

# ALV minimum steps

14 September 2025 15:03

```
ABAP Editor: Change Report ZALV_IN_MINIMUM_LINE
Report ZALV_IN_MINIMUM_LINE Active
1
2 REPORT zalv_in_minimum_line.
3
4 DATA: lt_flight TYPE TABLE OF sflight,
5       o_grid TYPE REF TO cl_gui_alv_grid.
6
7 SELECT * FROM sflight INTO TABLE lt_flight.
8
9 CREATE OBJECT o_grid EXPORTING i_parent = cl_gui_container=>default_screen.
10
11 CALL METHOD o_grid->set_table_for_first_display
12 EXPORTING
13   i_structure_name = 'SFLIGHT'
14   CHANGING
15     it_outtab      = lt_flight.
16
17 CALL SCREEN 0001. "Any screen number, just double click create it and provide a description
18                  "No pbo module activation and no screen container creation
```

\Out Put

Line Item	Material	Plant	Storage Location	Quantity	Unit of Measure	Base Unit of Measure	Base Quantity	Base Unit of Measure	Base Quantity
1	1000000	0001	0001	1000	KG	KG	1000	KG	1000
2	1000000	0001	0001	1000	KG	KG	1000	KG	1000
3	1000000	0001	0001	1000	KG	KG	1000	KG	1000
4	1000000	0001	0001	1000	KG	KG	1000	KG	1000
5	1000000	0001	0001	1000	KG	KG	1000	KG	1000
6	1000000	0001	0001	1000	KG	KG	1000	KG	1000
7	1000000	0001	0001	1000	KG	KG	1000	KG	1000
8	1000000	0001	0001	1000	KG	KG	1000	KG	1000
9	1000000	0001	0001	1000	KG	KG	1000	KG	1000
10	1000000	0001	0001	1000	KG	KG	1000	KG	1000
11	1000000	0001	0001	1000	KG	KG	1000	KG	1000
12	1000000	0001	0001	1000	KG	KG	1000	KG	1000
13	1000000	0001	0001	1000	KG	KG	1000	KG	1000
14	1000000	0001	0001	1000	KG	KG	1000	KG	1000
15	1000000	0001	0001	1000	KG	KG	1000	KG	1000
16	1000000	0001	0001	1000	KG	KG	1000	KG	1000
17	1000000	0001	0001	1000	KG	KG	1000	KG	1000
18	1000000	0001	0001	1000	KG	KG	1000	KG	1000
19	1000000	0001	0001	1000	KG	KG	1000	KG	1000
20	1000000	0001	0001	1000	KG	KG	1000	KG	1000
21	1000000	0001	0001	1000	KG	KG	1000	KG	1000
22	1000000	0001	0001	1000	KG	KG	1000	KG	1000
23	1000000	0001	0001	1000	KG	KG	1000	KG	1000
24	1000000	0001	0001	1000	KG	KG	1000	KG	1000
25	1000000	0001	0001	1000	KG	KG	1000	KG	1000
26	1000000	0001	0001	1000	KG	KG	1000	KG	1000
27	1000000	0001	0001	1000	KG	KG	1000	KG	1000
28	1000000	0001	0001	1000	KG	KG	1000	KG	1000
29	1000000	0001	0001	1000	KG	KG	1000	KG	1000
30	1000000	0001	0001	1000	KG	KG	1000	KG	1000
31	1000000	0001	0001	1000	KG	KG	1000	KG	1000
32	1000000	0001	0001	1000	KG	KG	1000	KG	1000
33	1000000	0001	0001	1000	KG	KG	1000	KG	1000
34	1000000	0001	0001	1000	KG	KG	1000	KG	1000
35	1000000	0001	0001	1000	KG	KG	1000	KG	1000
36	1000000	0001	0001	1000	KG	KG	1000	KG	1000
37	1000000	0001	0001	1000	KG	KG	1000	KG	1000
38	1000000	0001	0001	1000	KG	KG	1000	KG	1000
39	1000000	0001	0001	1000	KG	KG	1000	KG	1000
40	1000000	0001	0001	1000	KG	KG	1000	KG	1000
41	1000000	0001	0001	1000	KG	KG	1000	KG	1000
42	1000000	0001	0001	1000	KG	KG	1000	KG	1000
43	1000000	0001	0001	1000	KG	KG	1000	KG	1000
44	1000000	0001	0001	1000	KG	KG	1000	KG	1000
45	1000000	0001	0001	1000	KG	KG	1000	KG	1000
46	1000000	0001	0001	1000	KG	KG	1000	KG	1000
47	1000000	0001	0001	1000	KG	KG	1000	KG	1000
48	1000000	0001	0001	1000	KG	KG	1000	KG	1000
49	1000000	0001	0001	1000	KG	KG	1000	KG	1000
50	1000000	0001	0001	1000	KG	KG	1000	KG	1000

# Docking Container

14 September 2025 15:04

## OO ALV USING DOCKING CONTAINER

```
data : o_dock type ref to cl_gui_docking_container,  
o_alv type ref to cl_gui_alv_grid,  
it_spfli type table of spfli,  
wa_spfli type spfli,  
ok_code type sy-ucomm.  
select-options : p_carr for wa_spfli-carrid.  
start-of-selection.  
perform data_fetch.  
call screen 100.  
form DATA_FETCH .  
select * from spfli into table it_spfli where carrid in p_carr.  
endform. " DATA_FETCH
```

```
module STATUS_0100 output.  
SET PF-STATUS 'STATUS'.  
SET TITLEBAR 'TITLE'.  
endmodule. " STATUS_0100 OUTPUT  
module USER_COMMAND_0100 input.  
case ok_code.  
when 'BACK' OR 'EXIT' OR 'CANCEL'.  
LEAVE TO SCREEN 0.  
endcase.  
endmodule. " USER_COMMAND_0100 INPUT  
module DISPLAY_ALV output.  
if o_dock is initial.
```

\* EITHER PROVIDE SIDE AND EXTENSION OR RATIO TO CREATE A DOCKING CONTAINER\*

```
CREATE OBJECT o_dock  
EXPORTING  
side = o_dock->dock_at_left  
extension = 2000.  
* CREATE OBJECT o_dock  
* EXPORTING  
* ratio = 60.
```

```
endif.  
if o_alv is initial.
```

```
CREATE OBJECT o_alv  
EXPORTING  
i_parent = o_dock.  
CALL METHOD o_alv->set_table_for_first_display  
EXPORTING  
i_structure_name = 'SPFLI'  
CHANGING  
it_outtab = it_spfli.  
endif.
```

```
endmodule. " DISPLAY_ALV OUTPUT  
SCREEN FLOW LOGIC  
PROCESS BEFORE OUTPUT.  
MODULE STATUS_0100.  
MODULE DISPLAY_ALV.
```

```
PROCESS AFTER INPUT.  
MODULE USER_COMMAND_0100.
```

From <<https://sapcodes.com/2015/11/20/oo-alv-using-docking-container/>>

# Custom Container

14 September 2025 15:08

## OO ALV USING CUSTOM CONTAINER

```
DATA : o_cust TYPE REF TO cl_gui_custom_container,  
o_alv TYPE REF TO cl_gui_alv_grid,  
it_spfli TYPE TABLE OF spfli,  
wa_spfli TYPE spfli,  
ok_code TYPE sy-ucomm.  
SELECT-OPTIONS : p_carr FOR wa_spfli-carrid.  
START-OF-SELECTION.  
PERFORM data_fetch.  
CALL SCREEN 100.  
FORM data_fetch .  
SELECT * FROM spfli INTO TABLE it_spfli WHERE carrid IN p_carr.  
ENDFORM.          " DATA_FETCH
```

```
MODULE status_0100 OUTPUT.  
SET PF-STATUS 'STATUS'.  
SET TITLEBAR 'TITLE'.  
ENDMODULE.          " STATUS_0100 OUTPUT
```

```
MODULE user_command_0100 INPUT.  
CASE ok_code.  
WHEN 'BACK' OR 'EXIT' OR 'CANCEL'.  
LEAVE TO SCREEN 0.  
ENDCASE.  
ENDMODULE.          " USER_COMMAND_0100 INPUT
```

```
MODULE display_alv OUTPUT.  
CREATE OBJECT o_cust  
EXPORTING  
container_name = 'CUSTOM'.  
CREATE OBJECT o_alv  
EXPORTING  
i_parent = O_CUST.  
CALL METHOD o_alv->set_table_for_first_display  
EXPORTING  
i_structure_name = 'SPFLI'  
CHANGING  
it_outtab = IT_SPFLI.  
ENDMODULE.          " DISPLAY_ALV OUTPUT
```

### SCREEN FLOW LOGIC

```
PROCESS BEFORE OUTPUT.  
MODULE STATUS_0100.  
MODULE DISPLAY_ALV.
```

```
PROCESS AFTER INPUT.  
MODULE USER_COMMAND_0100.
```

