

Below are the list of OO ALV tutorials in SAP ABAP.You will find all the tutorials with source code and outputs.

Object Oriented ALV’s can be built using two standard classes.

* CL\_SALV\_TABLE is the factory class and easy to use to built ALV reports in ABAP.
* CL\_GUI\_ALV\_GRID is other class which can also be used to built OOP ALV reports in ABAP.

CL\_SALV\_TABLE is the factory class :

First, we define a global structure to store the data we are going to show in the ALV.

types: begin of ty\_material,

matnr type matnr,

maktx type maktx,

end of ty\_material.

data: g\_matnr like mara-matnr,

gt\_material type standard table of ty\_material.

Second step. We define a ‘handler’ class whose purpose is to control the double click event. In the class definition, we code the signature of the event double click and, in the implementation, we code the action or process that we want to happen when the event is raised.

class cl\_handler definition.

public section.

methods on\_double\_click for event double\_click of cl\_salv\_events\_table

importing row column.

endclass. "cl\_handler DEFINITION

class cl\_handler implementation.

method on\_double\_click.

if column eq 'MATNR'.

read table gt\_material into data(wa\_st\_data) index row.

\* Check that material exists

select count( \* ) from mara up to 1 rows where matnr eq wa\_st\_data-matnr.

if sy-subrc = 0. " Exists?

\* Load parameters

set parameter id 'MXX' field 'K'. " Default view

set parameter id 'MAT' field wa\_st\_data-matnr. " Material number

call transaction 'MM03' and skip first screen.

else. " No ?

call function 'CONVERSION\_EXIT\_ALPHA\_OUTPUT'

exporting

input = wa\_st\_data-matnr

importing

output = wa\_st\_data-matnr.

data(lv\_err) = `Material ` && wa\_st\_data-matnr && ` does not exist.`.

message lv\_err type 'I' display like 'E'.

endif.

else.

message text-002 type 'I'. " Invalid cell

endif.

endmethod.

endclass.

In this case, we will execute transaction ‘MM03’ and we will send the material code. Here we make sure that the parameter ‘column’ is ‘MATNR’, which is from the structure ‘ty\_material’ that we defined at the beginning of this program.

Then we verify that the material exists seeking it at table ‘MARA’. MARA is a standard table that stores the materials data.

If the material exists then the next step is to load the transaction settings in memory and finally execute it.

Otherwise, we show a message in the screen to indicate that material doesn’t exist.

3rd

Third step: Create the input screen. We have only one parameter; ‘s\_matnr’. S\_MATNR is a SELECT-OPTIONS structure where we can specify diverse criteria in order to use it in a SQL statement or at internal tables.

selection-screen: begin of block b1 with frame title text-001.

select-options: so\_matnr for g\_matnr.

selection-screen: end of block b1.

start-of-selection.

perform retrieve\_data.

perform display.

Fourth step: We then code the forms ‘retrieve\_data’ and ‘display’.

form retrieve\_data.

select matnr maktx from makt up to 100 rows " Retrieve only 100 records

into table gt\_material

where matnr in so\_matnr and spras eq sy-langu.

endform.

For example purposes, we’ll limit 100 max rows of the MARA table in the ‘retrieve\_data’ form.

form display.

data: lo\_gr\_alv type ref to cl\_salv\_table, " Variables for ALV properties

lo\_gr\_functions type ref to cl\_salv\_functions\_list.

data: lo\_event\_handler type ref to cl\_handler, " Variables for events

lo\_events type ref to cl\_salv\_events\_table.

data: lo\_grid type ref to cl\_salv\_form\_layout\_grid, " Variables for header

lo\_layout\_logo type ref to cl\_salv\_form\_layout\_logo,

lo\_content type ref to cl\_salv\_form\_element,

lv\_title type string,

lv\_rows type string.

data: lo\_layout type ref to cl\_salv\_layout, " Variables for enabling Save button

lv\_key type salv\_s\_layout\_key.

data: lo\_display type ref to cl\_salv\_display\_settings. " Variable for layout settings

data: lo\_selections type ref to cl\_salv\_selections, " Variables for selection mode and column properties

lo\_columns type ref to cl\_salv\_columns,

lo\_column type ref to cl\_salv\_column\_table.

\* Create the ALV object

try.

call method cl\_salv\_table=>factory

importing

r\_salv\_table = lo\_gr\_alv

changing

t\_table = gt\_material.

catch cx\_salv\_msg.

endtry.

\* Let's show all default buttons of ALV

lo\_gr\_functions = lo\_gr\_alv->get\_functions( ).

lo\_gr\_functions->set\_all( abap\_true ).

\* Fit the columns

lo\_columns = lo\_gr\_alv->get\_columns( ).

lo\_columns->set\_optimize( 'X' ).

\* Create header

describe table gt\_material lines lv\_rows.

concatenate 'Number of lv\_rows: ' lv\_rows into lv\_title separated by space.

create object lo\_grid.

create object lo\_layout\_logo.

lo\_grid->create\_label( row = 1 column = 1 text = lv\_title tooltip = lv\_title ).

lo\_layout\_logo->set\_left\_content( lo\_grid ).

lo\_content = lo\_layout\_logo.

lo\_gr\_alv->set\_top\_of\_list( lo\_content ).

\* Apply zebra style to lv\_rows

lo\_display = lo\_gr\_alv->get\_display\_settings( ).

lo\_display->set\_striped\_pattern( cl\_salv\_display\_settings=>true ).

\* Enable the save layout buttons

lv\_key-report = sy-repid.

lo\_layout = lo\_gr\_alv->get\_layout( ).

lo\_layout->set\_key( lv\_key ).

lo\_layout->set\_save\_restriction( if\_salv\_c\_layout=>restrict\_none ).

lo\_layout->set\_default( abap\_true ).

\* Register events

lo\_events = lo\_gr\_alv->get\_event( ).

create object lo\_event\_handler.

set handler lo\_event\_handler->on\_double\_click for lo\_events.

\* Enable cell selection mode

lo\_selections = lo\_gr\_alv->get\_selections( ).

lo\_selections->set\_selection\_mode( if\_salv\_c\_selection\_mode=>row\_column ).

try.

lo\_column ?= lo\_columns->get\_column( 'MAKTX' ). " Find the 'MAKTX' column ans change attributes

lo\_column->set\_visible( if\_salv\_c\_bool\_sap=>true ).

lo\_column->set\_long\_text( 'MyTitle' ).

lo\_column->set\_medium\_text( 'MyTitle' ).

lo\_column->set\_short\_text( 'MyTitle' ).

catch cx\_salv\_not\_found.

catch cx\_salv\_existing.

catch cx\_salv\_data\_error.

endtry.

lo\_gr\_alv->display( ).

endform.

The ‘display’ form is going to show the data in the ALV where we define several settings that are documented in the code itself. Just copy and paste it and try it.

An interesting property about CL\_SALV\_TABLE is that even though you order or filter the grid the internal table ‘it\_material’ reorganizes itself, so the ‘double\_click’ event will always be accurate to load the selected material.

So, this is it. An easy way to make reports in ABAP.

You can find the code in my [github repository](https://github.com/nmirandaghn/all_my_abap/tree/master/report_with_cl_salv_table" \t "_blank).