# ABAP Code

## Global Declaration

TYPE-POOLS: lvc, icon.  
  
" Selection screen  
SELECTION-SCREEN BEGIN OF BLOCK b\_sel 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 b\_sel.  
  
SELECTION-SCREEN BEGIN OF BLOCK b\_lay WITH FRAME.  
 PARAMETERS p\_layv TYPE disvariant-variant.  
SELECTION-SCREEN END OF BLOCK b\_lay.  
  
" Constants  
CONSTANTS:  
 c\_fcode\_close TYPE syucomm VALUE 'CLOSE',  
 c\_gui\_status TYPE sypfkey VALUE 'ZMM\_ALV',  
 c\_title TYPE sytitle VALUE 'Utility to Close Open POs',  
 c\_true TYPE abap\_bool VALUE abap\_true,  
 c\_false TYPE abap\_bool VALUE abap\_false,  
 c\_icon\_close TYPE icon\_d VALUE icon\_complete.  
  
" Types  
TYPES: BEGIN OF ty\_ekpo\_key,  
 ebeln TYPE ekpo-ebeln,  
 ebelp TYPE ekpo-ebelp,  
 END OF ty\_ekpo\_key.  
  
TYPES: BEGIN OF ty\_alv\_row,  
 sel TYPE char1, "Checkbox for bulk close  
 ebeln TYPE ekpo-ebeln,  
 ebelp TYPE ekpo-ebelp,  
 matnr TYPE ekpo-matnr,  
 txz01 TYPE ekpo-txz01,  
 werks TYPE ekpo-werks,  
 lifnr TYPE ekko-lifnr,  
 name1 TYPE lfa1-name1,  
 mtart TYPE mara-mtart,  
 menge TYPE ekpo-menge,  
 meins TYPE ekpo-meins,  
 elikz TYPE ekpo-elikz,  
 bedat TYPE ekko-bedat,  
 ekgrp TYPE ekko-ekgrp,  
 aedat TYPE ekko-aedat,  
 END OF ty\_alv\_row.  
  
TYPES:  
 ty\_r\_matnr TYPE RANGE OF mara-matnr,  
 ty\_r\_ebeln TYPE RANGE OF ekko-ebeln.  
  
" Data - ALV  
DATA:  
 go\_dock TYPE REF TO cl\_gui\_docking\_container,  
 go\_cont TYPE REF TO cl\_gui\_custom\_container,  
 go\_grid TYPE REF TO cl\_gui\_alv\_grid,  
 go\_handler TYPE REF TO lcl\_event\_handler.  
  
DATA:  
 gt\_fcat TYPE lvc\_t\_fcat,  
 gs\_fcat TYPE lvc\_s\_fcat,  
 gs\_layout TYPE lvc\_s\_layo,  
 gt\_sort TYPE lvc\_t\_sort,  
 gs\_variant TYPE disvariant,  
 gv\_save TYPE c LENGTH 1,  
 gs\_stable TYPE lvc\_s\_stbl.  
  
DATA:  
 gt\_selected\_rows TYPE lvc\_t\_row,  
 gs\_selected\_row TYPE lvc\_s\_row.  
  
" Data - Application  
DATA:  
 gt\_alv TYPE STANDARD TABLE OF ty\_alv\_row WITH DEFAULT KEY,  
 gs\_alv TYPE ty\_alv\_row,  
 gt\_to\_close TYPE STANDARD TABLE OF ty\_ekpo\_key WITH EMPTY KEY.  
  
DATA:  
 gt\_r\_matnr TYPE ty\_r\_matnr,  
 gt\_r\_ebeln TYPE ty\_r\_ebeln.  
  
DATA:  
 gv\_okcode TYPE syucomm,  
 gv\_repid TYPE syrepid VALUE sy-repid,  
 gv\_init TYPE abap\_bool.  
  
FIELD-SYMBOLS:  
 <fs\_alv> TYPE ty\_alv\_row.  
  
