

Scenario – Purchase order data to be displayed in the ALV tree output.

- 1) Create a program with the following attributes

The screenshot shows the 'ABAP: Program Attributes ZTREE_ALV_3 Change' dialog box. The 'Title' field is 'PO Tree'. The 'Original language' is 'EN' (English). The 'Created' date is '13.03.2011' and the 'Last changed by' is 'INVNAY'. The 'Status' is 'Active'. The 'Attributes' section includes: 'Type' (Executable program), 'Status' (Customer Production Program), 'Application' (empty), 'Authorization Group' (empty), 'Package' (\$TMP - Temporary Objects (never transported!)), 'Logical database' (empty), 'Selection screen' (empty), and checkboxes for 'Editor lock' (unchecked), 'Unicode checks active' (checked), 'Fixed point arithmetic' (checked), and 'Start using variant' (unchecked). The bottom toolbar contains icons for Save, Undo, Redo, Refresh, and Close.

- 2) The code for the program is in this object attachment.



3 alv tree PO.txt

- 3) This following screenshot shows the structure of the program in se80. It has a screen in which the tree is displayed and then it has a GUI status and all other dictionary structures.

ABAP Editor: Change Report ZTREE_ALV_3

The screenshot shows the ABAP Editor interface. On the left is the Repository Browser with the object ZTREE_ALV_3 (PO Tree) selected. On the right is the code editor showing the following code:

```
1  *-----*
2  * Report  ZTREE_ALV_3
3  *-----*
4  *
5  *
6  *-----*
7
8
9  REPORT  ZTREE_ALV_3.
10
11 *Data Declaration
12 *-----*
13 TABLES:  ekko.
14 TYPE-POOLS: slis.
15
16 *-----*
17 * TYPES: BEGIN OF t_ekko,
18 *         ebelp TYPE ekpo-ebelp,
19 *         statu TYPE ekpo-statu,
20 *         aedat TYPE ekpo-aedat,
21 *         matnr TYPE ekpo-matnr,
22 *         menge TYPE ekpo-menge,
23 *         meins TYPE ekpo-meins,
24 *         netpr TYPE ekpo-netpr,
25 *         peinh TYPE ekpo-peinh,
26 *         END OF t_ekko.
27
28 *-----*
29 * DATA: it_ekko TYPE STANDARD TABLE OF t_ekko INITIAL SIZE 0,
30 *         it_ekpo TYPE STANDARD TABLE OF t_ekko INITIAL SIZE 0,
31 *         it_emptytab TYPE STANDARD TABLE OF t_ekko INITIAL SIZE 0,
32 *         wa_ekko TYPE t_ekko,
33 *         wa_ekpo TYPE t_ekko.
34
35 *-----*
36 * DATA: ok_code like sy-ucomm,
37 *         name_ab like sy-ucomm
```

- 4) The start of selection event contains the data retrieval process. After the data selection process is done, the screen 100 is called.

