

Version 1.0
Date 27.06.2016
Status Completed
Number 2
Initial version by Igor Muntoreanu

Exercise Review:

User Details Application

Header

View 2



*User:

Search

View 1

View: [Standard View]

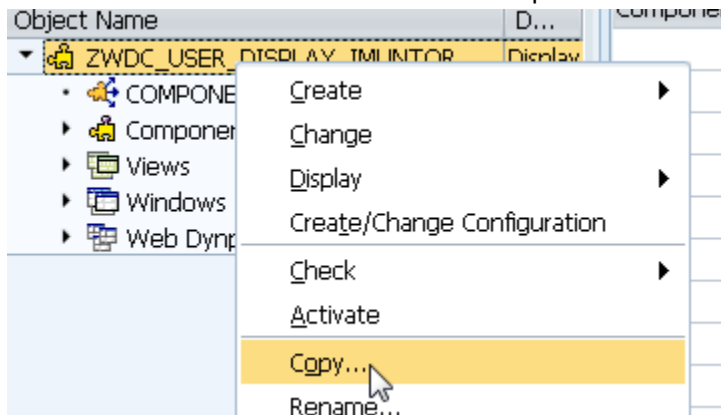
[Print Version](#)

Export ▾

[illegible]

Practical Guide:

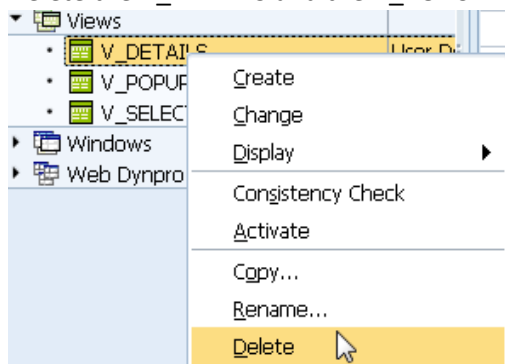
- 1) Save the last finished exercise to a new component:



Just choose a name of you like and save it. Just to have as backup.

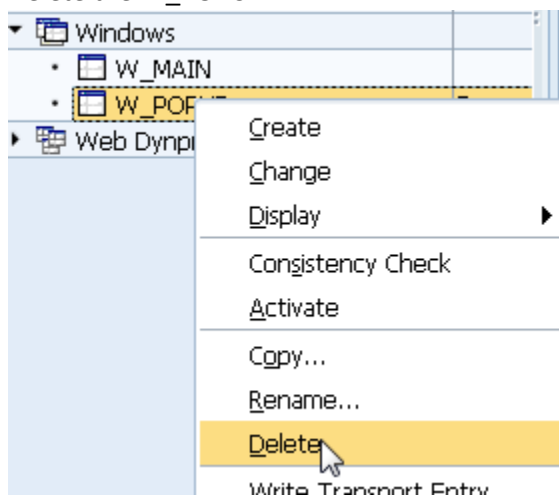
- 2) Deleting useless stuff:

- a. Delete the V_DETAILS and the V_POPUP Views



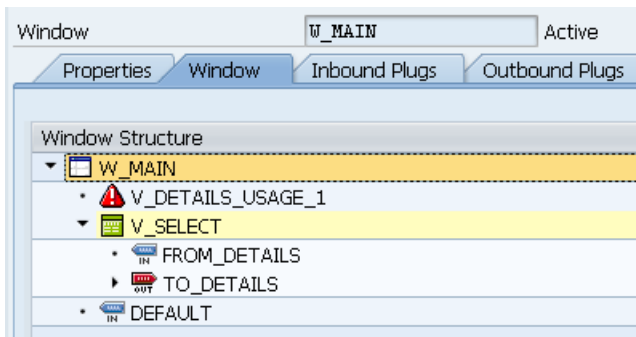
- b. Activate

- c. Delete the W_POPUP

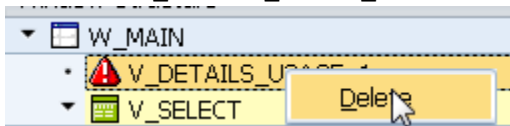


- d. Activate

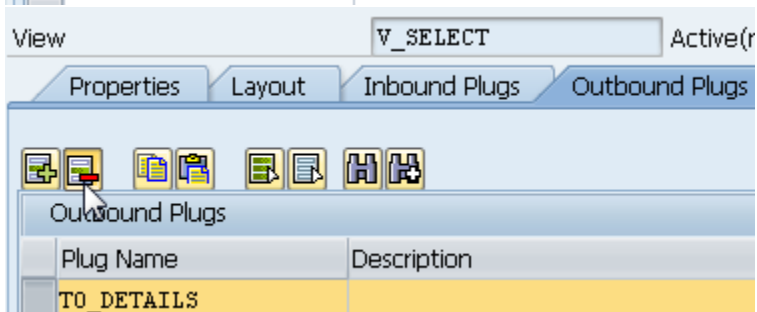
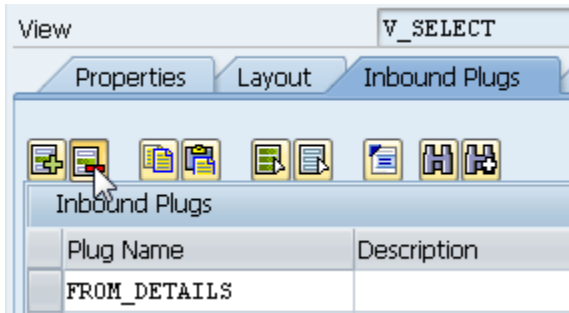
- e. Go the W_MAIN window, it will look like that:



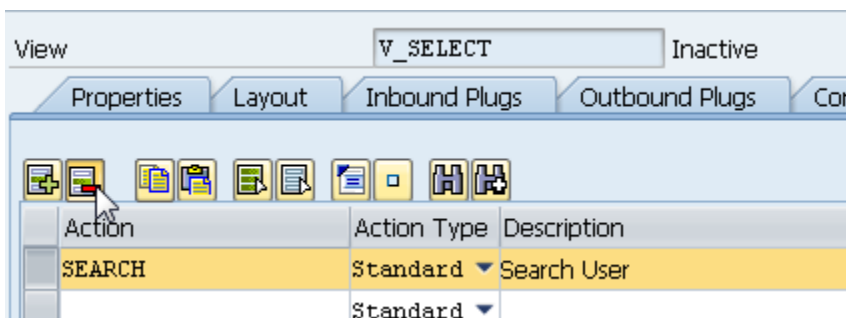
Delete the V_DETAILS_USAGE_1



- f. Go to the V_SELECT view and remove the plugs



- g. Delete the Action



- 3) Create an action for the SEARCH button again:
But this time keep the PLUG empty we will not use plugs in this example.

