# ABAP Code

## Global Declaration

TYPE-POOLS: abap, icon, lvc.  
  
" Selection Screen  
SELECT-OPTIONS:  
 s\_matnr FOR mara-matnr,  
 s\_werks FOR ekpo-werks OBLIGATORY,  
 s\_lifnr FOR ekko-lifnr,  
 s\_mtart FOR mara-mtart.  
  
" Constants  
CONSTANTS:  
 c\_ucomm\_close TYPE syucomm VALUE 'CLOSE',  
 c\_ucomm\_refresh TYPE syucomm VALUE 'REFR',  
 c\_pf\_status TYPE sypfkey VALUE 'ZMM\_PO\_CLOSE',  
 c\_title TYPE sytitle VALUE 'Utility to close open POs',  
 c\_icon\_close TYPE icon\_d VALUE icon\_locked,  
 c\_box\_field TYPE lvc\_fname VALUE 'SEL'.  
  
" Types  
TYPES: BEGIN OF ty\_item\_key,  
 ebeln TYPE ebeln,  
 ebelp TYPE ebelp,  
 END OF ty\_item\_key.  
  
TYPES: BEGIN OF ty\_output,  
 sel TYPE abap\_bool, " Checkbox for mass action  
 ebeln TYPE ebeln,  
 ebelp TYPE ebelp,  
 bukrs TYPE ekko-bukrs,  
 ekorg TYPE ekko-ekorg,  
 ekgrp TYPE ekko-ekgrp,  
 lifnr TYPE ekko-lifnr,  
 name1 TYPE lfa1-name1,  
 bedat TYPE ekko-bedat,  
 matnr TYPE mara-matnr,  
 mtart TYPE mara-mtart,  
 werks TYPE ekpo-werks,  
 menge TYPE ekpo-menge,  
 wemng TYPE ekpo-wemng,  
 open\_qty TYPE ekpo-menge,  
 meins TYPE ekpo-meins,  
 eindt TYPE eket-eindt,  
 elikz TYPE ekpo-elikz,  
 msg TYPE string,  
 END OF ty\_output.  
  
" Global Data  
DATA:  
 g\_repid TYPE syrepid VALUE sy-repid,  
 ok\_code TYPE syucomm,  
 save\_ok TYPE syucomm,  
 g\_save TYPE char1 VALUE 'A',  
 g\_is\_initialized TYPE abap\_bool.  
  
" Data buffers for selections derived from inputs  
DATA:  
 gt\_matnr TYPE SORTED TABLE OF mara-matnr WITH UNIQUE KEY table\_line,  
 gt\_ebeln TYPE SORTED TABLE OF ekko-ebeln WITH UNIQUE KEY table\_line.  
  
" Output table  
DATA:  
 gt\_out TYPE STANDARD TABLE OF ty\_output WITH DEFAULT KEY,  
 gs\_out TYPE ty\_output,  
 gt\_to\_close TYPE STANDARD TABLE OF ty\_item\_key WITH DEFAULT KEY.  
  
" ALV Objects and settings  
DATA:  
 gr\_dock TYPE REF TO cl\_gui\_docking\_container,  
 gr\_grid TYPE REF TO cl\_gui\_alv\_grid,  
 gt\_fcat TYPE lvc\_t\_fcat,  
 gs\_fcat TYPE lvc\_s\_fcat,  
 gs\_layout TYPE lvc\_s\_layo,  
 gt\_sort TYPE lvc\_t\_sort,  
 gt\_filter TYPE lvc\_t\_filt,  
 gs\_variant TYPE disvariant,  
 gt\_rows TYPE lvc\_t\_row,  
 gt\_row\_no TYPE lvc\_t\_roid.  
  
" Event Handler  
CLASS lcl\_event\_handler DEFINITION DEFERRED.  
DATA go\_events TYPE REF TO lcl\_event\_handler.  
  