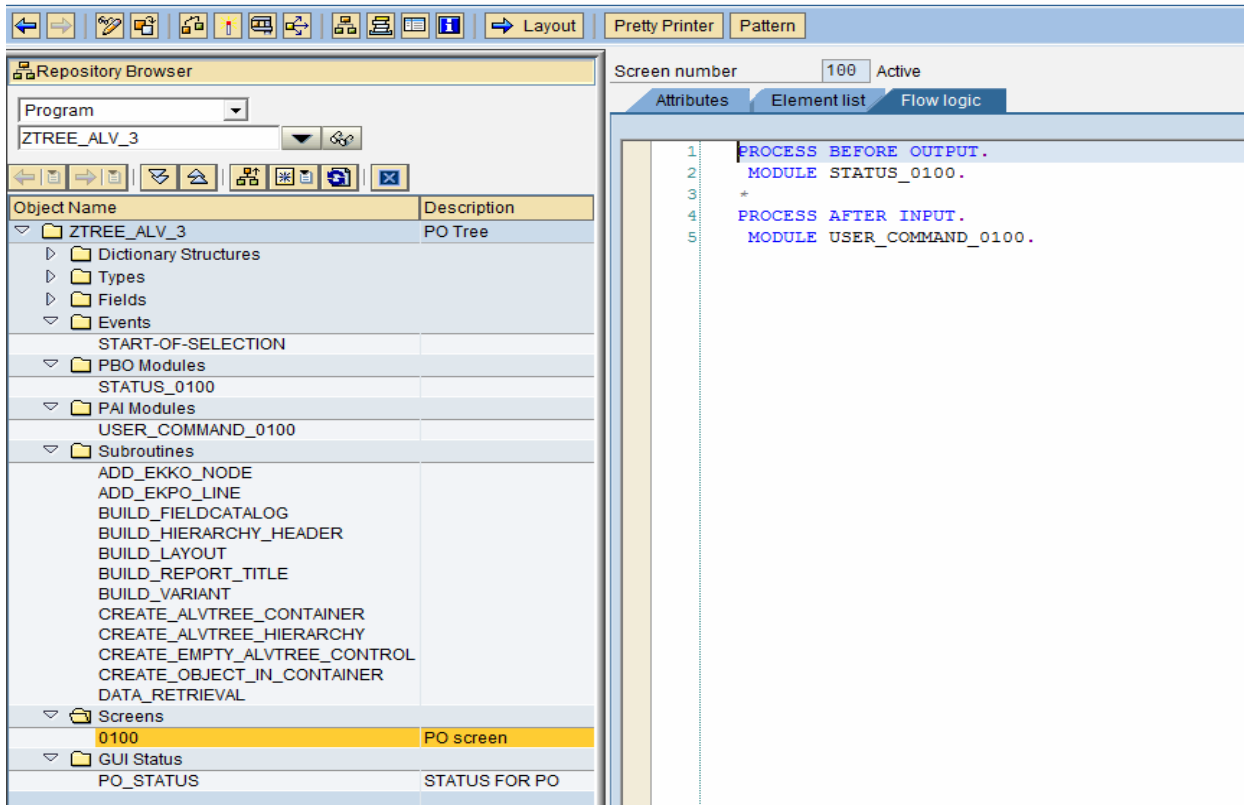
```
*-----*
*Start-of-selection.
start-of-selection.

* ALVtree setup data
perform data_retrieval.
perform build_fieldcatalog.
perform build_layout.
perform build_hierarchy_header changing gd_hierarchy_header.
perform build_report_title using gd_report_title gd_logo.
perform build_variant.

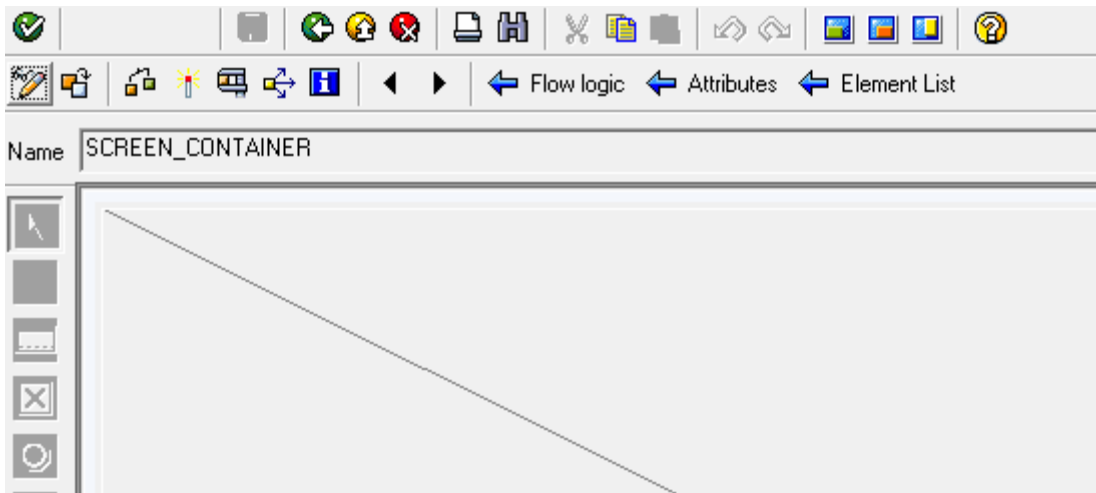
* Display ALVtree report
call screen 100.
```

- 5) The screen has the following code in it.

