

Web Dynpro ABAP: ALV and Table in Popup Window



Applies to:

SAP ECC 6.0

Summary

Normally in ABAP, we come across the scenario of displaying ALV in popup. This article tells about displaying data both in ALV and table format in popup window using Webdynpro ABAP.

Author: J.Jayanthi

Company: Siemens IT Solutions and Services Pvt. Ltd.

Created on: 11 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	4
Designing View	7
Embedding View	17
Component Usages	19
Creating Web Dynpro Application	20
Code	20
Output.....	22
Related Content	23
Disclaimer and Liability Notice	24

ALV in Web Dynpro ABAP

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

We are going to see how to design popup window with ALV and Table display format. In this article, Main window contains the input field of Personnel Number with two buttons named Address and Details. Address button will display the address of the employee in Table format in popup window. Details button will display the details of the employee as ALV in Popup window.

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.

Creating Web Dynpro

Go to SE80 and select Web Dynpro Comp./Intf. and provide the name(say ZZZ_JAYTEST7) to create. Then enter the description and choose the type as Web Dynpro Component.

Mention the Component Use as POPUP1 and Component as SALV_WD_TABLE in the Used Components tab in Web Dynpro (ZZZ_JAYTEST7).

Web Dynpro Component	ZZZ_JAYTEST7	Active	
Description	ALV in popup window		
Assistance Class			
Created By	JAYARAMAN . J	Created On	24 . 09 . 2010
Last Changed By	JAYARAMAN . J	Changed On	11 . 10 . 2010
Original Lang.	EN	Package	\$TMP
<input checked="" type="checkbox"/> Accessibility Checks Active			

Used Components		Implemented interfaces
Used Web Dynpro Components		
Component Use	Component	Description of Component
POPUP1	SALV_WD_TABLE	ALV Component

This will create a Component Usages by name POPUP1.

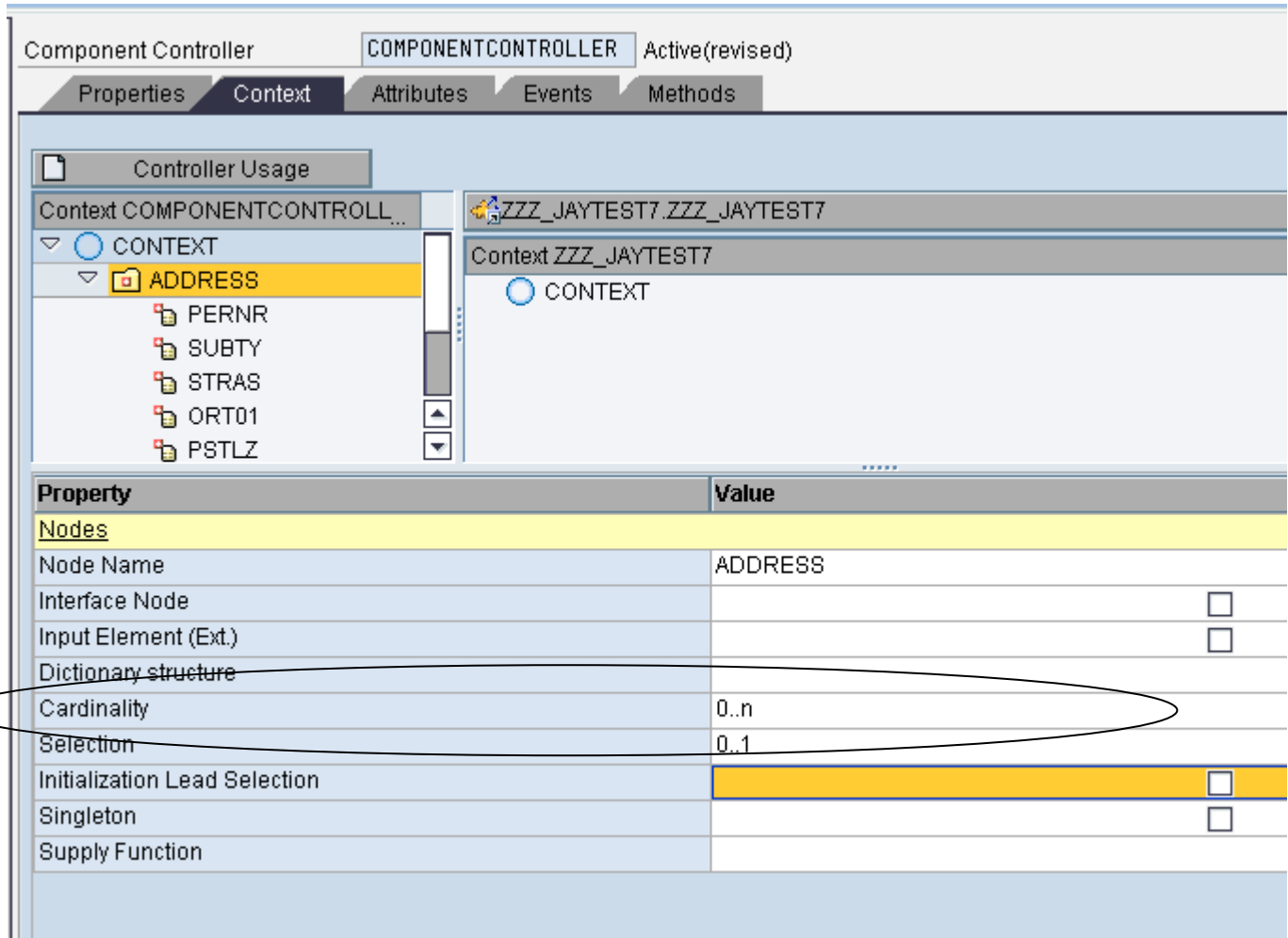
Component Controller

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

Create Nodes	
Node Name	ADDRESS
Interface Node	No
Input Element (Ext.)	No
Dictionary structure	pa0006
Cardinality	0..n
Selection	0..1
Init. Lead Selection	Yes
Singleton	No
Supply Function	
<input checked="" type="checkbox"/> <input checked="" type="checkbox"/> Add Attribute from Structure <input type="checkbox"/>	

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.



The screenshot shows the SAP Web Dynpro ABAP development environment. The 'Context' tab is active, and the 'ADDRESS' node is selected. The 'Property Value' table below shows the configuration for the selected node.

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

Similarly create another node Details using PA0002 table and add attributes from structure and then delete the dictionary structure after that.

Component Controller COMPONENTCONTROLLER Active(revised)

Properties **Context** Attributes Events Methods

Controller Usage

Context COMPONENTCONTROLL... ZZZ_JAYTEST7.ZZZ_JAYTEST7

CONTEXT

- ADDRESS
- DETAILS**
 - PERNR
 - ENDDA
 - BEGDA
 - NACHN
 - VORNA
 - CNAME
 - MIDNM

Context ZZZ_JAYTEST7

- CONTEXT

Property	Value
Nodes	
Node Name	DETAILS
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 type="checkbox"/>
Supply Function	

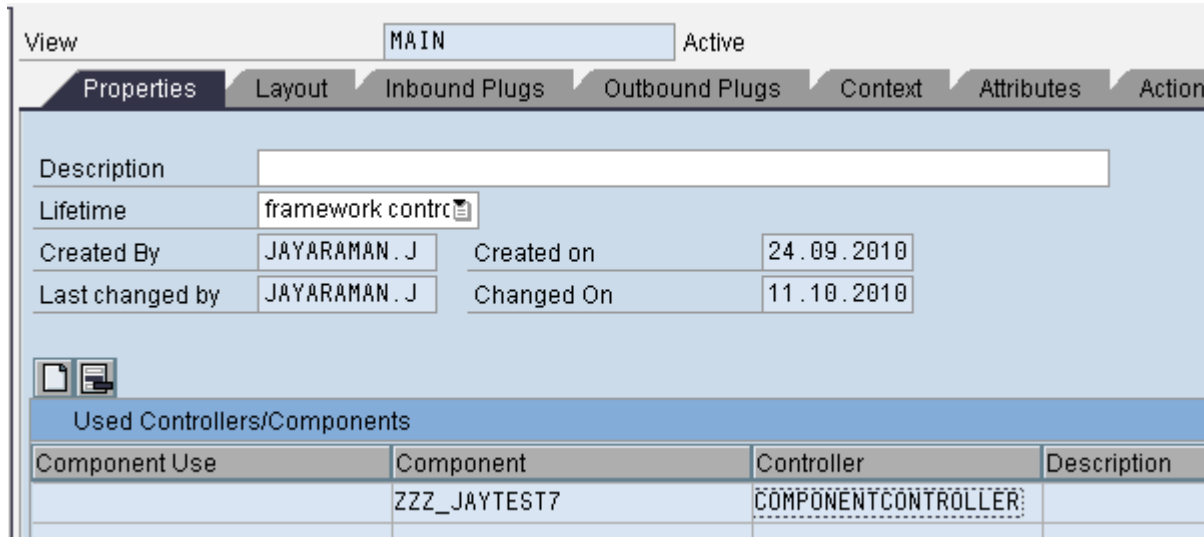
Create Attribute Pernr with properties as below.

Property	Value
Attribute	
Attribute Name	PERNR
Type assignment	Type
Type	PA0006-PERNR
Read-only	<input type="checkbox"/>
Default Value	
Null Value	<input type="checkbox"/>
Input Help Mode	Automatic
Determined Input Help	PA0003
Type of Input Help	Check table
Formatting	
Display Leading Zeros	Default Value

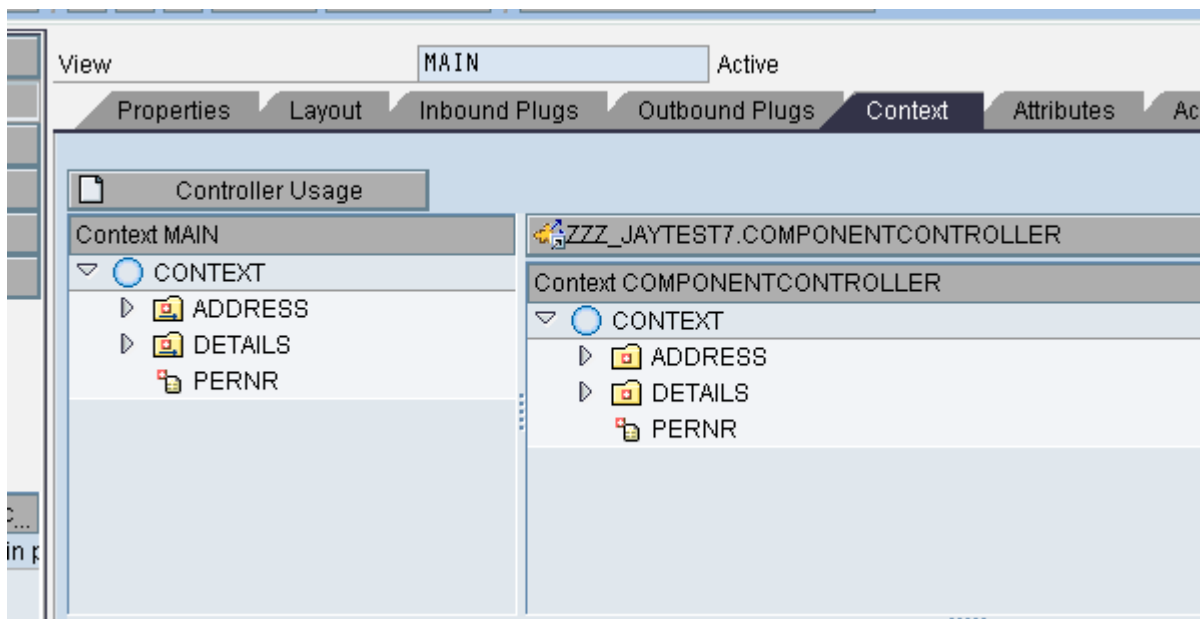
Designing View

We are going to design three views (Main, POPUP, POPUP1). POPUP will be used for Table display and POPUP1 will be used for ALV display.

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



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



The symbol in left side  shows it is mapped.

Then in layout, set the properties for ROOTUIELEMENTCONTAINER as follows. Go to the layout in view and right click the ROOTUIELEMENTCONTAINER and then choose Insert element. Create a Label with ID Employee and then InputField with ID PERNR.