CLASS lcl\_event\_handler DEFINITION PUBLIC FINAL.  
 PUBLIC SECTION.  
 METHODS:  
 handle\_toolbar  
 FOR EVENT toolbar OF cl\_gui\_alv\_grid  
 IMPORTING e\_object e\_interactive,  
 handle\_user\_command  
 FOR EVENT user\_command OF cl\_gui\_alv\_grid  
 IMPORTING e\_ucomm,  
 handle\_double\_click  
 FOR EVENT double\_click OF cl\_gui\_alv\_grid  
 IMPORTING e\_row e\_column es\_row\_no.  
ENDCLASS.

## 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-matart.  
SELECTION-SCREEN END OF BLOCK b1.

## Processing Logic

INITIALIZATION.  
 g\_is\_initialized = abap\_false.  
  
AT SELECTION-SCREEN.  
 " Additional validation if needed (S\_WERKS is already obligatory)  
  
START-OF-SELECTION.  
 PERFORM get\_data.  
 PERFORM build\_alv.  
  
END-OF-SELECTION.  
 PERFORM refresh\_grid.  
  
"======================================================================  
" Local forms  
"======================================================================  
  
FORM get\_data.  
 TYPES: BEGIN OF ty\_base,  
 ebeln TYPE ebeln,  
 ebelp TYPE ebelp,  
 bukrs TYPE ekko-bukrs,  
 ekorg TYPE ekko-ekorg,  
 ekgrp TYPE ekko-ekgrp,  
 lifnr TYPE ekko-lifnr,  
 name1 TYPE lfa1-name1,  
 bedat TYPE ekko-bedat,  
 matnr TYPE mara-matnr,  
 mtart TYPE mara-mtart,  
 werks TYPE ekpo-werks,  
 menge TYPE ekpo-menge,  
 meins TYPE ekpo-meins,  
 elikz TYPE ekpo-elikz,  
 END OF ty\_base.  
  
 DATA: lt\_base TYPE STANDARD TABLE OF ty\_base WITH DEFAULT KEY,  
 ls\_base TYPE ty\_base,  
 lt\_keys TYPE STANDARD TABLE OF ty\_item\_key WITH DEFAULT KEY,  
 lt\_eket TYPE STANDARD TABLE OF eket,  
 lt\_wemng TYPE HASHED TABLE OF eket  
 WITH UNIQUE KEY ebeln ebelp,  
 ls\_eket TYPE eket,  
 lt\_sum TYPE SORTED TABLE OF eket  
 WITH UNIQUE KEY ebeln ebelp,  
 ls\_sum TYPE eket.  
  
 CLEAR: gt\_out.  
  
 " Derive materials if MTART provided (as per requirement)  
 CLEAR gt\_matnr.  
 IF s\_mtart[] IS NOT INITIAL.  
 SELECT matnr  
 FROM mara  
 INTO TABLE @gt\_matnr  
 WHERE mtart IN @s\_mtart.  
 ENDIF.  
  
 " Derive PO numbers if LIFNR provided (as per requirement)  
 CLEAR gt\_ebeln.  
 IF s\_lifnr[] IS NOT INITIAL.  
 SELECT ebeln  
 FROM ekko  
 INTO TABLE @gt\_ebeln  
 WHERE lifnr IN @s\_lifnr.  
 ENDIF.  
  
 " Base selection from EKPO + EKKO + MARA + LFA1 (keep it broad, refine later)  
 SELECT a~ebeln,  
 a~ebelp,  
 b~bukrs,  
 b~ekorg,  
 b~ekgrp,  
 b~lifnr,  
 d~name1,  
 b~bedat,  
 a~matnr,  
 c~mtart,  
 a~werks,  
 a~menge,  
 a~meins,  
 a~elikz  
 FROM ekpo AS a  
 INNER JOIN ekko AS b ON a~ebeln = b~ebeln  
 LEFT JOIN mara AS c ON a~matnr = c~matnr  
 LEFT JOIN lfa1 AS d ON b~lifnr = d~lifnr  
 INTO TABLE @lt\_base  
 WHERE a~werks IN @s\_werks  
 AND a~elikz = @space.  
  
 IF lt\_base IS INITIAL.  
 RETURN.  
 ENDIF.  
  