Component: ZWDC_USER_DISPLAY_IMUNTOR

View: V_SELECT

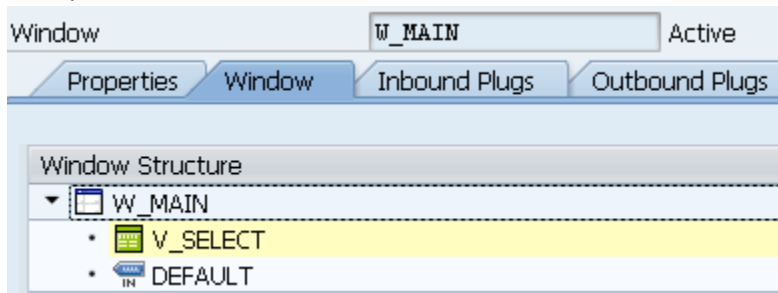
Action: SEARCH

Description: Search User

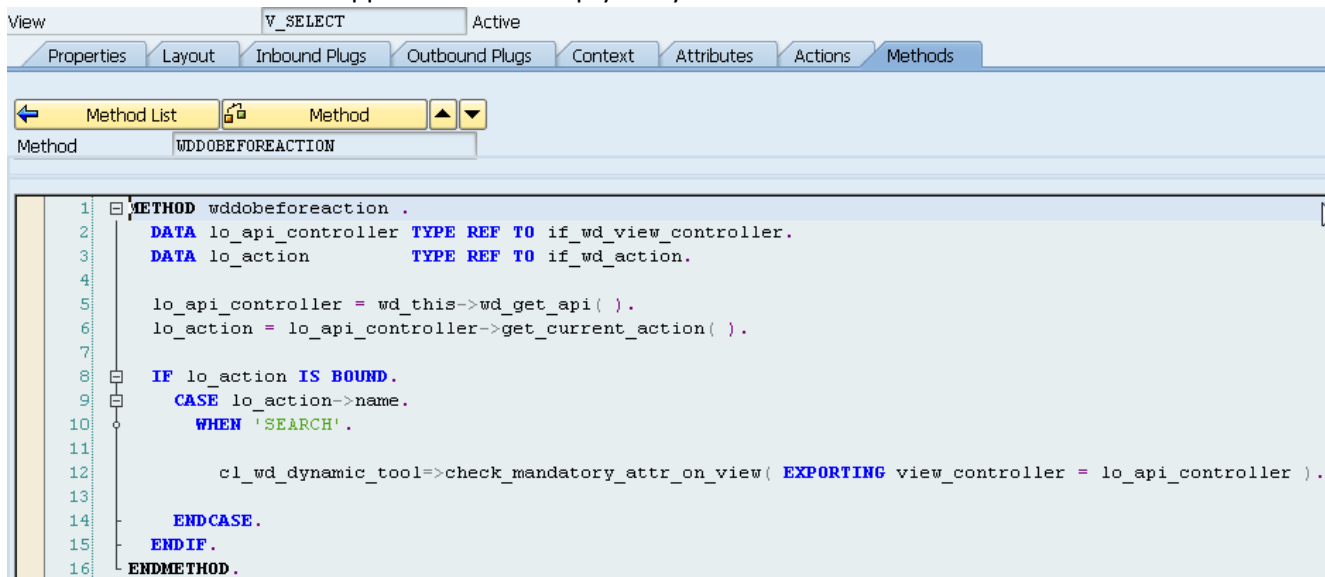
Select an outbound plug or enter an outbound plug for leaving the view by selecting the pushbutton

Outbound Plug:

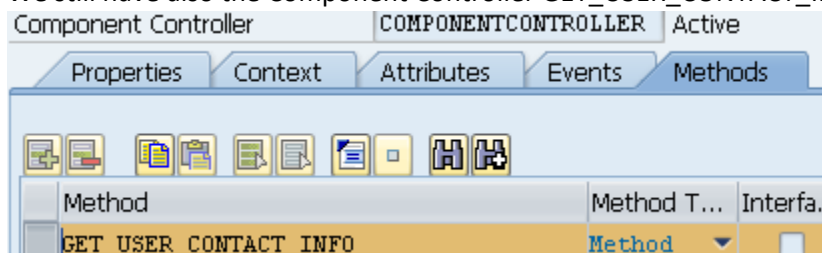
- 4) Make sure
- That your window is like shown below:



- We still have the method mapped in case of empty entry.



- We still have also the Component Controller GET_USER_CONTACT_INFO:



```

DATA: lo_nd_user_contact TYPE REF TO if_wd_context_node,
      lo_el_user_contact TYPE REF TO if_wd_context_element.

DATA: lv_username TYPE REF TO wd_this->element_user_contact-username.

DATA: ls_address TYPE bapiaddr3,
      lt_return  TYPE TABLE OF bapiret2.

DATA: ls_user_contact TYPE wd_this->element_user_contact.

" Read the selected user name into context:
lo_nd_user_contact = wd_context->get_child_node( name = wd_this-
>wdctx_user_contact ).
lo_el_user_contact = lo_nd_user_contact->get_element( ).

lv_username ?= lo_el_user_contact->get_attribute_ref('USERNAME').

" Lookup the user's contact information:
CALL FUNCTION 'BAPI_USER_GET_DETAIL'
EXPORTING
  username = lv_username->*
IMPORTING
  address  = ls_address
TABLES
  return   = lt_return.

" Copy the results to a structure and bind it to the context:

ls_user_contact-username      = lv_username->*.
ls_user_contact-firstname     = ls_address-firstname.
ls_user_contact-lastname      = ls_address-lastname.
ls_user_contact-street        = ls_address-street.
ls_user_contact-city          = ls_address-city.
ls_user_contact-region        = ls_address-region.
ls_user_contact-e_mail        = ls_address-e_mail.

" Bind it to the context
lo_el_user_contact->set_static_attributes( ls_user_contact ).

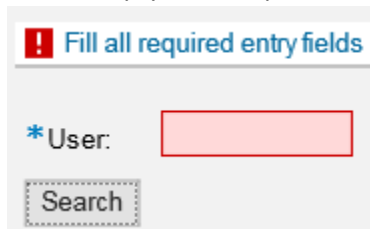
```

5) Taking a look how the programs works now:

- a. When SEARCH is pressed nothing happens:



- b. When empty value is pressed an error message appears:



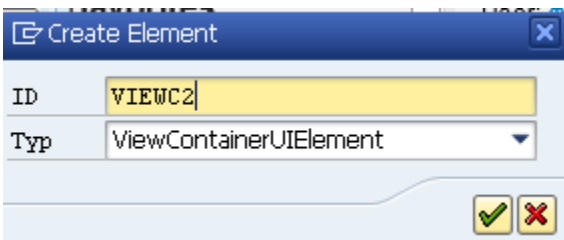
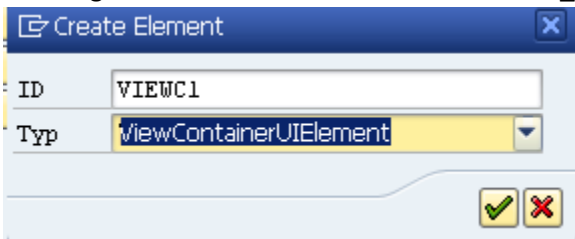
6) Creating the ALV:

- a. The first thing we need to understand is that you can use the same View to create N containers.
In our case, we are going to use the V_SELECT view to create two more containers inside it, the VIEWC1 and the VIEWC2. The VIEWC1, we will use to display an image.