From <<https://sapcodes.com/2015/11/20/oo-alv-using-custom-container/>>

# Double click event

14 September 2025

15:19

## OO ALV USING CUSTOM CONTAINER WITH DOUBLE CLICK EVENT

```
TYPES : BEGIN OF tab,
carrid TYPE spfli-carrid,
connid TYPE spfli-connid,
countryfr TYPE spfli-countryfr,
cityfrom TYPE spfli-cityfrom,
airpfrom TYPE spfli-airpfrom,
countryto TYPE spfli-countryto,
cityto TYPE spfli-cityto,
airpto TYPE spfli-airpto,
END OF tab.
DATA : it_spfli TYPE TABLE OF tab,
wa_spfli TYPE tab,
ok_code TYPE sy-ucomm,
o_cust TYPE REF TO cl_gui_custom_container,
o_alv TYPE REF TO cl_gui_alv_grid,
it_fcat TYPE lvc_t_fcat,
wa_fcat TYPE lvc_s_fcat .
DATA : IT_SFLIGHT TYPE TABLE OF SFLIGHT.
DATA : O_DOCK TYPE REF TO CL_GUI_DOCKING_CONTAINER,
O_ALV1 TYPE REF TO CL_GUI_ALV_GRID.
SELECT-OPTIONS : s_carr FOR wa_spfli-carrid.
CLASS HANDLE_EVENT DEFINITION.
PUBLIC SECTION.
CLASSMETHODS : HANDLE_DOUBLE_CLICK
FOR EVENT DOUBLE_CLICK OF CL_GUI_ALV_GRID IMPORTING e_row .
ENDCLASS.
CLASS HANDLE_EVENT IMPLEMENTATION.
METHOD handle_double_click.
READ TABLE IT_SPFLI INTO WA_SPFLI INDEX E_ROW.
SELECT * FROM SFLIGHT INTO TABLE IT_SFLIGHT WHERE CARRID = WA_SPFLI-CARRID.
CALL SCREEN 200.
ENDMETHOD.
ENDCLASS.
```

```
START-OF-SELECTION.
PERFORM DATA_FETCH.
PERFORM BUILD_FIELDCATALOG.
CALL SCREEN 100.
form DATA_FETCH .
SELECT carrid
connid
countryfr
cityfrom
airpfrom
countryto
cityto
airpto FROM spfli INTO TABLE it_spfli WHERE carrid IN s_carr.
endform.          " DATA_FETCH
form BUILD_FIELDCATALOG .
CLEAR wa_fcat.
WA_FCAT-col_pos = 1.
WA_FCAT-fieldname = 'CARRID'.
WA_FCAT-tabname = 'IT_SPFLI'.
WA_FCAT-COLTEXT = 'CARRID'.
APPEND wa_fcat TO IT_FCAT.
CLEAR wa_fcat.
WA_FCAT-col_pos = 2.
WA_FCAT-fieldname = 'CONNID'.
WA_FCAT-tabname = 'IT_SPFLI'.
WA_FCAT-COLTEXT = 'CONNID'.
APPEND wa_fcat TO IT_FCAT.
CLEAR wa_fcat.
WA_FCAT-col_pos = 3.
WA_FCAT-fieldname = 'COUNTRYFR'.
WA_FCAT-tabname = 'IT_SPFLI'.
WA_FCAT-COLTEXT = 'COUNTRY FROM'.
```

```

APPEND wa_fcat TO IT_FCAT.
CLEAR wa_fcat.
WA_FCAT-col_pos = 4.
WA_FCAT-fieldname = 'CITYFROM'.
WA_FCAT-tabname = 'IT_SPFLI'.
WA_FCAT-COLTEXT = 'COUNTRY TO'..
APPEND wa_fcat TO IT_FCAT.
CLEAR wa_fcat.
WA_FCAT-col_pos = 5.
WA_FCAT-fieldname = 'AIRPFROM'.
WA_FCAT-tabname = 'IT_SPFLI'.
WA_FCAT-COLTEXT = 'AIRPORT FROM'.
APPEND wa_fcat TO IT_FCAT.
CLEAR wa_fcat.
WA_FCAT-col_pos = 6.
WA_FCAT-fieldname = 'COUNTRYTO'.
WA_FCAT-tabname = 'IT_SPFLI'.
WA_FCAT-COLTEXT = 'COUNTRY TO'.
APPEND wa_fcat TO IT_FCAT.
CLEAR wa_fcat.
WA_FCAT-col_pos = 7.
WA_FCAT-fieldname = 'CITYTO'.
WA_FCAT-tabname = 'IT_SPFLI'.
WA_FCAT-COLTEXT = 'CITY TO'.
APPEND wa_fcat TO IT_FCAT.
CLEAR wa_fcat.
WA_FCAT-col_pos = 8.
WA_FCAT-fieldname = 'AIRPTO'.
WA_FCAT-tabname = 'IT_SPFLI'.
WA_FCAT-COLTEXT = 'AIRPORT TO'.
APPEND wa_fcat TO IT_FCAT.
CLEAR wa_fcat.
endform.          " BUILD_FIELDCATALOG
module STATUS_0100 output.
SET PF-STATUS 'STAT'.
SET TITLEBAR 'TITLT'.
endmodule.        " STATUS_0100 OUTPUT
module DISPLAY_ALV output.
IF o_cust IS NOT BOUND.
CREATE OBJECT o_cust
EXPORTING
container_name      = 'CONTAINER'.
ENDIF.
IF O_ALV IS NOT BOUND.
CREATE OBJECT o_alv
EXPORTING
i_parent            = O_CUST.
SET HANDLER HANDLE_EVENT=>HANDLE_DOUBLE_CLICK FOR O_ALV.
CALL METHOD o_alv->set_table_for_first_display
CHANGING
it_outtab           = IT_SPFLI
it_fieldcatalog     = IT_FCAT.
ENDIF.
endmodule.        " DISPLAY_ALV OUTPUT
module USER_COMMAND_0100 input.
CASE OK_CODE.
WHEN 'BACK'.
LEAVE TO SCREEN 0.
ENDCASE.
endmodule.        " USER_COMMAND_0100 INPUT
module STATUS_0200 output.
SET PF-STATUS 'STSA'.
* SET TITLEBAR 'xxx'.
IF O_DOCK IS NOT BOUND.
CREATE OBJECT o_dock
EXPORTING
side                = CL_GUI_DOCKING_CONTAINER=>dock_at_left
extension           = 5000.
ENDIF.

IF O_ALV1 IS NOT BOUND.
CREATE OBJECT o_alv1
EXPORTING
i_parent            = O_DOCK.
CALL METHOD o_alv1->set_table_for_first_display
EXPORTING
i_structure_name    = 'SFLIGHT'

```

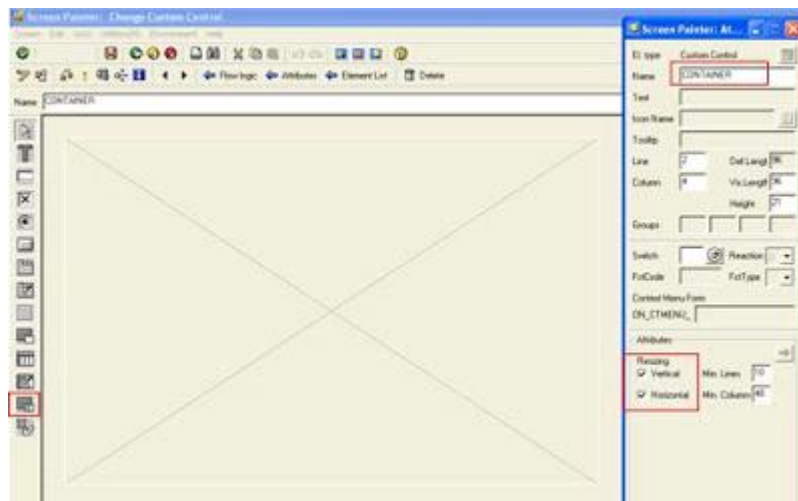
```

CHANGING
it_outtab          = IT_SFLIGHT.
ENDIF.
CALL METHOD o_alv1->refresh_table_display.

endmodule.          " STATUS_0200 OUTPUT
module USER_COMMAND_0200 input.
CASE OK_CODE.
WHEN 'BACK'.
LEAVE TO SCREEN 0.
ENDCASE.
endmodule.          " USER_COMMAND_0200 INPUT

_SCREEN 100 FLOW LOGIC
PROCESS BEFORE OUTPUT.
MODULE STATUS_0100.
MODULE DISPLAY_ALV.
PROCESS AFTER INPUT.
MODULE USER_COMMAND_0100.

```



```

_SCREEN 200 FLOW LOGIC
PROCESS BEFORE OUTPUT.
MODULE STATUS_0200.
PROCESS AFTER INPUT.
MODULE USER_COMMAND_0200.

```

From <<https://sapcodes.com/2015/11/20/oo-alv-using-custom-container-with-double-click-event/>>

