

# Web Dynpro ABAP: Supply Function and ALV



## Applies to:

SAP ECC 6.0

## Summary

The article tells about displaying data in an ALV based on the selected line in another ALV. This is achieved by using Supply function and Singleton property in Web Dynpro ABAP.

**Author:** J.Jayanthi

**Company:** Siemens IT Solutions and Services Pvt. Ltd.

**Created on:** 12 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 .....	4
Component Controller.....	5
Designing View .....	8
Embedding View .....	13
Component Usages .....	14
Creating Web Dynpro Application .....	16
Code .....	16
Output .....	17
Related Content.....	18
Disclaimer and Liability Notice.....	19

## ALV in Web Dynpro ABAP

ABAP Consultants are generally familiar in SAP List Viewer (ALV).

We are going to see how to display data in an ALV based on the selected line in another ALV. In this article, MAKT is the parent node and MARD is the child node. MARD has singleton property enabled. Based on the selected line in ALV (MAKT display), we are going to display the material information with plant and storage location in another ALV.

### 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.

#### Singleton

The property "Singleton" specifies the number of instances that can exist in a dependent context node (that does not belong directly to the root node). If the property "Singleton" is set, exactly one instance of the node exists. Its content changes when the lead selection of the parent node changes. If the property "Singleton" is not set, one instance per parent instance exists. The content of the instances does not change when the lead selection of the parent changes.

#### Supply Function

We can specify the name of a method which we can use to fill the context node at runtime in supply function.

## Creating Web Dynpro

Go to SE80 and select Web Dynpro Comp./Intf. and provide the name(say ZZZ\_JAYTEST8) to create. Then enter the description and choose the type as Web Dynpro Component.

Mention the Component Use as ALV1 and ALV2 and Component as SALV\_WD\_TABLE in the Used Components tab in Web Dynpro (ZZZ\_JAYTEST8).

Web Dynpro Component	ZZZ_JAYTEST8	Active
Description	Supply Function	
Assistance Class		
Created By	JAYARAMAN . J	Created On 11.10.2010
Last Changed By	JAYARAMAN . J	Changed On 12.10.2010
Original Lang.	EN	Package \$TMP
<input checked="" type="checkbox"/> Accessibility Checks Active		
<div style="display: flex; justify-content: space-between;"> <span>Used Components</span> <span>Implemented interfaces</span> </div>		
<div style="display: flex; align-items: center;"> </div>		
Used Web Dynpro Components		
Component Use	Component	Description of Component
ALV1	SALV_WD_TABLE	ALV Component
ALV2	SALV_WD_TABLE	ALV Component

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

## Component Controller

Go to Component Controller and Right click the context. Then select Create Node to create Parent Node MAKT and select the attributes as MATNR, MAKTX.

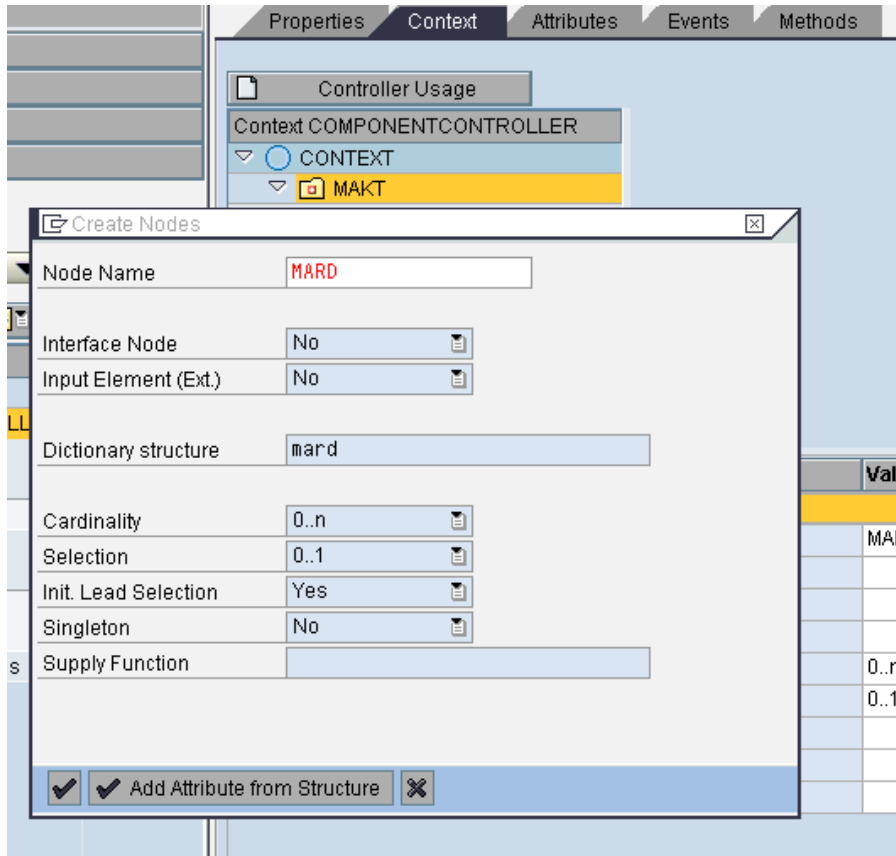
The screenshot shows the 'Create Nodes' dialog box with the following configuration:

Node Name	MAKT
Interface Node	No
Input Element (Ext.)	No
Dictionary structure	makt
Cardinality	0..n
Selection	0..1
Init. Lead Selection	Yes
Singleton	No
Supply Function	

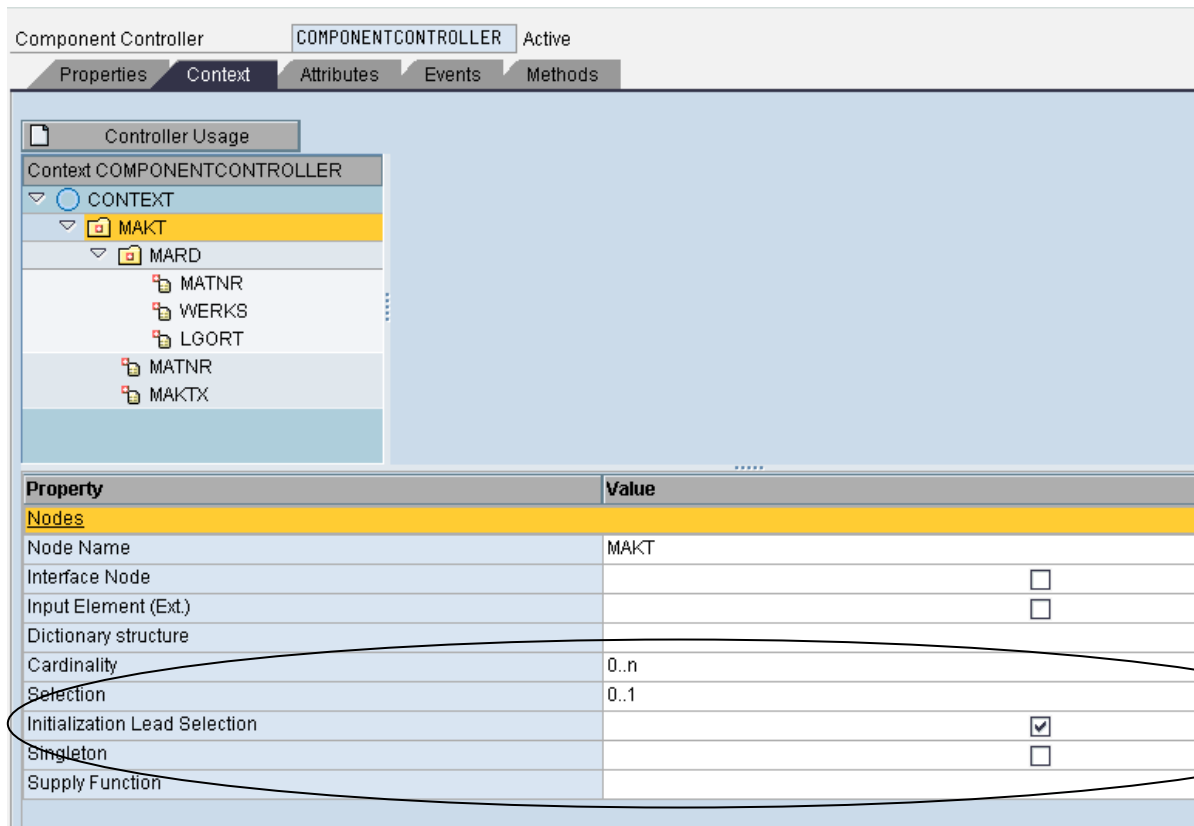
At the bottom of the dialog, there is a checked checkbox and a button labeled 'Add Attribute from Structure'.

