Web Dynpro: Multiple ALV Grids and Layouts in ALV

Applies to:
SAP ECC 6.0. For more information, visit the Web Dynpro ABAP homepage.

Summary
The article is designed in such a way that person with ABAP knowledge can work in Webdynpro ABAP after reading the article. This article provides the knowledge for understanding ALV in Web dynpro and how to display multiple alv grids in the same page and how to use create layouts in ALV.

Author: J.Jayanthi

Company: Siemens IT Solutions and Services Pvt. Ltd.

Created on: 05 October 2010

Author Bio
J.Jayanthi is a Certified ABAP consultant with HR ABAP knowledge.
# Table of Contents

ALV in Web Dynpro ABAP .................................................................................................................. 3
Prerequisites ........................................................................................................................................ 3
  Creating Web Dynpro ......................................................................................................................... 3
  Component Controller ........................................................................................................................ 4
Designing View ................................................................................................................................... 7
Embedding View ................................................................................................................................. 13
Component Usages ............................................................................................................................. 14
Creating Web Dynpro Application .................................................................................................... 15
Code ..................................................................................................................................................... 16
Output ................................................................................................................................................. 19
Related Content ................................................................................................................................. 22
Disclaimer and Liability Notice ......................................................................................................... 23
ALV in Web Dynpro ABAP

ABAP Consultants are generally familiar in SAP List Viewer (ALV) and aware of the ways to play with ALV by coloring a row, coloring a column, F4 help for a cell, displaying cell as button, coloring a cell, Multiple grids, Tabstrip, etc.

Being an ABAP consultant, we are going to see how to approach the Multiple ALV Grids in Web Dynpro in same page. We are going to display 4 grids (one for Planned working time, one grid for address, one for education and one grid for experience) in the same page. In the output section, how to create layout is explained.

Prerequisites

Component
The component is the central, reusable unit of the application project. You can create any number of views in a component and arrange them in any number of windows.

Component Usages
Web Dynpro components can be nested. This means that you can integrate any number of other, already existing components into a component.

View
The view is the smallest unit of a Web Dynpro application visible for the user. The layout elements and dialog elements - for example, tables, text fields, or buttons - required for the application are arranged in a view. The view contains a controller and a controller context in which the application data to be processed is stored in a hierarchical structure. This allows the linking of the graphical elements with the application data.

Window
A window is used to group multiple views and to specify the navigation between the views. A view can only displayed by the browser if the view is embedded in a window.

Creating Web Dynpro

Go to SE80 and select Web Dynpro Comp./Intf. and provide the name(say ZZZ_JAYTEST5) to create. Then enter the description and choose the type as Web Dynpro Component.
Mention the Component Use as ALV1, ALV2, ALV3 and ALV4 and Component as SALV_WD_TABLE in the Used Components tab in Web Dynpro (ZZZ_JAYTEST5).

This will create a Component Usages by name ALV1, ALV2, ALV3 and ALV4.

**Component Controller**

Go to Component Controller and Right click the context. Then select Create Node.

Then select the fields required from the structure using Add Attribute from Structure button.
Then Double click the node and then remove the dictionary structure, change the cardinality to 0:n. Uncheck the Initialization Lead selection.

Similarly create another node Address using PA0006 table and add attributes from structure and then delete the dictionary structure after that.
Create two other nodes Education and Experience using tables PA0022 and Pa0023 table and then add attributes and then set the properties like below.
In Properties tab, create like this using create icon.

Designing View
We are going to embed each Alv container in a group.
Go to the properties tab of view and then create as below.
Then in layout, set the properties for ROOTUIELEMENTCONTAINER as follows.