# Interactive ALV

14 September 2025 17:38

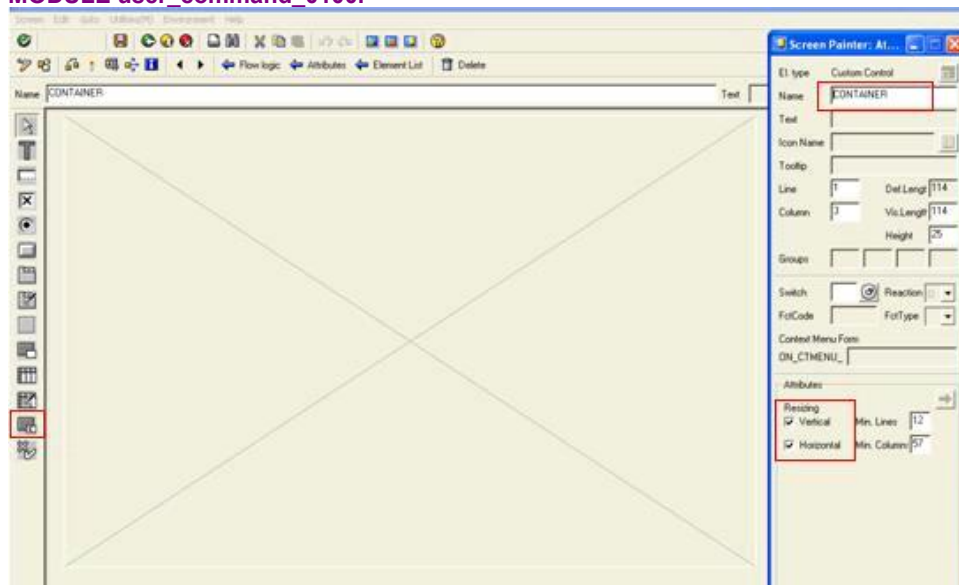
## SPLITTING CUSTOM CONTAINER BY SPLITTER CONTAINER AND DISPLAYING INTERACTIVE ALV

```
DATA : it_spfli TYPE TABLE OF spfli,
wa_spfli TYPE spfli,
it_sflight TYPE TABLE OF sflight,
wa_sflight TYPE sflight,
o_cust TYPE REF TO cl_gui_custom_container,
o_spli TYPE REF TO cl_gui_splitter_container,
o_ref1 TYPE REF TO cl_gui_container,
o_ref2 TYPE REF TO cl_gui_container,
o_alv1 TYPE REF TO cl_gui_alv_grid,
o_alv2 TYPE REF TO cl_gui_alv_grid,
ok_code TYPE sy-ucomm.
CLASS handle_event DEFINITION.
PUBLIC SECTION.
CLASS-METHODS handle_double_click
FOR EVENT double_click OF cl_gui_alv_grid
IMPORTING e_row.
ENDCLASS.
"HANDLE_EVENT DEFINITION
CLASS handle_event IMPLEMENTATION.
METHOD handle_double_click.
READ TABLE it_spfli INTO wa_spfli INDEX e_row.SELECT * FROM sflight INTO TABLE it_sflight WHERE c
arrid = wa_spfli-carrid.
IF o_alv2 IS NOT BOUND.
CREATE OBJECT o_alv2
EXPORTING
i_parent = o_ref2.
CALL METHOD o_alv2->set_table_for_first_display
EXPORTING
i_structure_name = 'SFLIGHT'
CHANGING
it_outtab = it_sflight.
ENDIF.
CALL METHOD o_alv2->refresh_table_display.
ENDMETHOD.
"HANDLE_DOUBLE_CLICK
ENDCLASS.
"HANDLE_EVENT IMPLEMENTATION
START-OF-SELECTION.
PERFORM fetch_data_from_spfli.
CALL SCREEN 100.
*-----*
FORM fetch_data_from_spfli .
SELECT * FROM spfli INTO TABLE it_spfli.
ENDFORM.
" FETCH_DATA_FROM_SPFLI
MODULE status_0100 OUTPUT.
SET PF-STATUS 'STATUS'.
SET TITLEBAR 'TITLE'.
ENDMODULE.
" STATUS_0100 OUTPUT
MODULE split_container OUTPUT.
IF o_cust IS NOT BOUND.
CREATE OBJECT o_cust
EXPORTING
container_name = 'CONTAINER'.
CREATE OBJECT o_spli
EXPORTING
parent = o_cust
rows = 2
columns = 1.
CALL METHOD o_spli->get_container
EXPORTING
row = 1
column = 1
RECEIVING
container = o_ref1.
CALL METHOD o_spli->get_container
EXPORTING
row = 2
column = 1
RECEIVING
container = o_ref2.
```

```

ENDIF.
ENDMODULE.          " SPLIT_CONTAINER OUTPUT
MODULE display_alv OUTPUT.
IF o_alv1 IS NOT BOUND.
CREATE OBJECT o_alv1
EXPORTING
i_parent = o_ref1.
SET HANDLER handle_event=>handle_double_click FOR o_alv1.
CALL METHOD o_alv1->set_table_for_first_display
EXPORTING
i_structure_name = 'SPFLI'
CHANGING
it_outtab      = it_spfli.
ENDIF.
ENDMODULE.          " DISPLAY_ALV OUTPUT
MODULE user_command_0100 INPUT.
CASE ok_code.
WHEN 'BACK'.
LEAVE PROGRAM.
ENDCASE.
ENDMODULE.          " USER_COMMAND_0100 INPUT
_SCREEN 100 FLOW LOGIC
PROCESS BEFORE OUTPUT.
MODULE status_0100.
MODULE split_container.
MODULE display_alv.
PROCESS AFTER INPUT.
MODULE user_command_0100.

```



From <<https://sapcodes.com/2015/11/20/splitting-custom-container-by-splitter-container-and-displaying-interactive-alv/>>



# Top of page

14 September 2025

17:53

## OO ALV WITH TOP\_OF\_PAGE EVENT TO DISPLAY HEADER

```
DATA : it_spfli TYPE TABLE OF spfli,
wa_spfli TYPE spfli,
o_cust TYPE REF TO cl_gui_custom_container,
o_spli TYPE REF TO cl_gui_splitter_container,
o_ref1 TYPE REF TO cl_gui_container,
o_ref2 TYPE REF TO cl_gui_container,
o_alv TYPE REF TO cl_gui_alv_grid,
o_docu TYPE REF TO cl_dd_document,
ok_code TYPE sy-ucomm.
*-----*

CLASS handle_event DEFINITION.
PUBLIC SECTION.
CLASSMETHODS : page_head FOR EVENT
                top_of_page OF cl_gui_alv_grid
                IMPORTING e_dyndoc_id table_index.

ENDCLASS.                "HANDLE_EVENT DEFINITION
*-----*

CLASS handle_event IMPLEMENTATION.
METHOD page_head.
*-----try this one-----*
    DATA : text TYPE sdydo_text_element.
    text = 'FLIGHT DETAILS'.
    CALL METHOD o_docu->add_text
    EXPORTING
        text      = text
        sap_color  = cl_dd_document=>list_positive
        sap_fontsize = cl_dd_document=>large.
    CALL METHOD o_docu->add_picture
    EXPORTING
        picture_id = 'HEADER'. " Object uploaded by OAER Transaction
    CALL METHOD o_docu->display_document
    EXPORTING
        parent = o_ref1.

*-----or try this -----*
    DATA : text TYPE sdydo_text_element .
    DATA : date TYPE char10.
    WRITE sy-datum TO date USING EDIT MASK '___/___/___'.
    CONCATENATE : 'FLIGHT DETAILS ON ' date INTO text SEPARATED BY space.
    CALL METHOD o_docu->add_text
    EXPORTING
        text      = text
        sap_style  = cl_dd_document=>heading
        sap_color  = cl_dd_document=>list_heading_int
        sap_fontsize = cl_dd_document=>large.
    CALL METHOD o_docu->new_line.
    CLEAR text.
    CONCATENATE : 'BY USER : ' sy-username INTO text SEPARATED BY space.
    CALL METHOD o_docu->add_text
    EXPORTING
        text      = text
        sap_style  = cl_dd_document=>heading
        sap_color  = cl_dd_document=>list_heading_int
        sap_fontsize = cl_dd_document=>large.
    CALL METHOD o_docu->set_document_background
    EXPORTING
        picture_id = 'HEADER1'. " Object uploaded by OAER Transaction
    CALL METHOD o_docu->display_document
    EXPORTING
        reuse_control = 'X'
        parent      = o_ref1.

ENDMETHOD.                "PAGE_HEAD
ENDCLASS.                "HANDLE_EVENT IMPLEMENTATION
START-OF-SELECTION.
```

```

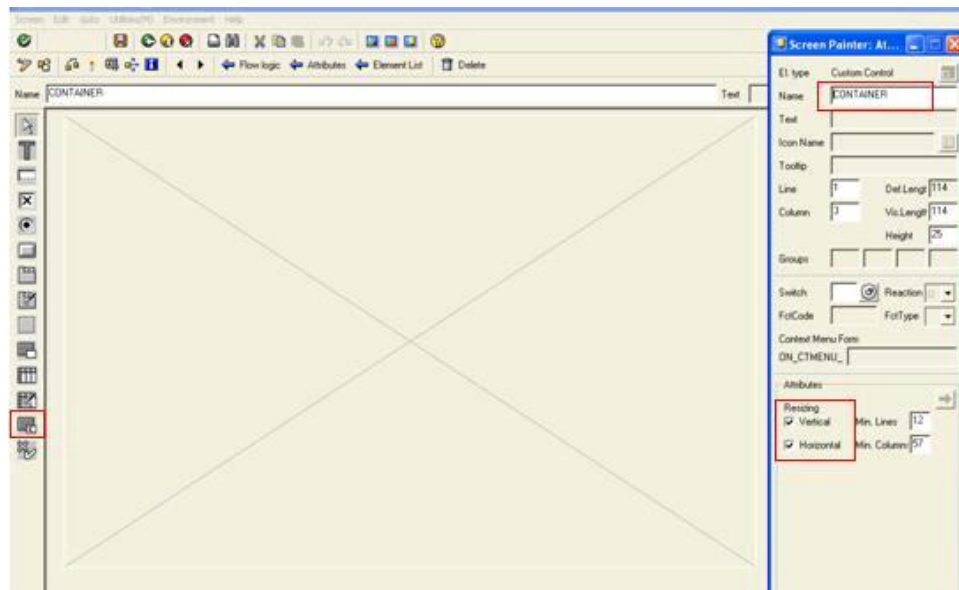
PERFORM fetch_data_from_spfli.
CALL SCREEN 100.
*&_____*
FORM fetch_data_from_spfli .
SELECT * FROM spfli INTO TABLE it_spfli.
ENDFORM.          " FETCH_DATA_FROM_SPFLI
*&_____*
MODULE status_0100 OUTPUT.
SET PF-STATUS 'STATUS'.
SET TITLEBAR 'TITLE'.
ENDMODULE.          " STATUS_0100 OUTPUT
*_____*

MODULE split_container OUTPUT.
IF o_cust IS NOT BOUND.
  CREATE OBJECT o_cust
  EXPORTING
    container_name = 'CONTAINER'.
  CREATE OBJECT o_spli
  EXPORTING
    parent = o_cust
    rows = 2
    columns = 1.
  CALL METHOD o_spli->get_container
  EXPORTING
    row = 1
    column = 1
  RECEIVING
    container = o_ref1.
  CALL METHOD o_spli->set_row_height
  EXPORTING
    id = 1
    height = 35.
  CALL METHOD o_spli->get_container
  EXPORTING
    row = 2
    column = 1
  RECEIVING
    container = o_ref2.
  CREATE OBJECT o_docu
  EXPORTING
    style = 'ALV_GRID'.
ENDIF.
ENDMODULE.          " SPLIT_CONTAINER OUTPUT
*_____*

MODULE display_alv OUTPUT.
IF o_alv IS NOT BOUND.
  CREATE OBJECT o_alv
  EXPORTING
    i_parent = o_ref2.
  SET HANDLER handle_event=>page_head FOR o_alv.
  CALL METHOD o_alv->set_table_for_first_display
  EXPORTING
    i_structure_name = 'SPFLI'
  CHANGING
    it_outtab = it_spfli.
  CALL METHOD o_alv->list_processing_events
  EXPORTING
    i_event_name = 'TOP_OF_PAGE'
    i_dyndoc_id = o_docu.
ENDIF.
ENDMODULE.          " DISPLAY_ALV OUTPUT
*_____*

MODULE user_command_0100 INPUT.
CASE ok_code.
WHEN 'BACK'.
  LEAVE PROGRAM.
ENDCASE.
ENDMODULE.          " USER_COMMAND_0100 INPUT
SCREEN 100 FLOW LOGIC
PROCESS BEFORE OUTPUT.
MODULE status_0100.
MODULE split_container.
MODULE display_alv.
PROCESS AFTER INPUT.
MODULE user_command_0100.

