# ABAP Code

## Global Declaration

TABLES: ekko, ekpo, eket, lfa1, mara.  
  
SELECTION-SCREEN BEGIN OF BLOCK b1 WITH FRAME.  
 SELECT-OPTIONS: s\_matnr FOR mara-matnr,  
 s\_werks FOR ekpo-werks OBLIGATORY,  
 s\_lifnr FOR ekko-lifnr,  
 s\_mtart FOR mara-mtart.  
SELECTION-SCREEN END OF BLOCK b1.  
  
TYPES: BEGIN OF ty\_ekpo\_out,  
 sel TYPE c LENGTH 1,  
 ebeln TYPE ekpo-ebeln,  
 ebelp TYPE ekpo-ebelp,  
 lifnr TYPE ekko-lifnr,  
 name1 TYPE lfa1-name1,  
 matnr TYPE ekpo-matnr,  
 mtart TYPE mara-mtart,  
 werks TYPE ekpo-werks,  
 menge TYPE ekpo-menge,  
 meins TYPE ekpo-meins,  
 elikz TYPE ekpo-elikz,  
 bedat TYPE ekko-bedat,  
 END OF ty\_ekpo\_out.  
  
TYPES ty\_t\_ekpo\_out TYPE STANDARD TABLE OF ty\_ekpo\_out WITH DEFAULT KEY.  
TYPES ty\_t\_matnr TYPE STANDARD TABLE OF mara-matnr WITH DEFAULT KEY.  
TYPES ty\_t\_ebeln TYPE STANDARD TABLE OF ekko-ebeln WITH DEFAULT KEY.  
  
TYPES: BEGIN OF ty\_to\_close,  
 ebeln TYPE ekpo-ebeln,  
 ebelp TYPE ekpo-ebelp,  
 END OF ty\_to\_close.  
TYPES ty\_t\_to\_close TYPE STANDARD TABLE OF ty\_to\_close WITH DEFAULT KEY.  
  
DATA: gt\_out TYPE ty\_t\_ekpo\_out,  
 gs\_out TYPE ty\_ekpo\_out,  
 gt\_to\_close TYPE ty\_t\_to\_close.  
  
DATA: gt\_ekpo TYPE STANDARD TABLE OF ekpo WITH DEFAULT KEY,  
 gt\_ekko TYPE STANDARD TABLE OF ekko WITH DEFAULT KEY,  
 gt\_mara TYPE STANDARD TABLE OF mara WITH DEFAULT KEY,  
 gt\_lfa1 TYPE STANDARD TABLE OF lfa1 WITH DEFAULT KEY.  
  
DATA: gt\_matnrs TYPE ty\_t\_matnr,  
 gt\_ebelns TYPE ty\_t\_ebeln.  
  
DATA: go\_dock TYPE REF TO cl\_gui\_docking\_container,  
 go\_grid TYPE REF TO cl\_gui\_alv\_grid.  
  
DATA: gs\_layout TYPE lvc\_s\_layo,  
 gt\_fieldcat TYPE lvc\_t\_fcat,  
 gt\_sort TYPE lvc\_t\_sort,  
 gt\_filter TYPE lvc\_t\_filt,  
 gs\_variant TYPE disvariant,  
 gv\_save TYPE c VALUE 'A'.  
  
DATA: gt\_rows TYPE lvc\_t\_row,  
 gs\_row TYPE lvc\_s\_row.  
  
CONSTANTS: c\_fcode\_close TYPE syucomm VALUE 'CLOSE\_PO',  
 c\_repid TYPE sy-repid VALUE 'ZMM\_PO\_CLOSE',  
 c\_x TYPE c VALUE 'X'.  
  
FIELD-SYMBOLS: <fs\_out> TYPE ty\_ekpo\_out.  
  