Select MATNR and MAKTX using Add attribute from structure.

Inside this node, create another node MARD, by right clicking the node MAKT.



Select MATNR, WERKS and LGORT using Add Attribute from Structure button. Change the property of MAKT and MARD as below.



Property	Value
<b>Nodes</b>	
Node Name	MARD
Interface Node	<input type="checkbox"/>
Input Element (Ext.)	<input type="checkbox"/>
Dictionary structure	
Cardinality	0..n
Selection	0..1
Initialization Lead Selection	<input type="checkbox"/>
Singleton	<input checked="" type="checkbox"/>
Supply Function	GET_DATA

Here we enabled singleton property so that there can be one instance of the child node MARD at runtime.

Double click the method GET\_DATA and code will be automatically generated with comment. Keep the required code as below and write logic for filling child node.

```

method GET_DATA .
* if necessary, get static attributes of parent element
DATA ls_parent_attributes TYPE wd_this->element_makt.
parent_element->get_static_attributes(
IMPORTING
static_attributes = ls_parent_attributes ).

* data declaration
DATA lt_mard TYPE wd_this->Elements_mard.
DATA ls_mard LIKE LINE OF lt_mard.
** @TODO compute values
** e.g. call a data providing FuBa
select matnr werks lgorrt into table lt_mard
from mard where matnr = ls_parent_attributes-matnr.
* bind all the elements
node->bind_table(
new_items = lt_mard
set_initial_elements = abap_true ).
endmethod.

```

## Designing View

Go to the properties tab of Main view and then create as below.

The screenshot shows the 'Properties' tab of the 'MAIN' view. The description is 'Singleton and Supply Function' and the lifetime is 'framework contr'. The view was created by 'JAYARAMAN . J' on '11.10.2010' and last changed by 'JAYARAMAN . J' on '12.10.2010'. Below the metadata is a table titled 'Used Controllers/Components'.

Component Use	Component	Controller	Description
	ZZZ_JAYTEST8	COMPONENTCONTROLLER	
ALV1	SALV_WD_TABLE		ALV Component
ALV1	SALV_WD_TABLE	INTERFACECONTROLLER	
ALV2	SALV_WD_TABLE		ALV Component
ALV2	SALV_WD_TABLE	INTERFACECONTROLLER	

In context tab, drag and drop the context which appears under component controller to view (Main is the view name).

The screenshot shows the 'Context' tab of the 'MAIN' view. On the left, the 'Context MAIN' tree shows a 'CONTEXT' node with a sub-node 'MAKT' containing 'MARD', 'MATNR', and 'MAKTX'. On the right, the 'Context COMPONENTCONTROLLER' tree shows a 'CONTEXT' node with a sub-node 'MAKT' containing 'MARD', 'MATNR', and 'MAKTX'. A blue arrow points from the 'CONTEXT' node in the controller's context to the 'CONTEXT' node in the view's context, indicating the mapping.

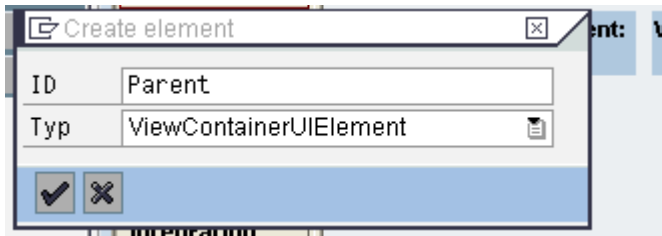
The symbol in left side  shows it is mapped.