```



From <[https://sapcodes.com/2015/11/20/oo-alv-with-top\\_of\\_page-event-to-display-header/](https://sapcodes.com/2015/11/20/oo-alv-with-top_of_page-event-to-display-header/)>

# HIDE Functions

14 September 2025

19:04

```
DATA : it_spfli      TYPE TABLE OF spfli,
      wa_spfli      TYPE spfli,
      ok_code       TYPE sy-ucomm,
      o_cust        TYPE REF TO cl_gui_custom_container,
      o_alv         TYPE REF TO cl_gui_alv_grid,
      lt_exclude     TYPE ui_functions,
      ls_exclude     TYPE ui_func,
      ls_layo       TYPE lvc_s_layo.
```

```
START-OF-SELECTION.
  PERFORM data_fetch.
  PERFORM build_layout.
  PERFORM fill_fcode.
```

```
END-OF-SELECTION.
```

```
  CALL SCREEN 100.
  *-----*
```

```
FORM data_fetch .
  SELECT * FROM spfli INTO TABLE it_spfli .
  SORT it_spfli BY carrid connid.
```

```
ENDFORM.                " DATA_FETCH
  *-----*
```

```
FORM build_layout .
```

```
*ls_layo-no_toolbar = 'X'. " To hide all toolbar buttons
```

```
ENDFORM.
  *-----*
```

```
FORM fill_fcode .
  ls_exclude = cl_gui_alv_grid=>mc_fc_subtot.
  APPEND ls_exclude TO lt_exclude.
  CLEAR ls_exclude.
```

```
ls_exclude = cl_gui_alv_grid=>mc_fc_sort_asc.
  APPEND ls_exclude TO lt_exclude.
  CLEAR ls_exclude.
ls_exclude = cl_gui_alv_grid=>mc_fc_sort_dsc.
  APPEND ls_exclude TO lt_exclude.
  CLEAR ls_exclude.
```

```
ls_exclude = cl_gui_alv_grid=>mc_fc_filter.
  APPEND ls_exclude TO lt_exclude.
  CLEAR ls_exclude.
```

```
ls_exclude = cl_gui_alv_grid=>mc_fc_detail.
  APPEND ls_exclude TO lt_exclude.
  CLEAR ls_exclude.
```

```
ENDFORM.                " FILL_FCODE
  *-----*
```

```
MODULE display_alv OUTPUT.
  IF o_cust IS NOT BOUND.
    CREATE OBJECT o_cust
    EXPORTING
      container_name = 'CONT'.
  ENDIF.
  IF o_alv IS NOT BOUND.
    CREATE OBJECT o_alv
```

```

EXPORTING
    i_parent = o_cust.
ENDIF.
CALL METHOD o_alv->set_table_for_first_display
EXPORTING
    i_structure_name    = 'SPFLI'
    it_toolbar_excluding = lt_exclude
    is_layout           = ls_layo
CHANGING
    it_outtab          = it_spfli.

ENDMODULE.          " DISPLAY_ALV OUTPUT
" BUILD_LAYOUT
SCREEN 100 FLOW LOGIC
PROCESS BEFORE OUTPUT.
* MODULE status_0100.
MODULE display_alv.
PROCESS AFTER INPUT.
* MODULE user_command_0100.

```

\*\_\_\_\_\_\*

#### OUTPUT BEFORE & AFTER HIDING TOOLBAR BUTTONS

ID	No.	Cty	Depart. city	De...	Cty	Arrival city	Apt	FlightT...
A/	17		NEW YORK	JFK		SAN FRANCISCO	SFO	361:00
AA	64		SAN FRANCISCO	SFO		NEW YORK	JFK	321:00
AC	820	DDD	FRANKFURT/MAIN	FRA	CA	MONTREAL-DORVAL	YUL	480:00
AF	820	SSS	FRANKFURT/MAIN	FRA	MQ	FORT DE FRANCE	FDF	660:00
AI	1000		AMSTERDAM			BOMBAY		0:00

ID	No.	Cty	Depart. city	De...	Cty	Arrival city	Apt	FlightT...
A/	17		NEW YORK	JFK		SAN FRANCISCO	SFO	361:00
AA	64		SAN FRANCISCO	SFO		NEW YORK	JFK	321:00
AC	820	DDD	FRANKFURT/MAIN	FRA	CA	MONTREAL-DORVAL	YUL	480:00

From <<https://sapcodes.com/2015/11/20/oo-alv-with-hiding-toolbar/>>

# Edit & Save

14 September 2025

19:12

## OO ALV WITH EDIT & SAVE FUNCTIONALITY & ADDING BUTTON IN ALV TOOLBAR

November 20, 2015 [coderobbot](#) [Uncategorized](#) [2 comments](#)

TYPE-POOLS : icon.

TYPES : BEGIN OF tab,

    carrid      TYPE spfli-carrid,  
    connid     TYPE spfli-connid,  
    countryfr   TYPE spfli-countryfr,  
    cityfrom    TYPE spfli-cityfrom,  
    airpfrom    TYPE spfli-airpfrom,  
    countryto   TYPE spfli-countryto,  
    cityto      TYPE spfli-cityto,  
    airpto      TYPE spfli-airpto,  
END OF tab.

DATA : it\_spfli      TYPE TABLE OF tab,  
      it\_spfli\_old  TYPE TABLE OF tab,  
      it\_spfli\_new  TYPE TABLE OF spfli,  
      wa\_spfli\_new   TYPE spfli,  
      wa\_spfli      TYPE tab,  
      ok\_code      TYPE sy-ucomm,  
      o\_cust       TYPE REF TO cl\_gui\_custom\_container,  
      o\_alv        TYPE REF TO cl\_gui\_alv\_grid,  
      it\_fcat       TYPE lvc\_t\_fcat,  
      is\_fcat       TYPE lvc\_s\_fcat,  
      is\_layo       TYPE lvc\_s\_layo,  
      check.

\*-----\*

\*      CLASS HANDLE\_EVENT DEFINITION

\*-----\*

\*

\*-----\*

CLASS handle\_event DEFINITION.

  PUBLIC SECTION.

    CLASS-METHODS : handle\_toolbar

      FOR EVENT toolbar OF cl\_gui\_alv\_grid

      IMPORTING e\_object

      e\_interactive.

    CLASS-METHODS : handle\_user\_command

      FOR EVENT user\_command OF cl\_gui\_alv\_grid

      IMPORTING e\_ucomm.

ENDCLASS.                  "HANDLE\_EVENT DEFINITION

\*-----\*

CLASS handle\_event IMPLEMENTATION.

  METHOD handle\_toolbar.

\*      \*ADDING A SAVE BUTTONN TO THE ALV TOOLBAR \*

    DATA : is\_btn TYPE stb\_button.

    is\_btn-function = 'SAVE'.

    is\_btn-icon = icon\_system\_save.

    is\_btn-text = 'SAVE'.

    is\_btn-quickinfo = 'SAVE'.

    is\_btn-disabled = ' '.

    APPEND is\_btn TO e\_object->mt\_toolbar.

  ENDMETHOD.                  "handle\_toolbar

  METHOD handle\_user\_command .

    CASE e\_ucomm.

      WHEN 'SAVE'. " WILL TRIGGER WHEN CLICK ON ALV TOOLBAR 'SAVE' BUTTON

        PERFORM update\_data\_base.

    ENDCASE.

  ENDMETHOD.                  "handle\_user\_command

ENDCLASS.                  "HANDLE\_EVENT IMPLEMENTATION

START-OF-SELECTION.

  PERFORM data\_fetch.

  PERFORM buld\_fieldcat.

  PERFORM buld\_layout.

END-OF-SELECTION.

  CALL SCREEN 100.

\*-----\*

```

FORM data_fetch .
  SELECT carrid
         connid
         countryfr
         cityfrom
         airpfrom
         countryto
         cityto
         airpto FROM spfli INTO TABLE it_spfli .
  SORT it_spfli BY carrid connid.
  it_spfli_old = it_spfli.
ENDFORM.                " DATA_FETCH
*-----*
MODULE display_alv OUTPUT.
  IF o_cust IS NOT BOUND.
    CREATE OBJECT o_cust
    EXPORTING
      container_name = 'CONT'.
  ENDIF.
  IF o_alv IS NOT BOUND.
    CREATE OBJECT o_alv
    EXPORTING
      i_parent = o_cust.
    SET HANDLER handle_event=>handle_toolbar FOR o_alv.
    SET HANDLER handle_event=>handle_user_command FOR o_alv.
    CALL METHOD o_alv->set_table_for_first_display
    EXPORTING
      is_layout      = is_layo
    CHANGING
      it_outtab      = it_spfli
      it_fieldcatalog = it_fcat.
    CALL METHOD o_alv->set_toolbar_interactive .
  ENDIF.
  CALL METHOD o_alv->refresh_table_display.
ENDMODULE.              " DISPLAY_ALV OUTPUT
*-----*
FORM buld_fieldcat .
  is_fcat-col_pos = 1 .
  is_fcat-fieldname = 'CARRID'.
  is_fcat-tabname = 'IT_SPFLI'.
  is_fcat-scrtext_l = 'Carrer No.'.
  is_fcat-key = 'X'.
  APPEND is_fcat TO it_fcat.
  CLEAR is_fcat.
  is_fcat-col_pos = 2 .
  is_fcat-fieldname = 'CONNID'.
  is_fcat-tabname = 'IT_SPFLI'.
  is_fcat-scrtext_l = 'Conn No.'.
  is_fcat-key = 'X'.
  APPEND is_fcat TO it_fcat.
  CLEAR is_fcat.
  is_fcat-col_pos = 3 .
  is_fcat-fieldname = 'COUNTRYFR'.
  is_fcat-tabname = 'IT_SPFLI'.
  is_fcat-scrtext_l = 'Country From'.
  APPEND is_fcat TO it_fcat.
  CLEAR is_fcat.
  is_fcat-col_pos = 4 .
  is_fcat-fieldname = 'CITYFROM'.
  is_fcat-tabname = 'IT_SPFLI'.
  is_fcat-scrtext_l = 'City From'.
  is_fcat-edit = 'X'. " MAKES ONLY THIS FIELD AS EDITABLE
  APPEND is_fcat TO it_fcat.
  CLEAR is_fcat.
  is_fcat-col_pos = 5 .
  is_fcat-fieldname = 'AIRP FROM'.
  is_fcat-tabname = 'IT_SPFLI'.
  is_fcat-scrtext_l = 'Airp From'.
  APPEND is_fcat TO it_fcat.
  CLEAR is_fcat.
  is_fcat-col_pos = 6.
  is_fcat-fieldname = 'COUNTRYTO'.
  is_fcat-tabname = 'IT_SPFLI'.
  is_fcat-scrtext_l = 'Country To'.