CLASS lcl\_events DEFINITION.  
 PUBLIC SECTION.  
 METHODS handle\_toolbar  
 FOR EVENT toolbar OF cl\_gui\_alv\_grid  
 IMPORTING e\_object e\_interactive.  
 METHODS handle\_user\_command  
 FOR EVENT user\_command OF cl\_gui\_alv\_grid  
 IMPORTING e\_ucomm.  
ENDCLASS.  
  
DATA go\_events TYPE REF TO lcl\_events.

## Selection Screen

SELECTION-SCREEN BEGIN OF BLOCK b1 WITH FRAME.  
 SELECT-OPTIONS: s\_matnr FOR mara-matnr,  
 s\_werks FOR ekpo-werks OBLIGATORY,  
 s\_lifnr FOR ekko-lifnr,  
 s\_mtart FOR mara-mtart.  
SELECTION-SCREEN END OF BLOCK b1.  
  
SELECTION-SCREEN BEGIN OF BLOCK b2 WITH FRAME.  
 PARAMETERS: p\_var TYPE disvariant-variant.  
SELECTION-SCREEN END OF BLOCK b2.  
  
AT SELECTION-SCREEN ON VALUE-REQUEST FOR p\_var.  
 DATA lv\_exit TYPE c.  
 gs\_variant-report = c\_repid.  
 CALL FUNCTION 'REUSE\_ALV\_VARIANT\_F4'  
 EXPORTING  
 is\_variant = gs\_variant  
 i\_save = gv\_save  
 IMPORTING  
 es\_variant = gs\_variant  
 e\_exit = lv\_exit  
 EXCEPTIONS  
 not\_found = 1  
 OTHERS = 2.  
 IF sy-subrc = 0 AND lv\_exit IS INITIAL.  
 p\_var = gs\_variant-variant.  
 ENDIF.

## Processing Logic

CLASS lcl\_events IMPLEMENTATION.  
 METHOD handle\_toolbar.  
 DATA ls\_tb TYPE stb\_button.  
  
 CLEAR ls\_tb.  
 ls\_tb-butn\_type = 3. " separator  
 APPEND ls\_tb TO e\_object->mt\_toolbar.  
  
 CLEAR ls\_tb.  
 ls\_tb-function = c\_fcode\_close.  
 ls\_tb-text = 'Close Items'.  
 ls\_tb-quickinfo = 'Set Delivery Completed (ELIKZ) = X'.  
 ls\_tb-icon = icon\_okay.  
 APPEND ls\_tb TO e\_object->mt\_toolbar.  
 ENDMETHOD.  
  
 METHOD handle\_user\_command.  
 DATA: lt\_to\_close TYPE ty\_t\_to\_close,  
 ls\_to\_close TYPE ty\_to\_close,  
 lv\_updates TYPE i.  
  
 IF e\_ucomm = c\_fcode\_close AND go\_grid IS BOUND.  
 CALL METHOD go\_grid->check\_changed\_data.  
  
 LOOP AT gt\_out ASSIGNING FIELD-SYMBOL(<ls\_out>) WHERE sel = c\_x AND elikz IS INITIAL.  
 ls\_to\_close-ebeln = <ls\_out>-ebeln.  
 ls\_to\_close-ebelp = <ls\_out>-ebelp.  
 APPEND ls\_to\_close TO lt\_to\_close.  
 ENDLOOP.  
  
 IF lt\_to\_close IS NOT INITIAL.  
 LOOP AT lt\_to\_close INTO ls\_to\_close.  
 UPDATE ekpo SET elikz = c\_x WHERE ebeln = ls\_to\_close-ebeln AND ebelp = ls\_to\_close-ebelp.  
 IF sy-subrc = 0.  
 lv\_updates = lv\_updates + 1.  
 ENDIF.  
 ENDLOOP.  
 IF lv\_updates > 0.  
 COMMIT WORK AND WAIT.  
 LOOP AT gt\_out ASSIGNING <ls\_out> WHERE sel = c\_x.  
 <ls\_out>-elikz = c\_x.  
 <ls\_out>-sel = space.  
 ENDLOOP.  
 IF go\_grid IS BOUND.  
 go\_grid->refresh\_table\_display( ).  
 ENDIF.  
 ELSE.  
 ROLLBACK WORK.  
 ENDIF.  
 ENDIF.  
 ENDIF.  
 ENDMETHOD.  
ENDCLASS.  
  
