



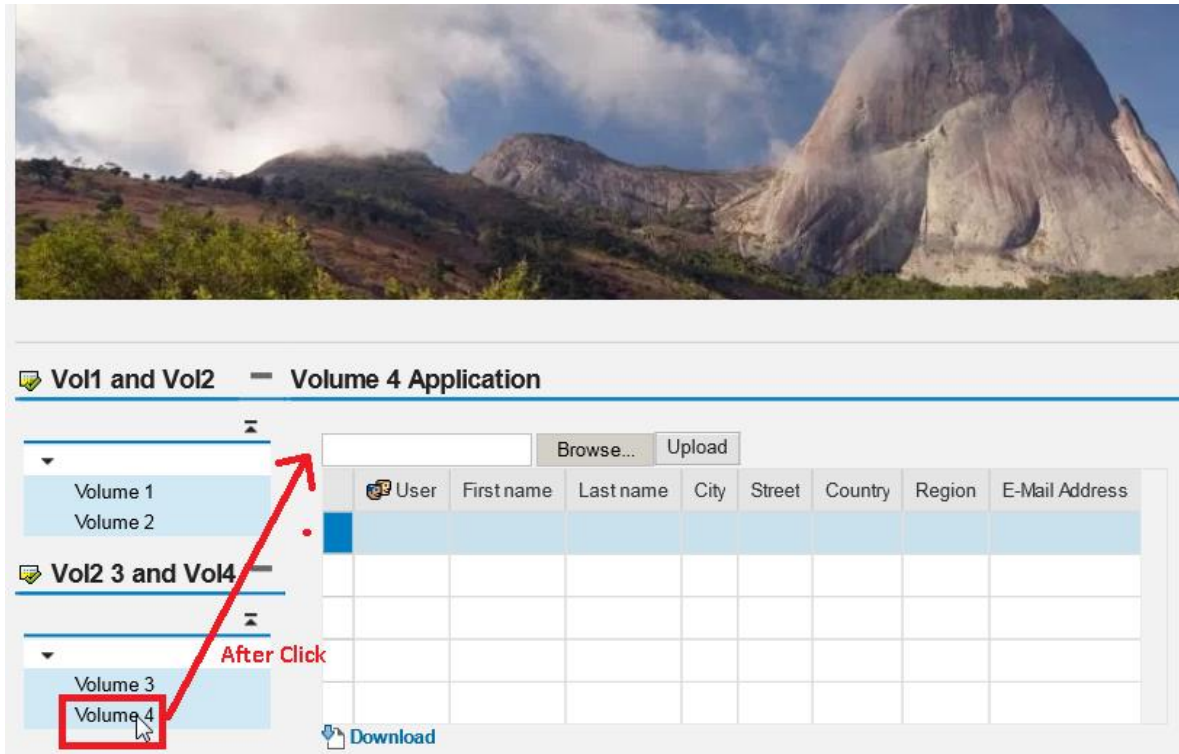
Version 1.0
Date 13.07.2016
Status Completed
Number 5
Initial version by Igor Muntoreanu

Exercise Review:

In this exercise we are going to create a WebDynpro Component with the following elements:

- a) Group
- b) Tray
- c) Trees
- d) IFRAME
- e) IMAGE

And it will have the function to load our last applications. Example:



Practical Guide:

- 1) Create a new WD Component:

Web Dynpro: Component / Create Interface

Name: ZWDC_IMUNTORE_TREE

Description: WD Tree Application

Type: ☒ Web Dynpro Component
☐ Web Dynpro Component Interface

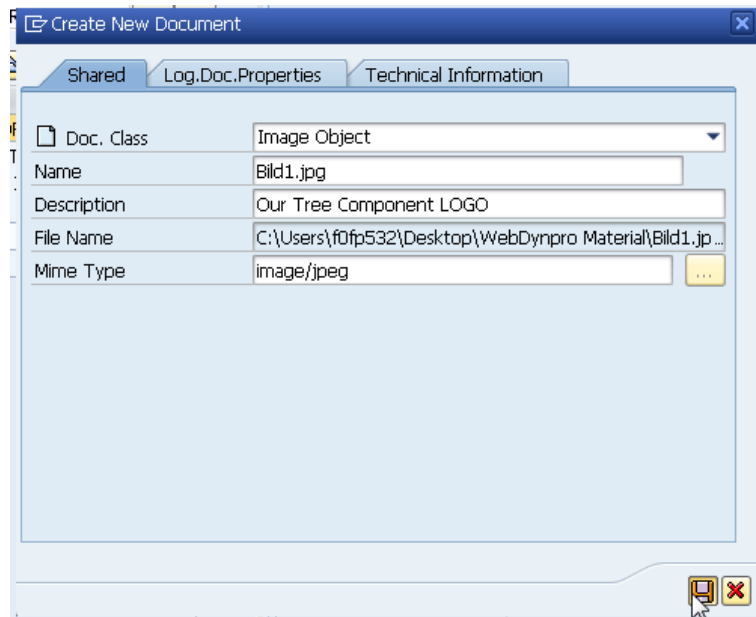
Window Name: ZWDC_IMUNTORE_TREE

View Name: MAIN

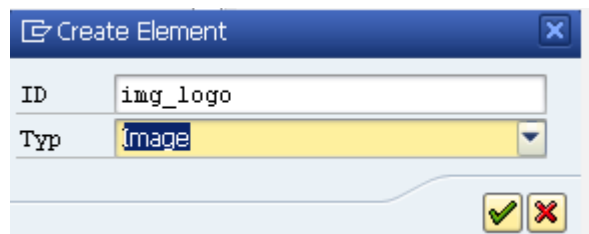
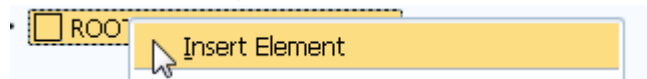
- 2) Save and Activate.
- 3) Upload the following image as a MIME Object:



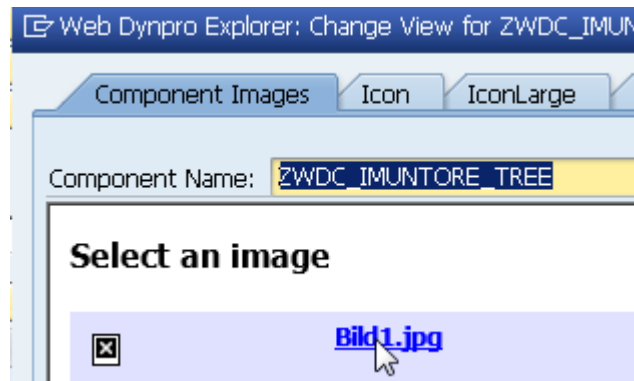
Object Name	D...	Component Use	Component	Description of Co
ZWDC_IMUNTOR				
COMPONENT		Create	Web Dynpro Component (Interface)	
Component I		Change	View	
Views		Display	Window	
MAIN		Create/Change Configuration	Custom Controller	
Windows		Check	Service Call	
		Activate	Mime Object	Create
		Copy...	Web Dynpro Application	Import
		Remove		



- 4) Go to the MAIN view
Insert the new Image:



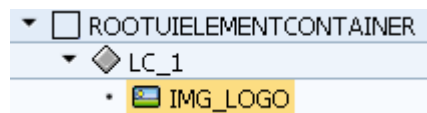
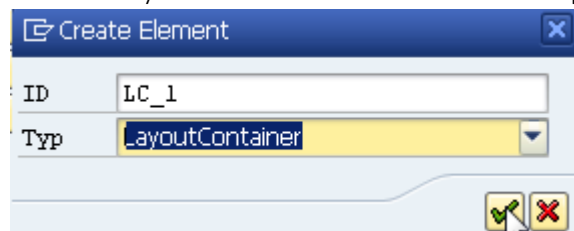
At the source property, select our uploaded image:



You View will look like this:



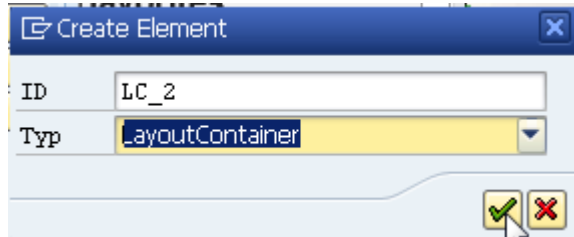
Create a Layout Container as shown below and put the IMAGE inside the new container:



Set the property of the layout container as demonstrated below:

Property	Value	B
Properties (LayoutContainer)		
ID	LC_1	
Layout	FlowLayout	
contextMenuBehaviour	Inherit	
contextMenuId		
enabled	<input checked="" type="checkbox"/>	
height		
tooltip		
visible	Visible	
width		
Layout (FlowLayout)		
wrapping	<input checked="" type="checkbox"/>	
Layout Data (FlowData)		
cellDesign	padless	
vGutter	None	

Create another Layout Container element:

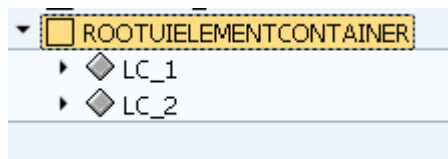


The 'Create Element' dialog box is shown. The 'ID' field contains 'LC_2' and the 'Type' dropdown is set to 'LayoutContainer'. At the bottom right, there are two buttons: a green checkmark and a red X.

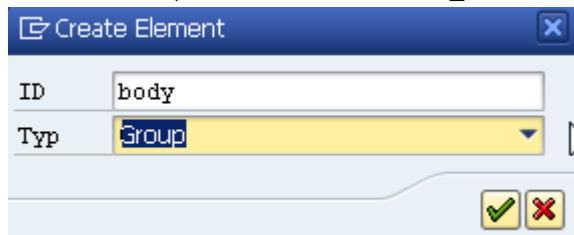
Set the Layout property as Matrix:

Property	Value	Bind
Properties (LayoutContainer)		
ID	LC_2	
Layout	MatrixLayout	

Result:



Create a Group Element inside the LC_2:

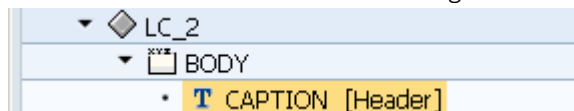











The 'Create Element' dialog box is shown. The 'ID' field contains 'body' and the 'Type' dropdown is set to 'Group'. At the bottom right, there are two buttons: a green checkmark and a red X.

Set the layout property as FlowLayout:

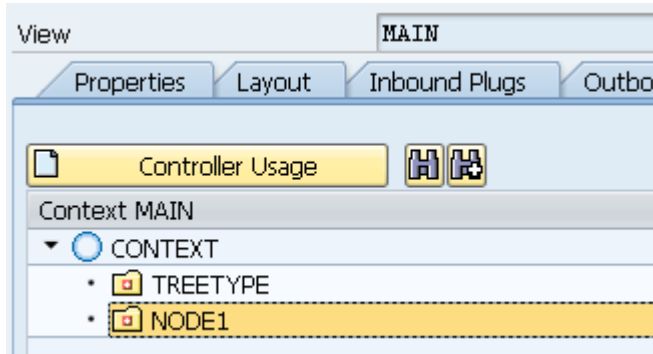
Property	Value	Bind
Properties (Group)		
ID	BODY	
Layout	FlowLayout	

Put some text in the automatic Label generated:

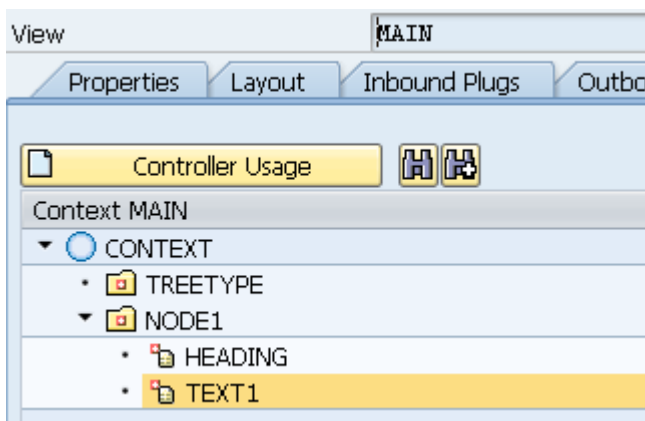


Property	Value	Bi
<u>Properties (Caption)</u>		
ID	CAPTION	
contextMenuBehaviour	Inherit	
contextMenuId		
enabled	<input checked="" type="checkbox"/>	
imageFirst	<input checked="" type="checkbox"/>	
imageIsDecorative	<input type="checkbox"/>	
imageSource		
isDragHandle	<input type="checkbox"/>	
text	Our First tree Application	
textDirection	Inherit	
tooltip		
visible	Visible	