```

```

APPEND is_fcat TO it_fcat.
CLEAR is_fcat.
is_fcat-col_pos = 7 .
is_fcat-fieldname = 'CITYTO'.
is_fcat-tabname = 'IT_SPFLI'.
is_fcat-scrtext_l = 'City To'.
is_fcat-edit = 'X'. " MAKES ONLY THIS FIELD AS EDITABLE
APPEND is_fcat TO it_fcat.
CLEAR is_fcat.
is_fcat-col_pos = 8 .
is_fcat-fieldname = 'AIRPTO'.
is_fcat-tabname = 'IT_SPFLI'.
is_fcat-scrtext_l = 'Airport To'.
APPEND is_fcat TO it_fcat.
CLEAR is_fcat.
ENDFORM.                " BULD_FIELDCAT
*-----*

FORM buld_layout .
  is_layo-col_opt = 'X'.
* is_layo-edit = 'X'. " ALL FIELDS ARE EDITABLE
ENDFORM.                " BULD_LAYOUT
*-----*

MODULE status_0100 OUTPUT.
  SET PF-STATUS 'STATUS'.
ENDMODULE.              " STATUS_0100 OUTPUT
*-----*

MODULE user_command_0100 INPUT.
  CASE ok_code.
    WHEN 'SAVE'. " WILL TRIGGER WHEN CLICK ON STANDARD TOOLBAR SAVE BUTTON
      PERFORM UPDATE_DATA_BASE.
    WHEN 'BACK' OR 'EXIT' OR 'CANCEL'.
      LEAVE PROGRAM.
  ENDCASE.
ENDMODULE.              " USER_COMMAND_0100 INPUT
*-----*

form UPDATE_DATA_BASE .
  CALL METHOD o_alv->check_changed_data
  IMPORTING
    e_valid = check.
  IF it_spfli NE it_spfli_old.
    LOOP AT it_spfli INTO wa_spfli.
      MOVE-CORRESPONDING wa_spfli TO wa_spfli_new.
      APPEND wa_spfli_new TO it_spfli_new.
    ENDLOOP.
    MODIFY spfli FROM TABLE it_spfli_new.
    IF sy-subrc = 0..
      MESSAGE 'DATABASE TABLE SPFLI UPDATED' TYPE 'S'.
    ENDIF.
  ENDIF.
endform.                " UPDATE_DATA_BASE

```

From <<https://sapcodes.com/2015/11/20/oo-alv-with-edit-save-functionality-adding-button-in-alv-toolbar/>>



# Sub Totals

28 September 2025 14:09

```
*&-----*
*& Report YPRO_OOPS_ALV1_640
*&-----*
*&
*&-----*
REPORT ypro_oops_alv1_640.

TYPES : BEGIN OF lty_header,
        ono      TYPE zdeono_640,
        odate    TYPE zdeodate_640,
        pm       TYPE zdepm_640,
        ta       TYPE zdeta_640,
        curr     TYPE zdecur_640,
        text1(20) TYPE c, " for Total Text
        text2(20) TYPE c, " For Subtotal Text
      END OF lty_header.

DATA : lt_header TYPE TABLE OF lty_header.

DATA : lo_splitter TYPE REF TO cl_gui_splitter_container,
      lo_alv1      TYPE REF TO cl_gui_alv_grid,
      lt_fieldcat TYPE lvc_t_fcat,
      LT_SORT TYPE LVC_T_SORT,
      LS_LAYO TYPE LVC_S_LAYO.

START-OF-SELECTION.
  SELECT ono odate pm ta curr FROM zord_h_12 INTO TABLE lt_header.
  SORT lt_header BY curr.

  DATA : curr TYPE zord_h_12-curr,
        txt(20) TYPE c.

  LOOP AT lt_header INTO DATA(ls_header).
    ls_header-text1 = 'TOTAL AMOUNT = '.
    IF ls_header-curr NE txt.
      curr = ls_header-curr.
      txt = |SUBTOTAL { curr } |.
    ENDIF.
    LS_HEADER-text2 = txt.
    MODIFY lt_header FROM ls_header.

  ENDLOOP.

  lt_fieldcat = VALUE #(
    ( col_pos = 1 fieldname = 'ONO' scrtext_s = 'Order Number' )
    ( col_pos = 2 fieldname = 'ODATE' scrtext_s = 'Order Date' )
    ( col_pos = 3 fieldname = 'PM' scrtext_s = 'Payment Mode' )
    ( col_pos = 4 fieldname = 'TA' scrtext_s = 'Total Amount' do_sum = 'X')
    ( col_pos = 5 fieldname = 'CURR' scrtext_s = 'Currency' )
    ( col_pos = 6 fieldname = 'TEXT1' no_out = 'X' tech = 'X')
    ( col_pos = 7 fieldname = 'TEXT2' no_out = 'X' tech = 'X') ).

  LT_SORT = VALUE #(
    ( SPOS = 1 FIELDNAME = 'TEXT1' UP = 'X' SUBTOT = 'X')
    ( SPOS = 2 FIELDNAME = 'TEXT2' UP = 'X' SUBTOT = 'X') ).

  LS_LAYO-no_totline = 'X'.
```

```
CALL SCREEN 100.
```

```
INCLUDE ypro_oops_alv1_640_containeo01.
```

```
INCLUDE ypro_oops_alv1_640_displayo01.
```

# Button add

28 September 2025 16:17

```
*&-----*
*& Report YPRO_ADD_BUTTON_640
*&-----*
*&
*&-----*
REPORT ypro_add_button_640.

TYPES : BEGIN OF lty_header,
        ono     TYPE zdeono_640,
        odate   TYPE zdeodate_640,
        pm      TYPE zdepm_640,
        ta      TYPE zdeta_640,
        curr    TYPE zdecur_640,
      END OF lty_header.

DATA : lt_header TYPE TABLE OF lty_header.

DATA : lo_splitter TYPE REF TO cl_gui_splitter_container,
      lo_alv1      TYPE REF TO cl_gui_alv_grid,
      lt_fieldcat  TYPE lvc_t_fcat,
      lt_sort      TYPE lvc_t_sort,
      wa_layout    TYPE lvc_s_layo,
      lt_exclude   TYPE ui_functions,
      lt_hyper     TYPE lvc_t_hype.

CLASS class1 DEFINITION.
  PUBLIC SECTION .
    CLASS-
METHODS handel_toolbar FOR EVENT toolbar OF cl_gui_alv_grid IMPORTING e_obj
ect e_interactive.
    CLASS-
METHODS user_command FOR EVENT user_command OF cl_gui_alv_grid IMPORTING e_
ucomm.
ENDCLASS.
CLASS class1 IMPLEMENTATION.
  METHOD handel_toolbar.
    DATA is_btn TYPE stb_button.

    is_btn-function = 'SAVE'.
    is_btn-icon = icon_system_save.
    is_btn-text = 'SAVE'.
    is_btn-quickinfo = 'SAVE'.
    is_btn-disabled = ' '.
    APPEND is_btn TO e_object->mt_toolbar.

  ENDMETHOD.
  METHOD user_command.
    CASE e_ucomm.

      WHEN 'SAVE'. " WILL TRIGGER WHEN CLICK ON ALV TOOLBAR 'SAVE' BUTTON

*          PERFORM update_data_base.

    ENDCASE.

  ENDMETHOD.
ENDCLASS.
```

```

START-OF-SELECTION.
  SELECT ono odate pm ta curr FROM zord_h_12 INTO TABLE lt_header.
  SORT lt_header BY ono.

  lt_fieldcat = VALUE #(
    ( col_pos = 1 fieldname = 'ONO' scrtext_s = 'Order Number' key = 'X' )
    ( col_pos = 2 fieldname = 'ODATE' scrtext_s = 'Order Date' )
    ( col_pos = 3 fieldname = 'PM' scrtext_s = 'Payment Mode' edit = 'X' )
    ( col_pos = 4 fieldname = 'TA' scrtext_s = 'Total Amount' do_sum = 'X')
    ( col_pos = 5 fieldname = 'CURR' scrtext_s = 'Currency' ) ).

*   wa_layout-no_toolbar = 'X'.
*   wa_layout-EDIT = 'X'.

*lt_exclude = VALUE #( ( cl_gui_alv_grid=>MC_FC_SORT_ASC ) ).

CALL SCREEN 100.

INCLUDE ypro_add_button_640_status_o01.

*-----*
***INCLUDE YPRO_ADD_BUTTON_640_STATUS_O01.
*-----*
*&-----*
*& Module STATUS_0100 OUTPUT
*&-----*
*&
*&-----*
MODULE status_0100 OUTPUT.
* SET PF-STATUS 'xxxxxxxxx'.
* SET TITLEBAR 'xxx'.

CREATE OBJECT lo_splitter
EXPORTING
  parent   = cl_gui_SPLITTER_container=>DEFAULT_screen
  rows     = 1
  columns  = 1.
CALL METHOD lo_splitter->get_container
EXPORTING
  row      = 1
  column   = 1
RECEIVING
  container = DATA(cont1).

CREATE OBJECT lo_alv1
EXPORTING
  i_parent = cont1.

SET HANDLER class1=>handel_toolbar FOR lo_alv1.
SET HANDLER class1=>user_command FOR lo_alv1.

CALL METHOD lo_alv1->set_table_for_first_display
EXPORTING
*   i_buffer_active           =
*   i_bypassing_buffer       =
*   i_consistency_check      =
*   i_structure_name         =
*   is_variant                =
*   i_save                    =
*   i_default                 = 'X'
  is_layout = wa_layout

```

```

*      is_print                                     =
*      it_special_groups                           =
      it_toolbar_excluding                         = lt_exclude
      it_hyperlink                                 = lt_hyper
*      it_alv_graphics                             =
*      it_except_qinfo                             =
*      ir_salv_adapter                             =
CHANGING
      it_outtab                                    = lt_header
      it_fieldcatalog                             = lt_fieldcat
*      it_sort                                       =
*      it_filter                                    =
EXCEPTIONS
      invalid_parameter_combination = 1
      program_error                  = 2
      too_many_lines                 = 3
      OTHERS                         = 4.