 " Apply selection-screen filters (material, vendor, material type, and vendor-derived POs)  
 IF s\_matnr[] IS NOT INITIAL.  
 DELETE lt\_base WHERE matnr NOT IN s\_matnr.  
 ENDIF.  
  
 IF s\_lifnr[] IS NOT INITIAL.  
 DELETE lt\_base WHERE lifnr NOT IN s\_lifnr.  
 ENDIF.  
  
 IF s\_mtart[] IS NOT INITIAL.  
 " If MTART provided but MARA join returned initial (e.g., service lines), filter them out  
 DELETE lt\_base WHERE mtart NOT IN s\_mtart.  
 ENDIF.  
  
 IF gt\_matnr[] IS NOT INITIAL.  
 " Further constrain by derived materials (intersection)  
 DELETE lt\_base WHERE matnr IS NOT INITIAL AND matnr NOT IN gt\_matnr.  
 ENDIF.  
  
 IF gt\_ebeln[] IS NOT INITIAL.  
 " Further constrain by derived POs from vendor (intersection)  
 DELETE lt\_base WHERE ebeln NOT IN gt\_ebeln.  
 ENDIF.  
  
 IF lt\_base IS INITIAL.  
 RETURN.  
 ENDIF.  
  
 " Collect keys for EKET aggregation  
 CLEAR lt\_keys.  
 lt\_keys = CORRESPONDING #( lt\_base MAPPING ebeln = ebeln ebelp = ebelp ).  
  
 " Aggregate delivered quantity (WEMNG) and a representative delivery date (take MIN(EINDT))  
 DATA: lt\_agg TYPE STANDARD TABLE OF  
 BEGIN OF ty\_agg,  
 ebeln TYPE ebeln,  
 ebelp TYPE ebelp,  
 wemng TYPE eket-wemng,  
 eindt TYPE eket-eindt,  
 END OF ty\_agg,  
 ls\_agg TYPE ty\_agg.  
  
 IF lt\_keys IS NOT INITIAL.  
 SELECT ebeln,  
 ebelp,  
 SUM( wemng ) AS wemng,  
 MIN( eindt ) AS eindt  
 FROM eket  
 INTO TABLE @lt\_agg  
 FOR ALL ENTRIES IN @lt\_keys  
 WHERE ebeln = @lt\_keys-ebeln  
 AND ebelp = @lt\_keys-ebelp  
 GROUP BY ebeln, ebelp.  
 ENDIF.  
  
 " Build output  
 LOOP AT lt\_base INTO ls\_base.  
 CLEAR gs\_out.  
 gs\_out-sel = abap\_false.  
 gs\_out-ebeln = ls\_base-ebeln.  
 gs\_out-ebelp = ls\_base-ebelp.  
 gs\_out-bukrs = ls\_base-bukrs.  
 gs\_out-ekorg = ls\_base-ekorg.  
 gs\_out-ekgrp = ls\_base-ekgrp.  
 gs\_out-lifnr = ls\_base-lifnr.  
 gs\_out-name1 = ls\_base-name1.  
 gs\_out-bedat = ls\_base-bedat.  
 gs\_out-matnr = ls\_base-matnr.  
 gs\_out-mtart = ls\_base-mtart.  
 gs\_out-werks = ls\_base-werks.  
 gs\_out-menge = ls\_base-menge.  
 gs\_out-meins = ls\_base-meins.  
 gs\_out-elikz = ls\_base-elikz.  
  
 READ TABLE lt\_agg INTO ls\_agg WITH KEY ebeln = ls\_base-ebeln ebelp = ls\_base-ebelp.  
 IF sy-subrc = 0.  
 gs\_out-wemng = ls\_agg-wemng.  
 gs\_out-eindt = ls\_agg-eindt.  
 ELSE.  
 gs\_out-wemng = 0.  
 CLEAR gs\_out-eindt.  
 ENDIF.  
  
 gs\_out-open\_qty = gs\_out-menge - gs\_out-wemng.  
  
 APPEND gs\_out TO gt\_out.  
 ENDLOOP.  
  