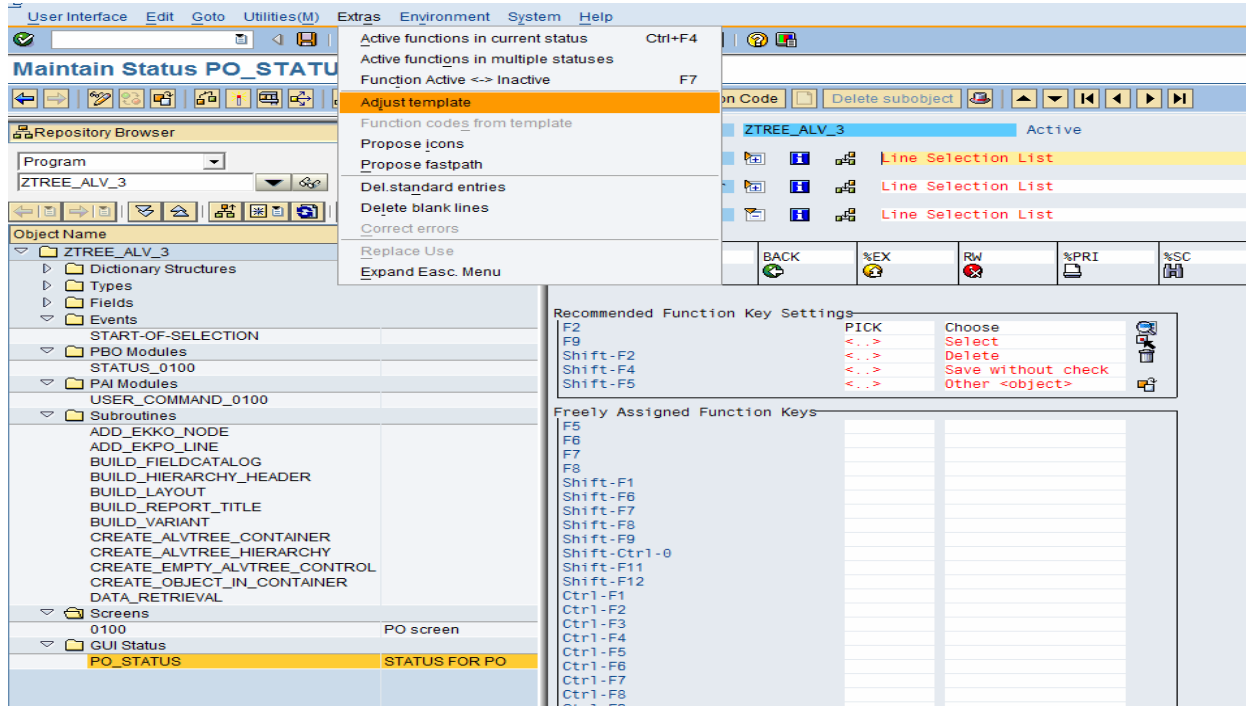
Screen Painter: Change Screen for ZTREE_ALV_3



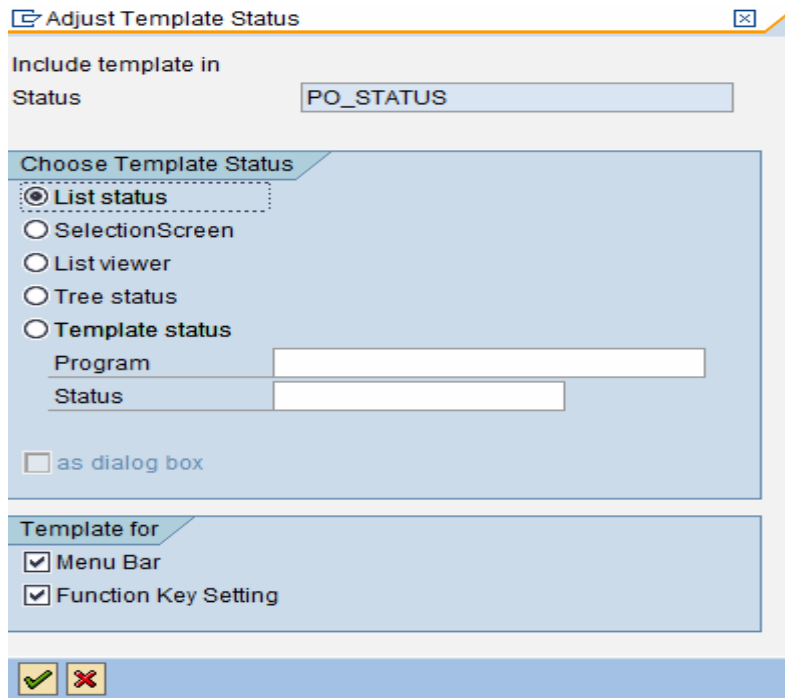
6) In the layout of the screen, build a container element with the name 'SCREEN_CONTAINER'.