START-OF-SELECTION.  
 DATA: lt\_data TYPE STANDARD TABLE OF ty\_ekpo\_out,  
 ls\_fcat TYPE lvc\_s\_fcat,  
 lv\_space TYPE c VALUE space.  
  
 " Get materials by MTART if provided  
 IF s\_mtart[] IS NOT INITIAL.  
 SELECT matnr  
 FROM mara  
 INTO TABLE @gt\_matnrs  
 WHERE mtart IN @s\_mtart  
 AND ( @s\_matnr[] IS INITIAL OR matnr IN @s\_matnr ).  
 SORT gt\_matnrs.  
 DELETE ADJACENT DUPLICATES FROM gt\_matnrs.  
 ENDIF.  
  
 " Prepare selection to EKPO/EKKO/MARA/LFA1  
 CLEAR gt\_out.  
  
 IF s\_mtart[] IS NOT INITIAL.  
 SELECT a~ebeln,  
 a~ebelp,  
 b~lifnr,  
 d~name1,  
 a~matnr,  
 c~mtart,  
 a~werks,  
 a~menge,  
 a~meins,  
 a~elikz,  
 b~bedat  
 FROM ekpo AS a  
 INNER JOIN ekko AS b ON a~ebeln = b~ebeln  
 LEFT OUTER JOIN mara AS c ON c~matnr = a~matnr  
 LEFT OUTER JOIN lfa1 AS d ON d~lifnr = b~lifnr  
 INTO CORRESPONDING FIELDS OF TABLE @gt\_out  
 WHERE a~werks IN @s\_werks  
 AND a~elikz = @lv\_space  
 AND a~matnr IN @gt\_matnrs  
 AND ( @s\_lifnr[] IS INITIAL OR b~lifnr IN @s\_lifnr ).  
 ELSE.  
 SELECT a~ebeln,  
 a~ebelp,  
 b~lifnr,  
 d~name1,  
 a~matnr,  
 c~mtart,  
 a~werks,  
 a~menge,  
 a~meins,  
 a~elikz,  
 b~bedat  
 FROM ekpo AS a  
 INNER JOIN ekko AS b ON a~ebeln = b~ebeln  
 LEFT OUTER JOIN mara AS c ON c~matnr = a~matnr  
 LEFT OUTER JOIN lfa1 AS d ON d~lifnr = b~lifnr  
 INTO CORRESPONDING FIELDS OF TABLE @gt\_out  
 WHERE a~werks IN @s\_werks  
 AND a~elikz = @lv\_space  
 AND ( @s\_matnr[] IS INITIAL OR a~matnr IN @s\_matnr )  
 AND ( @s\_lifnr[] IS INITIAL OR b~lifnr IN @s\_lifnr ).  
 ENDIF.  
  
 " Initialize selection flag  
 LOOP AT gt\_out ASSIGNING <fs\_out>.  
 <fs\_out>-sel = space.  
 ENDLOOP.  
  
 " Build ALV field catalog  
 CLEAR gt\_fieldcat.  
  
 CLEAR ls\_fcat.  
 ls\_fcat-fieldname = 'SEL'.  
 ls\_fcat-coltext = 'Sel'.  
 ls\_fcat-scrtext\_l = 'Select'.  
 ls\_fcat-checkbox = c\_x.  
 ls\_fcat-edit = c\_x.  
 ls\_fcat-outputlen = 4.  
 APPEND ls\_fcat TO gt\_fieldcat.  
  
 CLEAR ls\_fcat.  
 ls\_fcat-fieldname = 'EBELN'.  
 ls\_fcat-coltext = 'PO'.  
 ls\_fcat-scrtext\_l = 'PO Number'.  
 ls\_fcat-key = c\_x.  
 APPEND ls\_fcat TO gt\_fieldcat.  
  
 CLEAR ls\_fcat.  
 ls\_fcat-fieldname = 'EBELP'.  
 ls\_fcat-coltext = 'Item'.  
 ls\_fcat-scrtext\_l = 'Item'.  
 ls\_fcat-key = c\_x.  
 APPEND ls\_fcat TO gt\_fieldcat.  
  
 CLEAR ls\_fcat.  
 ls\_fcat-fieldname = 'LIFNR'.  
 ls\_fcat-coltext = 'Vendor'.  
 ls\_fcat-scrtext\_l = 'Vendor'.  
 APPEND ls\_fcat TO gt\_fieldcat.  
  
 CLEAR ls\_fcat.  
 ls\_fcat-fieldname = 'NAME1'.  
 ls\_fcat-coltext = 'Vendor Name'.  
 ls\_fcat-scrtext\_l = 'Vendor Name'.  
 ls\_fcat-outputlen = 25.  
 APPEND ls\_fcat TO gt\_fieldcat.  
  
 CLEAR ls\_fcat.  
 ls\_fcat-fieldname = 'MATNR'.  
 ls\_fcat-coltext = 'Material'.  
 ls\_fcat-scrtext\_l = 'Material'.  
 ls\_fcat-outputlen = 18.  
 APPEND ls\_fcat TO gt\_fieldcat.  
  
 CLEAR ls\_fcat.  
 ls\_fcat-fieldname = 'MTART'.  
 ls\_fcat-coltext = 'Mat. Type'.  
 ls\_fcat-scrtext\_l = 'Mat. Type'.  
 APPEND ls\_fcat TO gt\_fieldcat.  
  
 CLEAR ls\_fcat.  
 ls\_fcat-fieldname = 'WERKS'.  
 ls\_fcat-coltext = 'Plant'.  
 ls\_fcat-scrtext\_l = 'Plant'.  
 APPEND ls\_fcat TO gt\_fieldcat.  
  
 CLEAR ls\_fcat.  
 ls\_fcat-fieldname = 'MENGE'.  
 ls\_fcat-coltext = 'Quantity'.  
 ls\_fcat-scrtext\_l = 'Quantity'.  
 ls\_fcat-do\_sum = c\_x.  
 APPEND ls\_fcat TO gt\_fieldcat.  
  
 CLEAR ls\_fcat.  
 ls\_fcat-fieldname = 'MEINS'.  
 ls\_fcat-coltext = 'UoM'.  
 ls\_fcat-scrtext\_l = 'UoM'.  
 APPEND ls\_fcat TO gt\_fieldcat.  
  
 CLEAR ls\_fcat.  
 ls\_fcat-fieldname = 'ELIKZ'.  
 ls\_fcat-coltext = 'Closed'.  
 ls\_fcat-scrtext\_l = 'Closed'.  
 APPEND ls\_fcat TO gt\_fieldcat.  
  
 CLEAR ls\_fcat.  
 ls\_fcat-fieldname = 'BEDAT'.  
 ls\_fcat-coltext = 'PO Date'.  
 ls\_fcat-scrtext\_l = 'PO Date'.  
 ls\_fcat-outputlen = 10.  
 APPEND ls\_fcat TO gt\_fieldcat.  
  
 " Layout and variant  
 CLEAR gs\_layout.  
 gs\_layout-zebra = c\_x.  
 gs\_layout-cwidth\_opt = c\_x.  
 gs\_layout-sel\_mode = 'A'.  
 gs\_layout-edit = c\_x.  
  
 CLEAR gs\_variant.  
 gs\_variant-report = c\_repid.  
 gs\_variant-variant = p\_var.  
  
 " Create container and grid  
 IF go\_dock IS INITIAL.  
 CREATE OBJECT go\_dock  
 EXPORTING  
 ratio = 90  
 side = cl\_gui\_docking\_container=>dock\_at\_left.  
  