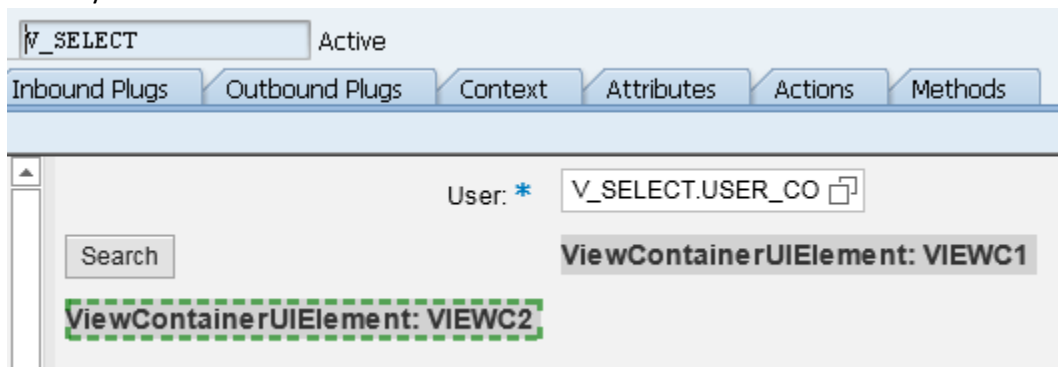


In the VIEWC2, we will use to display the ALV.

- b. Creating the Elements ViewContainers in the V_SELECT view.



Your layout will be like this:



It is messed up but don't about it yet.

7) Webdynpro ALV Component

An important concept in WebDynpro is the Component. WD is OO and Components oriented, that means you can use working tools such as ALV and Select-Options to aggregate your Application, they are reusable solutions. See the image below:

Click 2x in your WD Component:

Object Name
▼ ZWDC_USER_DISPLAY_IMUNTOR

SAP has many ready-to-use components to use. We are going to pick SALV_WD_TABLE for this exercise:

Web Dynpro Component: ZWDC_USER_DISPLAY_IMUNT... Active/revised
Description: Display User Contact Information
Assistance Class:
Created By: IMUNTORE Created On: 29.06.2016
Last Changed By: IMUNTORE Changed On: 29.06.2016
Original Lang.: EN Package: \$TMP
☐ Accessibility Checks Active

Your Name Convention

Used Components Implemented interfaces

Used Web Dynpro Components

Component Use	Component	Description of Component
	ALV	

Repository Info System: Web Dynpro Component Find (112 Hits)

Web Dynpro Component	Description
SALV_WD_TABLE	ALV Component
SALV_WD_TEST_COL_FIELD	
SALV_WD_TEST_CONF_CALLER	
SALV_WD_TEST_CONFIG_API	
SALV_WD_TEST_CONFIG_API2	
SALV_WD_TEST_CONFIG1	
SALV_WD_TEST_CR_2_ALVS	
SALV_WD_TEST_CV_DD_IDX	
SALV_WD_TEST_DATA	
SALV_WD_TEST_DATA_SEL	
SALV_WD_TEST_DATATYPES	
SALV_WD_TEST_DYN1	
SALV_WD_TEST_EXIT_PLUG	
SALV_WD_TEST_EXT_ORG	
SALV_WD_TEST_EXTENDED	
SALV_WD_TEST_F4_ALV_LAYO	
SALV_WD_TEST_FILE_UPLOAD	
SALV_WD_TEST_IMAGE1	
SALV_WD_TEST_MODIF1	
SALV_WD_TEST_NO_DDIC	
SALV_WD_TEST_NOM_PORTAL	
SALV_WD_TEST_P13N_NAVI	
SALV_WD_TEST_P9E_DYN	
SALV_WD_TEST_PFM	
SALV_WD_TEST_REGRESSION	

Web Dynpro Component: ZWDC_USER_DISPLAY_IMUNT... Active/revised
Description: Display User Contact Information
Assistance Class:
Created By: IMUNTORE Created On: 29.06.2016
Last Changed By: IMUNTORE Changed On: 29.06.2016
Original Lang.: EN Package: \$TMP
☒ Accessibility Checks Active

Used Components Implemented interfaces

Used Web Dynpro Components

Component Use	Component	Description of Component
ALV_USER_DETAILS	SALV_WD_TABLE	ALV Component

We will not use Assistance Class in this example.

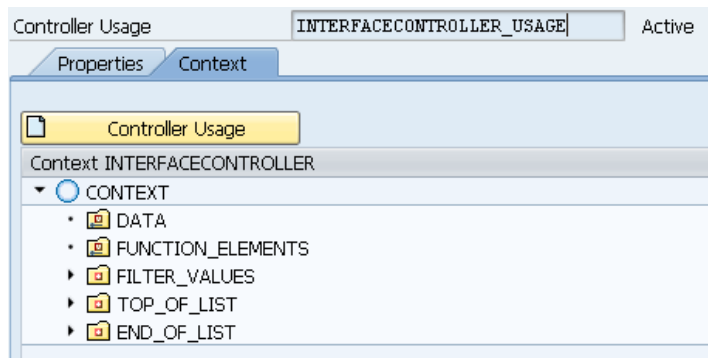
As you can see the "Component Usages" tree now showed up.