7) After that make sure the screen has the status bar and in the status bar, go to extras-Adjust template as shown in the below screen.



8) We get the following pop up and we get the standard Fcodes for the important functionalities and by default.



9) The screen has the following code –

```
Screen number 100 Active
Attributes Element list Flow logic
1 process before output.
2 module status_0100.
3 *
4 process after input.
5 module user_command_0100.
```

10) In the pbo, the following code is executed.

```
Report ZTREE_ALV_3 Active
274 module status_0100 output.
275   set pf-status 'PO_STATUS'.
276   * SET TITLEBAR 'xxx'.
277
278   * If ALVtree already exists then it must not be re
279   * will cause a runtime error.
280   if gd_tree is initial.
281
282   * Create ALVtree (must be performed within screen
283     perform create_alvtree_container.
284     perform create_object_in_container.
285     perform create_empty_alvtree_control.
286     perform create_alvtree_hierarchy.
287   endif.
288   call method cl_gui_cfw=>flush.
289
290
291 endmodule. " STATUS_0100 OUTPUT
292 *-----
```

- 11) The status when double clicked, goes in the screen showed in step 7 and the PF status is developed in steps 7 and 8
- 12) Now the object gd_custom_container is created as an instance of the class cl_gui_custom_container and the container SCREEN_CONTAINER (created in the screen 1000) is passed in to it.
- 13) Similarly and object of the instance cl_gui_alv_tree is created and is passed with the parameter gd_custom_container that is created in the step 12. Also another parameter node has to be passed. See the following code

```
create object gd_tree
  exporting
    parent          = gd_custom_container
    node_selection_mode = cl_gui_column_tree=>node_sel_mode_single
```

14) The method of the object is called now .

```
call method gd_tree->set_table_for_first_display
```

15) Now the final methods are to be called to display the report in the tree output, after writing the code for the nodes.

```
* calculate totals
call method gd_tree->update_calculations.

* this method must be called to send the data to the frontend
call method gd_tree->frontend_update.
```

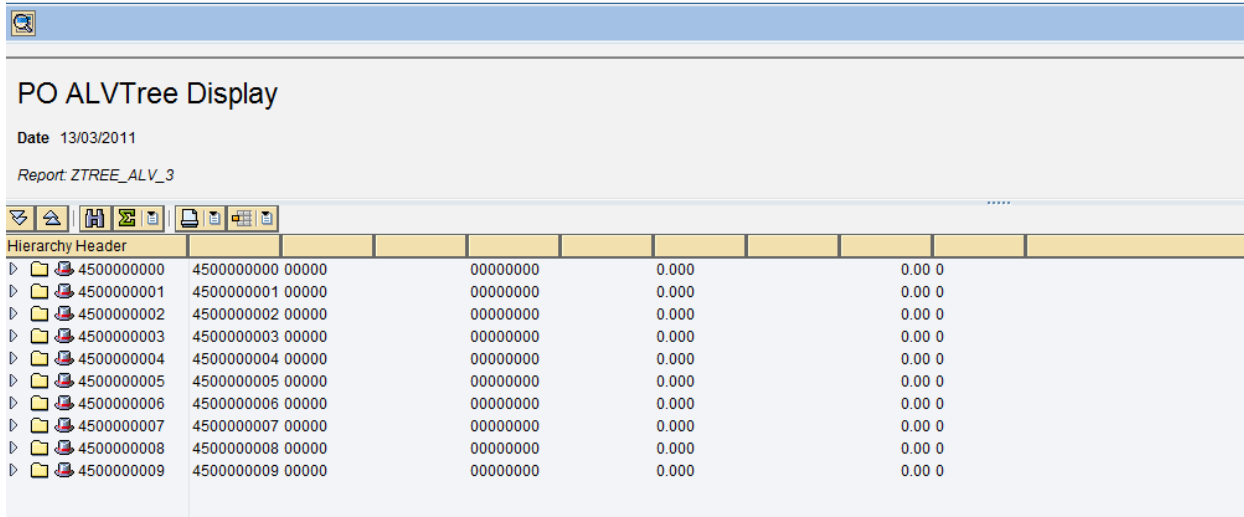
16) Execute the report. WE get the following output.