Then in layout, set the properties for ROOTUIELEMENTCONTAINER as follows.

Property	Value	Bi...
<b>Properties (TransparentContainer)</b>		
ID	ROOTUIELEMENTCC	
Layout	MatrixLayout	
accessibilityD...		
defaultButtonC...		
enabled	<input checked="" type="checkbox"/>	
height		
isLayoutConta...	<input checked="" type="checkbox"/>	
scrollingMode	none	
tooltip		
visible	Visible	
width		
<b>Layout (MatrixLayout)</b>		
stretchedHoriz	<input type="checkbox"/>	
stretchedVertic	<input type="checkbox"/>	

Right click the ROOTUIELEMENTCONTAINER and then choose Insert element. Create a Transparent Container with ID TC1 and then inside that Create ViewContainerUIElement.



Change the property of TC1 and Parent as below.

Property	Value	B
ID	TC1	
Layout	MatrixLayout	
Layout Data	MatrixHeadData	
accessibility		
defaultButton		
enabled	<input checked="" type="checkbox"/>	
height	80px	
isLayoutContainer	<input checked="" type="checkbox"/>	
scrollingMode	none	
tooltip		
visible	Visible	
width		
<b>Layout (MatrixLayout)</b>		
stretchedHorizontal	<input type="checkbox"/>	
stretchedVertical	<input type="checkbox"/>	
<b>Layout Data (MatrixHeadData)</b>		
cellBackground	transparent	
cellDesign	rPad	
colSpan	1	
height	70px	
hAlign	beginOfLine	
vAlign	baseline	
vGutter	none	
width	150px	

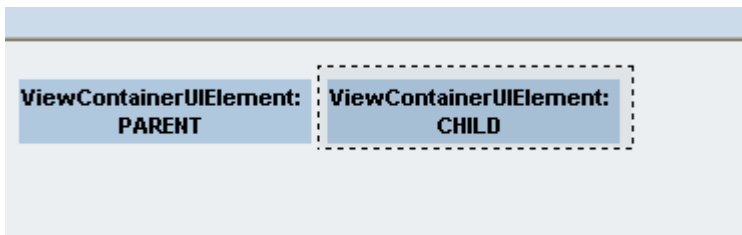
Property	Value	Bi...
<b>Properties (ViewContainerUIElement)</b>		
ID	PARENT	
Layout Data	MatrixHeadData	
enabled	<input checked="" type="checkbox"/>	
tooltip		
visible	Visible	
<b>Layout Data (MatrixHeadData)</b>		
cellBackground	transparent	
cellDesign	rPad	
colSpan	1	
height		
hAlign	beginOfLine	
vAlign	baseline	
vGutter	none	
width		

Similarly, create another transparent container TC2 and inside that create a ViewContainerUIElement by name Child.

Property	Value	Bi...
ID	TC2	
Layout	MatrixLayout	
Layout Data	MatrixData	
accessibility		
defaultButton		
enabled	<input checked="" type="checkbox"/>	
height		
isLayoutCon	<input checked="" type="checkbox"/>	
scrollingMod	none	
tooltip		
visible	Visible	
width		
<b>Layout (MatrixLayout)</b>		
stretchedHor	<input type="checkbox"/>	
stretchedVer	<input type="checkbox"/>	
<b>Layout Data (MatrixData)</b>		
cellBackgrou	transparent	
cellDesign	rPad	
colSpan	1	
height		
hAlign	beginOfLine	
vAlign	baseline	
vGutter	none	
width	150px	

Property	Value	Bi...
<b>Properties (ViewContainerUIElement)</b>		
ID	CHILD	
Layout Data	MatrixHeadData	
enabled	<input checked="" type="checkbox"/>	
tooltip		
visible	Visible	
<b>Layout Data (MatrixHeadData)</b>		
cellBackgrou	transparent	
cellDesign	rPad	
colSpan	1	
height	80px	
hAlign	beginOfLine	
vAlign	baseline	
vGutter	none	
width		

After that, layout will appear as below.



Select the method WDD0INIT (For Address Button) in methods tab.