ENDFORM.  
  
FORM build\_alv.  
 IF gr\_grid IS BOUND.  
 RETURN.  
 ENDIF.  
  
 " Create docking container and ALV grid  
 CREATE OBJECT gr\_dock  
 EXPORTING  
 repid = g\_repid  
 dynnr = sy-dynnr  
 side = cl\_gui\_docking\_container=>dock\_at\_left  
 extension = 400.  
  
 CREATE OBJECT gr\_grid  
 EXPORTING  
 i\_parent = gr\_dock.  
  
 " Event handler  
 CREATE OBJECT go\_events.  
 SET HANDLER go\_events->handle\_toolbar FOR gr\_grid.  
 SET HANDLER go\_events->handle\_user\_command FOR gr\_grid.  
 SET HANDLER go\_events->handle\_double\_click FOR gr\_grid.  
  
 " Field catalog  
 PERFORM build\_fcat.  
  
 " Layout and variant  
 CLEAR gs\_layout.  
 gs\_layout-zebra = abap\_true.  
 gs\_layout-cwidth\_opt = abap\_true.  
 gs\_layout-box\_fname = c\_box\_field.  
  
 CLEAR gs\_variant.  
 gs\_variant-report = g\_repid.  
  
 TRY.  
 CALL METHOD gr\_grid->set\_table\_for\_first\_display  
 EXPORTING  
 is\_layout = gs\_layout  
 i\_save = g\_save  
 is\_variant = gs\_variant  
 CHANGING  
 it\_fieldcatalog = gt\_fcat  
 it\_outtab = gt\_out.  
 CATCH cx\_salv\_msg.  
 " Ignore ALV exceptions to avoid dumps  
 ENDTRY.  
  
 g\_is\_initialized = abap\_true.  
  
ENDFORM.  
  
FORM build\_fcat.  
 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.  
 APPEND gs\_fcat TO gt\_fcat.  
 END-OF-DEFINITION.  
  
 " Selection Checkbox  
 add\_col c\_box\_field 'Select' 6 space abap\_true abap\_true.  
  
 add\_col 'EBELN' 'PO Number' 10 'X' space space.  
 add\_col 'EBELP' 'Item' 5 'X' space space.  
 add\_col 'BUKRS' 'Company Code' 8 space space space.  
 add\_col 'EKORG' 'Purch Org' 8 space space space.  
 add\_col 'EKGRP' 'Purch Group' 8 space space space.  
 add\_col 'LIFNR' 'Vendor' 10 space space space.  
 add\_col 'NAME1' 'Vendor Name' 20 space space space.  
 add\_col 'BEDAT' 'Doc Date' 10 space space space.  
 add\_col 'MATNR' 'Material' 18 space space space.  
 add\_col 'MTART' 'Mat. Type' 8 space space space.  
 add\_col 'WERKS' 'Plant' 8 space space space.  
 add\_col 'MENGE' 'PO Qty' 13 space space space.  
 add\_col 'WEMNG' 'Delivered' 13 space space space.  
 add\_col 'OPEN\_QTY' 'To Be Deliv.' 13 space space space.  
 add\_col 'MEINS' 'Unit' 6 space space space.  
 add\_col 'EINDT' 'Deliv. Date' 10 space space space.  
 add\_col 'ELIKZ' 'Closed' 6 space space space.  
 add\_col 'MSG' 'Message' 30 space space space.  
  
ENDFORM.  
  
FORM refresh\_grid.  
 IF gr\_grid IS BOUND.  
 TRY.  
 CALL METHOD gr\_grid->refresh\_table\_display  
 EXPORTING  
 is\_stable = VALUE lvc\_s\_stbl( row = abap\_true col = abap\_true ).  
 CATCH cx\_salv\_msg.  
 " ignore  
 ENDTRY.  
 ENDIF.  
ENDFORM.  
  
FORM close\_selected.  
 FIELD-SYMBOLS: <ls\_out> TYPE ty\_output.  
 DATA: lt\_close TYPE STANDARD TABLE OF ty\_item\_key WITH DEFAULT KEY,  
 ls\_key TYPE ty\_item\_key.  
  