Property	Value	Binding
Properties (TransparentContainer)		
ID	ROOTUIELEMENTCONTAINER	
Layout	MatrixLayout	
accessibilityDescription		
defaultButtonId		
enabled	<input checked="" type="checkbox"/>	
height		

CONTEXT_MENUS
 ROOTUIELEMENTCONTAINER

EMPLOYEE





PERNR
 ADDRESS
 DETAILS


Property	Value	Binding
Properties (Label)		
ID	EMPLOYEE	
Layout Data	MatrixHeadData	
design	standard	
enabled	<input checked="" type="checkbox"/>	
labelFor	PERNR	
text	Personnel Number	
textDirection	inherit	
tooltip		
visible	Visible	

Property	Value	Binding
Properties (InputField)		
ID	PERNR	
Layout Data	MatrixData	
alignment	auto	
enabled	<input checked="" type="checkbox"/>	
explanation		
length	0	
passwordField	<input type="checkbox"/>	
readOnly	<input type="checkbox"/>	
state	Normal Item	
textDirection	inherit	
tooltip		
value	MAIN.PERNR	
visible	Visible	
width		

Choose the yellow icon in the right side of the input field PERNR properties and choose the value as above. Then create two buttons Address and Details with events ONCLICK for Address and ONDETAILSCLICK for Details.

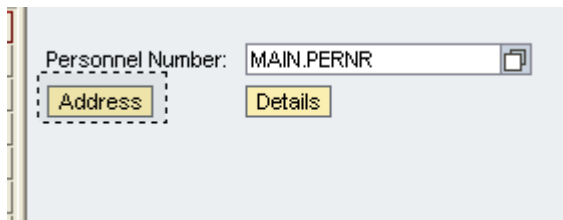
CONTEXT_MENUS
 ROOTUIELEMENTCONTAINER

-  EMPLOYEE
-  PERNR
-  ADDRESS
-  DETAILS

Property	Value	Binding
Properties (Button)		
ID	ADDRESS	
Layout Data	MatrixHeadData	
design	standard	
enabled	<input checked="" type="checkbox"/>	
explanation		
imageFirst	<input checked="" type="checkbox"/>	
imageSource		
text	Address	
textDirection	inherit	
tooltip		
visible	Visible	
width		
Events		
onAction	ONCLICK	

Property	Value	Binding
Properties (Button)		
ID	DETAILS	
Layout Data	MatrixData	
design	standard	
enabled	<input checked="" type="checkbox"/>	
explanation		
imageFirst	<input checked="" type="checkbox"/>	
imageSource		
text	Details	
textDirection	inherit	
tooltip		
visible	Visible	
width		
Events		
onAction	ONDETAILSCCLICK	
Layout Data (MatrixData)		
cellBackgroundDesign	transparent	

After that, layout will appear as below.



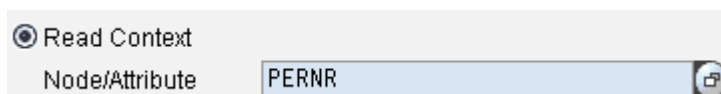
Select the method ONACTIONONCLICK (For Address Button) in methods tab.



Use Web Dynpro code wizard to generate code automatically.

Step a:

Choose the radio button Read context for the attribute PERNR.



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

```

DATA lo_el_context TYPE REF TO if_wd_context_element.
DATA ls_context TYPE wd_this->element_context.
DATA lv_pernr LIKE ls_context-pernr.
* get element via lead selection
lo_el_context = wd_context->get_element( ).
* get single attribute
lo_el_context->get_attribute(
  EXPORTING
    name = `PERNR`
  IMPORTING
    value = lv_pernr ).

```

Step b:

Again use Code Wizard as below for Address Node.

The screenshot shows the 'Code Wizard' dialog box. The 'Read Context' radio button is selected. Below it, the 'Node/Attribute' field contains the text 'ADDRESS'.

This will generate the code. Keep the below required code.

```

DATA lo_nd_address TYPE REF TO if_wd_context_node.
* Navigate from <CONTEXT> to <ADDRESS> via lead selection
lo_nd_address = wd_context->get_child_node( name =
  wd_this->wdctx_address ).

```

Step c:

To generate POPUP for Address details, do as below.

The screenshot shows the 'Code Wizard' dialog box. The 'Generate Popup' radio button is selected. Below it, the 'Component Use' field is empty, and the 'Window Name' field contains the text 'POP'.

```

* Generate POPUP for Address Display
DATA lo_window_manager TYPE REF TO if_wd_window_manager.
DATA lo_api_component TYPE REF TO if_wd_component.
DATA lo_window TYPE REF TO if_wd_window.

lo_api_component = wd_comp_controller->wd_get_api( ).
lo_window_manager = lo_api_component->get_window_manager( ).
lo_window = lo_window_manager->create_window(
  window_name = `POP`
  title = `Address Details`
  close_in_any_case = abap_true
  close_button = abap_true ).

lo_window->open( ).

```

Step d:

Then select the data by normal ABAP statement (declare t_0006 as required) and bind the table to the address node.

```

TYPES : BEGIN OF ty_0006,
        pernr TYPE pa0006-pernr,
        subty TYPE pa0006-subty,
        stras TYPE pa0006-stras,
        ort01 TYPE pa0006-ort01,
        pstlz TYPE pa0006-pstlz,
        END OF ty_0006.
DATA t_0006 TYPE STANDARD TABLE OF ty_0006.

SELECT pernr subty stras ort01 pstlz FROM pa0006 INTO TABLE t_0006
      WHERE pernr = lv_pernr.

* Binding table to the node
lv_nd_address->bind_table( t_0006 ).

```

Repeat the steps from a to d for Details button also in the event.

For the step c to generate popup ALV window for Details, do like below.