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
<th>Bind</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>ROOTUIELEMENTCONTAINER</td>
<td></td>
</tr>
<tr>
<td>Layout</td>
<td>MatrixLayout</td>
<td></td>
</tr>
<tr>
<td>accessibilityDesciption</td>
<td></td>
<td></td>
</tr>
<tr>
<td>defaultButtonType</td>
<td></td>
<td></td>
</tr>
<tr>
<td>enabled</td>
<td></td>
<td></td>
</tr>
<tr>
<td>height</td>
<td></td>
<td></td>
</tr>
<tr>
<td>isLayoutContainer</td>
<td></td>
<td></td>
</tr>
<tr>
<td>scrollingMode</td>
<td>none</td>
<td></td>
</tr>
<tr>
<td>tooltip</td>
<td></td>
<td></td>
</tr>
<tr>
<td>visible</td>
<td>Visible</td>
<td></td>
</tr>
<tr>
<td>width</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Go to the layout in view and right click the ROOTUIELEMENTCONTAINER and then choose Insert element. Fill the ID as G1 and typ as Group. Change the caption as required say Planned working time.

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
<th>Bind</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>G1</td>
<td></td>
</tr>
<tr>
<td>Typ</td>
<td>Group</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
<th>Bind</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td></td>
<td></td>
</tr>
<tr>
<td>enabled</td>
<td></td>
<td></td>
</tr>
<tr>
<td>imageFirst</td>
<td></td>
<td></td>
</tr>
<tr>
<td>imageSource</td>
<td></td>
<td></td>
</tr>
<tr>
<td>text</td>
<td>Planned Working Time</td>
<td></td>
</tr>
<tr>
<td>textDirection</td>
<td>inherit</td>
<td></td>
</tr>
<tr>
<td>tooltip</td>
<td></td>
<td></td>
</tr>
<tr>
<td>visible</td>
<td>Visible</td>
<td></td>
</tr>
</tbody>
</table>
Right click the group G1 and create ViewContainerUIElement as below.
Similarly create Group G2 and create a ViewContainerUIElement for Address.

Now create G3 for Education and G4 for Experience and then design viewcontainers also as explained.
In context tab, drag and drop the context which appears under component controller to view (Main is the view name).
Web Dynpro: Multiple ALV Grids and Layouts in ALV

Select the method WDODINIT in methods tab.

Use Web Dynpro code wizard to generate code automatically.

Step a:
Choose the radio button Read context and then press F4 to select the context Plannedworkingtime.

Keep the below generated code and delete the rest which is not required.

```plaintext
DATA lo_no_plannedworkingtime TYPE REF TO if_wd_context_node.
DATA lo_el_plannedworkingtime TYPE REF TO if_wd_context_element.
DATA lo_is_plannedworkingtime TYPE wd_this->element_plannedworkingtime.

* navigate from <UNIQ> to <PLANNEDWORKINGTIME> via lead selection.
lo_no_plannedworkingtime = wd_context->get_child_node( name = wd_this->wdctr_plannedworkingtime )

* get element via lead selection
lo_el_plannedworkingtime = lo_no_plannedworkingtime->get_element.
```

Step b:
Again use Code Wizard as below.

This will generate the below code. Since we are going to use 4 ALVs, lo_cmp_usage is changed to lo_cmp_usage1.

```plaintext
DATA lo_cmp_usage1 TYPE REF TO if_wd_component_usage.
lo_cmp_usage1 = wd_this->wd_cpuse_alv1().
IF lo_cmp_usage1->has_active_component() IS INITIAL.
lo_cmp_usage1->create_component().
ENDIF
```

Then use Method call in Used controller as below in Code Wizard. Change the generated code accordingly with prefix 1 so that we can use 4 alv coding with proper prefix.

Step c:

```plaintext
DATA lo_interfacecontroller1 TYPE REF TO iwdi_salv_wd_table.
lo_interfacecontroller1 = wd_this->wd_cpifc_alv1().

DATA lo_value1 TYPE REF TO cl_salv_wd_config_table.
lo_value1 = lo_interfacecontroller1->get_model().
```
Step d:
Then select the data by normal ABAP statement (declare t_0006 as required) and bind the table.

```abap
select pernr subty name2 stras ort01 pstlz land1
from pa0006 into table t_0006.
```

`lo_nd_plannedworkingtime->bind_table( t_0007 ).`

Repeat the steps from a to d for Address (use Address instead of Planned working time and ALV2 instead of ALV1), Education (use Education instead of Planned working time and ALV3 instead of ALV1) and Experience (use Experience instead of Planned working time and ALV4 instead of ALV1).