```

ENDMODULE.

# FACTORY METHODS

28 September 2025 17:31

```
*&-----*
*& Report YPRO_OOPS_ALV6_FACTORY_640
*&-----*
*&
*&-----*
REPORT ypro_oops_alv6_factory_640.

* SIMILAR TO REUSE_ALV_GRID_DISPLAY FUNCTION MODULE.

TABLES : vbak.

SELECT-OPTIONS : s_vbeln FOR vbak-vbeln.

DATA : gt_final TYPE TABLE OF ystr_order_640.

DATA : lo_alv TYPE REF TO cl_salv_table.

DATA : lo_value TYPE REF TO cl_salv_functions_list.

DATA lo_columns TYPE REF TO cl_salv_columns_table.

DATA : lo_text TYPE REF TO cl_salv_column.

DATA : lo_column TYPE REF TO cl_salv_column.

DATA : lo_sorts TYPE REF TO cl_salv_sorts.

DATA : lo_filters TYPE REF TO cl_salv_filters.

DATA : lo_filter TYPE REF TO cl_salv_filter.

SELECT a~vbeln
       a~erdat
       a~erzet
       a~ernam
       a~vbtyp
       b~posnr
       b~matnr
FROM vbak AS a JOIN vbap AS b
ON a~vbeln = b~vbeln
INTO TABLE Gt_final
WHERE a~vbeln IN s_vbeln.

TRY.
    CALL METHOD cl_salv_table=>factory
    *   EXPORTING
    *       list_display    = IF_SALV_C_BOOL_SAP=>FALSE
    *       r_container    =
    *       container_name =
    IMPORTING
        r_salv_table = lo_alv
    CHANGING
        t_table      = gt_final.
    CATCH cx_salv_msg.
ENDTRY.

CALL METHOD lo_alv->if_salv_gui_on_table_info~get_functions
RECEIVING
```

```

        value = lo_value.

*CALL METHOD lt_value->set_default
* EXPORTING
*   value = IF_SALV_C_BOOL_SAP=>TRUE
*   .

CALL METHOD lo_value->set_all
EXPORTING
    value = if_salv_c_bool_sap=>True.

CALL METHOD lo_alv->if_salv_gui_om_table_info~get_columns
RECEIVING
    value = lo_columns.

CALL METHOD lo_columns->set_column_position
EXPORTING
    columnname = 'ERDAT'
    position   = 3.

CALL METHOD lo_columns->set_column_position
EXPORTING
    columnname = 'ERZET'
    position   = 2.

TRY.
    CALL METHOD lo_columns->get_column
    EXPORTING
        columnname = 'VBELN'
    RECEIVING
        value      = lo_COLUMN.
    CATCH cx_salv_not_found.
ENDTRY.

CALL METHOD lo_column->set_long_text
EXPORTING
    value = 'Documnet Number'.

CALL METHOD lo_alv->if_salv_gui_om_table_info~get_sorts
RECEIVING
    value = lo_sorts.

TRY.
    CALL METHOD lo_sorts->add_sort
    EXPORTING
        columnname = 'VBELN'
*       position   =
        sequence    = if_salv_c_sort=>sort_down
*       subtotal   = IF_SALV_C_BOOL_SAP=>FALSE
*       group      = IF_SALV_C_SORT=>GROUP_NONE
*       obligatory = IF_SALV_C_BOOL_SAP=>FALSE
*   receiving
*       value      = LO_SORT
    .
    CATCH cx_salv_not_found.
    CATCH cx_salv_existing.
    CATCH cx_salv_data_error.
ENDTRY.

*TRY.
*CALL METHOD lo_sorts->add_sort
* EXPORTING
*   columnname = 'ERNAM'

```

```

**      position      =
*      sequence      = IF_SALV_C_SORT=>SORT_UP
**      subtotal      = IF_SALV_C_BOOL_SAP=>FALSE
**      group         = IF_SALV_C_SORT=>GROUP_NONE
***      obligatory   = IF_SALV_C_BOOL_SAP=>FALSE
**      receiving
**      value         = LO_SORT
*      .
*      CATCH cx_salv_not_found.
*      CATCH cx_salv_existing.
*      CATCH cx_salv_data_error.
*ENDTRY.

CALL METHOD lo_alv->if_salv_gui_om_table_info~get_filters
  RECEIVING
    value = lo_filters.

TRY.
  CALL METHOD lo_filters->add_filter
    EXPORTING
      columnname = 'POSNR'
      sign       = 'I'
      option     = 'EQ'
      low        = '10'
*      high      =
  receiving
    value       = lo_filter
  .
  CATCH cx_salv_not_found.
  CATCH cx_salv_data_error.
  CATCH cx_salv_existing.
ENDTRY.

TRY.
CALL METHOD lo_filter->add_selopt
  EXPORTING
    sign   = 'I'
    option = 'EQ'
    low    = '20'
*      high =
*  RECEIVING
*    value =
  .
  CATCH cx_salv_data_error.
ENDTRY.

TRY.
CALL METHOD lo_filter->add_selopt
  EXPORTING
    sign   = 'I'
    option = 'EQ'
    low    = '40'
*      high =
*  RECEIVING
*    value =
  .
  CATCH cx_salv_data_error.
ENDTRY.

CALL METHOD lo_alv->if_salv_gui_om_table_action~display.

*****START OF HOT SPOT ON *****
o_cols = o_salv->get_columns( ).
TRY.
  o_col ?= o_cols->get_column( 'VBELN' ). " sales reder number

```



```

    CATCH cx_salv_not_found.
ENDTRY.
o_col->set_cell_type( if_salv_c_cell_type=>hotspot ).
* o_col->set_cell_type( value = 5 ).
*****END OF HOT SPOT ON *****
*****START OF HIDING A COLUMN *****
TRY.
    o_col ?= o_cols->get_column( 'ERZET' ). " Entry time
    CATCH cx_salv_not_found.
ENDTRY.
o_col->set_visible( value = if_salv_c_bool_sap=>false ).
*****END OF HIDING A COLUMN *****

```

From <[https://sapcodes.com/2015/11/20/hiding-a-column-with-cl\\_salv\\_table/](https://sapcodes.com/2015/11/20/hiding-a-column-with-cl_salv_table/)>

# Color Rows

29 September 2025 12:09

```
*&-----*
*& Report ZPRO_SAMPLE4_640
*&-----*
*&
*&-----*
REPORT zpro_sample4_640.

TYPES: BEGIN OF lty_header,
        ono      TYPE zord_h_12-ono, "order number
        odate    TYPE zord_h_12-odate, "order date
        pm       TYPE zord_h_12-pm, "payment mode
        ta       TYPE zord_h_12-ta, "total amount
        curr     TYPE zord_h_12-curr, "currency
        color(4),
    END OF lty_header.

DATA : lt_header TYPE TABLE OF lty_header,
      ls_header TYPE lty_header.
DATA : lo_alv1    TYPE REF TO cl_gui_alv_grid,
      lo_splitter TYPE REF TO cl_gui_splitter_container.

DATA : ls_layout TYPE lvc_s_layo.

DATA : lt_filedcat TYPE lvc_t_fcat.

START-OF-SELECTION.
    SELECT  ono
           odate
           pm
           ta
           curr
    FROM zord_h_12 INTO TABLE lt_header.

    SORT lt_header BY ono.

    CONCATENATE 'C' col_group '10' INTO ls_header-color.
    MODIFY lt_header FROM ls_header TRANSPORTING color WHERE pm = 'D'.

    CLEAR ls_header.

    CONCATENATE 'C' col_key '10' INTO ls_header-color.
    MODIFY lt_header FROM ls_header TRANSPORTING color WHERE pm = 'N'.

    lt_filedcat = VALUE #(
        ( col_pos = 1 fieldname = 'ONO' tabname = 'LT_HEADER' )
        ( col_pos = 2 fieldname = 'ODATE' tabname = 'LT_HEADER' )
        ( col_pos = 3 fieldname = 'PM' tabname = 'LT_HEADER' )
        ( col_pos = 4 fieldname = 'TA' tabname = 'LT_HEADER' )
        ( col_pos = 5 fieldname = 'CURR' tabname = 'LT_HEADER' ) ).

    ls_layout-info_fname = 'COLOR'.

END-OF-SELECTION.
    CALL SCREEN 50.

    INCLUDE zpro_sample4_640_status_005o01.
```

```

*-----*
***INCLUDE ZPRO_SAMPLE4_640_STATUS_005001.
*-----*
*&-----*
*& Module STATUS_0050 OUTPUT
*&-----*
*&
*&-----*
MODULE status_0050 OUTPUT.
* SET PF-STATUS 'xxxxxxxx'.
* SET TITLEBAR 'xxx'.

CREATE OBJECT lo_splitter
EXPORTING
    parent = cl_gui_container=>screen0
    rows   = 1
    columns = 1.
CALL METHOD lo_splitter->get_container
EXPORTING
    row       = 1
    column    = 1
RECEIVING
    container = DATA(cont1).

CREATE OBJECT lo_alv1
EXPORTING
    i_parent = cont1.
CALL METHOD lo_alv1->set_table_for_first_display
EXPORTING
*      i_buffer_active           =
*      i_bypassing_buffer       =
*      i_consistency_check      =
*      i_structure_name         = 'ZORD_H_12'
*      is_variant               =
*      i_save                   =
*      i_default                = 'X'
*      is_layout                = ls_layout
*      is_print                 =
*      it_special_groups        =
*      it_toolbar_excluding     =
*      it_hyperlink             =
*      it_alv_graphics          =
*      it_except_qinfo          =
*      ir_salv_adapter          =
CHANGING
    it_outtab = lt_header
    it_fieldcatalog = lt_filedcat
*      it_sort =
*      it_filter =
EXCEPTIONS
    invalid_parameter_combination = 1
    program_error                 = 2
    too_many_lines                 = 3
OTHERS                            = 4.

ENDMODULE.