 CREATE OBJECT go\_grid  
 EXPORTING  
 i\_parent = go\_dock.  
  
 go\_grid->register\_edit\_event( cl\_gui\_alv\_grid=>mc\_evt\_modified ).  
  
 CREATE OBJECT go\_events.  
 SET HANDLER go\_events->handle\_toolbar FOR go\_grid.  
 SET HANDLER go\_events->handle\_user\_command FOR go\_grid.  
  
 go\_grid->set\_table\_for\_first\_display(  
 EXPORTING  
 is\_layout = gs\_layout  
 is\_variant = gs\_variant  
 i\_save = gv\_save  
 CHANGING  
 it\_outtab = gt\_out  
 it\_fieldcatalog = gt\_fieldcat  
 it\_sort = gt\_sort  
 it\_filter = gt\_filter ).  
  
 go\_grid->set\_ready\_for\_input( c\_x ).  
 cl\_gui\_cfw=>flush( ).  
 ELSE.  
 go\_grid->refresh\_table\_display( ).  
 ENDIF.

## Output Display

DATA: ls\_fcat TYPE lvc\_s\_fcat,  
 ls\_sort TYPE lvc\_s\_sort,  
 ls\_stbl TYPE lvc\_s\_stbl.  
  
" Build ALV field catalog  
CLEAR gt\_fieldcat.  
  
CLEAR ls\_fcat.  
ls\_fcat-fieldname = 'SEL'.  
ls\_fcat-coltext = 'Sel'.  
ls\_fcat-scrtext\_l = 'Select'.  
ls\_fcat-checkbox = c\_x.  
ls\_fcat-edit = c\_x.  
ls\_fcat-outputlen = 4.  
APPEND ls\_fcat TO gt\_fieldcat.  
  
CLEAR ls\_fcat.  
ls\_fcat-fieldname = 'EBELN'.  
ls\_fcat-coltext = 'PO'.  
ls\_fcat-scrtext\_l = 'PO Number'.  
ls\_fcat-key = c\_x.  
APPEND ls\_fcat TO gt\_fieldcat.  
  
CLEAR ls\_fcat.  
ls\_fcat-fieldname = 'EBELP'.  
ls\_fcat-coltext = 'Item'.  
ls\_fcat-scrtext\_l = 'Item'.  
ls\_fcat-key = c\_x.  
APPEND ls\_fcat TO gt\_fieldcat.  
  
CLEAR ls\_fcat.  
ls\_fcat-fieldname = 'LIFNR'.  
ls\_fcat-coltext = 'Vendor'.  
ls\_fcat-scrtext\_l = 'Vendor'.  
APPEND ls\_fcat TO gt\_fieldcat.  
  
CLEAR ls\_fcat.  
ls\_fcat-fieldname = 'NAME1'.  
ls\_fcat-coltext = 'Vendor Name'.  
ls\_fcat-scrtext\_l = 'Vendor Name'.  
ls\_fcat-outputlen = 25.  
APPEND ls\_fcat TO gt\_fieldcat.  
  
CLEAR ls\_fcat.  
ls\_fcat-fieldname = 'MATNR'.  
ls\_fcat-coltext = 'Material'.  
ls\_fcat-scrtext\_l = 'Material'.  
ls\_fcat-outputlen = 18.  
APPEND ls\_fcat TO gt\_fieldcat.  
  
CLEAR ls\_fcat.  
ls\_fcat-fieldname = 'MTART'.  
ls\_fcat-coltext = 'Mat. Type'.  
ls\_fcat-scrtext\_l = 'Mat. Type'.  
APPEND ls\_fcat TO gt\_fieldcat.  
  
CLEAR ls\_fcat.  
ls\_fcat-fieldname = 'WERKS'.  
ls\_fcat-coltext = 'Plant'.  
ls\_fcat-scrtext\_l = 'Plant'.  
APPEND ls\_fcat TO gt\_fieldcat.  
  