Use Web Dynpro code wizard  to generate code automatically.

**Step a:**

Choose the radio button Read context for the node MAKT.

The screenshot shows the 'Read Context' radio button selected. Below it, a text field contains the node name 'MAKT'.

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

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

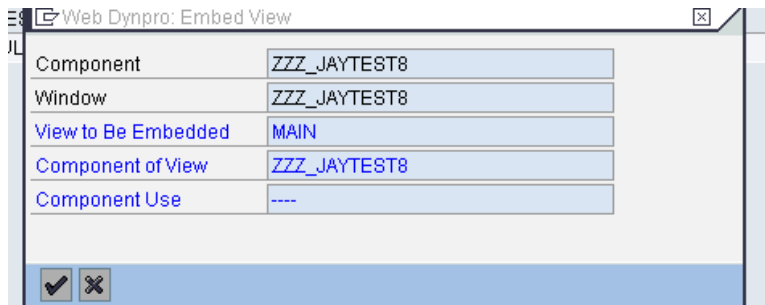
**Step b:**

Then select the data by normal ABAP statement (declare t\_alv1 as required) and bind the table to the makt node.

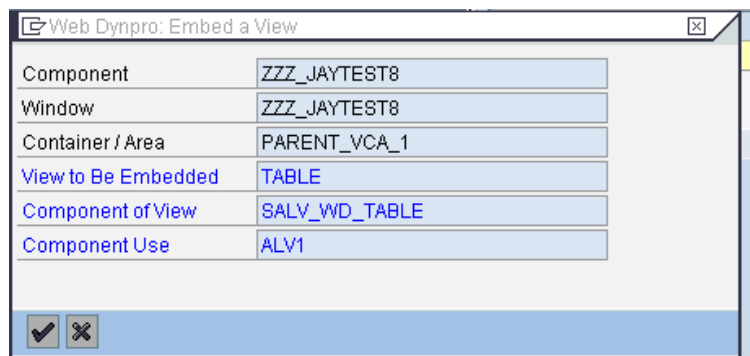
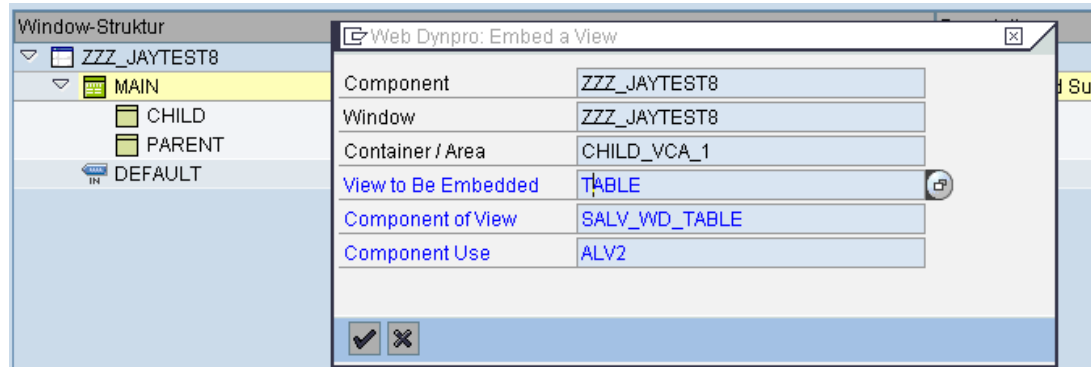
```
types : begin of ty_alv1,  
       matnr type makt-matnr,  
       maktx type makt-maktx,  
       end of ty_alv1.  
  
data : t_alv1 type standard table of ty_alv1,  
       wa_alv1 type ty_alv1.  
  
select matnr maktx from makt into table t_alv1  
up to 10 rows where spras = sy-langu.  
  
lo_nd_makt->bind_table( t_alv1 ).
```