Generate Popup

Component Use	POPUP1
Window Name	TABLE

This will generate the code and change it as below.

```

DATA lo_window_manager TYPE REF TO if_wd_window_manager.
DATA lo_api_component  TYPE REF TO if_wd_component.
DATA lo_window         TYPE REF TO if_wd_window.

lo_api_component = wd_comp_controller->wd_get_api( ).
lo_window_manager = lo_api_component->get_window_manager( ).
lo_window = lo_window_manager->create_window_for_cmp_usage(
    interface_view_name = 'TABLE'
    component_usage_name = 'POPUP1'
    title                = 'Employee Details'
    close_in_any_case   = abap_true
    message_display_mode =
if_wd_window=>co_msg_display_mode_selected
).
lo_window->open( ).

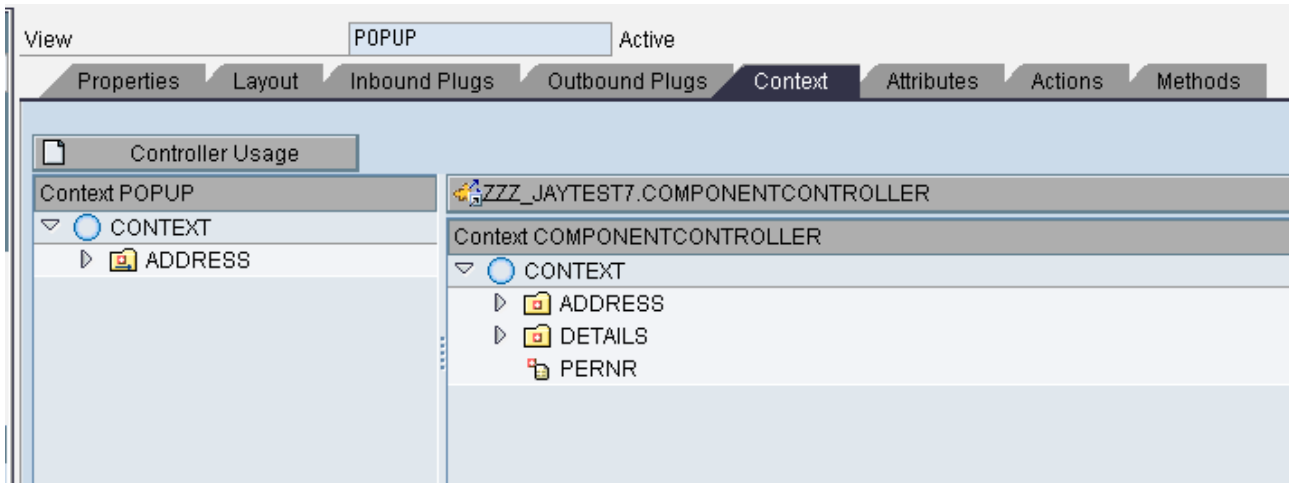
```

Create another view say POPUP.

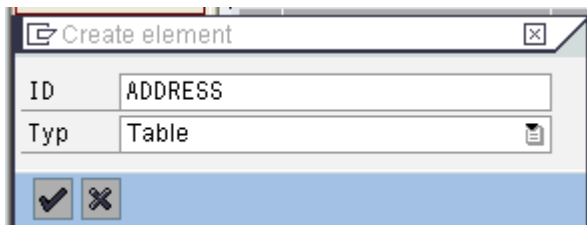
Design the properties as below.

Component Use	Component	Controller	Description
ZZZ_JAYTEST7	COMPONENTCONTROLLER		

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



Go to Layout of this view and create a table by using Insert element in ROOTUIELEMENTCONTAINER.



For every column, in the text editor, select the value by choosing the yellow color icon in the right side as required.

Property	Value	Binding
Properties (InputField)		
ID	ADDRESS_PERNR_EDITOR	
alignment	auto	
enabled	<input checked="" type="checkbox"/>	
explanation		
length	0	
passwordField	<input type="checkbox"/>	
readOnly	<input type="checkbox"/>	
state	Normal Item	
textDirection	inherit	
tooltip		
value	POPUP.ADDRESS.PERNR	

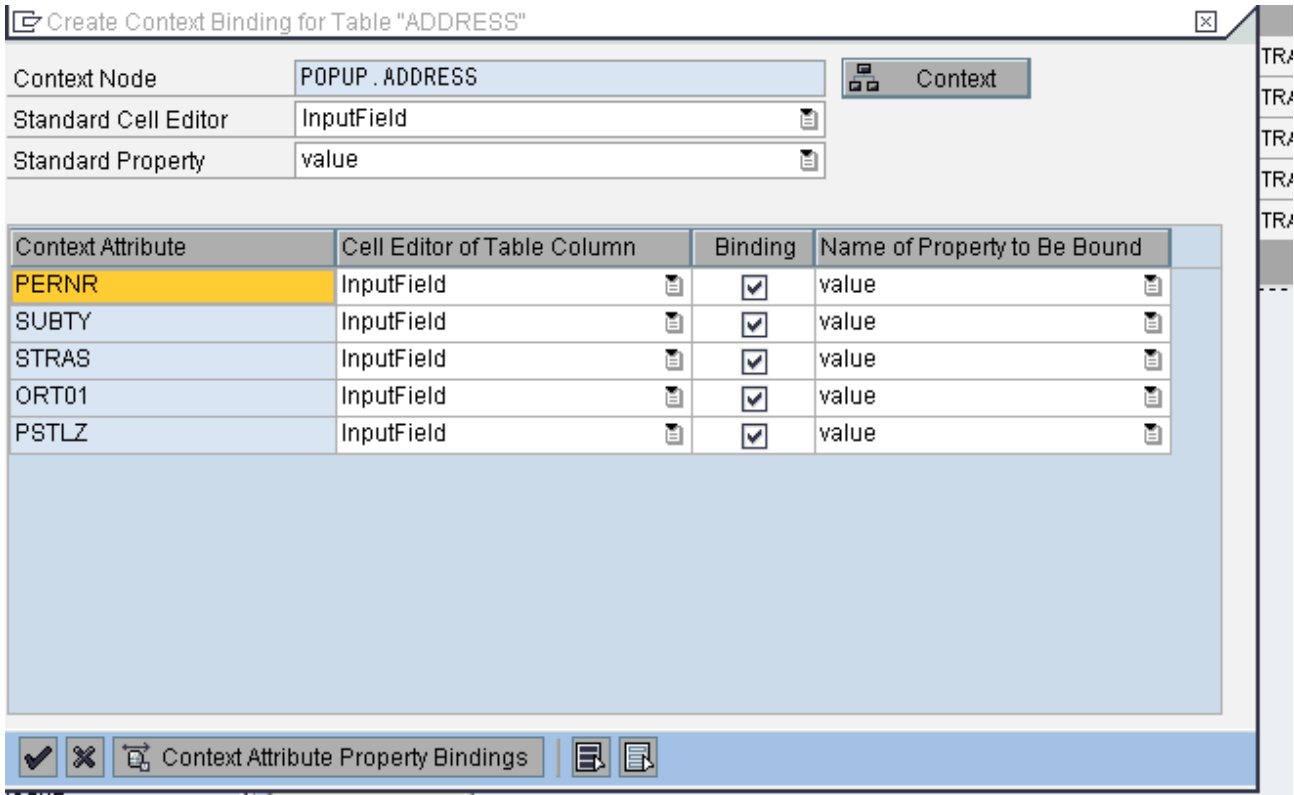
Similarly do for other columns.