" Local event handler (declaration only)  
CLASS lcl\_event\_handler DEFINITION.  
 PUBLIC SECTION.  
 METHODS on\_toolbar  
 FOR EVENT toolbar OF cl\_gui\_alv\_grid  
 IMPORTING e\_object e\_interactive.  
 METHODS on\_user\_command  
 FOR EVENT user\_command OF cl\_gui\_alv\_grid  
 IMPORTING e\_ucomm.  
 METHODS on\_after\_user\_command  
 FOR EVENT after\_user\_command OF cl\_gui\_alv\_grid  
 IMPORTING e\_ucomm.  
 METHODS on\_hotspot\_click  
 FOR EVENT hotspot\_click OF cl\_gui\_alv\_grid  
 IMPORTING e\_row\_id e\_column\_id es\_row\_no.  
ENDCLASS.

## Selection Screen

SELECTION-SCREEN BEGIN OF BLOCK b\_sel 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 b\_sel.  
  
SELECTION-SCREEN BEGIN OF BLOCK b\_lay WITH FRAME.  
 PARAMETERS:  
 p\_layv TYPE disvariant-variant,  
 p\_save TYPE c LENGTH 1 DEFAULT 'U'.  
SELECTION-SCREEN END OF BLOCK b\_lay.

## Processing Logic

INITIALIZATION.  
 gs\_variant-report = sy-repid.  
  
START-OF-SELECTION.  
 gv\_save = p\_save.  
 gs\_variant-variant = p\_layv.  
  
 PERFORM build\_data.  
 PERFORM display\_alv.  
  
FORM build\_data.  
 CLEAR gt\_alv.  
  
 SELECT  
 ekpo~ebeln,  
 ekpo~ebelp,  
 ekpo~matnr,  
 ekpo~txz01,  
 ekpo~werks,  
 ekko~lifnr,  
 lfa1~name1,  
 mara~mtart,  
 ekpo~menge,  
 ekpo~meins,  
 ekpo~elikz,  
 ekko~bedat,  
 ekko~ekgrp,  
 ekko~aedat  
 FROM ekpo  
 INNER JOIN ekko ON ekko~ebeln = ekpo~ebeln  
 LEFT OUTER JOIN mara ON mara~matnr = ekpo~matnr  
 LEFT OUTER JOIN lfa1 ON lfa1~lifnr = ekko~lifnr  
 INTO CORRESPONDING FIELDS OF TABLE @gt\_alv  
 WHERE ekpo~elikz = @space  
 AND ekpo~werks IN @s\_werks  
 AND ( @s\_matnr IS INITIAL OR ekpo~matnr IN @s\_matnr )  
 AND ( @s\_lifnr IS INITIAL OR ekko~lifnr IN @s\_lifnr )  
 AND ( @s\_mtart IS INITIAL OR mara~mtart IN @s\_mtart ).  
  
ENDFORM.  
  
FORM build\_fcat.  
 CLEAR gt\_fcat.  
  
 " Selection checkbox  
 CLEAR gs\_fcat.  
 gs\_fcat-fieldname = 'SEL'.  
 gs\_fcat-coltext = 'Sel'.  
 gs\_fcat-checkbox = abap\_true.  
 gs\_fcat-edit = abap\_true.  
 gs\_fcat-outputlen = 3.  
 APPEND gs\_fcat TO gt\_fcat.  
  
 " EBELN  
 CLEAR gs\_fcat.  
 gs\_fcat-fieldname = 'EBELN'.  
 gs\_fcat-ref\_table = 'EKPO'.  
 gs\_fcat-ref\_field = 'EBELN'.  
 gs\_fcat-coltext = 'PO'.  
 gs\_fcat-key = abap\_true.  
 gs\_fcat-hotspot = abap\_true.  
 APPEND gs\_fcat TO gt\_fcat.  
  
 " EBELP  
 CLEAR gs\_fcat.  
 gs\_fcat-fieldname = 'EBELP'.  
 gs\_fcat-ref\_table = 'EKPO'.  
 gs\_fcat-ref\_field = 'EBELP'.  
 gs\_fcat-coltext = 'Item'.  
 gs\_fcat-key = abap\_true.  
 APPEND gs\_fcat TO gt\_fcat.  
  