**Embedding View**

Embed the table by right clicking each view in the window.

After embedding, it will look like below.
Component Usages

Properties of Component usage should be as below.

Right click the component Usage (here with name ALV1) and Create controller Usage.

Drag and drop the node (Plannedworkingtime in right side) from Component Controller context to Data (in left side) in Controller Usage Context.
Similarly do for other component usages ALV2, ALV3 and ALV4.

**Creating Web Dynpro Application**

Create Web Dynpro Application by right clicking the Webdynpro (ZZZ_JAYTEST5).

Right click the Web Dynpro component and activate.
Code

method WDDOINIT.
  * For address
  TYPES : BEGIN OF ty_0006,
    pernr TYPE pa0006-pernr,
    subty TYPE pa0006-subty,
    name2 TYPE pa0006-name2,
    stras TYPE pa0006-stras,
    ort01 TYPE pa0006-ort01,
    pstl2 TYPE pa0006-pstl2,
    land1 TYPE pa0006-land1,
  END OF ty_0006,
  * For Planned Working time
  BEGIN OF ty_0007,
    pernr TYPE pa0007-pernr,
    subty TYPE pa0007-subty,
    begda TYPE pa0007-begda,
    endda TYPE pa0007-endda,
    schkz TYPE pa0007-schkz,
    mostd TYPE pa0007-mostd,
    wostd TYPE pa0007-wostd,
  END OF ty_0007,
  * For Education
  BEGIN OF ty_0022,
    pernr TYPE pa0022-pernr,
    subty TYPE pa0022-subty,
    slart TYPE pa0022-slart,
    insti TYPE pa0022-insti,
    sland TYPE pa0022-sland,
    slabs TYPE pa0022-slabs,
    anzk1 TYPE pa0022-anzk1,
    emark TYPE pa0022-emark,
  END OF ty_0022,
  * For Experience
  BEGIN OF ty_0023,
    pernr TYPE pa0023-pernr,
    begda TYPE pa0023-begda,
    endda TYPE pa0023-endda,
    arrgb TYPE pa0023-arrgb,
    ort01 TYPE pa0023-ort01,
    land1 TYPE pa0023-land1,
    branc TYPE pa0023-branc,
    taete TYPE pa0023-taete,
  END OF ty_0023.
DATA : t_0006 TYPE STANDARD TABLE OF ty_0006,
       t_0007 TYPE STANDARD TABLE OF ty_0007,
       t_0022 TYPE STANDARD TABLE OF ty_0022,
       t_0023 TYPE STANDARD TABLE OF ty_0023.

* Selecting from tables
SELECT pernr subj try2 stras ort01 pstlz landl FROM pa0006 INTO TABLE t_0006.
SELECT pernr subj try2 begda schkz mostd wostd FROM pa0007 INTO TABLE t_0007.
SELECT pernr subj try2 insti sland slabs anzk1 earm FROM pa0022 INTO TABLE t_0022.
SELECT pernr begda endda arbgb ort01 land1 branc taepe FROM pa0023 INTO TABLE t_0023.

* For ALV1 Planned Working time
  * Read Context
DATA lo_en_plannedworkingtime TYPE REF TO if_wd_context_node.
DATA lo_el_plannedworkingtime TYPE REF TO if_wd_context_element.
  * navigate from <CONTEXT> to <PLANNEDWORKINGTIME> via lead selection
lo_en_plannedworkingtime = wd_context->get_child_node( name = wd_this->wdctx_plannedworkingtime ).
  * get element via lead selection
lo_el_plannedworkingtime = lo_en_plannedworkingtime->get_element( ).

  * Instantiate used component
DATA lo_cmp_usage1 TYPE REF TO if_wd_component_usage.
lo_cmp_usage1 = wd_this->wd_couse_alv1( ).
IF lo_cmp_usage1->has_active_component() IS INITIAL.
  lo_cmp_usage1->create_component().
ENDIF.