We can change the text for each column as below.

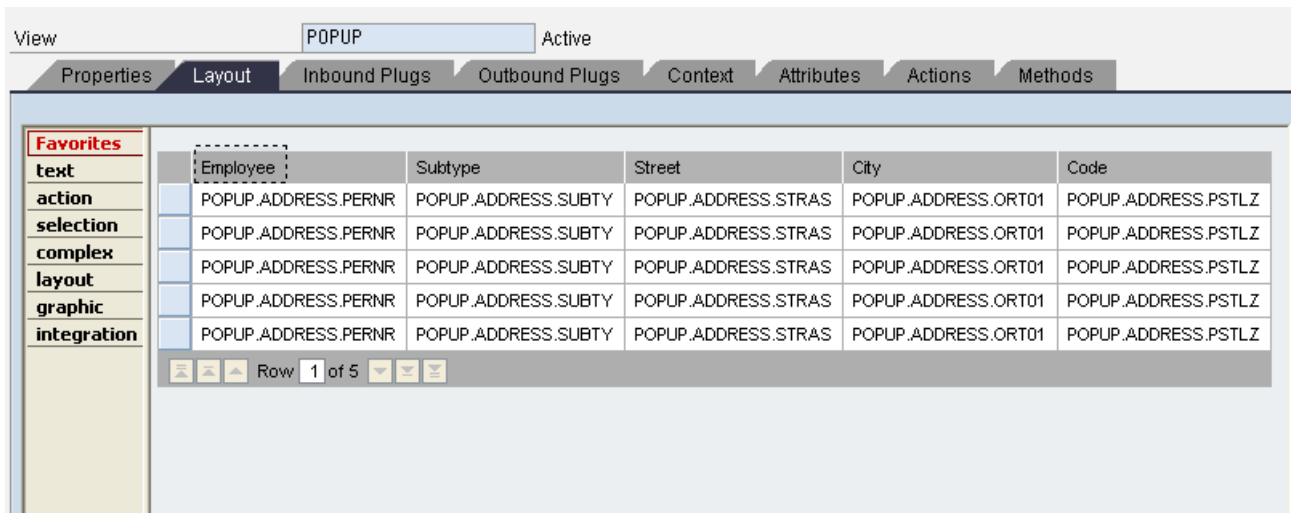
Property	Value	Binding
Properties (Caption)		
ID	ADDRESS_PERNR_HEADER	
enabled	<input checked="" type="checkbox"/>	
imageFirst	<input checked="" type="checkbox"/>	
imageSource		
text	Employee	
textDirection	inherit	
tooltip		
visible	Visible	

Then right click the table and create binding.

Employee	Subtype	Street	City
POPOP.ADDRESS.PERNR	POPOP.ADDRESS.SUBTY	POPOP.ADDRESS.STRAS	POPOP
POPOP.ADDRESS.PERNR	POPOP.ADDRESS.SUBTY	POPOP.ADDRESS.STRAS	POPOP
POPOP.ADDRESS.PERNR	POPOP.ADDRESS.SUBTY	POPOP.ADDRESS.STRAS	POPOP
POPOP.ADDRESS.PERNR	POPOP.ADDRESS.SUBTY	POPOP.ADDRESS.STRAS	POPOP
POPOP.ADDRESS.PERNR	POPOP.ADDRESS.SUBTY	POPOP.ADDRESS.STRAS	POPOP

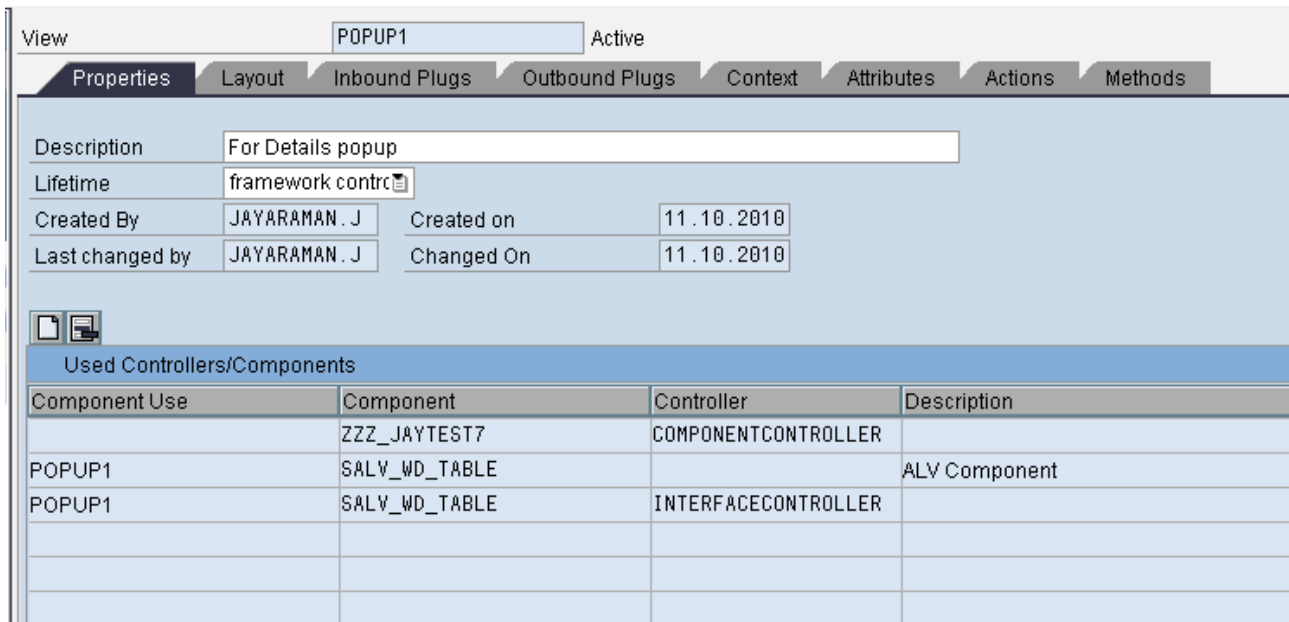


Layout will look like below.