 " MATNR  
 CLEAR gs\_fcat.  
 gs\_fcat-fieldname = 'MATNR'.  
 gs\_fcat-ref\_table = 'EKPO'.  
 gs\_fcat-ref\_field = 'MATNR'.  
 gs\_fcat-coltext = 'Material'.  
 APPEND gs\_fcat TO gt\_fcat.  
  
 " TXZ01  
 CLEAR gs\_fcat.  
 gs\_fcat-fieldname = 'TXZ01'.  
 gs\_fcat-ref\_table = 'EKPO'.  
 gs\_fcat-ref\_field = 'TXZ01'.  
 gs\_fcat-coltext = 'Short Text'.  
 APPEND gs\_fcat TO gt\_fcat.  
  
 " WERKS  
 CLEAR gs\_fcat.  
 gs\_fcat-fieldname = 'WERKS'.  
 gs\_fcat-ref\_table = 'EKPO'.  
 gs\_fcat-ref\_field = 'WERKS'.  
 gs\_fcat-coltext = 'Plant'.  
 APPEND gs\_fcat TO gt\_fcat.  
  
 " LIFNR  
 CLEAR gs\_fcat.  
 gs\_fcat-fieldname = 'LIFNR'.  
 gs\_fcat-ref\_table = 'EKKO'.  
 gs\_fcat-ref\_field = 'LIFNR'.  
 gs\_fcat-coltext = 'Vendor'.  
 APPEND gs\_fcat TO gt\_fcat.  
  
 " NAME1  
 CLEAR gs\_fcat.  
 gs\_fcat-fieldname = 'NAME1'.  
 gs\_fcat-ref\_table = 'LFA1'.  
 gs\_fcat-ref\_field = 'NAME1'.  
 gs\_fcat-coltext = 'Vendor Name'.  
 gs\_fcat-outputlen = 25.  
 APPEND gs\_fcat TO gt\_fcat.  
  
 " MTART  
 CLEAR gs\_fcat.  
 gs\_fcat-fieldname = 'MTART'.  
 gs\_fcat-ref\_table = 'MARA'.  
 gs\_fcat-ref\_field = 'MTART'.  
 gs\_fcat-coltext = 'Mat. Type'.  
 APPEND gs\_fcat TO gt\_fcat.  
  
 " MENGE  
 CLEAR gs\_fcat.  
 gs\_fcat-fieldname = 'MENGE'.  
 gs\_fcat-ref\_table = 'EKPO'.  
 gs\_fcat-ref\_field = 'MENGE'.  
 gs\_fcat-coltext = 'Quantity'.  
 APPEND gs\_fcat TO gt\_fcat.  
  
 " MEINS  
 CLEAR gs\_fcat.  
 gs\_fcat-fieldname = 'MEINS'.  
 gs\_fcat-ref\_table = 'EKPO'.  
 gs\_fcat-ref\_field = 'MEINS'.  
 gs\_fcat-coltext = 'UoM'.  
 APPEND gs\_fcat TO gt\_fcat.  
  
 " ELIKZ  
 CLEAR gs\_fcat.  
 gs\_fcat-fieldname = 'ELIKZ'.  
 gs\_fcat-ref\_table = 'EKPO'.  
 gs\_fcat-ref\_field = 'ELIKZ'.  
 gs\_fcat-coltext = 'Closed'.  
 APPEND gs\_fcat TO gt\_fcat.  
  
 " BEDAT  
 CLEAR gs\_fcat.  
 gs\_fcat-fieldname = 'BEDAT'.  
 gs\_fcat-ref\_table = 'EKKO'.  
 gs\_fcat-ref\_field = 'BEDAT'.  
 gs\_fcat-coltext = 'PO Date'.  
 APPEND gs\_fcat TO gt\_fcat.  
  