## Embedding View

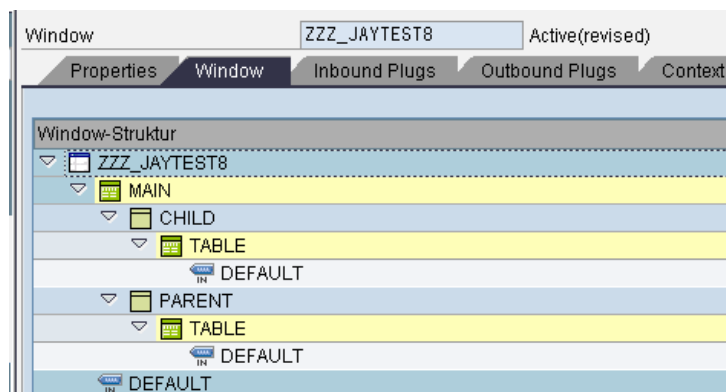
Right click and embed the view Main.



Then inside Child and Parent, embed the table.

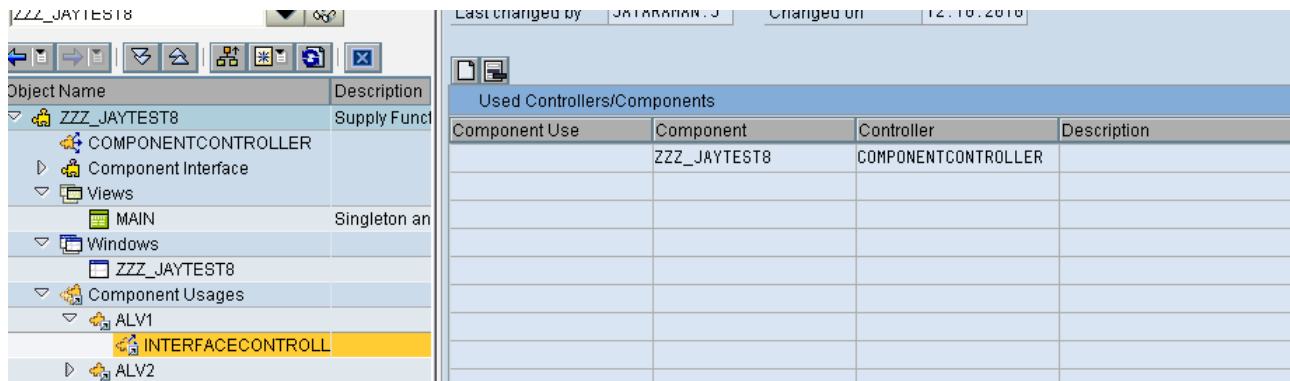


After embedding, it will appear like this.



## Component Usages

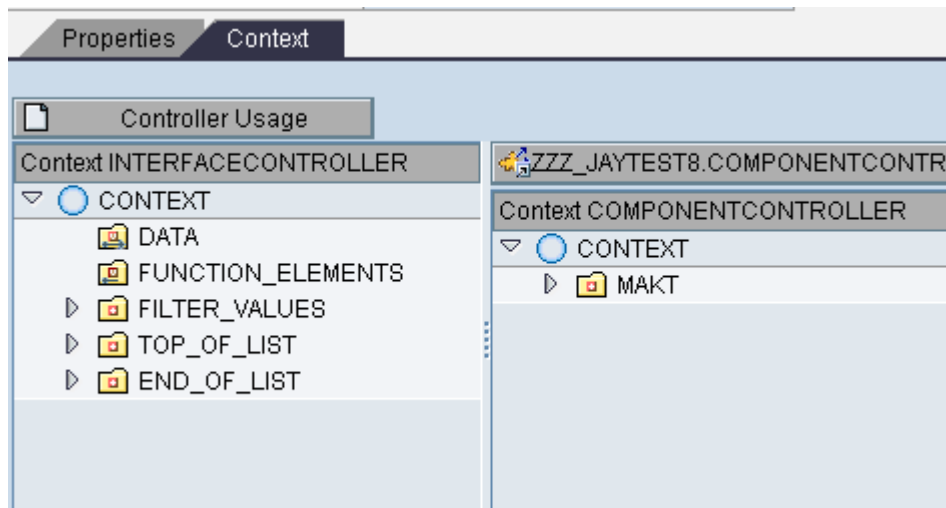
Properties of Component usage should be as below.