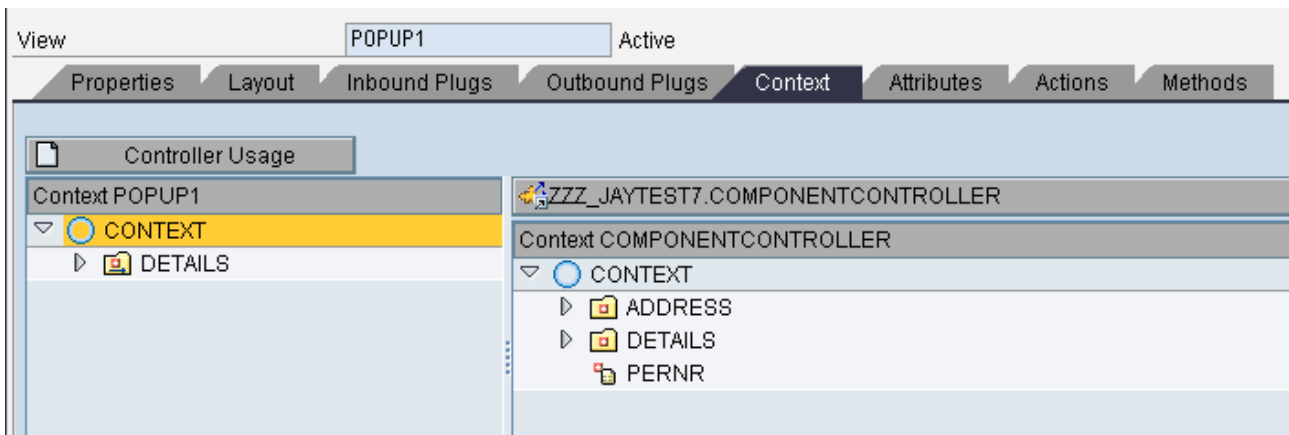


Create another view POPUP1 for ALV display of employee details.

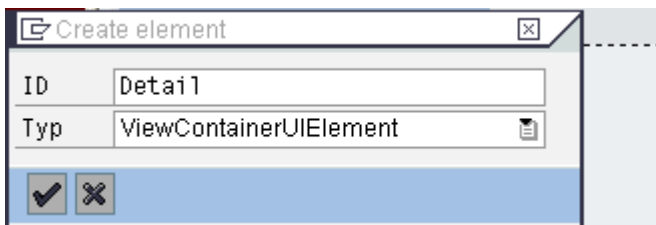
Design the properties of the view POPUP1 as below.



Do the mapping in the context by drag and drop the Details Node.



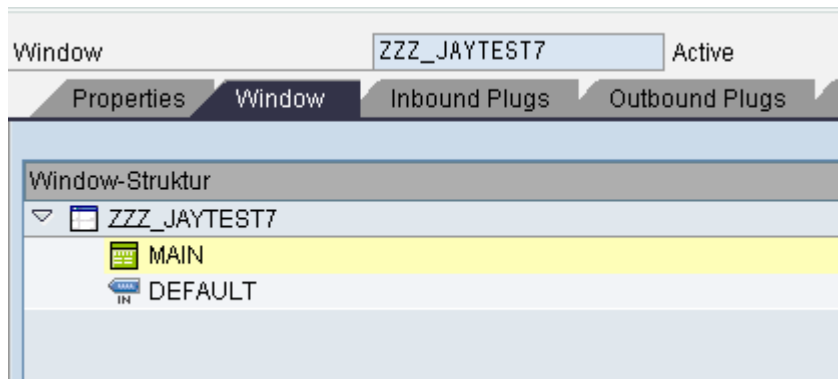
In layout, create ViewContainerUIElement by using Insert Element on right clicking the ROOTUIELEMENTCONTIANER.



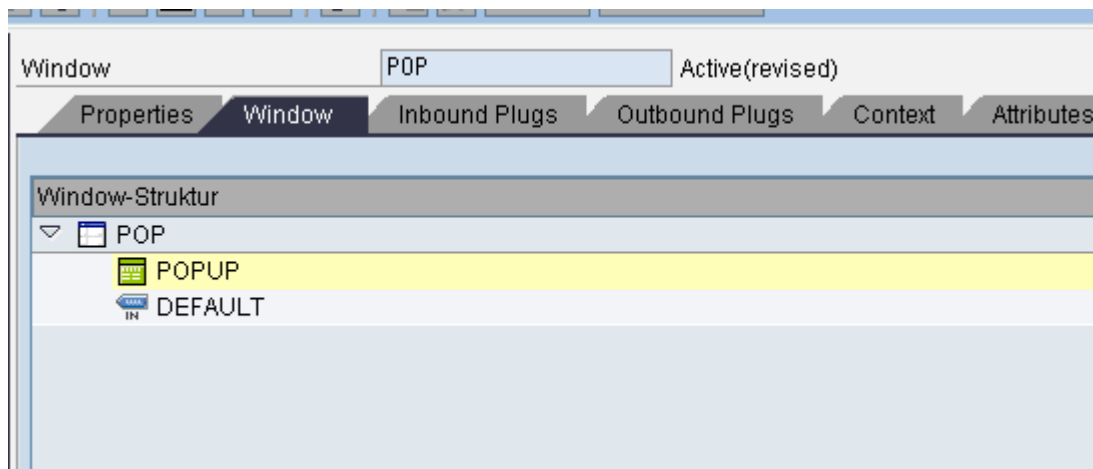
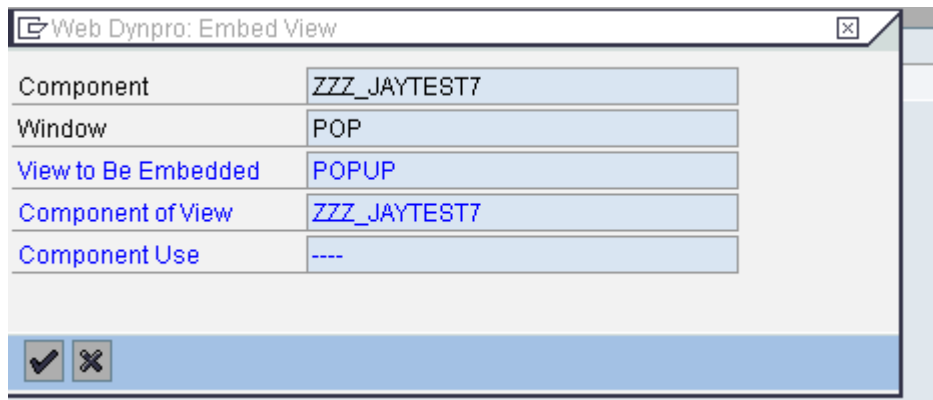
Embedding View

Create two windows in addition to the default (ZZZ_JAYTEST7) window.