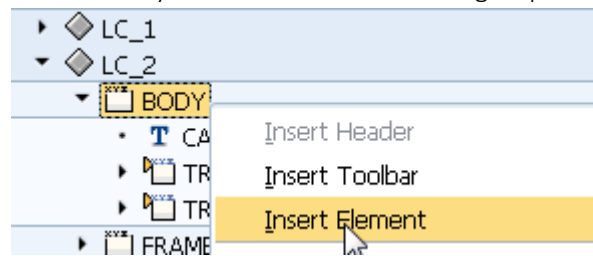
Go to the Context tab and create two nodes with the standard specifications:



Create 2 Attributes inside NODE1, both STRING type:



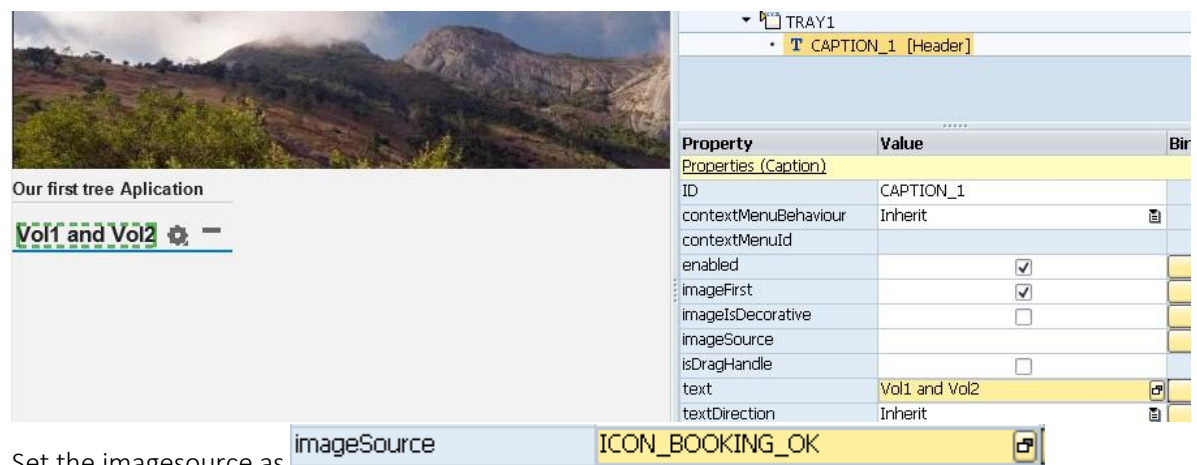
Create a Tray element inside the BODY group:



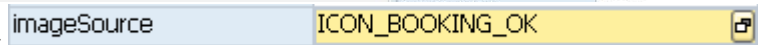
Set the layout property as below:

Property	Value	
Properties (Tray)		
ID	TRAY1	
Layout	FlowLayout	

In the auto generated label, type the text:



Set the imageSource as



Result:



Create the TRAY2 element also inside the BODY group:

The 'Create Element' dialog box shows the ID set to 'tray2' and the Type set to 'Tray'. Below the dialog, a table lists the properties for the 'Tray' element.

Property	Value	Binding
Properties (Tray)		
ID	TRAY2	
Layout	FlowLayout	

In the auto generated label, type the text:

The application interface shows a tree view with the following structure:

- Our first tree Application
 - Vol1 and Vol2
 - Vol2 3 and Vol4

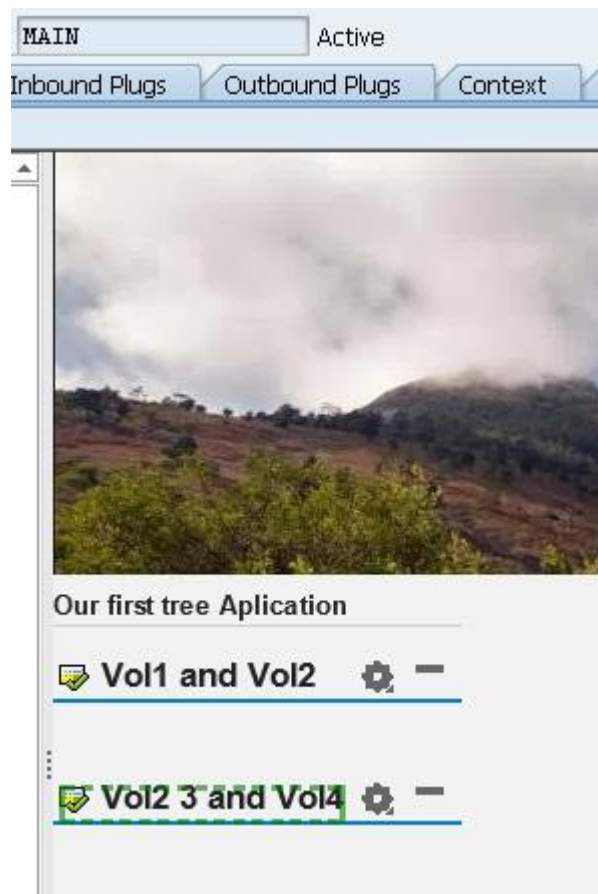
The properties table for 'CAPTION_2 [Header]' is shown on the right:

Property	Value	Binding
Properties (Caption)		
ID	CAPTION_2	
contextMenuBehaviour	Inherit	
contextMenuId		
enabled	<input checked="" type="checkbox"/>	
imageFirst	<input checked="" type="checkbox"/>	
imageIsDecorative	<input type="checkbox"/>	
imageSource	ICON_BOOKING_OK	
isDragHandle	<input type="checkbox"/>	
text	Vol2 3 and Vol4	
textDirection	Inherit	
tooltip		
visible	Visible	

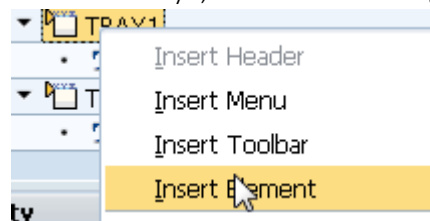
Set the imagesource as

Property	Value	Binding
imageSource	ICON_BOOKING_OK	

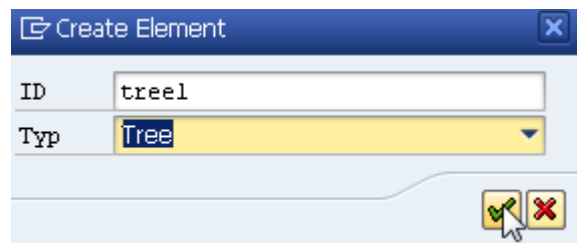
Result:



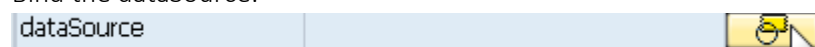
Inside the Tray1, insert the following element:

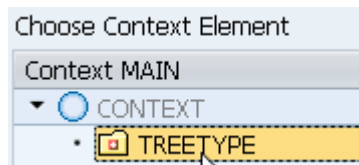


Create the Tree1 as shown below:

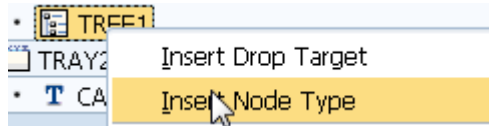


Bind the dataSource:

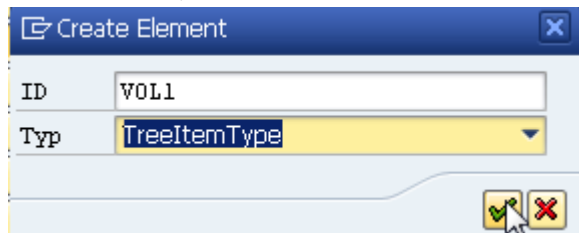




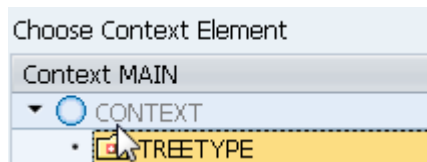
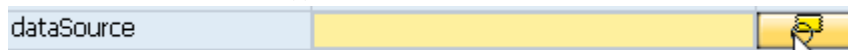
After the TREE1 created, insert the Node Type inside the TREE element.



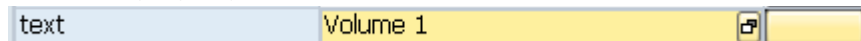
Name as VOL1, as shown below:



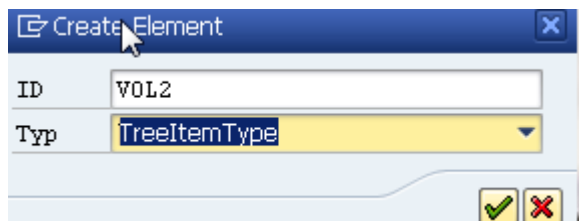
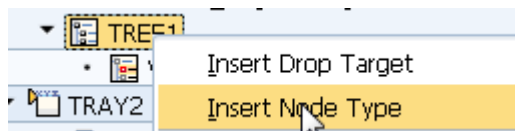
Bind also with the Treetype

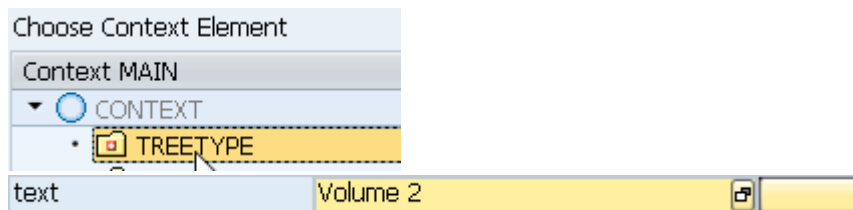


In the text property, set Volume 1.

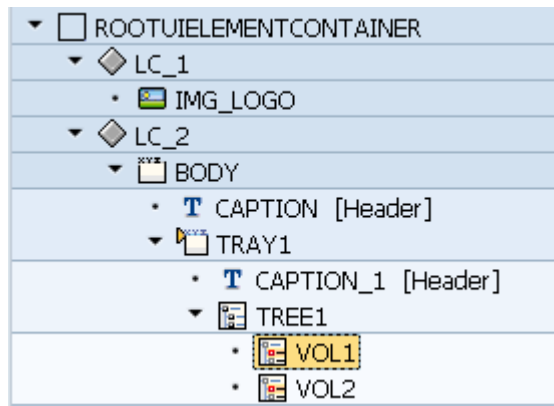


Do this step again for TreeItemType VOL2, as shown below:

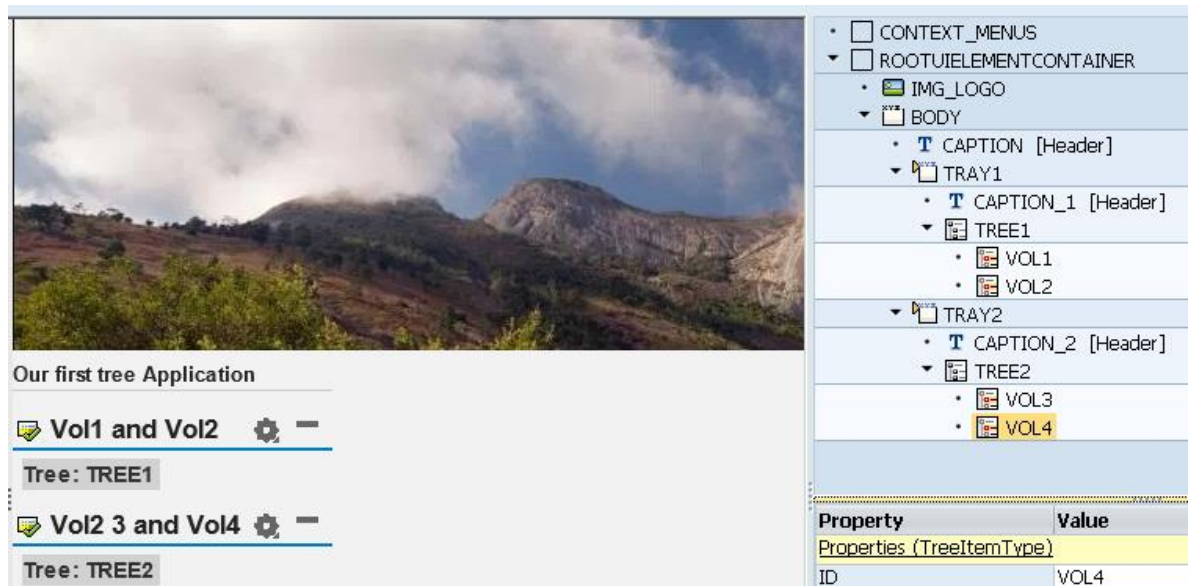




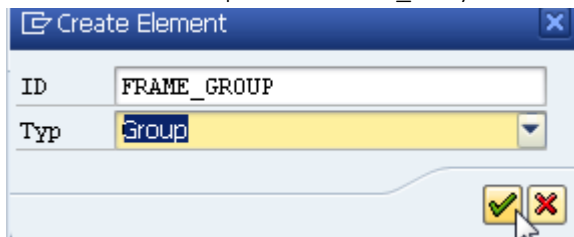
Until now, your layout must look like this:



Do exactly the same steps above to the TRAY2, the final result will look like this:



Insert a new Group inside the LC_2 Layout Container:

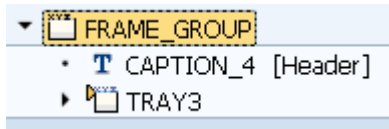


The 'Create Element' dialog box is shown. The 'ID' field contains 'FRAME_GROUP' and the 'Type' dropdown is set to 'Group'. There are 'OK' and 'Cancel' buttons at the bottom right.

FRAME_GROUP properties:

Property	Value
<u>Properties (Group)</u>	
ID	FRAME_GROUP
Layout	FormLayout
Layout Data	MatrixData

Inside the FRAME_GROUP, create the TRAY3:

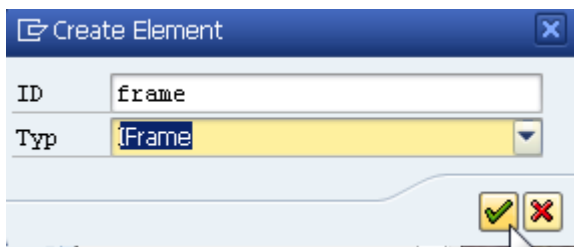


The 'FRAME_GROUP' tree view is shown. It contains a 'CAPTION_4 [Header]' and a 'TRAY3' sub-element.

TRAY3 properties:

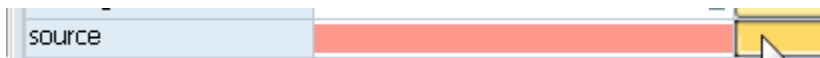
Property	Value	Bind
<u>Properties (Tray)</u>		
ID	TRAY3	
Layout	FormLayout	

Insert a new element inside the TRAY3:

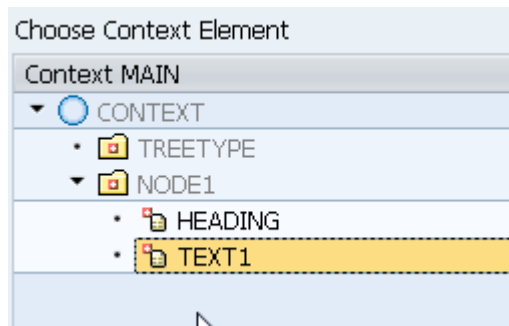


The 'Create Element' dialog box is shown. The 'ID' field contains 'frame' and the 'Type' dropdown is set to 'Frame'. There are 'OK' and 'Cancel' buttons at the bottom right.

Bind it with the TEXT1 attribute:



A binding bar is shown with the text 'source' on the left and a red bar on the right, indicating a binding to a 'TEXT1' attribute.

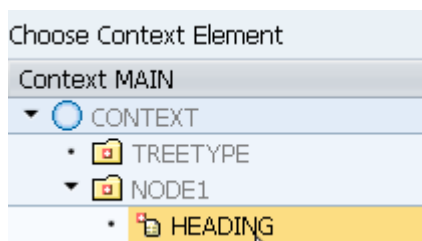


In the auto generated Caption for the Tray3, bind the text as shown below:

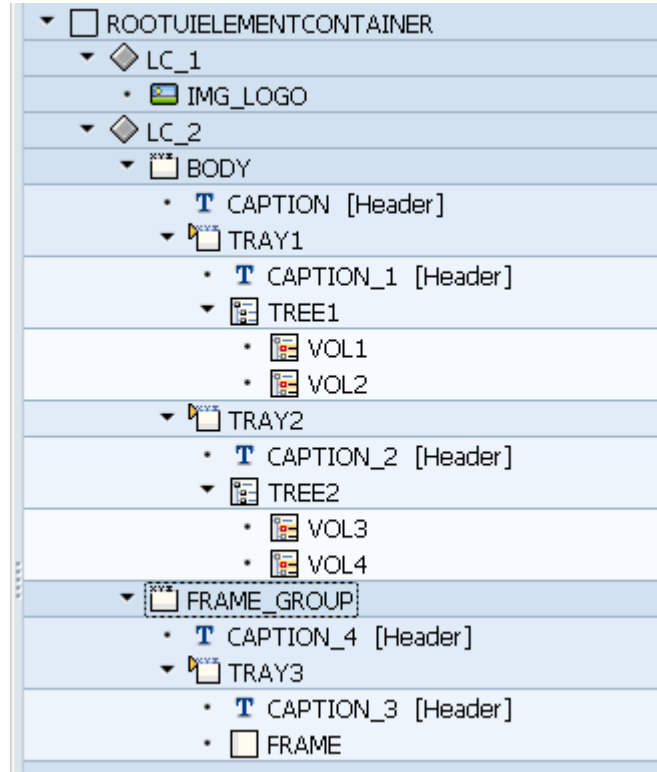
TRAY3

- CAPTION_3 [Header]
- FRAME

Property	Value	Binding
<u>Properties (Caption)</u>		
ID	CAPTION_3	
contextMenuBehaviour	Inherit	
contextMenuId		
enabled	<input checked="" type="checkbox"/>	
imageFirst	<input checked="" type="checkbox"/>	
imageIsDecorative	<input type="checkbox"/>	
imageSource		
sDragHandle	<input type="checkbox"/>	
text		



Before we create the actions, let's do some layout adjustments.
First of all, your ROOT must look like this:

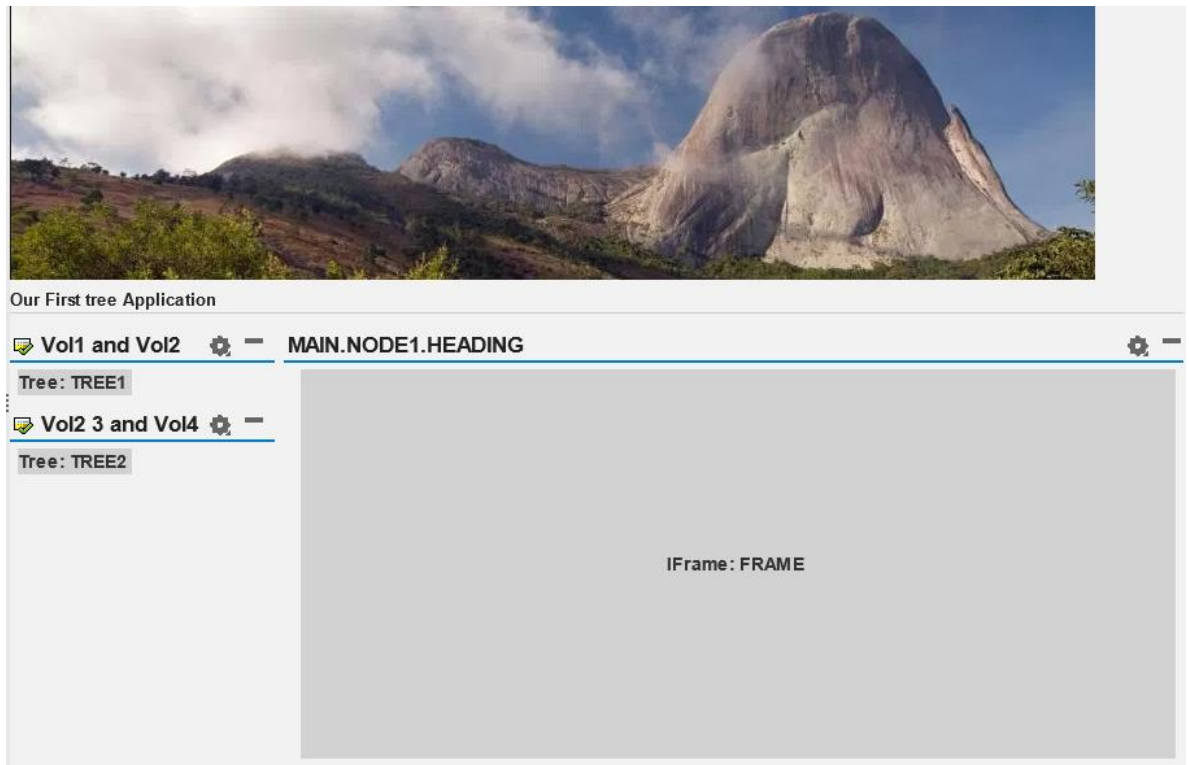


Change the following properties of the following elements:

▶	FRAME_GROUP
height	500px
width	700px

▶	TRAY3
height	500px
width	700px

5) Your layout should look like this now:

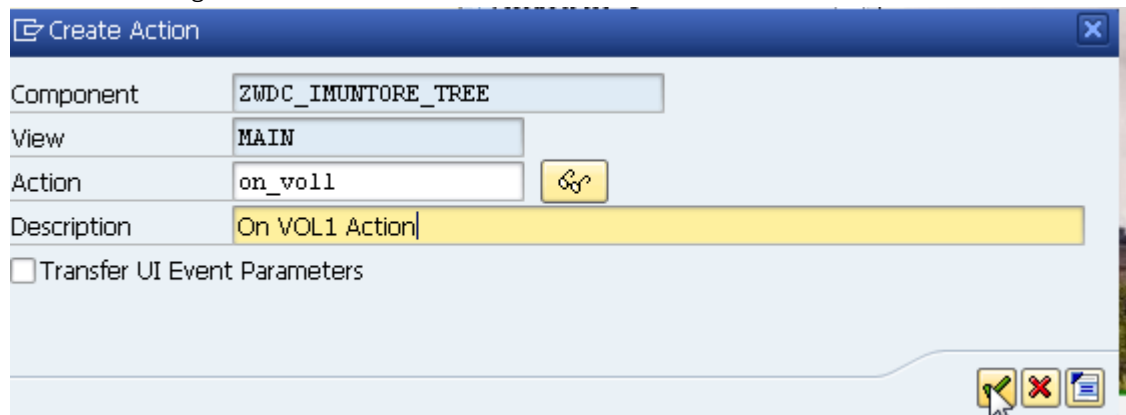


6) Creating the TreeItemTypes Actions

We will now create the actions for the following elements:



Do the following for the all 4 elements:



Your actions tab must look like this:

View			MAIN	Active
Properties Layout Inbound Plugs Outbound Plugs Context Attributes Actions				
Action	Action Type	Description		
ON_VOL1	Standard	On VOL1 Action		
ON_VOL2	Standard	On VOL2 Action		
ON_VOL3	Standard	On VOL3 Action		
ON_VOL4	Standard	On VOL4 Action		

7) Time to Code!

a. ON_VOL1 action:

Copy and past the code below. Remember that you can use the Wizard to do it:

The image shows two screenshots from the SAP ABAP development environment. The left screenshot displays the 'Event Handler' for the 'ONACTIONON_VOL1' event, containing the following ABAP code:

```

1 METHOD onactionon_vol1 .
2
3 DATA lo_nd_model TYPE REF TO if_wd_context_node.
4 DATA lo_el_model TYPE REF TO if_wd_context_element.
5
6 *** Read the NODE1 from the CONTEXT
7 lo_nd_model = wd_context->get_child_node( name = wd_this->wdctx_node1 ).
8 lo_el_model = lo_nd_model->get_element( ).
9
10 *** SET TEXT1 value
11 DATA lv_text1 TYPE wd_this->element_model-text1.
12
13 lo_nd_model = wd_context->get_child_node( name = wd_this->wdctx_node1 ).
14 lo_el_model = lo_nd_model->get_element( ).
15 lv_text1 = 'http://q4dshuys77.nl.s1-s1.com:8000/sap/bc/webdynpro/sap/zwdc_user_display_copy'.
16 lo_el_model->set_attribute(
17   name = 'TEXT1'
18   value = lv_text1 ).
19
20 *** SET HEADING value
21 DATA lv_heading TYPE wd_this->element_model-heading.
22
23 lo_nd_model = wd_context->get_child_node( name = wd_this->wdctx_node1 ).
24 lo_el_model = lo_nd_model->get_element( ).
25 lv_heading = 'Volume 1 Application'.
26 lo_el_model->set_attribute(
27   name = 'HEADING'
28   value = lv_heading ).
29
30 ENDMETHOD.

```

The right screenshot shows the 'Context' configuration for the 'E-CONTEXT_VIEWFR' view. It displays the 'Operation on Context' tab with the following settings:

- Node/Attribute: E-CONTEXT_VIEWFR
- Operation on Context: Read
- Choose Context Element: Context MAIN
- CONTEXT: TREE TYPE
- NODE1: HEADING
- TEXT1

Red arrows indicate the mapping between the code and the context configuration. Specifically, the 'TEXT1' attribute in the code is linked to the 'TEXT1' attribute in the context configuration, and the 'HEADING' attribute is linked to the 'HEADING' attribute in the context configuration.

METHOD onactionon_vol1 .

```

DATA lo_nd_model TYPE REF TO if_wd_context_node.
DATA lo_el_model TYPE REF TO if_wd_context_element.

*** Read the NODE1 from the CONTEXT
lo_nd_model = wd_context->get_child_node( name = wd_this-
>wdctx_node1 ).
lo_el_model = lo_nd_model->get_element( ).

*** SET TEXT1 value
DATA lv_text1 TYPE wd_this->element_model-text1.

lo_nd_model = wd_context->get_child_node( name = wd_this-
>wdctx_node1 ).

```

```

lo_el_node1 = lo_nd_node1->get_element( ).
lv_text1 = 'http://q4de3gsy577.m1.sl-
si.com:8000/sap/bc/webdynpro/sap/zwdc_user_display_copy'.
lo_el_node1->set_attribute(
    name = `TEXT1`
    value = lv_text1 ).

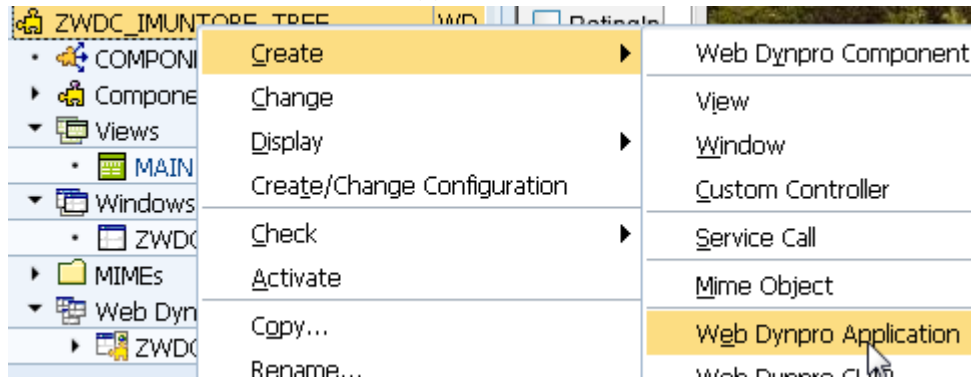
DATA lv_heading TYPE wd_this->element_node1-heading.
*** SET HEADING value
lo_nd_node1 = wd_context->get_child_node( name = wd_this-
>wdctx_node1 ).
lo_el_node1 = lo_nd_node1->get_element( ).
lv_heading = 'Volume 1 Application'.
lo_el_node1->set_attribute(
    name = `HEADING`
    value = lv_heading ).

ENDMETHOD.

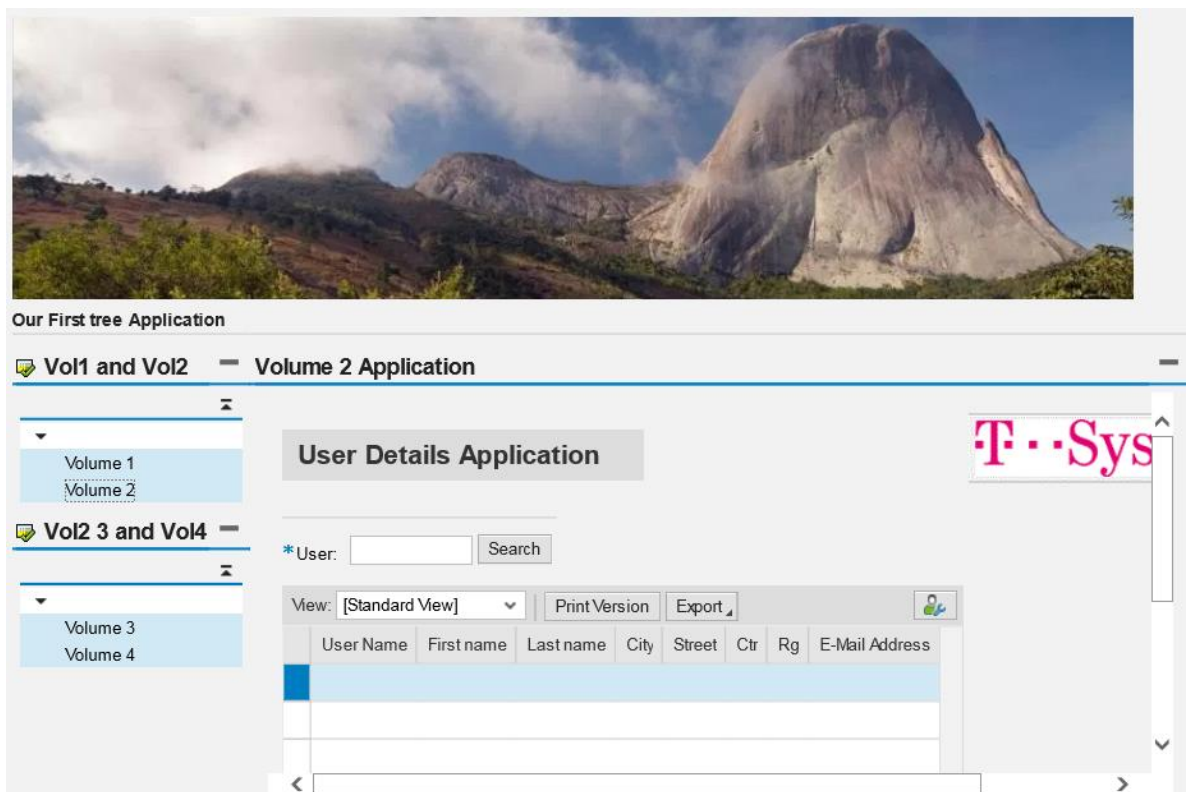
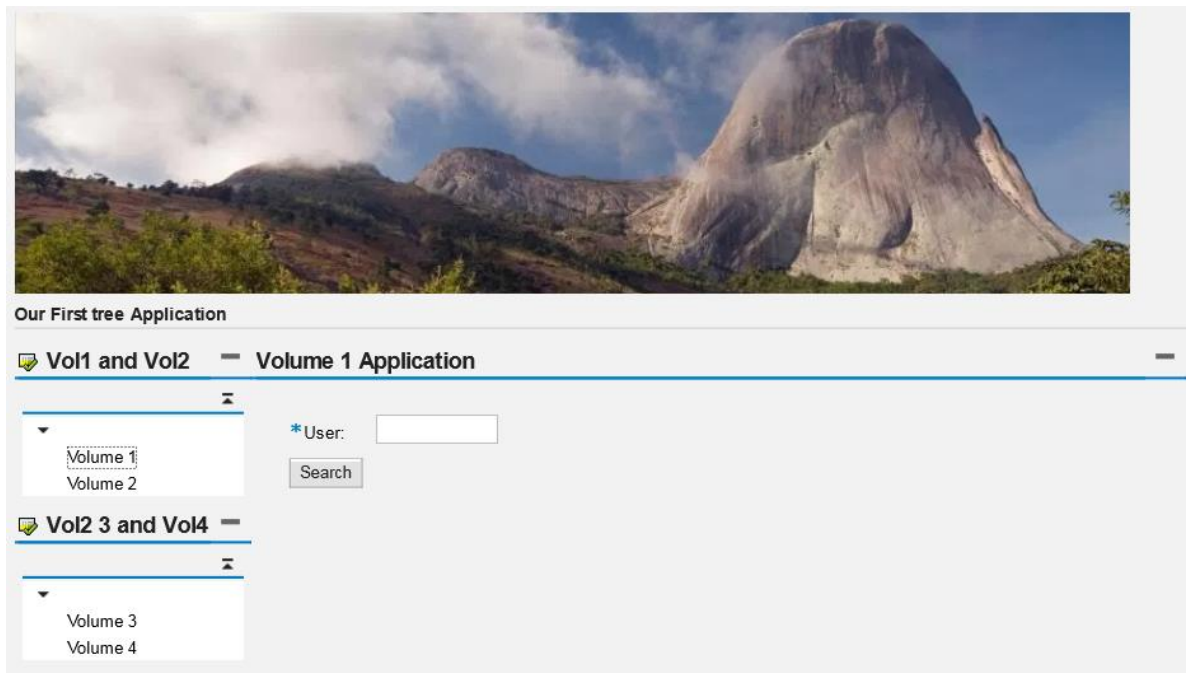
```

- b. ON_VOL2 action:
Just do the same but with lv_text1 = http://q4de3gsy577.m1.sl-si.com:8000/sap/bc/webdynpro/sap/zwdc_user_display_imuntor
- c. ON_VOL3 action:
Just do the same but with lv_text1 = http://q4de3gsy577.m1.sl-si.com:8000/sap/bc/webdynpro/sap/zwdc_imuntor_select_optio
- d. ON_VOL4 action:
Just do the same but with lv_text1 = http://q4de3gsy577.m1.sl-si.com:8000/sap/bc/webdynpro/sap/zwdc_imuntore_excel

8) Create the WDA and execute the application:



9) Final Result:





Our First tree Application

Vol1 and Vol2 Volume 3 Application

- Volume 1
 - Volume 2
- Vol2 3 and Vol4
- Volume 3
 - Volume 4

Created on: To

Country Key:

Search

Search Results

User	First na...	Last na...	Street	City	Region	E-Mail Addre...	Co...	Crea...



Our First tree Application

Vol1 and Vol2 Volume 4 Application

- Volume 1
 - Volume 2
- Vol2 3 and Vol4
- Volume 3
 - Volume 4

Browse... Upload

User	First name	Last name	City	Street	Country	Region	E-Mail Address

Download