DATA lo_interfacecontroller1 TYPE REF TO ifwci_salv_wd_table.
lo_interfacecontroller1 = wd_this->wd_cpfic_alv1( ).
  * get model
DATA lo_value1 TYPE REF TO cl_salv_wd_config_table.
lo_value1 = lo_interfacecontroller1->get_model( ).
  * Bind table
lo_en_plannedworkingtime->bind_table( t_0007 ).

  * For ALV2 Address
DATA lo_en_address TYPE REF TO if_wd_context_node.
DATA lo_el_address TYPE REF TO if_wd_context_element.
  * navigate from <CONTEXT> to <ADDRESS> via lead selection
lo_en_address = wd_context->get_child_node( name = wd_this->wdctx_address ).
  * get element via lead selection
lo_el_address = lo_en_address->get_element( ).
  * Instantiate used component
DATA lo_cmp_usage2 TYPE REF TO if_wd_component_usage.
lo_cmp_usage2 = wd_this->wd_cpuas_alv2( ).
IF lo_cmp_usage2->has_active_component() IS INITIAL.
  lo_cmp_usage2->create_component().
ENDIF.
DATA lo_interfacecontroller2 TYPE REF TO ifwci_salv_wd_table.
lo_interfacecontroller2 = wd_this->wd_cpfic_alv2( ).
  * Get the model
DATA lo_value2 TYPE REF TO cl_salv_wd_config_table.
lo_value2 = lo_interfacecontroller2->get_model( ).
  * Bind table
lo_en_address->bind_table( t_0006 ).
Web Dynpro: Multiple ALV Grids and Layouts in ALV

* For ALV3 Education
DATA lo_nd_education TYPE REF TO if_wd_context_node.
DATA lo_el_education TYPE REF TO if_wd_context_element.
* navigate from <CONTEXT> to <EDUCATION> via lead selection
lo_nd_education = wd_context->get_child_node( name = wd_this->wdctx_education ).
* get element via lead selection
lo_el_education = lo_nd_education->get_element( ).
* Instantiate used component
DATA lo_cmp_usage3 TYPE REF TO if_wd_component_usage.
lo_cmp_usage3 = wd_this->wd_cpuse_alv3( ).
IF lo_cmp_usage3->has_active_component( ) IS INITIAL.
  lo_cmp_usage3->create_component( ).
ENDIF
DATA lo_interfacecontroller3 TYPE REF TO iwci_salv_wd_table.
lo_interfacecontroller3 = wd_this->wd_cpifc_alv3( ).
* Get the model
DATA lo_value3 TYPE REF TO cl_salv_wd_config_table.
lo_value3 = lo_interfacecontroller3->get_model( ).
* Bind table
lo_nd_education->bind_table( t_0022 ).

* For ALV4 Experience
DATA lo_nd_experience TYPE REF TO if_wd_context_node.
DATA lo_el_experience TYPE REF TO if_wd_context_element.
* navigate from <CONTEXT> to <EXPERIENCE> via lead selection
lo_nd_experience = wd_context->get_child_node( name = wd_this->wdctx_experience ).
* get element via lead selection
lo_el_experience = lo_nd_experience->get_element( ).
* Instantiate used component
DATA lo_cmp_usage4 TYPE REF TO if_wd_component_usage.
lo_cmp_usage4 = wd_this->wd_cpuse_alv4( ).
IF lo_cmp_usage4->has_active_component( ) IS INITIAL.
  lo_cmp_usage4->create_component( ).
ENDIF
DATA lo_interfacecontroller4 TYPE REF TO iwci_salv_wd_table.
lo_interfacecontroller4 = wd_this->wd_cpifc_alv4( ).
* Get the model
DATA lo_value4 TYPE REF TO cl_salv_wd_config_table.
lo_value4 = lo_interfacecontroller4->get_model( ).
* Bind table
lo_nd_experience->bind_table( t_0023 ).
Here in any grid select settings, we can set the user specific layout as follows.
Then press Save as. Give a description. If Initial View is checked, then first page will be defaulted for the user. If it is unchecked, user still has the option to choose the view. But Standard view will be by default.

Here we can see the start and end date is hidden in Planned working time and the view selected is automatically Jaytest.
If we want to select all the fields, we can choose Standard view from the drop down.
Related Content
For more information, visit the Web Dynpro 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.