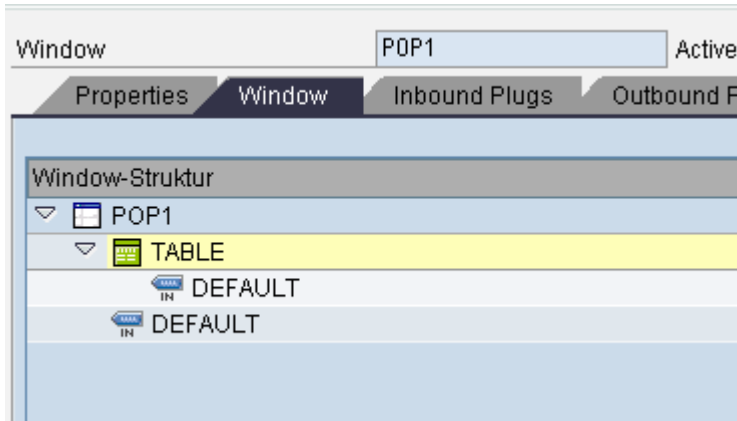
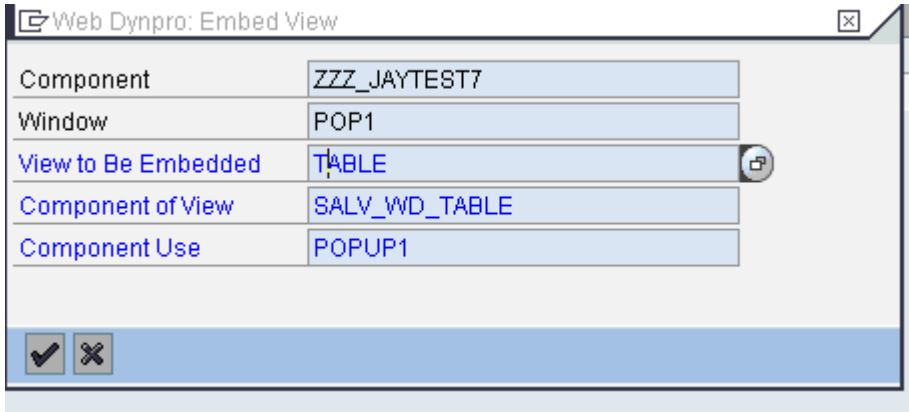
In the default window (ZZZ_JAYTEST7), embed Main view as below.



In the window POP, embed table as below.

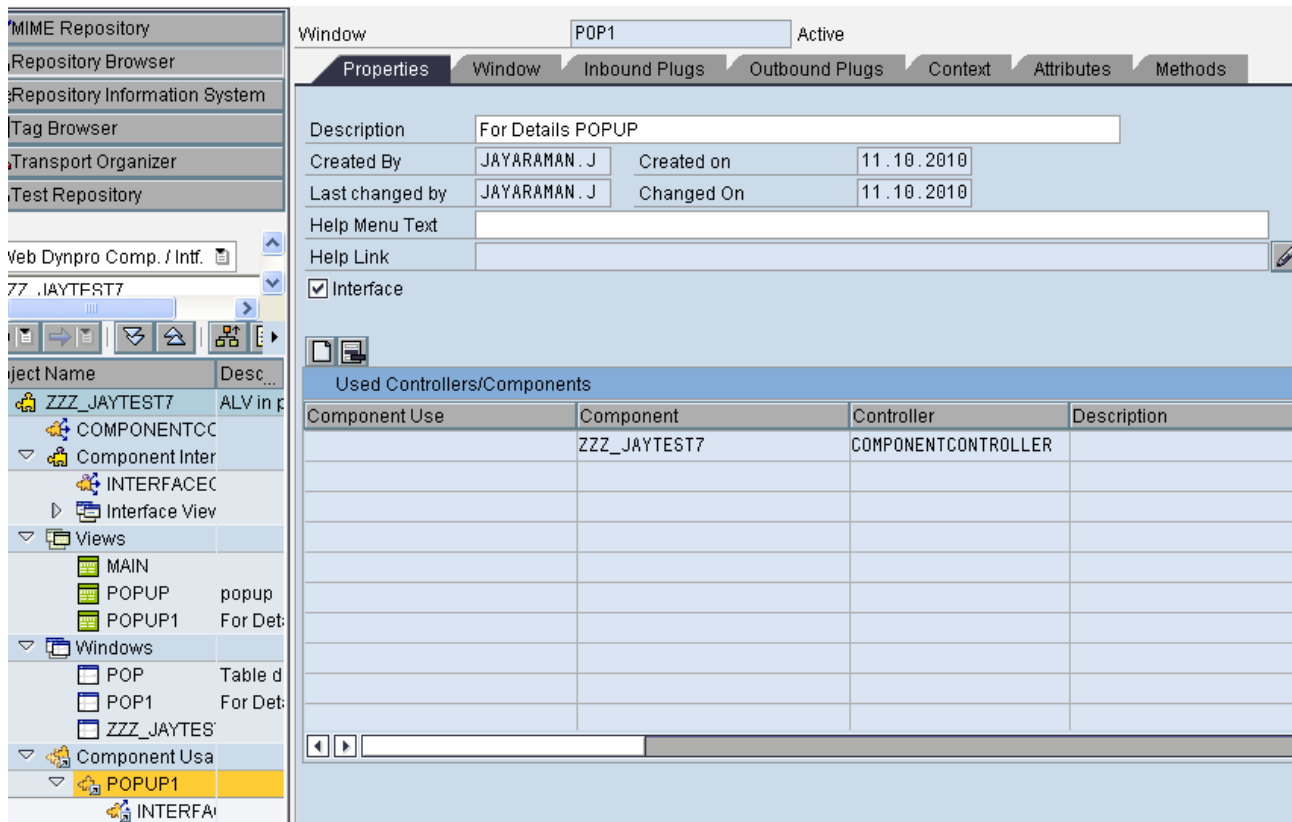


In the window POP1, embed the ALV as below.