Object Name

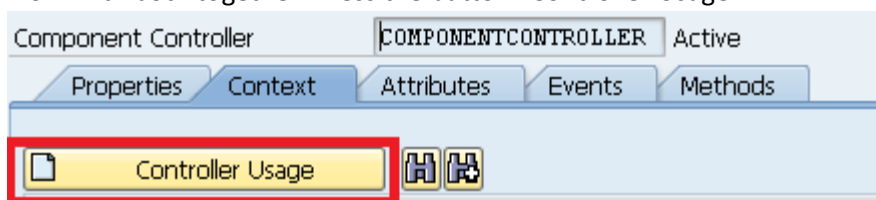
▼ ZWDC_USER_DISPLAY_IMUNTOR

- COMPONENTCONTROLLER
- Component Interface
- Views
- Windows
- Component Usages**
 - ALV_USER_DETAILS
 - INTERFACECONTROLLER_USAGE
- Web Dynpro Applications

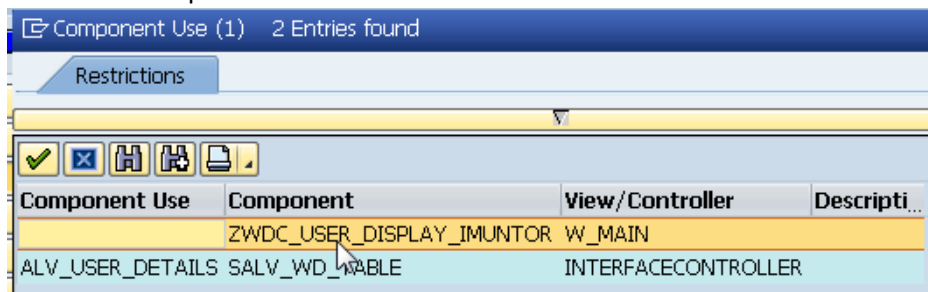
The “INTERFACECONTROLLER_USAGE” contains the own context of the SALV_WD_TABLE Component, lets open it:



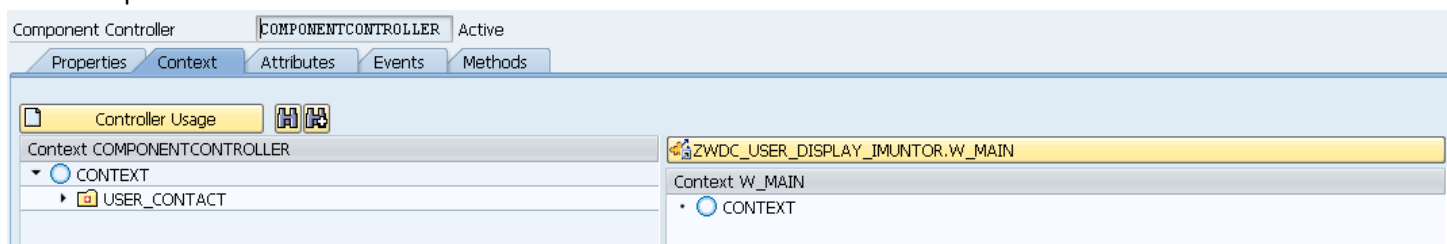
Now we have to integrate our Controller Nodes Attributes with the ALV Component. This way we will be able to work with both together. Press the button “Controller Usage”:



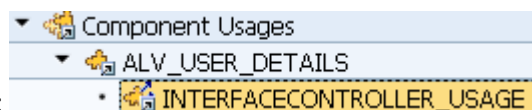
Select our Component:



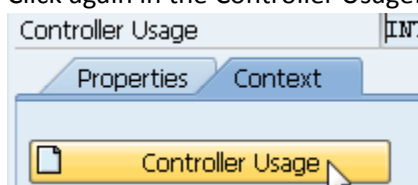
Your Component Controller will look like this now:



Go back to the:



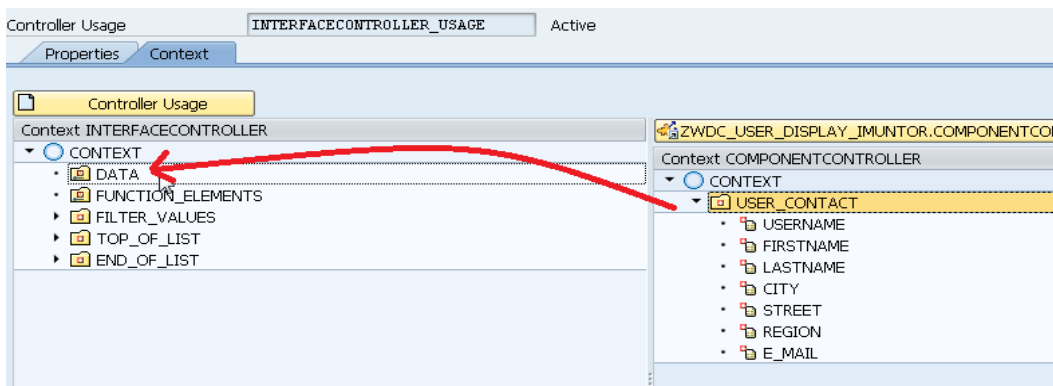
Click again in the Controller Usage:



Select our component:

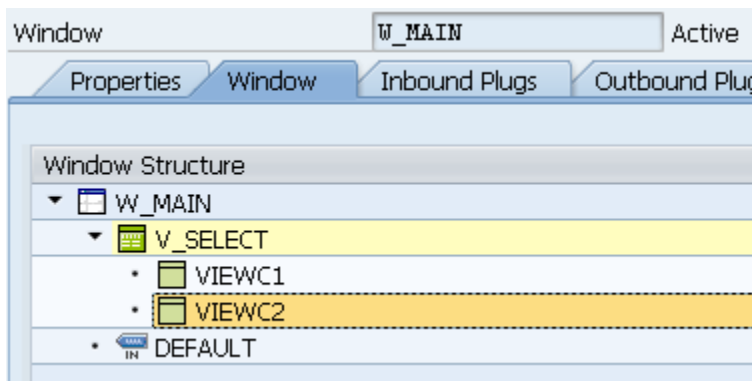
Component	View/Controller	Description
ZWDC_USER_DISPLAY_IMUNTOR	COMPONENTCONTROLLER	Component Controller

Now the component controller will open in the right side panel as shown below. Drag and drop the USER_CONTACT into the DATA node of the Interface Controller. This will declare a mapping. Meaning we have just declared which node is going to be displayed in the ALV.

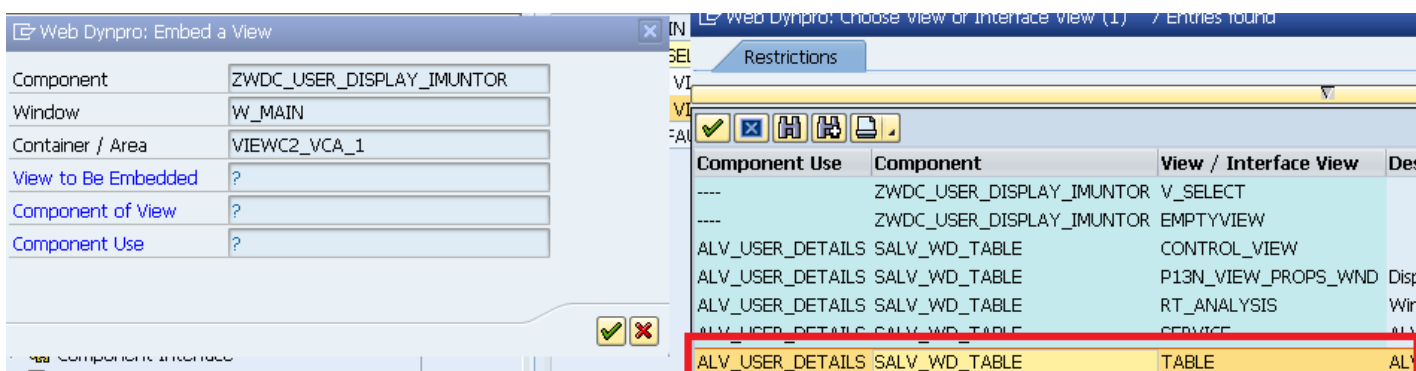
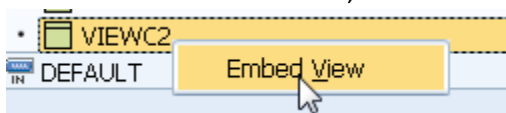


