  
 gs\_fcat-do\_sum = abap\_true.  
 MODIFY gt\_fcat FROM gs\_fcat INDEX sy-tabix.  
 ENDIF.  
ENDFORM.  
  
FORM build\_sort.  
 CLEAR gt\_sort.  
  
 CLEAR gs\_sort.  
 gs\_sort-fieldname = 'LIFNR'.  
 gs\_sort-up = abap\_true.  
 gs\_sort-subtot = abap\_true.  
 APPEND gs\_sort TO gt\_sort.  
  
 CLEAR gs\_sort.  
 gs\_sort-fieldname = 'EBELN'.  
 gs\_sort-up = abap\_true.  
 gs\_sort-subtot = abap\_true.  
 APPEND gs\_sort TO gt\_sort.  
  
 CLEAR gs\_sort.  
 gs\_sort-fieldname = 'WERKS'.  
 gs\_sort-up = abap\_true.  
 gs\_sort-subtot = abap\_true.  
 APPEND gs\_sort TO gt\_sort.  
  
 CLEAR gs\_sort.  
 gs\_sort-fieldname = 'EBELP'.  
 gs\_sort-up = abap\_true.  
 APPEND gs\_sort TO gt\_sort.  
ENDFORM.  
  
FORM setup\_layout.  
 CLEAR gs\_layo.  
 gs\_layo-zebra = abap\_true.  
 gs\_layo-cwidth\_opt = abap\_true.  
 gs\_layo-grid\_title = gv\_title.  
 gs\_layo-sel\_mode = 'A'. " multiple row selection  
 gs\_layo-totals\_bef = abap\_true.  
 gs\_layo-totals\_after = abap\_true.  
ENDFORM.  
  
FORM display\_alv.  
 PERFORM build\_fieldcatalog.  
 PERFORM build\_sort.  
 PERFORM setup\_layout.  
  
 IF go\_grid IS INITIAL.  
 TRY.  
 CREATE OBJECT go\_dock  
 EXPORTING  
 repid = sy-repid  
 dynnr = sy-dynnr  
 side = cl\_gui\_docking\_container=>dock\_at\_left  
 extension = 3000.  
  
 CREATE OBJECT go\_grid  
 EXPORTING  
 i\_parent = go\_dock.  
  
 CREATE OBJECT go\_handler.  
  
 SET HANDLER go\_handler->on\_toolbar FOR go\_grid.  
 SET HANDLER go\_handler->on\_user\_command FOR go\_grid.  
 SET HANDLER go\_handler->on\_after\_user\_command FOR go\_grid.  
 SET HANDLER go\_handler->on\_data\_changed FOR go\_grid.  
 SET HANDLER go\_handler->on\_hotspot\_click FOR go\_grid.  
  
 CALL METHOD go\_grid->set\_table\_for\_first\_display  
 EXPORTING  
 is\_layout = gs\_layo  
 is\_variant = gs\_variant  
 i\_save = g\_save  
 CHANGING  
 it\_outtab = gt\_out  
 it\_fieldcatalog = gt\_fcat  
 it\_sort = gt\_sort  
 it\_filter = gt\_filt.  
  
 CALL METHOD go\_grid->register\_edit\_event  
 EXPORTING  
 i\_event\_id = cl\_gui\_alv\_grid=>mc\_evt\_modified.  
  
 SET PF-STATUS c\_pf\_status.  
 SET TITLEBAR c\_title.  
  
 CATCH cx\_salv\_msg INTO DATA(lx\_salv).  
 MESSAGE lx\_salv->get\_text( ) TYPE 'S' DISPLAY LIKE 'E'.  
 CATCH cx\_root INTO DATA(lx\_root).  
 MESSAGE lx\_root->get\_text( ) TYPE 'S' DISPLAY LIKE 'E'.  
 ENDTRY.  
 ELSE.  
 TRY.  
 go\_grid->set\_frontend\_layout( gs\_layo ).  
 go\_grid->set\_variant( gs\_variant ).  
 go\_grid->refresh\_table\_display( is\_stable = VALUE lvc\_s\_stbl( row = abap\_true col = abap\_true ) ).  
 CATCH cx\_salv\_msg INTO lx\_salv.  
 MESSAGE lx\_salv->get\_text( ) TYPE 'S' DISPLAY LIKE 'E'.  
 CATCH cx\_root INTO lx\_root.  
 MESSAGE lx\_root->get\_text( ) TYPE 'S' DISPLAY LIKE 'E'.  
 ENDTRY.  
 ENDIF.  
ENDFORM.