```

# Show Edit Data in Separate

12 December 2025 15:10

```
*&-----*
*& Report Z633_OOPS_EDITABLE_DOWNL_ALV
*&-----*
*&
*&-----*
REPORT z633_oops_editable_downl_alv.
TABLES zordrh_27.

CLASS cl_editable DEFINITION.
  PUBLIC SECTION.
    DATA : gt_h_data TYPE ztordrh_27,
            gs_h_data TYPE zordrh_27,
            gt_fieldcat TYPE lvc_t_fcat,
            gs_fieldcat TYPE lvc_s_fcat,
            lo_alv_grid TYPE REF TO cl_gui_alv_grid.

    DATA: lo_grid TYPE REF TO cl_salv_form_layout_grid.

    DATA: gt_edited_keys TYPE SORTED TABLE OF zordrh_27-ordernum
            WITH UNIQUE KEY table_line.

    METHODS : display_alv IMPORTING iv_ono TYPE zt633
      _ordernum_selectoption,
      fetch_data IMPORTING iv_ono TYPE zt633_ordernum_selectoption
        EXPORTING et_h TYPE STANDARD TABLE,
      fieldcatlog,
      save_data,
      download_data.

    METHODS: handle_data_changed
      FOR EVENT data_changed OF cl_gui_alv_grid
        IMPORTING er_data_changed.

    METHODS get_edited_rows.

ENDCLASS.

CLASS cl_editable IMPLEMENTATION.
  METHOD fetch_data.
    SELECT *
      FROM zordrh_27
      INTO TABLE et_h
      WHERE ordernum IN iv_ono.
    SORT et_h.
  ENDMETHOD.

  METHOD fieldcatlog.
    gs_fieldcat-col_pos = '1'.
    gs_fieldcat-fieldname = 'ORDERNUM'.
    gs_fieldcat-scrtext_m = 'Order_num'.
    APPEND gs_fieldcat TO gt_fieldcat.
    CLEAR gs_fieldcat.

    gs_fieldcat-col_pos = '2'.
    gs_fieldcat-fieldname = 'ODATE'.
    gs_fieldcat-scrtext_m = 'Order_Date'.
  
```

```

APPEND gs_fieldcat TO gt_fieldcat.
CLEAR gs_fieldcat.

gs_fieldcat-col_pos ='3'.
gs_fieldcat-fieldname ='PAYMENTMETHOD'.
gs_fieldcat-scrtext_m ='Payment_method'.
gs_fieldcat-edit = 'X'.
APPEND gs_fieldcat TO gt_fieldcat.
CLEAR gs_fieldcat.

gs_fieldcat-col_pos ='4'.
gs_fieldcat-fieldname ='TOTALAMOUNT'.
gs_fieldcat-scrtext_m ='Grand_total'.
gs_fieldcat-edit = 'X'.
APPEND gs_fieldcat TO gt_fieldcat.
CLEAR gs_fieldcat.

gs_fieldcat-col_pos ='5'.
gs_fieldcat-fieldname ='CURRENCY'.
gs_fieldcat-scrtext_m ='Currency'.
APPEND gs_fieldcat TO gt_fieldcat.
CLEAR gs_fieldcat.

ENDMETHOD.

METHOD display_alv.

me->fetch_data(
    EXPORTING
        iv_ono = iv_ono
    IMPORTING
        et_h   = gt_h_data
).

me->fieldcatlog( ).

DATA(lo_logo_CONTAINER) = NEW cl_gui_custom_container(
    container_name           = 'LOGO' " Name of the Screen CustCtrl Name to Link Container To
).

DATA(lo_logo) = NEW cl_dd_document( ).
lo_logo->add_picture(
    picture_id               = 'ZLT626_LT_LOGO'
).

lo_logo->display_document(
    EXPORTING
        parent               = lo_logo_container " Container Object Already Exists
).

DATA(lo_top_container) = NEW cl_gui_custom_container( container_name = 'TOP' ).
DATA(lo_top) = NEW cl_dd_document( ).

lo_top->new_line( ).
lo_top->add_text(
    text                     = 'Order Details Report'

```

```

        sap_style = cl_dd_document=>heading ).

lo_top->new_line( ).
lo_top->add_text( text = |Executed By: { sy-uname }| ).

lo_top->new_line( ).
lo_top->add_text( text = |Date: { sy-datlo DATE = ISO } / Time: { sy-
timlo TIME = ISO }| ).

lo_top->display_document(
    EXPORTING
        parent          = lo_top_container          " Contain Obj
ect Already Exists
    ).

DATA(lo_container) = NEW cl_gui_custom_container( container_name = 'CON
T' ).
lo_alv_grid = NEW cl_gui_alv_grid(
    i_parent          = lo_container          " Parent Conta
iner
    ).

SET HANDLER me->handle_data_changed FOR lo_alv_grid.

lo_alv_grid->register_edit_event(
    EXPORTING i_event_id = cl_gui_alv_grid=>mc_evt_enter
    ).

lo_alv_grid->set_table_for_first_display(
*   EXPORTING
*   is_layout          = ls_layout
    CHANGING
        it_outtab          = gt_h_data          " Output
Table
        it_fieldcatalog    = gt_fieldcat
    ).

ENDMETHOD.

METHOD handle_data_changed.

DATA: ls_mod_cell TYPE lvc_s_modi,
      ls_data      TYPE zordrh_27.

CLEAR gt_edited_keys.
LOOP AT er_data_changed->mt_mod_cells INTO ls_mod_cell.

    READ TABLE gt_h_data INTO ls_data INDEX ls_mod_cell-row_id.

    IF sy-subrc = 0.
        INSERT ls_data-ordernum INTO TABLE gt_edited_keys.
    ENDIF.
ENDLOOP.
ENDMETHOD.

METHOD save_data.
*does NOT automatically update your internal table.
*The new value stays only in the frontend buffer until you explicitly synch
ronize it.
*forces the ALV to validate and transfer the modified cell values back to y
our internal table.
lo_alv_grid->check_changed_data( ).
LOOP AT gt_h_data INTO gs_h_data.

```

```

        UPDATE zordrh_27
            SET totalamount = gs_h_data-totalamount
              paymentmethod = gs_h_data-paymentmethod
            WHERE ordernum  = gs_h_data-ordernum.
    ENDLOOP.
    IF sy-subrc = 0.
        MESSAGE 'updated successfully' TYPE 'S'.
    ELSE.
        MESSAGE 'error in update' TYPE 'E'.
    ENDIF.

    COMMIT WORK.

ENDMETHOD.

METHOD download_data.
    DATA: lv_filename TYPE string,
           lv_path      TYPE string,
           lv_fullpath  TYPE string,
           lv_action    TYPE i.

    " Show save dialog
    CALL METHOD cl_gui_frontend_services=>file_save_dialog
        EXPORTING
            window_title           = 'Save Order Data'
            default_extension      = 'xls'
            default_file_name      = 'orders.xls'
            prompt_on_overwrite    = 'X'
        CHANGING
            filename               = lv_filename
            path                   = lv_path
            fullpath               = lv_fullpath
            user_action            = lv_action
        EXCEPTIONS
            cntl_error             = 1
            error_no_gui           = 2
            not_supported_by_gui   = 3
            invalid_default_file_name = 4
            OTHERS                 = 5.

    IF sy-subrc <> 0 OR lv_action <> cl_gui_frontend_services=>action_ok.
        MESSAGE 'File save canceled' TYPE 'I'.
        RETURN.
    ENDIF.

    " Download internal table to chosen file
    CALL METHOD cl_gui_frontend_services=>gui_download
        EXPORTING
            filename               = lv_fullpath
            filetype               = 'DAT' " or 'CSV'
        CHANGING
            data_tab               = gt_h_data
        EXCEPTIONS
            file_write_error       = 1
            no_batch               = 2
            gui_refuse_filetransfer = 3
            invalid_type           = 4
            no_authority           = 5
            unknown_error          = 6
            header_not_allowed     = 7
            separator_not_allowed  = 8
            filesize_not_allowed   = 9
            dp_error_create        = 10
            dp_error_send          = 11

```

```

        dp_error_write           = 12
        unknown_dp_error        = 13
        access_denied           = 14
        dp_out_of_memory         = 15
        disk_full                = 16
        dp_timeout               = 17
        OTHERS                   = 18.

IF sy-subrc = 0.
    MESSAGE 'File downloaded successfully' TYPE 'S'.
ELSE.
    MESSAGE 'Error during file download' TYPE 'E'.
ENDIF.
ENDMETHOD.

METHOD get_edited_rows.
    DATA: lt_final_edited_records TYPE STANDARD TABLE OF zordrh_27.

    lo_alv_grid->check_changed_data( ).

    IF gt_edited_keys IS INITIAL.
        MESSAGE 'No rows have been edited.' TYPE 'S'.
        RETURN.
    ENDIF.

    SELECT * FROM zordrh_27
        INTO TABLE lt_final_edited_records
        FOR ALL ENTRIES IN gt_edited_keys
        WHERE ordernum = gt_edited_keys-table_line.

    cl_demo_output=>display( lt_final_edited_records ).
    refresh lt_final_edited_records.

ENDMETHOD.

ENDCLASS.

SELECT-OPTIONS s_ono FOR zordrh_27-ordernum.

START-OF-SELECTION.
    DATA(lo_editable) = NEW cl_editable( ).
    lo_editable->display_alv( iv_ono = s_ono[] ).

    CALL SCREEN 100.

    INCLUDE z633_oops_editable_pbo_pbi.

*-----*
***INCLUDE Z633_OOPS_EDITABLE_PBO_PBI.
*-----*
*&-----*
*& Module STATUS_0100 OUTPUT
*&-----*
*&
*&-----*
MODULE status_0100 OUTPUT.
    SET PF-STATUS 'EDITABLE_ALV'.
    * SET TITLEBAR 'xxx'.
ENDMODULE.
*&-----*
*& Module USER_COMMAND_0100 INPUT
*&-----*
* text
*-----*

```



```

MODULE user_command_0100 INPUT.
CASE sy-ucomm.

    WHEN 'BACK'.
        LEAVE TO SCREEN 0.

    WHEN 'SAVE'.
        lo_editable->save_data( ).

    WHEN 'DOWNLOAD'.

        DATA lv_answer TYPE c.
        CALL FUNCTION 'POPUP_TO_CONFIRM'
            EXPORTING
                text_question = 'Do you want to save changes before downloading?'
                text_button_1 = 'Yes'
                text_button_2 = 'No'
                default_button = '1'
            IMPORTING
                answer = lv_answer.

        IF lv_answer = '1'.
            lo_editable->save_data( ).
            lo_editable->download_data( ).
        ELSE.

        ENDIF.

    WHEN 'EDIT_ROWS'.

        lo_editable->get_edited_rows( ).

ENDCASE.

ENDMODULE.

