ALV Report without creating screen using OOPS ABAP

Author: Sneha Jadhav
Submitted: 09/03/2015

When I started learning OOPS ABAP the very first assignment I got to work on is creation of ALV. I searched on google which class should I use for creation of ALV using OOPS concepts. I found one class cl_gui_alv_grid and its sample code. I started understanding code given in the example but it was very time consuming for me to understand the exact steps because using that class we have to create screens and containers. I got succeeded developing the ALV using class cl_gui_alv_grid. After that I got thought of developing the ALV without creation of screen. But unfortunately I didn't find any such example.

This document will explain the class and steps to be followed which will help us to develop the ALV without creating class.

Class Name: cl_SALV_Table

Description:
Create a ALV report using OOPS concepts for displaying the materials from table MARA. after double clicking on each material number you should be navigated to MM03 TCODE.

Code:

REPORT yat_alvoops_salv.

TYPES :
  BEGIN OF ty_mara,    " structure of table mara
    matnr TYPE matnr,
    mtart TYPE mtart,
    matkl TYPE matkl,
    meins TYPE meins,
    brgew TYPE brgew,
    ntgew TYPE ntgew,
  END OF ty_mara,

  BEGIN OF ty_makt,     " structure of table makt
    matnr TYPE matnr,
    maktx TYPE maktx,
  END OF ty_makt,

  BEGIN OF ty_final,    " structure of final output table
    matnr TYPE matnr,
    mtart TYPE mtart,
    matkl TYPE matkl,
    meins TYPE meins,
    brgew TYPE brgew,
    ntgew TYPE ntgew,
    maktx TYPE maktx,
  END OF ty_final.

DATA :
  it_mara TYPE TABLE OF ty_mara,
  wa_mara TYPE ty_mara,
  it_makt TYPE TABLE OF ty_makt,
  wa_makt TYPE ty_makt,
  it_final TYPE STANDARD TABLE OF ty_final,
  wa_final TYPE ty_final,
  o_alv TYPE REF TO cl_salv_table,  "Object of class cl_salv_table
  lv_msg TYPE REF TO cx_salv_msg, ="#EC NEEDED "Catching exceptions
  o_function TYPE REF TO cl_salv_functions_list,  "For setting PF-Status
  If_events TYPE REF TO cl_salv_events_table,  "For handling double click event

SOURCE CODE:
PUBLIC SECTION.
METHODS:
  on_double_click FOR EVENT double_click OF cl_salv_events_table
    IMPORTING row.
ENDCLASS.  "lcl_handle_events DEFINITION

*----------------------------------------------------------------------*
*       CLASS lcl_handle_events IMPLEMENTATION
*----------------------------------------------------------------------*
**   Double click event Handling
*----------------------------------------------------------------------*
CLASS lcl_handle_events IMPLEMENTATION.
METHOD on_double_click.
  DATA: row_c(4) TYPE c.

  row_c = row.
  READ TABLE it_final INDEX row_c INTO wa_final. "it_output is the output internal table.
  SET PARAMETER ID 'MAT' FIELD wa_final-matnr.
  CALL TRANSACTION 'MM03' AND SKIP FIRST SCREEN.
ENDMETHOD. "on_double_click
ENDCLASS.  "lcl_handle_events IMPLEMENTATION

START-OF-SELECTION.
SELECT matnr mtart matkl means brgew ntgew FROM mara INTO TABLE it_mara UP TO 10 ROWS. "Fetching upto 10 rows from table MARA
SELECT matnr maktx FROM makt INTO TABLE it_makt FOR ALL ENTRIES IN it_mara WHERE matnr = it_mara-matnr. "Fectching material description from Makt table
CLEAR it_final.
"Filling Final Internal table
LOOP AT it_mara INTO wa_mara.
  READ TABLE it_makt INTO wa_makt WITH KEY matnr = wa_mara-matnr.
  MOVE-CORRESPONDING wa_mara TO wa_final.
  MOVE-CORRESPONDING wa_makt TO wa_final.
  APPEND wa_final TO it_final.
  CLEAR: wa_final, wa_mara, wa_makt.
ENDLOOP.
" Calling Factory Method of the class, it will return the ALV object
TRY.
  CALL METHOD cl_salv_table=>factory
    IMPORTING
      r_salv_table = o_alv
    CHANGING
      t_table = it_final.
  CATCH cx_salv_msg INTO lv_msg. "#EC NO_HANDLER
ENDTRY.

"Double click event handling(Calling the method)
DATA: lo_event_handler TYPE REF TO lcl_handle_events.
If_events = o_alv->get_event( ).
CREATE OBJECT lo_event_handler.
SET HANDLER lo_event_handler->on_double_click FOR If_events.

"Setting Default PF-Status
o_function = o_alv->get_functions( ).
o_function->set_all( ).

* Calling Display method
CALL METHOD o_alv->display( ).

output: