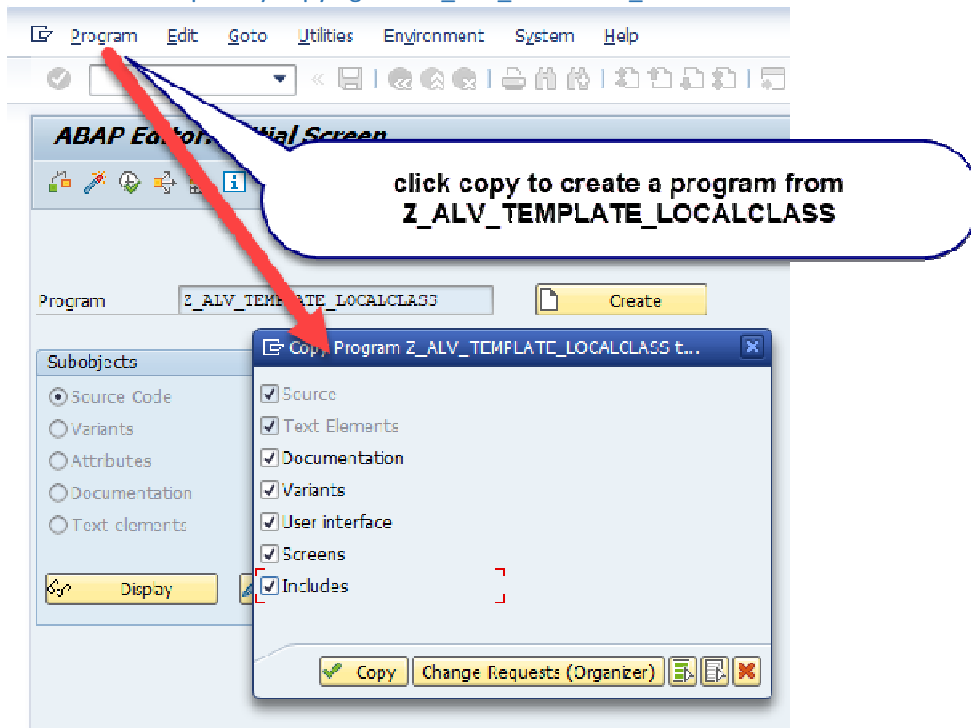# Create an ALV Report in No Time

Wednesday, April 3, 2019   2:56 PM

- Create an ALV report by copying from Z_ALV_TEMPLATE_LOCALCLASS

- [MVC](#) design pattern. Search
  "initialize the alv container and pareparing the data " to see how the objects
  are initialized.
  There are two custom abap classes involved, `ZCL_DATA(REF_DATA_OBJECT)` and
  `ZCL_ALV_SERVICE(DIST_ALV_DISPLAY)`. ZCL_DATA is the model and `ZCL_ALV_SERVICE` is
  the controller. The view is `DIST_ALV_DISPLAY`, which imports the custom control we created in
  the Screen 1001.

- Change Report Name.
  ```
  REPORT  ZPP_COGI_ALV
          MESSAGE-ID ZALV
          NO STANDARD PAGE HEADING
          LINE-SIZE 276
          LINE-COUNT 65 .
  ```
  Please change the report name first if you copy it from template.

- Set all the display title below as you expected.

- REPORT name
- Program Attribute

- GUI Title - ALV template title(ALV_TITLE)

- GS TITLE - Report Name Here

```
431     *************** End of ALV Model and Controller classes***************
432
433
434     DATA REF_DATA_OBJECT TYPE REF TO ZCL_DATA.
435     *************** ALV related *****************************
436     ****TODO: Type the report name here.
437     DATA: GS_TITLE(50) TYPE C VALUE 'Report Name Here'.
438     *-- Layout Structure
439     DATA GS_LAYOUT TYPE LVC_S_LAYO.
440     *-- ALV Service CLASS, display the ALV
441     DATA: DIST_ALV_DISPLAY TYPE REF TO ZCL_ALV_SERVICE.
442     *-- Field Category
443     DATA GT_FIELDCAT TYPE LVC_T_FCAT.
444     *-- File Name
445     DATA: FNAME(128) TYPE C.
446
447
448
```

- Search "TODO: Your own global types and data section ". Prepare data type and data definition, especially the internal table to display. Here it is ITAB_GIITEM, which will be used below to hook up with the GUI container
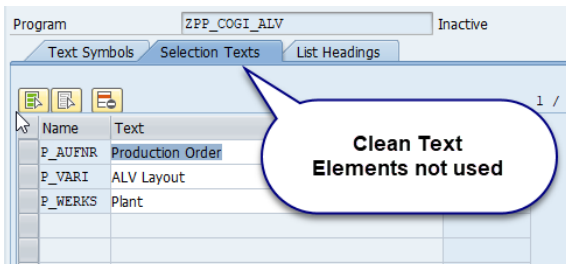
```
    ****TODO: Your own global types and data section
 TYPES: BEGIN OF TY_GIITEM,
           FLAG(1) TYPE C,
           AUFNR    LIKE RESB-AUFNR,  " Production Order
           MATNR    LIKE RESB-MATNR,  " Material
           WERKS    LIKE RESB-WERKS,  " Plant
           LGORT    LIKE RESB-LGORT,  " Storage Location
           RSNUM    LIKE RESB-RSNUM,  " Reservation Quantity
           BDMNG    LIKE RESB-BDMNG,  " Requirement Quantity
           ENMNG    LIKE RESB-ENMNG,  " Withdrawn Quantity
           RNMNG    LIKE RESB-ENMNG.  " Issue Quantity
 TYPES: END OF  TY_GIITEM.


    DATA: ITAB_GIITEM TYPE STANDARD TABLE OF TY_GIITEM,
          WA_GIITEM    TYPE TY_GIITEM.
    CONSTANTS : C_CHECK(1) VALUE 'X'. " value used to set X for a field
```

- Finish the select screen and clean the Text elements

Program	ZPP_COGI_ALV	Inactive

Text Symbols / Selection Texts / List Headings

1 / 3

Name	Text
P_AUFNR	Production Order
P_VARI	ALV Layout
P_WERKS	Plant

Clean Text Elements not used

- Search "TODO: prepare the model data here. " and prepare the data to display.

- Search "TODO: Delete the code below ". This code is only for demo alv template.

- Search "TODO: Prepare ALV Field Category " and set field category for fields to display in ALV grid.

```
319    ****TODO: Prepare ALV Field Category (column name, type etc)
320 □  METHOD GET_FIELDCAT.
321        DATA WA_PT_FCAT TYPE LVC_S_FCAT.
322        CLEAR WA_PT_FCAT.
323        WA_PT_FCAT-FIELDNAME = 'CHECKBOX'.
324        WA_PT_FCAT-CHECKBOX = 'X'.
325        WA_PT_FCAT-EDIT = 'X'.
326        WA_PT_FCAT-COLTEXT = 'Select'.
327        WA_PT_FCAT-SELTEXT = 'Select A Row'.
328        APPEND WA_PT_FCAT TO PT_FIELDCAT.
329
330        CLEAR WA_PT_FCAT.
331        WA_PT_FCAT-FIELDNAME = 'MATNR'.
332        WA_PT_FCAT-REF_TABLE = 'ITAB_GIITEMS'.
333        WA_PT_FCAT-REF_FIELD = 'MATNR'.
334        WA_PT_FCAT-OUTPUTLEN = '18'.
335        WA_PT_FCAT-JUST = 'L'.
336        WA_PT_FCAT-COLTEXT = 'Material'.
337        WA_PT_FCAT-SELTEXT = 'Material'.
338        APPEND WA_PT_FCAT TO PT_FIELDCAT.
```

- ALV output or Download. Search "*ALV output or Background job download.*"
  This template provide code sample for either ALV output or file download. Comment out
  DOWNLOAD_OUTPUT if no needed, both DOWNLOAD_OUTPUT, DOWNLOAD_HEADER and
  DOWNLOAD_DETAIL forms and the form call.

```
□ * ALV output or Background job download.
▶□ *  IF RB_SCRN = 'X'.
        PERFORM ALV_OUTPUT.
▶□ *  ELSE.
▶  | *    PERFORM DOWNLOAD_OUTPUT.
▶  └ *  ENDIF.
```

- Search "TODO: set your table ". Link your table to the view

```
□MODULE STATUS_1001 OUTPUT.
    SET PF-STATUS 'ALV'. "Defined in the GUI Status folder
    SET TITLEBAR 'ALV_TITLE'. "Defined in the GUI Title folder
□**|**TODO: set your table
 * display it
    CALL METHOD DIST_ALV_DISPLAY->DISPLAY_ALV
      CHANGING
        GT_FIELDCAT = GT_FIELDCAT
        GS_LAYOUT    = GS_LAYOUT
        GS_TAB       = ITAB_GIITEM[]
        GS_VARIANT   = P_VARI.
 ENDMODULE.                 " STATUS_1001  OUTPUT
```

> Your internal table variable

- Add Buttons to the Application Tab.
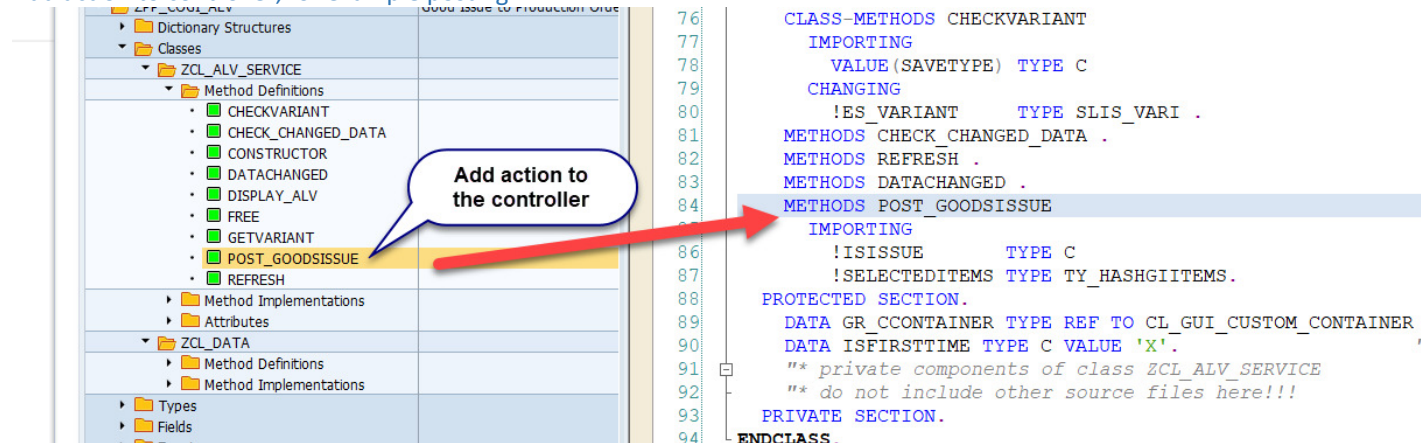 Add Item into GUI Status ALV.

Add code to " `MODULE USER_COMMAND_1001 INPUT.` "

```abap
MODULE USER_COMMAND_1001 INPUT.
  CASE SY-UCOMM.
    WHEN '%ISU'. " Issue Goods
      REFRESH: ITAB_SELECTEDGIITEM.
*** Get input
      CALL METHOD DIST_ALV_DISPLAY->CHECK_CHANGED_DATA.

      CALL METHOD REF_DATA_OBJECT->GETSELECTED
        EXPORTING
          ITEMS         = ITAB_GIITEM
        IMPORTING
          SELECTEDITEMS = ITAB_SELECTEDGIITEM.
      IF LINES( ITAB_SELECTEDGIITEM ) > 0.
        CALL METHOD DIST_ALV_DISPLAY->POST_GOODSISSUE
          EXPORTING
            ISISSUE       = 'X'
            SELECTEDITEMS = ITAB_SELECTEDGIITEM.
        CALL METHOD REF_DATA_OBJECT->GET_DATA.
      ENDIF.
    WHEN '%USU'. " Goods Withdraw
      REFRESH: ITAB_SELECTEDGIITEM.
*** Get Input
      CALL METHOD DIST_ALV_DISPLAY->CHECK_CHANGED_DATA.

      CALL METHOD REF_DATA_OBJECT->GETSELECTED
        EXPORTING
          ITEMS         = ITAB_GIITEM
        IMPORTING
          SELECTEDITEMS = ITAB_SELECTEDGIITEM.
      IF LINES( ITAB_SELECTEDGIITEM ) > 0.
        CALL METHOD DIST_ALV_DISPLAY->POST_GOODSISSUE
          EXPORTING
            ISISSUE       = ''
            SELECTEDITEMS = ITAB_SELECTEDGIITEM.
        CALL METHOD REF_DATA_OBJECT->GET_DATA.
      ENDIF.
    WHEN 'BACK' OR 'EXIT' OR 'RETURN' OR 'CANCEL' OR '&F03' OR '&F12' OR
'&F15'.
      LEAVE TO SCREEN '0'.
      CALL METHOD DIST_ALV_DISPLAY->FREE.
    WHEN 'DISPLAY'.
      LEAVE TO SCREEN 1001.
  ENDCASE.
ENDMODULE.                    " USER_COMMAND_1001  INPUT
```

- Add action to controller, for example posting.

Add action to the controller

Extra Abap Page 5