8) Changing the Window W_MAIN Structure

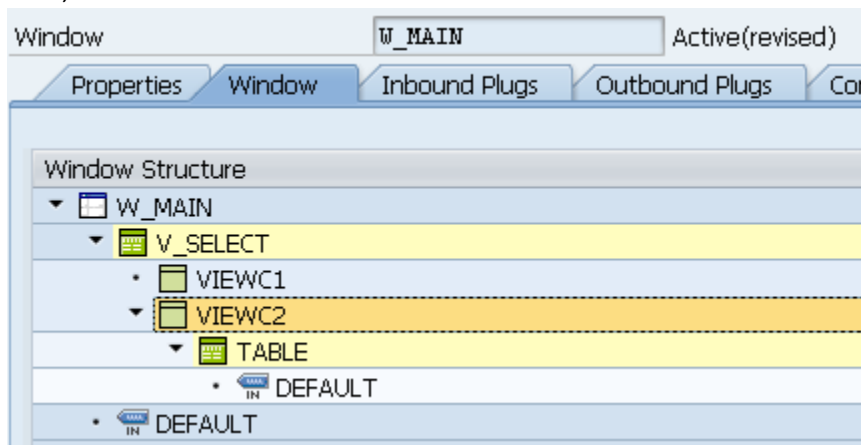
a. Your window will look like this:



b. The ALV will be in VIEWC2, so let's EMBE it:



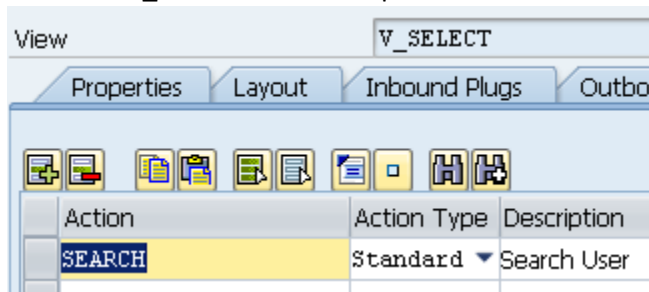
Now, it will look like this:



9) Save and Activate everything.

10) Coding the App

a. Go to the V_SELECT View and open the Search button action:

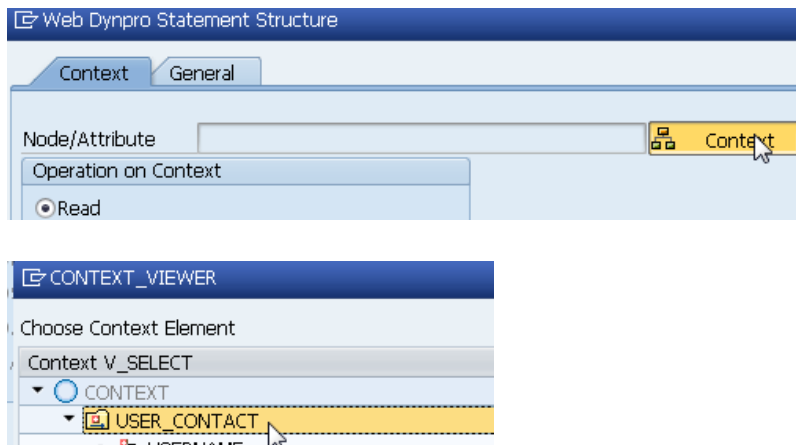


The first thing we need to do is to read the user through a BAPI. We already did that on the VOL1 of this training. So let's take advantage of that and use here:

```
1  METHOD onactionsearch .  
2  
3      " Get the user details  
4      wd_comp_controller->get_user_contact_info( ).  
...
```

After that, let's read the structure of the node created that we named "USER_CONTACT".

Press the Wizard Button: .



The following code will be generated:

```

6  " Wizard
7  DATA lo_nd_user_contact TYPE REF TO if_wd_context_node.
8
9  DATA lo_el_user_contact TYPE REF TO if_wd_context_element.
10 DATA ls_user_contact TYPE wd_this->element_user_contact.
11
12 * navigate from <CONTEXT> to <USER_CONTACT> via lead selection
13 lo_nd_user_contact = wd_context->get_child_node( name = wd_this->wdctx_user_contact ).
14
15 * @TODO handle non existant child
16 * IF lo_nd_user_contact IS INITIAL.
17 *   ENDIF.
18
19 * get element via lead selection
20 lo_el_user_contact = lo_nd_user_contact->get_element( ).
21 * @TODO handle not set lead selection
22 IF lo_el_user_contact IS INITIAL.
23   ENDIF.
24
25 * get all declared attributes
26 lo_el_user_contact->get_static_attributes(
27   IMPORTING
28     static_attributes = ls_user_contact ).

```

Now, lets pass the value of the strucuture ls_user_contact to the ALV:

```

31 " Fill ALV
32 DATA:
33   node_node_alv TYPE REF TO if_wd_context_node.
34
35 * navigate from <CONTEXT> to <NODE_ALV> via lead selection
36 node_node_alv = wd_context->get_child_node( name = if_v_select=>wdctx_user_contact ).
37
38 * get all declared attributes
39 node_node_alv->bind_structure( ls_user_contact ).

```

As we can see though debug, the variable "node_node_alv" is feeded with the 'USER_CONTACT' node, and later the values were BINDED.