```

# Edit Row

12 December 2025 15:09

Table Type	YTT_ORD	Active
Short text	table type	
<div> <div>Attributes</div> <div>Line Type</div> <div>Initialization and Access</div> <div>Primary Key</div> <div>Secondary Key</div> </div>		
Line Type	YSTR ORDS 640	

Structure

YSTR\_ORDS\_640

Active

Short Description

str

Attributes

Components

Input Help/Check

Currency/quantity fields

```

*&-----*
*& Report YALV_EDIT_ROW1_640
*&-----*
*&
*&-----*
REPORT yalv_edit_row1_640.

DATA : lo_cont      TYPE REF TO cl_gui_custom_container,
       lo_alv       TYPE REF TO cl_gui_alv_grid,
       lt_fieldcat   TYPE lvc_t_fcat,
       lt_data       TYPE ytt_hord_640,
       gs_layout     TYPE lvc_s_layo,
       lt_index_rows TYPE lvc_t_row,
       lt_row_no     TYPE lvc_t_roid,
       ls_row_no     TYPE lvc_s_roid.

DATA : lt_header TYPE ytt_ord.

DATA: ls_stylerow TYPE lvc_s_styl,
      lt_stylerow TYPE TABLE OF lvc_s_styl.

CLASS class1 DEFINITION.
  PUBLIC SECTION.
    CLASS-
METHODS : buttons FOR EVENT toolbar OF cl_gui_alv_grid IMPORTING e_object e
_interactive,
          user_comd FOR EVENT user_command OF cl_gui_alv_grid IMPORTING e_ucomm
,
          change_data1 FOR EVENT data_changed OF cl_gui_alv_grid .

ENDCLASS.

CLASS class1 IMPLEMENTATION.
  METHOD buttons.
    *
    *ADDING A SAVE BUTTONN TO THE ALV TOOLBAR *
    DATA : it_btn TYPE TABLE OF stb_button.

    it_btn = VALUE #(
      ( function = 'Save' icon = icon_system_save text = 'Save' quickinfo = '
Save' disabled = ' ' )
      ( function = 'Edit' icon = icon_change_text text = 'Edit' quickinfo = '
Edit Record' disabled = ' ' )
    ).
    APPEND LINES OF it_btn TO e_object->mt_toolbar.
  ENDMETHOD.

  METHOD user_comd.

    CASE e_ucomm.
      WHEN 'Edit'.
        PERFORM edit_row.
    
```

```

        WHEN 'Save'.
            PERFORM save_edited_record.
        ENDCASE.

    ENDMETHOD.

METHOD change_data1.

ENDMETHOD.

ENDCLASS.

START-OF-SELECTION.

    CALL SCREEN 100.

*-----*
* Module STATUS_0100 OUTPUT
*-----*
*
*-----*
MODULE status_0100 OUTPUT.
    SET PF-STATUS 'STATUS_PF'.
    * SET TITLEBAR 'xxx'.

    CREATE OBJECT lo_cont
        EXPORTING
            container_name = 'CONTAINER'.

    CREATE OBJECT lo_alv
        EXPORTING
            i_parent = lo_cont.          " Parent Container

    ycl_singleton_class1_640=>get_instance(
        RECEIVING
            r_object = DATA(lo_obj)    " SINGLETON_CLASS
    ).

    lo_obj->get_data(
        RECEIVING
            r_data = lt_data            " table type
    ).

    lt_header = CORRESPONDING #( lt_data ).
    lt_fieldcat = VALUE #( ( col_pos = 1 fieldname = 'ONO' scrtext_1 = 'Order
Number' )
        ( col_pos = 2 fieldname = 'ODATE' scrtext_1 = 'Date'
        )
        ( col_pos = 3 fieldname = 'PM' scrtext_1 = 'Payment
Mode' )
        ( col_pos = 4 fieldname = 'TA' scrtext_1 = 'Amount'
        )
        ( col_pos = 5 fieldname = 'CURR' scrtext_1 = 'Curren
cy' ) ).

    gs_layout-sel_mode = 'A'.
    gs_layout-stylefname = 'CELLSTYLE'.
    * gs_layout-no_ = 'X'.

    SET HANDLER class1=>buttons FOR lo_alv.
    SET HANDLER class1=>user_comd FOR lo_alv.

    CALL METHOD lo_alv->set_table_for_first_display
        EXPORTING
            is_layout      = gs_layout
            i_default      = 'X'
    *i_structure_name = 'MARA'
        CHANGING
            it_outtab      = lt_header
            it_fieldcatalog = lt_fieldcat.

ENDMODULE.

INCLUDE yalv_edit_row1_640_user_comi01.
*-----*
* Form EDIT_ROW
*-----*
* text
*-----*
* --> p1      text
* <-- p2      text

```

```

*&-----*
FORM edit_row .

CALL METHOD lo_alv->get_selected_rows
IMPORTING
*   et_index_rows = lt_index_rows
    et_row_no = lt_row_no.

IF lt_row_no IS NOT INITIAL.
  LOOP AT lt_header INTO DATA(ls_data).
    READ TABLE lt_row_no INTO ls_row_no WITH KEY row_id = sy-tabix.
    IF sy-subrc = 0.

      lt_stylerow = VALUE #(
        ( fieldname = 'ONO'   style = cl_gui_alv_grid=>mc_style_disabled )
        ( fieldname = 'ODATE' style = cl_gui_alv_grid=>mc_style_enabled  )
        ( fieldname = 'PM'    style = cl_gui_alv_grid=>mc_style_enabled  )
        ( fieldname = 'TA'    style = cl_gui_alv_grid=>mc_style_enabled  )
        ( fieldname = 'CURR'  style = cl_gui_alv_grid=>mc_style_enabled  )
      ).

      CLEAR: ls_DATA-cellstyle.
      INSERT LINES OF lt_stylerow INTO TABLE ls_DATA-cellstyle.
      MODIFY lt_header FROM ls_DATA TRANSPORTING cellstyle.
      CLEAR: lt_stylerow.

    ELSE.

      lt_stylerow = VALUE #(
        ( fieldname = 'ONO'   style = cl_gui_alv_grid=>mc_style_disabled )
        ( fieldname = 'ODATE' style = cl_gui_alv_grid=>mc_style_disabled )
        ( fieldname = 'PM'    style = cl_gui_alv_grid=>mc_style_disabled )
        ( fieldname = 'TA'    style = cl_gui_alv_grid=>mc_style_disabled )
        ( fieldname = 'CURR'  style = cl_gui_alv_grid=>mc_style_disabled )
      ).

      CLEAR: ls_DATA-cellstyle.
      INSERT LINES OF lt_stylerow INTO TABLE ls_DATA-cellstyle.
      MODIFY lt_header FROM ls_DATA TRANSPORTING cellstyle.
      CLEAR: lt_stylerow.

    ENDIF.

  ENDLOOP.
ELSE .
  MESSAGE |Please Select a Record | TYPE 'S'.
ENDIF.

CLEAR lt_row_no.
CLEAR ls_row_no.

IF lo_alv->is_ready_for_input( ) EQ 0.
  CALL METHOD lo_alv->set_ready_for_input
  EXPORTING
    i_ready_for_input = 1.
ELSE.
  CALL METHOD lo_alv->set_ready_for_input
  EXPORTING
    i_ready_for_input = 0.
ENDIF.
* lo_alv->refresh_table_display( ).

ENDFORM.
*&-----*
*& Form save_edited_record
*&-----*
*& text
*&-----*
*& --> p1      text
*& <-- p2      text
*&-----*
FORM save_edited_record .

* lo_alv->check_changed_data( ).
* CALL METHOD lo_alv->get_selected_rows
*   IMPORTING
**   et_index_rows = lt_index_rows
*   et_row_no = lt_row_no.
* IF lt_index_rows IS NOT INITIAL.
*   DATA(ls_row) = lt_index_rows[ 1 ].
*
*   READ TABLE lt_header INTO DATA(wa_header) INDEX ls_row_no-row_id .
*
*   DATA(wa_header1) = VALUE zord_h_12( ono = wa_header-

```

```

ono odate = wa_header-odate pm = wa_header-pm
*
*                               ta = wa_header-
ta curr = wa_header-curr ) .
*   IF sy-subrc = 0.
*       UPDATE ZORD_h_12 FROM wa_header1.
*       MESSAGE |Record { wa_header1-ono } is Updated | TYPE 'S'.
*   ENDIF.
*   lo_alv->refresh_table_display( ).
*
*   ENDIF.

lt_data = CORRESPONDING #( lt_header ).
UPDATE zord_h_12 FROM TABLE lt_data.
IF sy-subrc = 0.
    MESSAGE |Record is Updated | TYPE 'S'.
ELSE.
    MESSAGE |Record is Fail Updated | TYPE 'E'.
ENDIF.
LOOP AT lt_header INTO DATA(ls_data).
    lt_stylerow = VALUE #(
    ( fieldname = 'ONO'   style = cl_gui_alv_grid=>mc_style_disabled )
    ( fieldname = 'ODATE' style = cl_gui_alv_grid=>mc_style_disabled )
    ( fieldname = 'PM'   style = cl_gui_alv_grid=>mc_style_disabled )
    ( fieldname = 'TA'   style = cl_gui_alv_grid=>mc_style_disabled )
    ( fieldname = 'CURR' style = cl_gui_alv_grid=>mc_style_disabled )
    ).

    CLEAR: ls_DATA-cellstyle.
    INSERT LINES OF lt_stylerow INTO TABLE ls_DATA-cellstyle.
    MODIFY lt_header FROM ls_DATA TRANSPORTING cellstyle.
    CLEAR: lt_stylerow.
ENDLOOP.

lo_alv->refresh_table_display( ).

ENDFORM.

```

# Edit Row In POPUP

12 December 2025

15:12