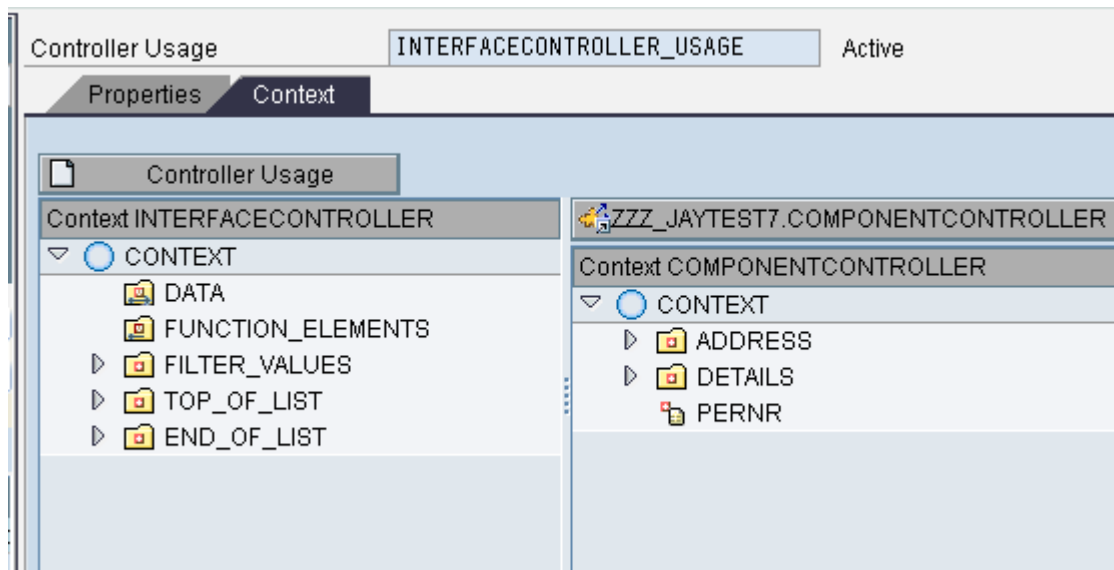
Component Usages

Properties of Component usage should be as below.



Right click the component Usage (here with name POPUP1) 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.



Creating Web Dynpro Application

Create Web Dynpro Application by right clicking the Webdynpro (ZZZ_JAYTEST7).

Right click the Web Dynpro component and activate.

Code

```

METHOD onactiononclick .
* To get the value from input field PERNR
  DATA lo_el_context TYPE REF TO if_wd_context_element.
  DATA ls_context TYPE wd_this->element_context.
  DATA lv_pernr LIKE ls_context-pernr.
* get element via lead selection
  lo_el_context = wd_context->get_element( ).
* get single attribute
  lo_el_context->get_attribute(
    EXPORTING
      name = `PERNR`
    IMPORTING
      value = lv_pernr ).

  DATA lo_nd_address TYPE REF TO if_wd_context_node.
* Navigate from <CONTEXT> to <ADDRESS> via lead selection
  lo_nd_address = wd_context->get_child_node( name =
    wd_this->wdctx_address ).

  TYPES : BEGIN OF ty_0006,
          pernr TYPE pa0006-pernr,
          subty TYPE pa0006-subty,
          stras TYPE pa0006-stras,
          ort01 TYPE pa0006-ort01,
          pstlz TYPE pa0006-pstlz,
          END OF ty_0006.
  DATA t_0006 TYPE STANDARD TABLE OF ty_0006.

  SELECT pernr subty stras ort01 pstlz FROM pa0006 INTO TABLE t_0006
    WHERE pernr = lv_pernr.

* Binding table to the node
  lo_nd_address->bind_table( t_0006 ).

* Generate POPUP for Address Display
  DATA lo_window_manager TYPE REF TO if_wd_window_manager.
  DATA lo_api_component TYPE REF TO if_wd_component.
  DATA lo_window TYPE REF TO if_wd_window.

  lo_api_component = wd_comp_controller->wd_get_api( ).
  lo_window_manager = lo_api_component->get_window_manager( ).
  lo_window = lo_window_manager->create_window(
    window_name = 'POP'
    title = 'Address Details'
    close_in_any_case = abap_true
    close_button = abap_true ).

  lo_window->open( ).
ENDMETHOD.

```

```

method ONACTIONONDETAILSCLICK .

* Read the input from the attribute Pernr
DATA lo_el_context TYPE REF TO if_wd_context_element.
DATA ls_context TYPE wd_this->element_context.
DATA lv_pernr LIKE ls_context-pernr.
* get element via lead selection
lo_el_context = wd_context->get_element( ).

* get single attribute
lo_el_context->get_attribute(
  EXPORTING
    name = `PERNR`
  IMPORTING
    value = lv_pernr ).

DATA lo_nd_details TYPE REF TO if_wd_context_node.
* navigate from <CONTEXT> to <DETAILS> via lead selection
lo_nd_details = wd_context->get_child_node( name =
wd_this->wdctx_details ).

DATA lo_window_manager TYPE REF TO if_wd_window_manager.
DATA lo_api_component TYPE REF TO if_wd_component.
DATA lo_window TYPE REF TO if_wd_window.

lo_api_component = wd_comp_controller->wd_get_api( ).
lo_window_manager = lo_api_component->get_window_manager( ).
lo_window = lo_window_manager->create_window_for_cmp_usage(
  interface_view_name = 'TABLE'
  component_usage_name = 'POPUP1'
  title = 'Employee Details'
  close_in_any_case = abap_true
  message_display_mode =
if_wd_window=>co_msg_display_mode_selected
).
lo_window->open( ).

types : begin of ty_0002,
  pernr type pa0002-pernr,
  begda type pa0002-begda,
  endda type pa0002-endda,
  nachn type pa0002-nachn,
  vorna type pa0002-vorna,
  midnm type pa0002-midnm,
  cname type pa0002-cname,
  end of ty_0002.
data t_0002 type standard table of ty_0002.

select pernr begda endda nachn
  vorna midnm cname from pa0002 into table t_0002
  where pernr = lv_pernr.

lo_nd_details->bind_table( t_0002 ).

endmethod.

```

Output

Personnel Number:

When Address button is clicked, popup will appear as below.

Address Details

Employee	Subtype	Street	City	Code
90000028	1	12	123456	1234

Row 1 of 1

When Details button is clicked, Popup will appear as below.

Employee Details

View [Standard View] Filter Settings

PersNo	End Date	Start Date	Last name	First name	Full Name	Middle name
90000028	30.06.2009	16.04.2009	Training	Jaytest1		
90000028	31.12.9999	01.07.2009	Training	Jaytest2		

Row 1 of 2

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.