 " EKGRP  
 CLEAR gs\_fcat.  
 gs\_fcat-fieldname = 'EKGRP'.  
 gs\_fcat-ref\_table = 'EKKO'.  
 gs\_fcat-ref\_field = 'EKGRP'.  
 gs\_fcat-coltext = 'Purch. Grp'.  
 APPEND gs\_fcat TO gt\_fcat.  
  
 " AEDAT  
 CLEAR gs\_fcat.  
 gs\_fcat-fieldname = 'AEDAT'.  
 gs\_fcat-ref\_table = 'EKKO'.  
 gs\_fcat-ref\_field = 'AEDAT'.  
 gs\_fcat-coltext = 'Changed On'.  
 APPEND gs\_fcat TO gt\_fcat.  
  
ENDFORM.  
  
FORM build\_sort.  
 CLEAR gt\_sort.  
  
 DATA ls\_sort TYPE lvc\_s\_sort.  
  
 CLEAR ls\_sort.  
 ls\_sort-fieldname = 'EBELN'.  
 ls\_sort-up = abap\_true.  
 APPEND ls\_sort TO gt\_sort.  
  
 CLEAR ls\_sort.  
 ls\_sort-fieldname = 'EBELP'.  
 ls\_sort-up = abap\_true.  
 APPEND ls\_sort TO gt\_sort.  
  
ENDFORM.  
  
FORM set\_layout.  
 CLEAR gs\_layout.  
 gs\_layout-cwidth\_opt = abap\_true.  
 gs\_layout-zebra = abap\_true.  
 gs\_layout-sel\_mode = 'A'.  
 gs\_layout-box\_fname = 'SEL'.  
ENDFORM.  
  
FORM display\_alv.  
 IF go\_dock IS INITIAL.  
 CREATE OBJECT go\_dock  
 EXPORTING  
 side = cl\_gui\_docking\_container=>dock\_at\_left  
 ratio = 90.  
  
 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\_hotspot\_click FOR go\_grid.  
 ENDIF.  
  
 PERFORM build\_fcat.  
 PERFORM build\_sort.  
 PERFORM set\_layout.  
  
 gs\_variant-report = sy-repid.  
 gs\_variant-variant = p\_layv.  
  
 CALL METHOD go\_grid->set\_table\_for\_first\_display  
 EXPORTING  
 is\_layout = gs\_layout  
 is\_variant = gs\_variant  
 i\_save = gv\_save  
 CHANGING  
 it\_outtab = gt\_alv  
 it\_fieldcatalog = gt\_fcat  
 it\_sort = gt\_sort.  
ENDFORM.  
  
FORM close\_selected.  
 DATA: lv\_success TYPE i VALUE 0,  
 lv\_failed TYPE i VALUE 0.  
  
 IF go\_grid IS BOUND.  
 go\_grid->check\_changed\_data( ).  
 ENDIF.  
  
 CLEAR gt\_to\_close.  
  
 LOOP AT gt\_alv ASSIGNING <fs\_alv> WHERE sel = 'X' AND elikz = space.  
 APPEND VALUE ty\_ekpo\_key( ebeln = <fs\_alv>-ebeln ebelp = <fs\_alv>-ebelp ) TO gt\_to\_close.  
 ENDLOOP.  
  
 " Fallback to row selection if checkbox not used  
 IF gt\_to\_close IS INITIAL AND go\_grid IS BOUND.  
 CLEAR gt\_selected\_rows.  
 go\_grid->get\_selected\_rows(  
 IMPORTING  
 et\_index\_rows = gt\_selected\_rows ).  
 LOOP AT gt\_selected\_rows INTO gs\_selected\_row.  
 READ TABLE gt\_alv ASSIGNING <fs\_alv> INDEX gs\_selected\_row-index.  
 IF sy-subrc = 0 AND <fs\_alv>-elikz = space.  
 APPEND VALUE ty\_ekpo\_key( ebeln = <fs\_alv>-ebeln ebelp = <fs\_alv>-ebelp ) TO gt\_to\_close.  
 ENDIF.  
 ENDLOOP.  
 ENDIF.  
  