Used Controllers/Components			
Component Use	Component	Controller	Description
	ZZZ_JAYTEST8	COMPONENTCONTROLLER	

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

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



Similarly, for ALV2, set the properties as explained above and map MARD.

Properties Context

Controller Usage

Context INTERFACECONTROLLER

- CONTEXT
  - DATA
  - FUNCTION\_ELEMENTS
  - FILTER\_VALUES
  - TOP\_OF\_LIST
  - END\_OF\_LIST

ZZZ\_JAYTEST8.COMPONENTCONTROLLER

Context COMPONENTCONTROLLER

- CONTEXT
  - MAKT
    - MARD
    - MATNR
    - MAKTX

Property	Value
<b>Nodes</b>	
Node Name	DATA
Interface Node	<input type="checkbox"/>
Input Element (Ext.)	<input checked="" type="checkbox"/>
Dictionary structure	
Cardinality	0..n
Selection	0..1
Initialization Lead Selection	<input checked="" type="checkbox"/>
Singleton	<input checked="" type="checkbox"/>
Supply Function	
Mapping Path	ZZZ_JAYTEST8.COMPONENTCONTROLLER.MAKT.MARD

## Creating Web Dynpro Application

Create Web Dynpro Application by right clicking the Webdynpro (ZZZ\_JAYTEST8).  
Right click the Web Dynpro component and activate.

### Code

Supply Function method Get data in component controller.

```

method GET_DATA .
  * if necessary, get static attributes of parent element
  DATA ls_parent_attributes TYPE wd_this->element_makt.
  parent_element->get_static_attributes(
    IMPORTING
      static_attributes = ls_parent_attributes ).

  * data declaration
  DATA lt_mard TYPE wd_this->Elements_mard.
  DATA ls_mard LIKE LINE OF lt_mard.
  ** @TODO compute values
  ** e.g. call a data providing FuBa
  select matnr werks lgort into table lt_mard
    from mard where matnr = ls_parent_attributes-matnr.
  * bind all the elements
  node->bind_table(
    new_items          = lt_mard
    set_initial_elements = abap_true ).
endmethod.

```

---

```

method WDDOINIT .

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

  * get element via lead selection
  lo_el_makt = lo_nd_makt->get_element( ).

types : begin of ty_alv1,
        matnr type makt-matnr,
        maktx type makt-maktx,
        end of ty_alv1.

data : t_alv1 type standard table of ty_alv1,
        wa_alv1 type ty_alv1.

select matnr maktx from makt into table t_alv1
up to 10 rows where spras = sy-langu.

lo_nd_makt->bind_table( t_alv1 ).
endmethod.

```



## Output

Material	Material Description	Material	Plnt	SLoc
0000000001	Test CATT BPZ:WFW20.D110 /Uni-WZ 110mm	0000000001	1021	001A
0000000002	Test CATT ERSATZ-MATERIAL FÜR FDB	0000000001	1021	REPL
0000000004	Test CATT ERSATZ-MATERIAL FÜR FDB	0000000001	1025	0023
0000000018	Test CATT - EN Text	0000000001	1025	REPL
0000000022	test material	0000000001	1025	Z3
0000000024	hhh	0000000001	1125	REPL
0000000025	test material	0000000001	1227	0023
0000000026	test material			
0000000027	test material			
0000000029	test material			

If selected line in first alv is changed, second alv will be refreshed automatically.

Material	Material Description	Material	Plnt	SLoc
0000000001	Test CATT BPZ:WFW20.D110 /Uni-WZ 110mm			
0000000002	Test CATT ERSATZ-MATERIAL FÜR FDB			
0000000004	Test CATT ERSATZ-MATERIAL FÜR FDB	0000000004	1025	0023
0000000018	Test CATT - EN Text			
0000000022	test material			
0000000024	hhh			
0000000025	test material			
0000000026	test material			
0000000027	test material			
0000000029	test material			

Since we enabled Initialization lead selection property, first line is selected always.

## Related Content

[Web Dynpro: Coloring Table Conditionally](#)

[Web Dynpro: Column Coloring in ALV](#)

## 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.