 IF gr\_grid IS BOUND.  
 CALL METHOD gr\_grid->check\_changed\_data.  
 ENDIF.  
  
 CLEAR lt\_close.  
 LOOP AT gt\_out ASSIGNING <ls\_out> WHERE sel = abap\_true OR sel = 'X'.  
 ls\_key-ebeln = <ls\_out>-ebeln.  
 ls\_key-ebelp = <ls\_out>-ebelp.  
 APPEND ls\_key TO lt\_close.  
 ENDLOOP.  
  
 IF lt\_close IS INITIAL.  
 RETURN.  
 ENDIF.  
  
 LOOP AT lt\_close INTO ls\_key.  
 UPDATE ekpo  
 SET elikz = 'X'  
 WHERE ebeln = @ls\_key-ebeln  
 AND ebelp = @ls\_key-ebelp  
 AND elikz = @space.  
  
 READ TABLE gt\_out ASSIGNING <ls\_out> WITH KEY ebeln = ls\_key-ebeln ebelp = ls\_key-ebelp.  
 IF sy-subrc = 0.  
 IF sy-dbcnt > 0 AND sy-subrc = 0.  
 <ls\_out>-msg = |Closed|.  
 <ls\_out>-elikz = 'X'.  
 ELSE.  
 <ls\_out>-msg = |No update|.  
 ENDIF.  
 ENDIF.  
 ENDLOOP.  
  
 COMMIT WORK AND WAIT.  
  
 " Refresh data after update  
 PERFORM get\_data.  
 PERFORM refresh\_grid.  
  
ENDFORM.  
  
FORM on\_refresh.  
 PERFORM get\_data.  
 PERFORM refresh\_grid.  
ENDFORM.  
  
"======================================================================  
" Event Handler Implementation  
"======================================================================  
CLASS lcl\_event\_handler IMPLEMENTATION.  
  
 METHOD handle\_toolbar.  
 DATA ls\_button TYPE stb\_button.  
  
 CLEAR ls\_button.  
 ls\_button-function = c\_ucomm\_close.  
 ls\_button-text = 'Close Selected'.  
 ls\_button-quickinfo = 'Set Delivery Completed (ELIKZ) for selected items'.  
 ls\_button-icon = c\_icon\_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-text = 'Refresh'.  
 ls\_button-quickinfo = 'Refresh Data'.  
 ls\_button-butn\_type = 0.  
 APPEND ls\_button TO e\_object->mt\_toolbar.  
 ENDMETHOD.  
  
 METHOD handle\_user\_command.  
 CASE e\_ucomm.  
 WHEN c\_ucomm\_close.  
 PERFORM close\_selected.  
 WHEN c\_ucomm\_refresh.  
 PERFORM on\_refresh.  
 WHEN OTHERS.  
 " do nothing  
 ENDCASE.  
 ENDMETHOD.  
  
 METHOD handle\_double\_click.  
 " Toggle selection checkbox on double-click  
 DATA ls\_row TYPE lvc\_s\_row.  
 ls\_row = e\_row.  
 FIELD-SYMBOLS <ls\_out> TYPE ty\_output.  
 READ TABLE gt\_out ASSIGNING <ls\_out> INDEX ls\_row-index.  
 IF sy-subrc = 0.  
 <ls\_out>-sel = COND abap\_bool( WHEN <ls\_out>-sel = abap\_true THEN abap\_false ELSE abap\_true ).  
 IF gr\_grid IS BOUND.  
 CALL METHOD gr\_grid->refresh\_table\_display.  
 ENDIF.  
 ENDIF.  
 ENDMETHOD.  
  
ENDCLASS.