608 **DATA:**
609 node_node_alv **TYPE REF TO** if_wd_context_node.
610
611 ** navigate from <CONTEXT> to <NODE_ALV> via lead selection*
612 node_node_alv = wd_context->get_child_node(name = if_v_select=>wdctx_user_contact).
613
614 ** get all declared attributes*
615 node_node_alv->bind_structure(ls_user_contact).
616
617 **ENDMETHOD.**
618
619 **endclass.**
620

Scope \CLASS CL_V_SELECT_Ctr\METHOD onactionsearch

Variables 1 Variables 2 Locals Globals Auto Memory Analysis

S...	Variable	V...	Val.	C...	Hexadecimal Value	Technical Type	Absolute Ty
	IF_V_SELECT=>WDCTX_...		USER_CONTACT		55005300450052005F004300...	CString{12}	\TYPE=STR
	NODE_NODE_ALV		{0:INITIAL}			Ref to IF_WD_CO...	\TYPE=%_T

Your final code should be more or less like this:

```
" Get the user details
wd_comp_controller->get_user_contact_info( ).

" Wizard
DATA lo_nd_user_contact TYPE REF TO if_wd_context_node.

DATA lo_el_user_contact TYPE REF TO if_wd_context_element.
DATA ls_user_contact TYPE wd_this->element_user_contact.

* navigate from <CONTEXT> to <USER_CONTACT> via lead selection
lo_nd_user_contact = wd_context->get_child_node( name = wd_this-
>wdctx_user_contact ).

* @TODO handle non existant child
* IF lo_nd_user_contact IS INITIAL.
* ENDIF.

* get element via lead selection
lo_el_user_contact = lo_nd_user_contact->get_element( ).
* @TODO handle not set lead selection
IF lo_el_user_contact IS INITIAL.
ENDIF.

* get all declared attributes
lo_el_user_contact->get_static_attributes(
IMPORTING
static_attributes = ls_user_contact ).

" Fill ALV
DATA:
node_node_alv TYPE REF TO if_wd_context_node.

* navigate from <CONTEXT> to <NODE_ALV> via lead selection
node_node_alv = wd_context-
```

```
>get_child_node( name = if_v_select=>wdctx_user_contact ).

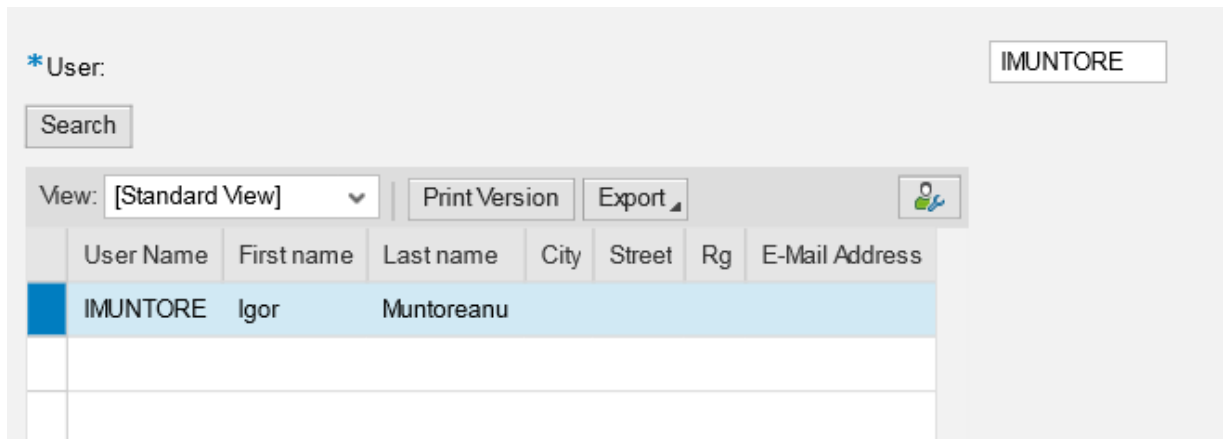
* get all declared attributes
node_node_alv->bind_structure( ls_user_contact ).
```

11) Checking the results after running the WDA:

- a. Go to the WDA and execute it:

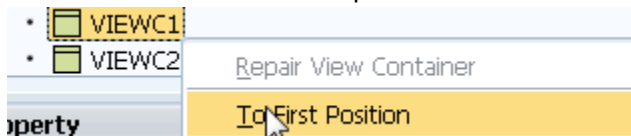


As we can see in the SS below, the layout is messed up, but it is working:

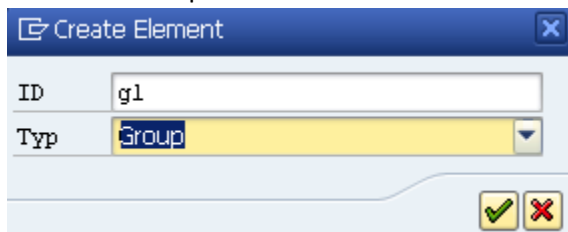


12) Adjusting the layout for best Appearance:

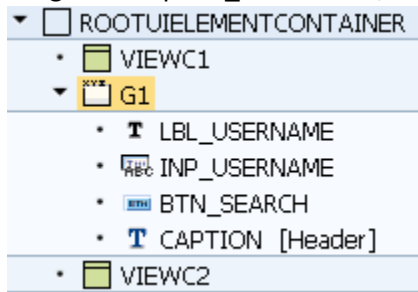
- a. Send the VIEWC1 to the first position



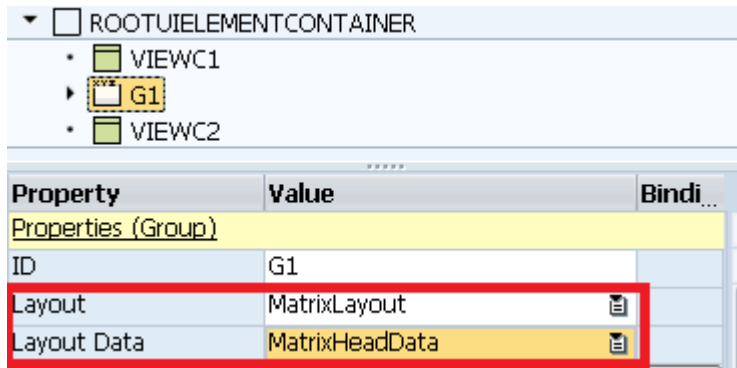
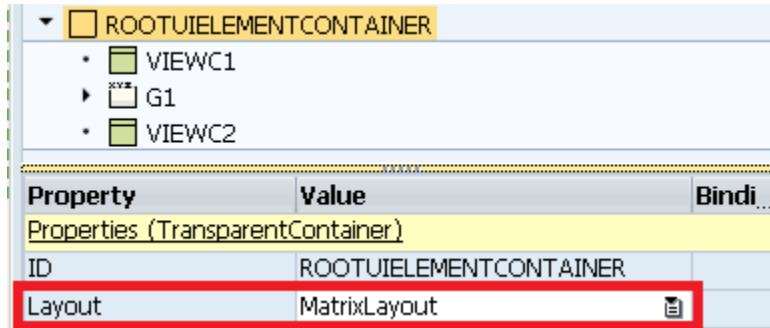
- b. Create an Group Element in the ROOT:



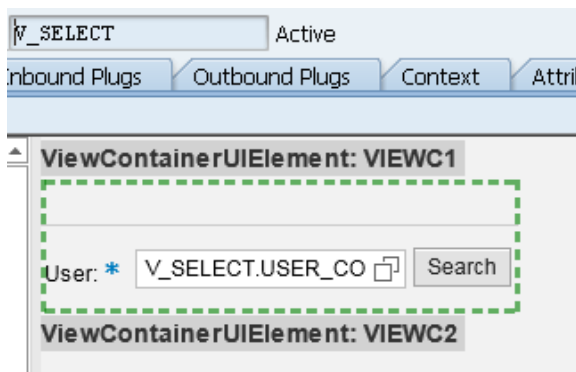
- c. Drag and drop LBL_USERNAME, LBL_USERNAME and LBL_USERNAME to the group created:



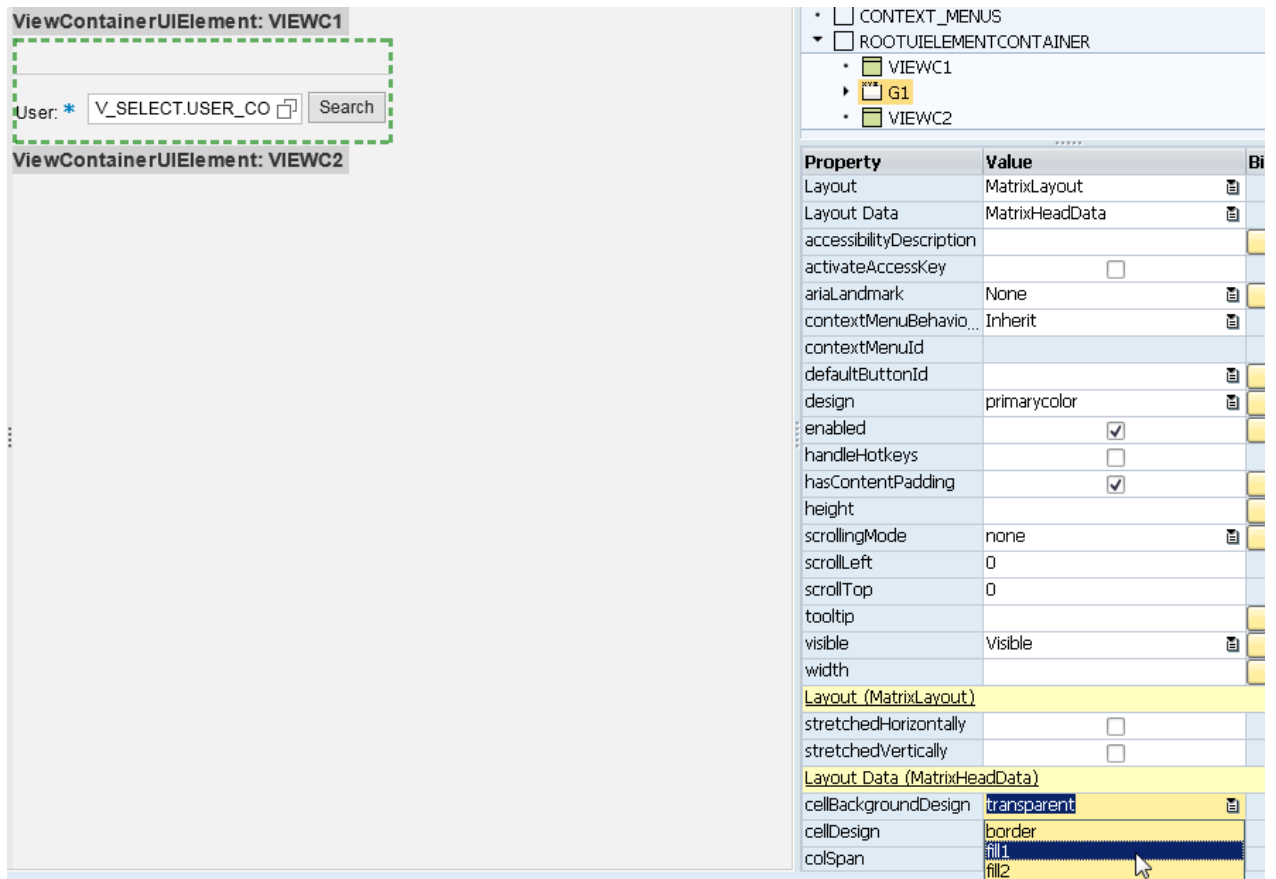
- d. Change all the Layouts from GRID to MATRIX:



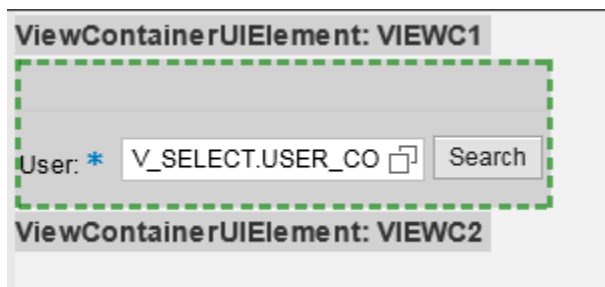
- e. The result should be like this:



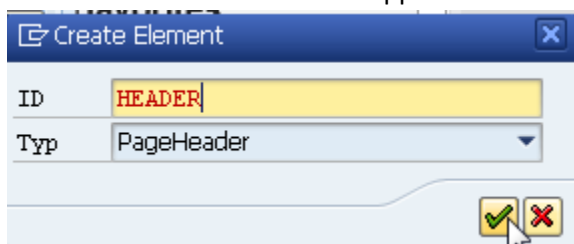
- f. One tip here, when you want to change a layout property just press ENTER and check the results, example:
Selecting "fill1" as background color:



Press ENTER:



- g. Lets create now a title for our Application:

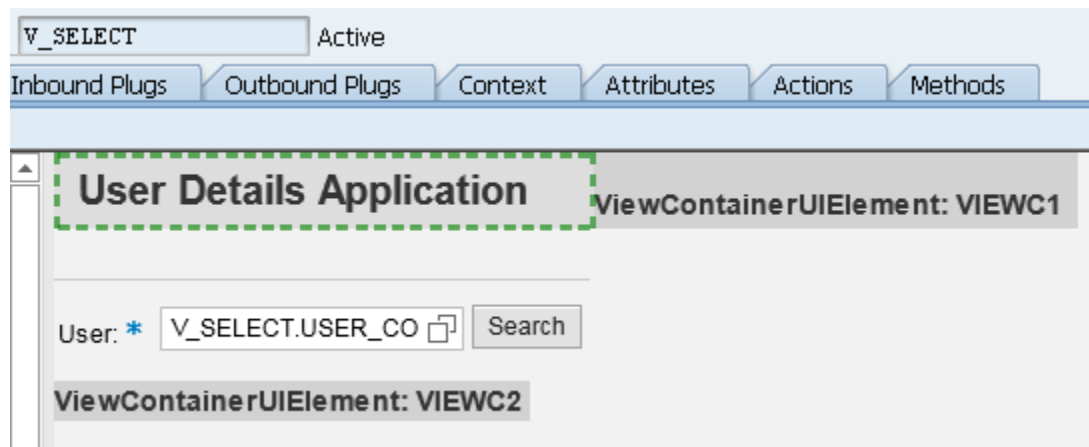


Put this in the top of the List of the ROOT.
Name the title as: "User Details Application"