CLEAR ls\_fcat.  
ls\_fcat-fieldname = 'MENGE'.  
ls\_fcat-coltext = 'Quantity'.  
ls\_fcat-scrtext\_l = 'Quantity'.  
ls\_fcat-do\_sum = c\_x. " enable totals/subtotals  
APPEND ls\_fcat TO gt\_fieldcat.  
  
CLEAR ls\_fcat.  
ls\_fcat-fieldname = 'MEINS'.  
ls\_fcat-coltext = 'UoM'.  
ls\_fcat-scrtext\_l = 'UoM'.  
APPEND ls\_fcat TO gt\_fieldcat.  
  
CLEAR ls\_fcat.  
ls\_fcat-fieldname = 'ELIKZ'.  
ls\_fcat-coltext = 'Closed'.  
ls\_fcat-scrtext\_l = 'Closed'.  
APPEND ls\_fcat TO gt\_fieldcat.  
  
CLEAR ls\_fcat.  
ls\_fcat-fieldname = 'BEDAT'.  
ls\_fcat-coltext = 'PO Date'.  
ls\_fcat-scrtext\_l = 'PO Date'.  
ls\_fcat-outputlen = 10.  
APPEND ls\_fcat TO gt\_fieldcat.  
  
" Sorting and subtotals  
CLEAR gt\_sort.  
  
CLEAR ls\_sort.  
ls\_sort-fieldname = 'LIFNR'.  
ls\_sort-up = c\_x.  
APPEND ls\_sort TO gt\_sort.  
  
CLEAR ls\_sort.  
ls\_sort-fieldname = 'EBELN'.  
ls\_sort-up = c\_x.  
ls\_sort-subtot = c\_x. " show subtotals per PO  
APPEND ls\_sort TO gt\_sort.  
  
" Layout and variant  
CLEAR gs\_layout.  
gs\_layout-zebra = c\_x.  
gs\_layout-cwidth\_opt = c\_x.  
gs\_layout-sel\_mode = 'A'.  
gs\_layout-edit = c\_x.  
gs\_layout-totals\_text = 'Subtotal'.  
gs\_layout-grid\_title = |Open PO Items ({ lines( gt\_out ) })|.  
  
CLEAR gs\_variant.  
gs\_variant-report = c\_repid.  
gs\_variant-variant = p\_var.  
  
" Create container and grid  
IF go\_dock IS INITIAL.  
 CREATE OBJECT go\_dock  
 EXPORTING  
 ratio = 90  
 side = cl\_gui\_docking\_container=>dock\_at\_left.  
  
 CREATE OBJECT go\_grid  
 EXPORTING  
 i\_parent = go\_dock.  
  
 go\_grid->register\_edit\_event( cl\_gui\_alv\_grid=>mc\_evt\_modified ).  
 go\_grid->set\_toolbar\_interactive( ).  
  
 IF go\_events IS INITIAL.  
 CREATE OBJECT go\_events.  
 ENDIF.  
 SET HANDLER go\_events->handle\_toolbar FOR go\_grid.  
 SET HANDLER go\_events->handle\_user\_command FOR go\_grid.  
  
 go\_grid->set\_table\_for\_first\_display(  
 EXPORTING  
 is\_layout = gs\_layout  
 is\_variant = gs\_variant  
 i\_save = gv\_save  
 CHANGING  
 it\_outtab = gt\_out  
 it\_fieldcatalog = gt\_fieldcat  
 it\_sort = gt\_sort  
 it\_filter = gt\_filter ).  
  
 go\_grid->set\_ready\_for\_input( c\_x ).  
 cl\_gui\_cfw=>flush( ).  
ELSE.  
 ls\_stbl-row = c\_x.  
 ls\_stbl-col = c\_x.  
 go\_grid->refresh\_table\_display( EXPORTING is\_stable = ls\_stbl ).  
ENDIF.