 IF gt\_to\_close IS INITIAL.  
 MESSAGE 'No items selected for closing.' TYPE 'S'.  
 RETURN.  
 ENDIF.  
  
 LOOP AT gt\_to\_close ASSIGNING FIELD-SYMBOL(<ls\_key>).  
 UPDATE ekpo  
 SET elikz = 'X'  
 WHERE ebeln = @<ls\_key>-ebeln  
 AND ebelp = @<ls\_key>-ebelp  
 AND elikz = @space.  
 IF sy-subrc = 0.  
 ADD 1 TO lv\_success.  
 " Reflect change in ALV buffer  
 LOOP AT gt\_alv ASSIGNING <fs\_alv>  
 WHERE ebeln = <ls\_key>-ebeln  
 AND ebelp = <ls\_key>-ebelp.  
 <fs\_alv>-elikz = 'X'.  
 <fs\_alv>-sel = space.  
 ENDLOOP.  
 ELSE.  
 ADD 1 TO lv\_failed.  
 ENDIF.  
 ENDLOOP.  
  
 COMMIT WORK.  
  
 IF go\_grid IS BOUND.  
 gs\_stable-row = abap\_true.  
 gs\_stable-col = abap\_true.  
 go\_grid->refresh\_table\_display( EXPORTING is\_stable = gs\_stable ).  
 ENDIF.  
  
 MESSAGE |Closed: { lv\_success } item(s). Failed: { lv\_failed }.| TYPE 'S'.  
ENDFORM.  
  
CLASS lcl\_event\_handler IMPLEMENTATION.  
 METHOD on\_toolbar.  
 DATA ls\_toolbar TYPE stb\_button.  
  
 CLEAR ls\_toolbar.  
 ls\_toolbar-function = c\_fcode\_close.  
 ls\_toolbar-icon = c\_icon\_close.  
 ls\_toolbar-quickinfo = 'Close selected items'.  
 ls\_toolbar-text = 'Close'.  
 ls\_toolbar-butn\_type = 0.  
 APPEND ls\_toolbar TO e\_object->mt\_toolbar.  
 ENDMETHOD.  
  
 METHOD on\_user\_command.  
 CASE e\_ucomm.  
 WHEN c\_fcode\_close.  
 PERFORM close\_selected.  
 WHEN 'BACK' OR 'CANC' OR 'EXIT'.  
 LEAVE PROGRAM.  
 ENDCASE.  
 ENDMETHOD.  
  
 METHOD on\_after\_user\_command.  
 IF go\_grid IS BOUND.  
 gs\_stable-row = abap\_true.  
 gs\_stable-col = abap\_true.  
 go\_grid->refresh\_table\_display( EXPORTING is\_stable = gs\_stable ).  
 ENDIF.  
 ENDMETHOD.  
  
 METHOD on\_hotspot\_click.  
 READ TABLE gt\_alv INDEX e\_row\_id-index ASSIGNING <fs\_alv>.  
 IF sy-subrc = 0 AND e\_column\_id-fieldname = 'EBELN'.  
 IF <fs\_alv>-sel = 'X'.  
 <fs\_alv>-sel = space.  
 ELSE.  
 <fs\_alv>-sel = 'X'.  
 ENDIF.  
 IF go\_grid IS BOUND.  
 go\_grid->refresh\_table\_display( ).  
 ENDIF.  
 ENDIF.  
 ENDMETHOD.  
ENDCLASS.

## Output Display

FORM build\_fcat.  
 CLEAR gt\_fcat.  
  
 " Selection checkbox  
 CLEAR gs\_fcat.  
 gs\_fcat-fieldname = 'SEL'.  
 gs\_fcat-coltext = 'Sel'.  
 gs\_fcat-checkbox = abap\_true.  
 gs\_fcat-edit = abap\_true.  
 gs\_fcat-outputlen = 3.  
 APPEND gs\_fcat TO gt\_fcat.  
  
 " EBELN  
 CLEAR gs\_fcat.  
 gs\_fcat-fieldname = 'EBELN'.  
 gs\_fcat-ref\_table = 'EKPO'.  
 gs\_fcat-ref\_field = 'EBELN'.  
 gs\_fcat-coltext = 'PO'.  
 gs\_fcat-key = abap\_true.  
 gs\_fcat-hotspot = abap\_true.  
 APPEND gs\_fcat TO gt\_fcat.  
  
 " EBELP  
 CLEAR gs\_fcat.  
 gs\_fcat-fieldname = 'EBELP'.  
 gs\_fcat-ref\_table = 'EKPO'.  
 gs\_fcat-ref\_field = 'EBELP'.  
 gs\_fcat-coltext = 'Item'.  
 gs\_fcat-key = abap\_true.  
 APPEND gs\_fcat TO gt\_fcat.  
  
 " MATNR  
 CLEAR gs\_fcat.  
 gs\_fcat-fieldname = 'MATNR'.  
 gs\_fcat-ref\_table = 'EKPO'.  
 gs\_fcat-ref\_field = 'MATNR'.  
 gs\_fcat-coltext = 'Material'.  
 APPEND gs\_fcat TO gt\_fcat.  
  
 " TXZ01  
 CLEAR gs\_fcat.  
 gs\_fcat-fieldname = 'TXZ01'.  
 gs\_fcat-ref\_table = 'EKPO'.  
 gs\_fcat-ref\_field = 'TXZ01'.  
 gs\_fcat-coltext = 'Short Text'.  
 APPEND gs\_fcat TO gt\_fcat.  
  
 " WERKS  
 CLEAR gs\_fcat.  
 gs\_fcat-fieldname = 'WERKS'.  
 gs\_fcat-ref\_table = 'EKPO'.  
 gs\_fcat-ref\_field = 'WERKS'.  
 gs\_fcat-coltext = 'Plant'.  
 APPEND gs\_fcat TO gt\_fcat.  
  
 " LIFNR  
 CLEAR gs\_fcat.  
 gs\_fcat-fieldname = 'LIFNR'.  
 gs\_fcat-ref\_table = 'EKKO'.  
 gs\_fcat-ref\_field = 'LIFNR'.  
 gs\_fcat-coltext = 'Vendor'.  
 APPEND gs\_fcat TO gt\_fcat.  
  
 " NAME1  
 CLEAR gs\_fcat.  
 gs\_fcat-fieldname = 'NAME1'.  
 gs\_fcat-ref\_table = 'LFA1'.  
 gs\_fcat-ref\_field = 'NAME1'.  
 gs\_fcat-coltext = 'Vendor Name'.  
 gs\_fcat-outputlen = 25.  
 APPEND gs\_fcat TO gt\_fcat.  
  
 " MTART  
 CLEAR gs\_fcat.  
 gs\_fcat-fieldname = 'MTART'.  
 gs\_fcat-ref\_table = 'MARA'.  
 gs\_fcat-ref\_field = 'MTART'.  
 gs\_fcat-coltext = 'Mat. Type'.  
 APPEND gs\_fcat TO gt\_fcat.  
  
 " MENGE  
 CLEAR gs\_fcat.  
 gs\_fcat-fieldname = 'MENGE'.  
 gs\_fcat-ref\_table = 'EKPO'.  
 gs\_fcat-ref\_field = 'MENGE'.  
 gs\_fcat-coltext = 'Quantity'.  
 gs\_fcat-do\_sum = abap\_true.  
 APPEND gs\_fcat TO gt\_fcat.  
  
 " MEINS  
 CLEAR gs\_fcat.  
 gs\_fcat-fieldname = 'MEINS'.  
 gs\_fcat-ref\_table = 'EKPO'.  
 gs\_fcat-ref\_field = 'MEINS'.  
 gs\_fcat-coltext = 'UoM'.  
 APPEND gs\_fcat TO gt\_fcat.  
  
 " ELIKZ  
 CLEAR gs\_fcat.  
 gs\_fcat-fieldname = 'ELIKZ'.  
 gs\_fcat-ref\_table = 'EKPO'.  
 gs\_fcat-ref\_field = 'ELIKZ'.  
 gs\_fcat-coltext = 'Closed'.  
 APPEND gs\_fcat TO gt\_fcat.  
  
