

Splitting the Custom Container & Display more than one ALV



Applies to:

This document applies to SAP ECC 6.0, SAP Netweaver 2004s. For more information, visit the [ABAP homepage](#).

Summary

This article contains the step by step procedure to guide you to display more than one Internal Table in a Custom Container by splitting it.

The requirement is to display 3 different reports in single container.

Author: Chinnaiya Pandian

Company: HCL AXON

Created on: 09 February 2011

Author Bio



Chinnaiya Pandian is working for HCL AXON. He is Computer Science Graduate and working on Technologies like ABAP, Web Dynpro ABAP, Floor Plan Manager & SAP Interactive forms

Table of Contents

Scenario.....	3
Step by step Solution.....	3
Step 1: Creating Screen.....	3
Step 2: Flow Logic.....	4
Step 3: Programming Report.....	4
Output.....	7
Related Content.....	8
Disclaimer and Liability Notice.....	9

Scenario

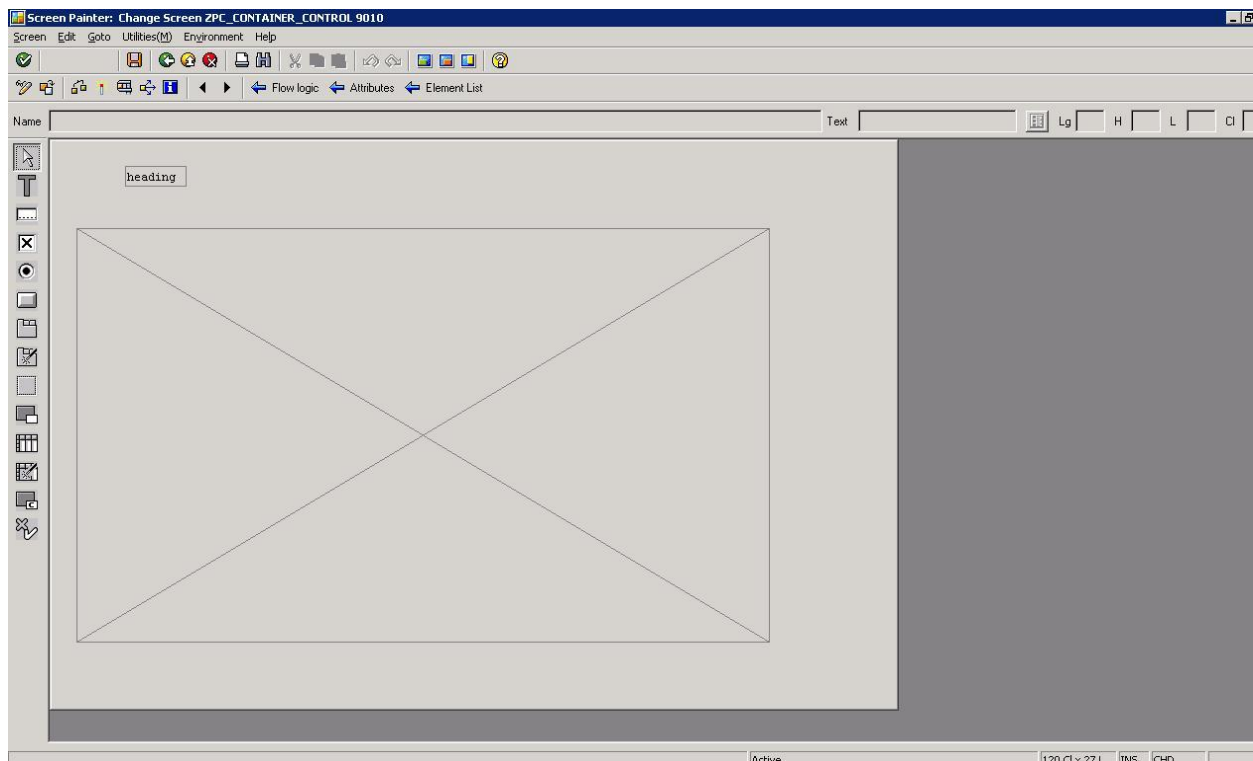
Displaying more than one table in a report by splitting the custom container