## Output Display

" ALV Objects and settings  
DATA:  
 gr\_dock TYPE REF TO cl\_gui\_docking\_container,  
 gr\_grid TYPE REF TO cl\_gui\_alv\_grid,  
 gt\_fcat TYPE lvc\_t\_fcat,  
 gs\_fcat TYPE lvc\_s\_fcat,  
 gs\_layout TYPE lvc\_s\_layo,  
 gt\_sort TYPE lvc\_t\_sort,  
 gt\_filter TYPE lvc\_t\_filt,  
 gs\_variant TYPE disvariant,  
 gt\_rows TYPE lvc\_t\_row,  
 gt\_row\_no TYPE lvc\_t\_roid.  
  
" Event Handler  
CLASS lcl\_event\_handler DEFINITION DEFERRED.  
DATA go\_events TYPE REF TO lcl\_event\_handler.  
  
CLASS lcl\_event\_handler DEFINITION PUBLIC FINAL.  
 PUBLIC SECTION.  
 METHODS:  
 handle\_toolbar  
 FOR EVENT toolbar OF cl\_gui\_alv\_grid  
 IMPORTING e\_object e\_interactive,  
 handle\_user\_command  
 FOR EVENT user\_command OF cl\_gui\_alv\_grid  
 IMPORTING e\_ucomm,  
 handle\_double\_click  
 FOR EVENT double\_click OF cl\_gui\_alv\_grid  
 IMPORTING e\_row e\_column es\_row\_no.  
ENDCLASS.  
  
FORM build\_alv.  
 IF gr\_grid IS BOUND.  
 TRY.  
 CALL METHOD gr\_grid->set\_frontend\_layout  
 EXPORTING  
 is\_layout = VALUE lvc\_s\_layo( zebra = abap\_true cwidth\_opt = abap\_true box\_fname = c\_box\_field ).  
 CATCH cx\_salv\_msg.  
 ENDTRY.  
 RETURN.  
 ENDIF.  
  
 SET TITLEBAR c\_title.  
 SET PF-STATUS c\_pf\_status.  
  
 CREATE OBJECT gr\_dock  
 EXPORTING  
 repid = g\_repid  
 dynnr = sy-dynnr  
 side = cl\_gui\_docking\_container=>dock\_at\_left  
 extension = 400.  
  
 CREATE OBJECT gr\_grid  
 EXPORTING  
 i\_parent = gr\_dock.  
  
 " Event handler  
 CREATE OBJECT go\_events.  
 SET HANDLER go\_events->handle\_toolbar FOR gr\_grid.  
 SET HANDLER go\_events->handle\_user\_command FOR gr\_grid.  
 SET HANDLER go\_events->handle\_double\_click FOR gr\_grid.  
  
 " Field catalog  
 PERFORM build\_fcat.  
  
 " Layout and variant  
 CLEAR gs\_layout.  
 gs\_layout-zebra = abap\_true.  
 gs\_layout-cwidth\_opt = abap\_true.  
 gs\_layout-box\_fname = c\_box\_field.  
 gs\_layout-sel\_mode = 'A'. " Multiple selection in ALV  
  
 CLEAR gs\_variant.  
 gs\_variant-report = g\_repid.  
  
 TRY.  
 CALL METHOD gr\_grid->set\_table\_for\_first\_display  
 EXPORTING  
 is\_layout = gs\_layout  
 i\_save = g\_save  
 is\_variant = gs\_variant  
 CHANGING  
 it\_fieldcatalog = gt\_fcat  
 it\_outtab = gt\_out.  
 CATCH cx\_salv\_msg.  
 ENDTRY.  
ENDFORM.  
  
FORM build\_fcat.  
 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.  
 APPEND gs\_fcat TO gt\_fcat.  
 END-OF-DEFINITION.  
  
 " Selection Checkbox  
 add\_col c\_box\_field 'Select' 6 space abap\_true abap\_true.  
  
 add\_col 'EBELN' 'PO Number' 10 'X' space space.  
 add\_col 'EBELP' 'Item' 5 'X' space space.  
 add\_col 'BUKRS' 'Company Code' 8 space space space.  
 add\_col 'EKORG' 'Purch Org' 8 space space space.  
 add\_col 'EKGRP' 'Purch Group' 8 space space space.  
 add\_col 'LIFNR' 'Vendor' 10 space space space.  
 add\_col 'NAME1' 'Vendor Name' 20 space space space.  
 add\_col 'BEDAT' 'Doc Date' 10 space space space.  
 add\_col 'MATNR' 'Material' 18 space space space.  
 add\_col 'MTART' 'Mat. Type' 8 space space space.  
 add\_col 'WERKS' 'Plant' 8 space space space.  
  