SAP

PO ALVTree Display

Date 13/03/2011

Report ZTREE_ALV_3



| Hierarchy Header | | | | | | |
|------------------|------------------|--|----------|-------|--|--------|
| 4500000000 | 4500000000 00000 | | 00000000 | 0.000 | | 0.00 0 |
| 4500000001 | 4500000001 00000 | | 00000000 | 0.000 | | 0.00 0 |
| 4500000002 | 4500000002 00000 | | 00000000 | 0.000 | | 0.00 0 |
| 4500000003 | 4500000003 00000 | | 00000000 | 0.000 | | 0.00 0 |
| 4500000004 | 4500000004 00000 | | 00000000 | 0.000 | | 0.00 0 |
| 4500000005 | 4500000005 00000 | | 00000000 | 0.000 | | 0.00 0 |
| 4500000006 | 4500000006 00000 | | 00000000 | 0.000 | | 0.00 0 |
| 4500000007 | 4500000007 00000 | | 00000000 | 0.000 | | 0.00 0 |
| 4500000008 | 4500000008 00000 | | 00000000 | 0.000 | | 0.00 0 |
| 4500000009 | 4500000009 00000 | | 00000000 | 0.000 | | 0.00 0 |

17) When expanded we can see the linei item details of the PO

PO ALVTree Display

Date 13/03/2011

Report: ZTREE_ALV_3



| Hierarchy Header | | | | | | | |
|------------------|------------|-------|---|----------|-------------|------------|-------------|
| 4500000000 | 4500000000 | 00000 | | 00000000 | | 0.000 | 0.00 0 |
| 00001 | 4500000000 | 00001 | | 20090703 | | 1.000 | 25,000.00 1 |
| 4500000001 | 4500000001 | 00000 | | 00000000 | | 0.000 | 0.00 0 |
| 00010 | 4500000001 | 00010 | | 20090706 | 00000000... | 4.000 | 7,074.00 1 |
| 4500000002 | 4500000002 | 00000 | | 00000000 | | 0.000 | 0.00 0 |
| 00010 | 4500000002 | 00010 | | 20090723 | 00000000... | 613000.000 | 3.00 1 |
| 4500000003 | 4500000003 | 00000 | | 00000000 | | 0.000 | 0.00 0 |
| 00010 | 4500000003 | 00010 | F | 20091214 | 00000000... | 168790.000 | 17.16 10 |
| 00020 | 4500000003 | 00020 | F | 20090804 | 00000000... | 45350.000 | 0.95 1 |
| 4500000004 | 4500000004 | 00000 | | 00000000 | | 0.000 | 0.00 0 |
| 00010 | 4500000004 | 00010 | | 20090723 | 00000000... | 3.000 | 477.00 1 |
| 4500000005 | 4500000005 | 00000 | | 00000000 | | 0.000 | 0.00 0 |
| 00010 | 4500000005 | 00010 | | 20090723 | 00000000... | 4.000 | 19,166.61 1 |
| 00020 | 4500000005 | 00020 | | 20090723 | 00000000... | 2.000 | 34,938.70 1 |
| 00030 | 4500000005 | 00030 | | 20090723 | 00000000... | 2.000 | 23,007.29 1 |
| 00040 | 4500000005 | 00040 | | 20090723 | 00000000... | 2.000 | 19,502.38 1 |
| 4500000006 | 4500000006 | 00000 | | 00000000 | | 0.000 | 0.00 0 |
| 4500000007 | 4500000007 | 00000 | | 00000000 | | 0.000 | 0.00 0 |
| 4500000008 | 4500000008 | 00000 | | 00000000 | | 0.000 | 0.00 0 |
| 4500000009 | 4500000009 | 00000 | | 00000000 | | 0.000 | 0.00 0 |