TABLE 1	TABLE 2
	TABLE 3

Step by step Solution

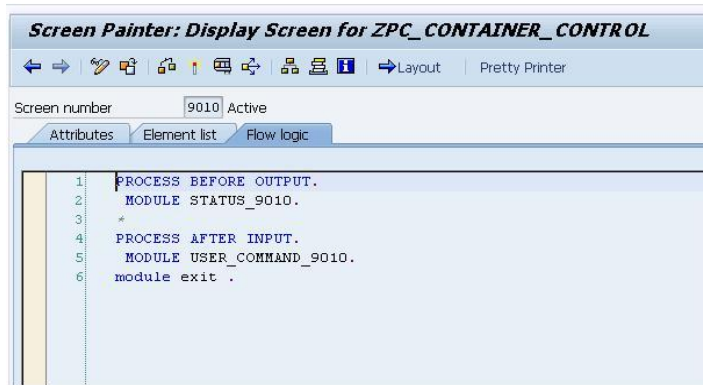
Step 1: Creating Screen

Go to Screen painter Transaction Code SE51 ,Create a screen with no 9010,provide description for the screen and click on the Layout Button , Place a Custom container UI element and give name as 'CCONTAINER' now you will have a screen with custom container as below screen shot . Activate the Object.



Step 2: Flow Logic

Go to the Flow Logic Tab to write coding for PBO & PAI.



Step 3: Programming Report

Go to SE38 and create a program in Z name space and paste the below code snippet.

```

*&-----*
*& Report  ZPC_CONTAINER_CONTROL
*&
*&-----*
*&
*&
*&-----*

REPORT zpc_container_control.
*~~~ Declaration
DATA splitter_1 TYPE REF TO cl_gui_splitter_container.
DATA splitter_2 TYPE REF TO cl_gui_splitter_container.
DATA container TYPE REF TO cl_gui_custom_container.
DATA container_1 TYPE REF TO cl_gui_container.
DATA container_2 TYPE REF TO cl_gui_container.
DATA container_3 TYPE REF TO cl_gui_container.
DATA : gt_sflight_1 TYPE TABLE OF sflight,
      gt_sflight_2 TYPE TABLE OF sflight,
      gt_sflight_3 TYPE TABLE OF sflight,
      g_container TYPE scrfname VALUE 'CCONTAINER',
      grid1 TYPE REF TO cl_gui_alv_grid,
      grid2 TYPE REF TO cl_gui_alv_grid,
      grid3 TYPE REF TO cl_gui_alv_grid.

DATA : ok_code TYPE sy-ucomm.

* fetching data from table for different internal tables
SELECT * FROM sflight INTO TABLE gt_sflight_1. " itab 1
SELECT * FROM sflight INTO TABLE gt_sflight_2 UP TO 3 ROWS. "itab 2
SELECT * FROM sflight INTO TABLE gt_sflight_3 UP TO 2 ROWS ." itab 3

CALL SCREEN 9010.

*&-----*
*&      Module STATUS_9010  OUTPUT
*&-----*

```

```

*          text
*-----*
MODULE status_9010 OUTPUT.
  SET PF-STATUS 'TEST'.
*   SET TITLEBAR 'TESTBAR'.

* creating object reference for container
  CREATE OBJECT container
  EXPORTING
    container_name = 'CONTAINER'. " pass name of the container created in
Screen no 9010

*Splitting the main container in to 1 row & 2 coloum
  CREATE OBJECT splitter_1
  EXPORTING
    parent = container
    rows   = 1
    columns = 2.

*getting the reference for the splited container( row 1 & col 1 container )
  CALL METHOD splitter_1->get_container
  EXPORTING
    row       = 1
    column    = 1
  RECEIVING
    container = container_1.

*getting the reference for the splited container (row 1 & col 2 container)
  CALL METHOD splitter_1->get_container
  EXPORTING
    row       = 1
    column    = 2
  RECEIVING
    container = container_2.

* now splitting the 2nd coloum container in to 2 rows & 1 coloum
  CREATE OBJECT splitter_2
  EXPORTING
    parent = container_2
    rows   = 2
    columns = 1.

*getting the reference for the splited container2 ( row 1 & col 2 container )
  CALL METHOD splitter_2->get_container
  EXPORTING
    row       = 1
    column    = 1
  RECEIVING
    container = container_2.

*getting the reference for the splited container2 ( row 2 & col 1 container )
  CALL METHOD splitter_2->get_container
  EXPORTING
    row       = 2
    column    = 1
  RECEIVING
    container = container_3.

*   Populating first internal table to the container
  CREATE OBJECT container

