7 Steps for Creating ALV

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

Summary
The article is written to support the ABAP consultants who want to develop their first ALV in Webdynpro ABAP.

Author: J.Jayanthi
Company: Siemens Information Processing Services Pvt. Ltd.
Created on: 04 November, 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  
   Step 1: Creating Web Dynpro ................................................................................................. 3  
   Step 2: Component Controller ............................................................................................... 4  
   Step 3: Component Usages ..................................................................................................... 6  
   Step 4: Designing View .......................................................................................................... 6  
   Step 5: Code ........................................................................................................................... 8  
   Step 6: Window ...................................................................................................................... 11  
   Step 7: Creating Web Dynpro Application ............................................................................ 11  
Output ......................................................................................................................................... 12  
Related Content ......................................................................................................................... 13  
Disclaimer and Liability Notice .................................................................................................. 14
ALV in Web Dynpro ABAP
ABAP consultants will be familiar with ALV. This article will help them to create ALV in Webdynpro ABAP.

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 be displayed by the browser if the view is embedded in a window.

Step 1: Creating Web Dynpro
Go to SE80 and select Web Dynpro Comp./Intf. and provide the name (say ZZZ_JAYTEST13) and create. Enter the description and choose the type as Web Dynpro Component.

Mention the Component Use as ALV and Component as SALV_WD_TABLE in the Used Components tab in Web Dynpro (ZZZ_JAYTEST13).
This will create a Component Usages by name ALV.

**Step 2: Component Controller**

Go to Component Controller and Right click the context. Then select Create Node MAKT with dictionary structure MAKT and select the required attributes from MAKT by using Add Attribute from Structure.
Remove the dictionary structure MAKT from the node MAKT and set the properties as below (Cardinality, Lead selection, etc.).

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Node Name</td>
<td>MAKT</td>
</tr>
<tr>
<td>Interface Node</td>
<td></td>
</tr>
<tr>
<td>Input Element (Ext)</td>
<td></td>
</tr>
<tr>
<td>Dictionary Structure</td>
<td></td>
</tr>
<tr>
<td>- Cardinality</td>
<td>0..n</td>
</tr>
<tr>
<td>- Selection</td>
<td>0..1</td>
</tr>
<tr>
<td>Initialization Lead Selection</td>
<td></td>
</tr>
<tr>
<td>Singleton</td>
<td></td>
</tr>
<tr>
<td>Supply/Function</td>
<td></td>
</tr>
</tbody>
</table>
Step 3: Component Usages

Right click the component Usage (here with name ALV) and Click controller Usage button for creating the same.

Drag and drop the node (MAKT in right side) from Component Controller context to Data (in left side) in Controller Usage Context.

Once mapping is done to Data, message will appear as follows.

External mapping for context element DATA was defined

Step 4: Designing View

Go to the Context in the view and drag and drop the MAKT node which appears in Right side (Component Controller) to View.

After mapping, it will appear as below. symbol shows it is mapped.
Go to the layout in view and right click the ROOTUIELEMENTCONTAINER and then choose Insert element.

This will make the layout appear as below.

In the properties, use button to define as below.
Step 5: Code

Go to Methods in View and double click WDDOINIT method to write code. We are using this method because we are going to load the data initially.
Use the Webdynpro code Code wizard icon highlighted above to generate code.
First we need to read the context MAKT.

Keep the below code and delete the rest.

```
DATA lo_nd_makt TYPE REF TO if_wd_context_node.
  * navigate from <CONTEXT> to <MAKT> via lead selection
  lo_nd_makt = wd_context->get_child_node( name = wd_this->wdctx_makt ).
```

Then we need to instantiate the used component ALV.

This will generate the below code.

```
DATA lo_cmp_usage TYPE REF TO if_wd_component_usage.
  lo_cmp_usage = wd_this->wd_components_alv().
  IF lo_cmp_usage->has_active_component() IS INITIAL.
    lo_cmp_usage->create_component().
  ENDIF.
```

Then use the GET_MODEL method in the used controller.

Do the selection logic as required.

```
types : begin of ty_makt,
  makt_r type makt-makt,
  maktx type makt-maktx,
end of ty_makt

data t_makt type standard table of ty_makt.
select makt_r maktx into table t_makt from makt up to 10 rows.
```

Bind the table to the context node using BIND_TABLE method.
Complete Code:

```
* Bind table
lo_nd_makt->bind_table(t_makt).

method WDODOINIT .
* Read context node MAKT
  DATA lo_nd_makt TYPE REF TO if_wd_context_node.
  * navigate from <CONTEXT> to <MAKT> via lead selection
  lo_nd_makt = wd_context->get_child_node( name = wd_this->wocctx_makt ).
  * Instantiate Used component
  DATA lo_cmp_usage TYPE REF TO if_wd_component_usage.
  lo_cmp_usage = wd_this->wd_cpuse_alv().
  IF lo_cmp_usage->has_active_component( ) IS INITIAL.
    lo_cmp_usage->create_component( ).
  ENDIF.
* Get the model from used controller
  DATA lo_interfacecontroller TYPE REF TO iwci_salv_wd_table.
    lo_interfacecontroller = wd_this->wd_cpifc_alv().
    DATA lo_value TYPE REF TO cl_salv_wd_config_table.
    lo_value = lo_interfacecontroller->get_model( ).

  types : begin of ty_makt ,
    matnr type makt-matnr ,
    maktx type makt-maktx ,
  end of ty_makt.
  data t_makt type standard table of ty_makt.
  select matnr maktx into table t_makt from makt up to 10 rows.

* Bind table
lo_nd_makt->bind_table(t_makt).
endmethod
```
Step 6: Window
The window will appear like below. Right click C1 to embed ALV table into the view.

After that, the window will appear as follows.

Step 7: Creating Web Dynpro Application
Create Web Dynpro Application by right clicking the Webdynpro(ZZZ_JAYTEST13).
Right click the Web Dynpro component and activate.
Output

Double click the Web Dynpro Application and then Press F8 to execute.

Enter username and password of the application and then the output can be seen.
Related Content

SAP List Viewer in Web Dynpro ABAP

Creating a WebDynpro ALV Application in 30 easy steps

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.