 " BEDAT  
 CLEAR gs\_fcat.  
 gs\_fcat-fieldname = 'BEDAT'.  
 gs\_fcat-ref\_table = 'EKKO'.  
 gs\_fcat-ref\_field = 'BEDAT'.  
 gs\_fcat-coltext = 'PO Date'.  
 APPEND gs\_fcat TO gt\_fcat.  
  
 " EKGRP  
 CLEAR gs\_fcat.  
 gs\_fcat-fieldname = 'EKGRP'.  
 gs\_fcat-ref\_table = 'EKKO'.  
 gs\_fcat-ref\_field = 'EKGRP'.  
 gs\_fcat-coltext = 'Purch. Grp'.  
 APPEND gs\_fcat TO gt\_fcat.  
  
 " AEDAT  
 CLEAR gs\_fcat.  
 gs\_fcat-fieldname = 'AEDAT'.  
 gs\_fcat-ref\_table = 'EKKO'.  
 gs\_fcat-ref\_field = 'AEDAT'.  
 gs\_fcat-coltext = 'Changed On'.  
 APPEND gs\_fcat TO gt\_fcat.  
  
ENDFORM.  
  
FORM build\_sort.  
 CLEAR gt\_sort.  
  
 DATA ls\_sort TYPE lvc\_s\_sort.  
  
 CLEAR ls\_sort.  
 ls\_sort-fieldname = 'EBELN'.  
 ls\_sort-up = abap\_true.  
 APPEND ls\_sort TO gt\_sort.  
  
 CLEAR ls\_sort.  
 ls\_sort-fieldname = 'EBELP'.  
 ls\_sort-up = abap\_true.  
 APPEND ls\_sort TO gt\_sort.  
  
ENDFORM.  
  
FORM set\_layout.  
 CLEAR gs\_layout.  
 gs\_layout-cwidth\_opt = abap\_true.  
 gs\_layout-zebra = abap\_true.  
 gs\_layout-sel\_mode = 'A'.  
 gs\_layout-box\_fname = 'SEL'.  
ENDFORM.  
  
FORM display\_alv.  
 IF go\_dock IS INITIAL.  
 CREATE OBJECT go\_dock  
 EXPORTING  
 side = cl\_gui\_docking\_container=>dock\_at\_left  
 ratio = 90.  
  
 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\_hotspot\_click FOR go\_grid.  
  
 go\_grid->register\_edit\_event( i\_event\_id = cl\_gui\_alv\_grid=>mc\_evt\_modified ).  
 ENDIF.  
  
 PERFORM build\_fcat.  
 PERFORM build\_sort.  
 PERFORM set\_layout.  
  
 gs\_variant-report = sy-repid.  
 gs\_variant-variant = p\_layv.  
  
 CALL METHOD go\_grid->set\_table\_for\_first\_display  
 EXPORTING  
 is\_layout = gs\_layout  
 is\_variant = gs\_variant  
 i\_save = gv\_save  
 CHANGING  
 it\_outtab = gt\_alv  
 it\_fieldcatalog = gt\_fcat  
 it\_sort = gt\_sort.  
ENDFORM.  
  
FORM close\_selected.  
 DATA: lv\_success TYPE i VALUE 0,  
 lv\_failed TYPE i VALUE 0.  
  
 IF go\_grid IS BOUND.  
 go\_grid->check\_changed\_data( ).  
 ENDIF.  
  
 CLEAR gt\_to\_close.  
  
 LOOP AT gt\_alv ASSIGNING <fs\_alv> WHERE sel = 'X' AND elikz = space.  
 APPEND VALUE ty\_ekpo\_key( ebeln = <fs\_alv>-ebeln ebelp = <fs\_alv>-ebelp ) TO gt\_to\_close.  
 ENDLOOP.  
  
