How to Create Top of List and End of List of the ALV Output in Web Dynpro for ABAP

Applies to:
SAP Netweaver 2004S: Web Dynpro for ABAP. For more information, visit the User Interface Technology homepage.

Summary
This document will explains about how to create a Top of list nothing but a Header area and End of list nothing but a Footer area of ALV Table.

Author: Suman Kumar Chinnam
Company: YASH Technologies
Created on: 24 September 2008

Author Bio
Suman Kumar is a Consultant for YASH Technologies, Pvt., and Ltd. Based in India. He has 2 years of IT experience. He has good knowledge of ABAP, BSP and Web Dynpro for ABAP.
How to Create Top of List and End of List of the ALV Output in Web Dynpro for ABAP

I am assuming that users have already finished basic exercises on ALV. So I am straight away explaining how to create Header and Footer Areas for ALV Output. To do this, we have to use design objects.

We can use as many elements of different element types as you want in order to form a design object. Then we display it at desired position.

**Element Types and Layout Forms**

You can use elements of the following element types for your design object:

1. Header element (header info).
2. Text element with or without label (text label).
3. Action information.

These elements can be arranged within your design object. To do this we can choose the two Forms of layout:

1. Single element
   Generate an element and display it in the required position.
2. Row-type layout
   Arrange as many elements as you want in a row, one after the other.
3. Table-type layout
   Arrange as many elements as you want in rows and columns.

**Context Nodes TOP_OF_LIST and END_OF_LIST**

The ALV component provides the two context nodes TOP_OF_LIST and END_OF_LIST. Each contains an attribute CONTENT. These context nodes hold the data of your design objects for the header and footer areas of the ALV output.

Define context mapping to a node with the same name in the context of your application and set the attribute CONTEXT of your context node to your design object.
Note: Sometimes you may not be able to directly map the contexts; then right-click on the context node of your view and select define mapping from list, then click OK. A pop-up appears with Component and Interface Controllers, select interface controller (ALV) and then select TOP_OF_LIST.

### Setting up a Design Object

Proceed as follows to set up a design object for the header and footer areas:
- Map the context nodes TOP_OF_LIST and END_OF_LIST of the ALV component to the context of your application.
- Generate a design object. To do this, use either the row-type or the table-type layout.
- You insert the required elements into this layout.
- Set the design object as the value of the attribute CONTENT in the context node.

### Creating a Design Object

First define the basic layout for your design object. Decide whether to display a single element, a sequence of elements in a row, or elements arranged in multiple rows and columns. You use one of the following classes for this purpose.

#### Classes for the Layout of Design Objects

<table>
<thead>
<tr>
<th>Layout</th>
<th>Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single element</td>
<td>See <em>Elements Types and Their Classes</em> (except for the element type <em>Label</em>)</td>
</tr>
<tr>
<td>Row-type layout</td>
<td>CL_SALV_FORM_LAYOUT_FLOW</td>
</tr>
<tr>
<td>Table-type layout</td>
<td>CL_SALV_FORM_LAYOUT_GRID</td>
</tr>
</tbody>
</table>
Creating an Element
After creation of layout for a design object (see above), you can use methods to Create elements with the various element types: These methods can be found in the classes CL_SALV_FORM_LAYOUT_FLOW and CL_SALV_FORM_LAYOUT_GRID.

Methods for Creating Elements for a Design Object

<table>
<thead>
<tr>
<th>Function</th>
<th>Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create row-type layout (for nested layouts)</td>
<td>CREATE_FLOW</td>
</tr>
<tr>
<td>Create table-type layout (for nested layouts)</td>
<td>CREATE_GRID</td>
</tr>
<tr>
<td>Create text element</td>
<td>CREATE_TEXT</td>
</tr>
<tr>
<td>Create a label for a specific text element</td>
<td>CREATE_LABEL</td>
</tr>
<tr>
<td>Create a header element</td>
<td>CREATE_HEADER_INFORMATION</td>
</tr>
<tr>
<td>Create action information</td>
<td>CREATE_ACTION_INFORMATION</td>
</tr>
</tbody>
</table>

Write the code to get TOP_OF_LIST and END_OF_LIST of ALV Output

Write the below code in WDDOINIT method.

```
method wddoinit.
  *
  *ALV Component usage
  *data: l_ref_cmp_usage type ref to if_wd_component_usage.
  l_ref_cmp_usage = wd_this->wd_cpuse_alv().
  if l_ref_cmp_usage->has_active_component() is initial.
    l_ref_cmp_usage->create_component().
  endif.
  *
  *Call Get_model() of Interface Controller to set changes to the output
  *table
  *
  *data: l_ref_interfacecontroller type ref to iwci_salv_wd_table.
  l_ref_interfacecontroller = wd_this->wd_cpiifc_alv().
  *
  *data:
    l_value type ref to cl_salv_wd_config_table.
    l_value = l_ref_interfacecontroller->get_model().
```

How to Create Top of List and End of List of the ALV Output in Web Dynpro for ABAP

DATA:
  node_top_of_list TYPE REF TO if wd_context_node,
  node_end_of_list TYPE REF TO if wd_context_node,
  elem_top_of_list TYPE REF TO if wd_context_element,
  stru_top_of_list TYPE if_search=>element_top_of_list.

* Declare variables for Layout and Elements
data: lr_grid type ref to cl_salv_form_layout_grid,
  lr_value type ref to cl_salv_form_label,
  lr_value1 type ref to cl_salv_form_text.

*Create Layout
CREATE OBJECT lr_grid.

*Create Elements
lr_value1 = lr_grid->create_text(
  row = 1
  column = 2
  * ROWSPAN = ROWSPAN
  * COLSPAN = COLSPAN
  TEXT = 'TEXT1'
  * TOOLTIP = TOOLTIP
).

lr_value = lr_grid->create_label(
  row = 1
  column = 1
  * ROWSPAN = ROWSPAN
  * COLSPAN = COLSPAN
  TEXT = 'Label1'
  * TOOLTIP = TOOLTIP
  _LABEL_FOR = lr_value1
).

lr_value1 = lr_grid->create_text(
  row = 2
  column = 2
  * ROWSPAN = ROWSPAN
  * COLSPAN = COLSPAN
  TEXT = 'TEXT2'
  * TOOLTIP = TOOLTIP
).

lr_value = lr_grid->create_label(
    row = 2
    column = 1
* ROWSPAN = ROWSPAN
* COLSPAN = COLSPAN
* TEXT = 'Label2'
* TOOLTIP = TOOLTIP
    R_LABEL_FOR = lr_value!
).

* navigate from <CONTEXT> to <TOP_OF_LIST> via lead selection
  node_top_of_list = wd_context->get_child_node( 'TOP_OF_LIST' ).
* navigate from <CONTEXT> to <END_OF_LIST> via lead selection
  node_end_of_list = wd_context->get_child_node( 'END_OF_LIST' ).

*Set Attribute CONTENT of node Top of List
node_top_of_list->set_attribute(  
    VALUE = lr_grid
    name = 'CONTENT'
).

*Set Attribute CONTENT of node End of List
node_end_of_list->set_attribute(  
    VALUE = lr_grid
    name = 'CONTENT'
).

*Set Top of list
l_value->if_salv_wd_table_settings-set_top_of_list_visible(  
    value = abap_true
)

*Set End of list
l_value->if_salv_wd_table_settings-set_end_of_list_visible(  
    value = abap_true
).

Endmethod.
Result

Header Area

Footer Area
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.