 " Quantities with summation  
 CLEAR gs\_fcat.  
 gs\_fcat-fieldname = 'MENGE'.  
 gs\_fcat-coltext = 'PO Qty'.  
 gs\_fcat-outputlen = 13.  
 gs\_fcat-do\_sum = abap\_true.  
 APPEND gs\_fcat TO gt\_fcat.  
  
 CLEAR gs\_fcat.  
 gs\_fcat-fieldname = 'WEMNG'.  
 gs\_fcat-coltext = 'Delivered'.  
 gs\_fcat-outputlen = 13.  
 gs\_fcat-do\_sum = abap\_true.  
 APPEND gs\_fcat TO gt\_fcat.  
  
 CLEAR gs\_fcat.  
 gs\_fcat-fieldname = 'OPEN\_QTY'.  
 gs\_fcat-coltext = 'To Be Deliv.'.  
 gs\_fcat-outputlen = 13.  
 gs\_fcat-do\_sum = abap\_true.  
 APPEND gs\_fcat TO gt\_fcat.  
  
 add\_col 'MEINS' 'Unit' 6 space space space.  
 add\_col 'EINDT' 'Deliv. Date' 10 space space space.  
 add\_col 'ELIKZ' 'Closed' 6 space space space.  
 add\_col 'MSG' 'Message' 30 space space space.  
ENDFORM.  
  
FORM refresh\_grid.  
 IF gr\_grid IS BOUND.  
 TRY.  
 CALL METHOD gr\_grid->refresh\_table\_display  
 EXPORTING  
 is\_stable = VALUE lvc\_s\_stbl( row = abap\_true col = abap\_true ).  
 CATCH cx\_salv\_msg.  
 ENDTRY.  
 ENDIF.  
ENDFORM.  
  
CLASS lcl\_event\_handler IMPLEMENTATION.  
 METHOD handle\_toolbar.  
 DATA ls\_button TYPE stb\_button.  
  
 CLEAR ls\_button.  
 ls\_button-function = c\_ucomm\_close.  
 ls\_button-text = 'Close Selected'.  
 ls\_button-quickinfo = 'Set Delivery Completed (ELIKZ) for selected items'.  
 ls\_button-icon = c\_icon\_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-text = 'Refresh'.  
 ls\_button-quickinfo = 'Refresh Data'.  
 ls\_button-butn\_type = 0.  
 APPEND ls\_button TO e\_object->mt\_toolbar.  
 ENDMETHOD.  
  
 METHOD handle\_user\_command.  
 CASE e\_ucomm.  
 WHEN c\_ucomm\_close.  
 PERFORM close\_selected.  
 WHEN c\_ucomm\_refresh.  
 PERFORM on\_refresh.  
 WHEN OTHERS.  
 ENDCASE.  
 ENDMETHOD.  
  
 METHOD handle\_double\_click.  
 DATA ls\_row TYPE lvc\_s\_row.  
 ls\_row = e\_row.  
 FIELD-SYMBOLS <ls\_out> TYPE ty\_output.  
 READ TABLE gt\_out ASSIGNING <ls\_out> INDEX ls\_row-index.  
 IF sy-subrc = 0.  
 <ls\_out>-sel = COND abap\_bool( WHEN <ls\_out>-sel = abap\_true THEN abap\_false ELSE abap\_true ).  
 IF gr\_grid IS BOUND.  
 TRY.  
 CALL METHOD gr\_grid->refresh\_table\_display  
 EXPORTING  
 is\_stable = VALUE lvc\_s\_stbl( row = abap\_true col = abap\_true ).  
 CATCH cx\_salv\_msg.  
 ENDTRY.  
 ENDIF.  
 ENDIF.  
 ENDMETHOD.  
ENDCLASS.