 IF gt\_to\_close IS INITIAL AND go\_grid IS BOUND.  
 CLEAR gt\_selected\_rows.  
 go\_grid->get\_selected\_rows(  
 IMPORTING  
 et\_index\_rows = gt\_selected\_rows ).  
 LOOP AT gt\_selected\_rows INTO gs\_selected\_row.  
 READ TABLE gt\_alv ASSIGNING <fs\_alv> INDEX gs\_selected\_row-index.  
 IF sy-subrc = 0 AND <fs\_alv>-elikz = space.  
 APPEND VALUE ty\_ekpo\_key( ebeln = <fs\_alv>-ebeln ebelp = <fs\_alv>-ebelp ) TO gt\_to\_close.  
 ENDIF.  
 ENDLOOP.  
 ENDIF.  
  
 IF gt\_to\_close IS INITIAL.  
 MESSAGE 'No items selected for closing.' TYPE 'S'.  
 RETURN.  
 ENDIF.  
  
 LOOP AT gt\_to\_close ASSIGNING FIELD-SYMBOL(<ls\_key>).  
 UPDATE ekpo  
 SET elikz = 'X'  
 WHERE ebeln = @<ls\_key>-ebeln  
 AND ebelp = @<ls\_key>-ebelp  
 AND elikz = @space.  
 IF sy-subrc = 0.  
 ADD 1 TO lv\_success.  
 LOOP AT gt\_alv ASSIGNING <fs\_alv>  
 WHERE ebeln = <ls\_key>-ebeln  
 AND ebelp = <ls\_key>-ebelp.  
 <fs\_alv>-elikz = 'X'.  
 <fs\_alv>-sel = space.  
 ENDLOOP.  
 ELSE.  
 ADD 1 TO lv\_failed.  
 ENDIF.  
 ENDLOOP.  
  
 COMMIT WORK.  
  
 IF go\_grid IS BOUND.  
 gs\_stable-row = abap\_true.  
 gs\_stable-col = abap\_true.  
 go\_grid->refresh\_table\_display( EXPORTING is\_stable = gs\_stable ).  
 ENDIF.  
  
 MESSAGE |Closed: { lv\_success } item(s). Failed: { lv\_failed }.| TYPE 'S'.  
ENDFORM.  
  
CLASS lcl\_event\_handler IMPLEMENTATION.  
 METHOD on\_toolbar.  
 DATA ls\_toolbar TYPE stb\_button.  
  
 CLEAR ls\_toolbar.  
 ls\_toolbar-function = c\_fcode\_close.  
 ls\_toolbar-icon = c\_icon\_close.  
 ls\_toolbar-quickinfo = 'Close selected items'.  
 ls\_toolbar-text = 'Close'.  
 ls\_toolbar-butn\_type = 0.  
 APPEND ls\_toolbar TO e\_object->mt\_toolbar.  
 ENDMETHOD.  
  
 METHOD on\_user\_command.  
 CASE e\_ucomm.  
 WHEN c\_fcode\_close.  
 PERFORM close\_selected.  
 WHEN 'BACK' OR 'CANC' OR 'EXIT'.  
 LEAVE PROGRAM.  
 ENDCASE.  
 ENDMETHOD.  
  
 METHOD on\_after\_user\_command.  
 IF go\_grid IS BOUND.  
 gs\_stable-row = abap\_true.  
 gs\_stable-col = abap\_true.  
 go\_grid->refresh\_table\_display( EXPORTING is\_stable = gs\_stable ).  
 ENDIF.  
 ENDMETHOD.  
  
 METHOD on\_hotspot\_click.  
 READ TABLE gt\_alv INDEX e\_row\_id-index ASSIGNING <fs\_alv>.  
 IF sy-subrc = 0 AND e\_column\_id-fieldname = 'EBELN'.  
 IF <fs\_alv>-sel = 'X'.  
 <fs\_alv>-sel = space.  
 ELSE.  
 <fs\_alv>-sel = 'X'.  
 ENDIF.  
 IF go\_grid IS BOUND.  
 go\_grid->refresh\_table\_display( ).  
 ENDIF.  
 ENDIF.  
 ENDMETHOD.  
ENDCLASS.