```

```

EXPORTING
  container_name = g_container.
CREATE OBJECT grid1
EXPORTING
  i_parent = container_1.
CALL METHOD grid1->set_table_for_first_display
EXPORTING
  i_structure_name = 'SFLIGHT'
CHANGING
  it_outtab      = gt_sflight_1.

*   Populating second internal table

CREATE OBJECT container
EXPORTING
  container_name = g_container.
CREATE OBJECT grid2
EXPORTING
  i_parent = container_2.
CALL METHOD grid2->set_table_for_first_display
EXPORTING
  i_structure_name = 'SFLIGHT'
CHANGING
  it_outtab      = gt_sflight_2.

**  Populating third internal table
CREATE OBJECT container
EXPORTING
  container_name = g_container.
CREATE OBJECT grid3
EXPORTING
  i_parent = container_3.
CALL METHOD grid3->set_table_for_first_display
EXPORTING
  i_structure_name = 'SFLIGHT'
CHANGING
  it_outtab      = gt_sflight_3.

*
*   ENDIF.
ENDMODULE.                " STATUS_9010  OUTPUT
*&-----*
*&      Module  EXIT  INPUT
*&-----*
*      text
*-----*
MODULE exit INPUT.
* free the container memory when exit
CALL METHOD container->free.
LEAVE PROGRAM.

ENDMODULE.                " EXIT  INPUT
*&-----*
*&      Module  USER_COMMAND_9010  INPUT
*&-----*
*      text
*-----*

```

```

MODULE user_command_9010 INPUT.
  CALL METHOD cl_gui_cfw=>dispatch.
  CASE ok_code.
    WHEN 'BACK'.
      LEAVE SCREEN.
    WHEN 'CANCEL'.
      LEAVE PROGRAM.
    WHEN 'EXIT'.
      LEAVE SCREEN .
  ENDCASE .

ENDMODULE.                " USER_COMMAND_9010  INPUT

```

Output

The report will display as below screen shot.

The screenshot shows an SAP ALV report window with a table of flight data. The table is split into two panes. The left pane shows a full list of flights, and the right pane shows a filtered view of the first two rows.

ID	No.	Flight Date	Airfare	Curr.	Pk
A	820	20.12.2002	1,222.00	CAD	A3
AF	820	23.12.2002	2,222.00	EUR	A3
LH	400	28.02.1995	899.00	DEM	A3
LH	454	17.11.1995	1,499.00	DEM	A3
LH	455	06.06.1995	1,090.00	USD	A3
LH	3577	28.04.1995	6,000.00	LIT	A3
LH	9981	21.12.2002	222.00	EUR	A3
SQ	26	28.02.1995	849.00	DEM	DC

Related Content

[SAP HELP](#)

[ABAP Wiki](#)

[WDAbap – OVS search Help In Select Option](#)

For more information, visit the [ABAP homepage](#).

Disclaimer and Liability Notice

This document may discuss sample coding or other information that does not include SAP official interfaces and therefore is not supported by SAP. Changes made based on this information are not supported and can be overwritten during an upgrade.

SAP will not be held liable for any damages caused by using or misusing the information, code or methods suggested in this document, and anyone using these methods does so at his/her own risk.

SAP offers no guarantees and assumes no responsibility or liability of any type with respect to the content of this technical article or code sample, including any liability resulting from incompatibility between the content within this document and the materials and services offered by SAP. You agree that you will not hold, or seek to hold, SAP responsible or liable with respect to the content of this document.