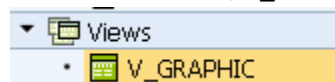
Properties:

Property	Value
Properties (PageHeader)	
ID	HEADER
contextMenuBehaviour	Inherit
contextMenuId	
design	Small
enabled	<input checked="" type="checkbox"/>
textDirection	Inherit
title	User Details Application
tooltip	
visible	Visible
Layout Data (MatrixHeadData)	
cellBackgroundDesign	transparent
cellDesign	padless
colSpan	1
dragData	
height	
hAlign	forcedLeft
vAlign	baseline
vGutter	None
width	270px

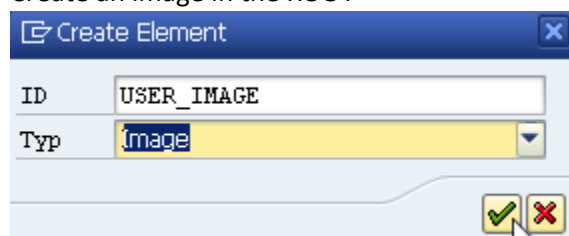
Result:



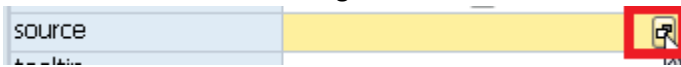
- h. The VIEWC1 for is useless, lets assign an image for that View
Create a new view, V_GRAPHIC



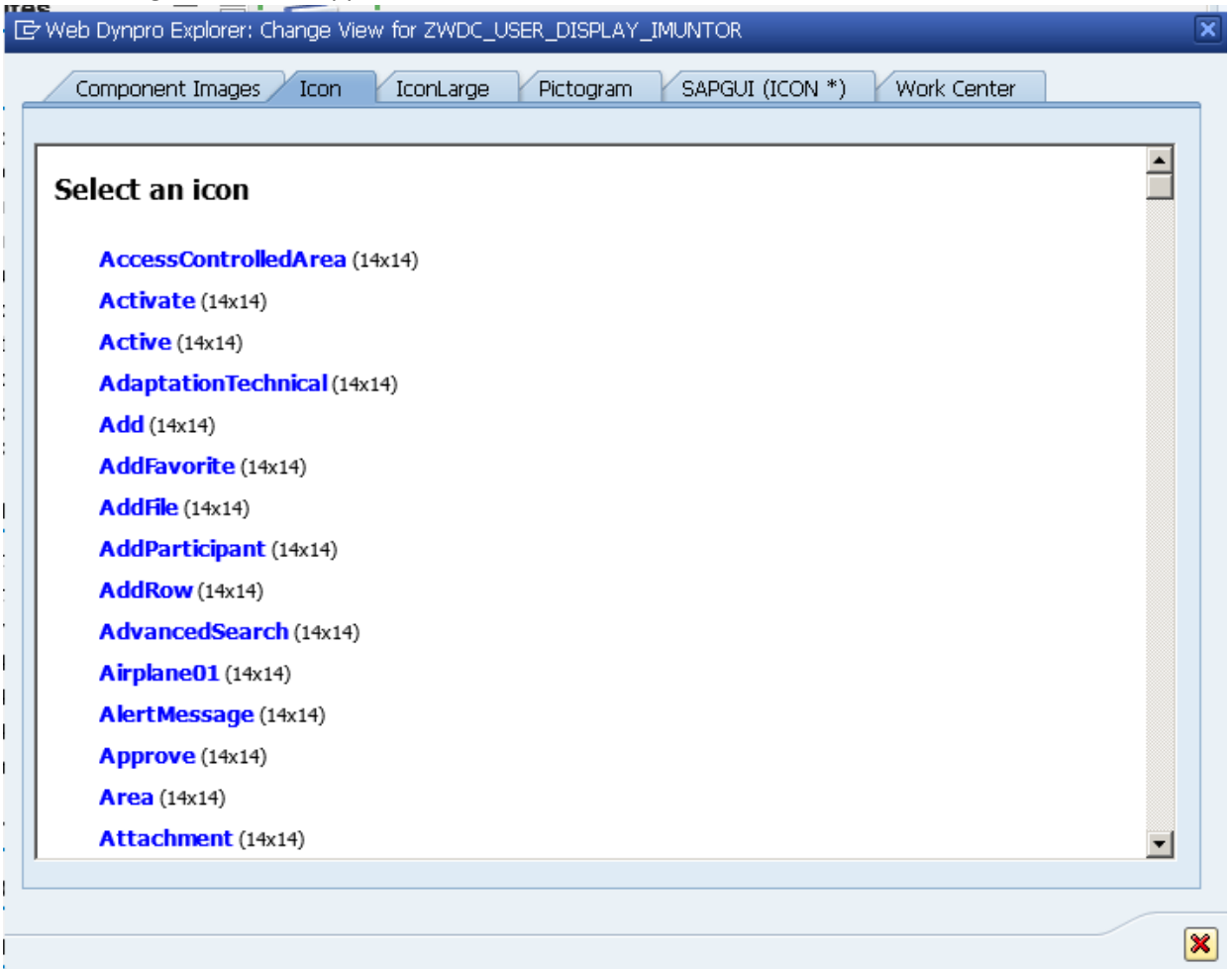
Create an image in the ROOT



Select the source of the image

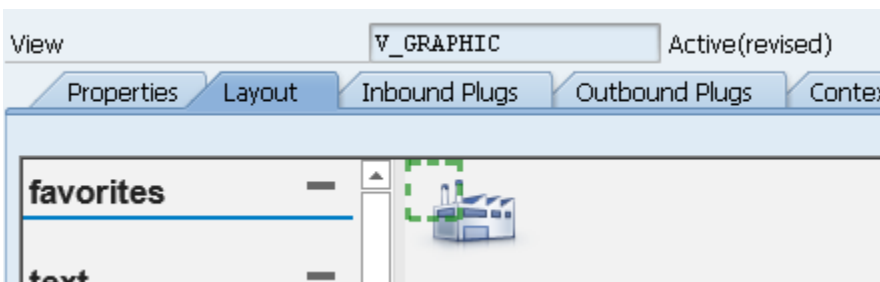


The following window will appears:

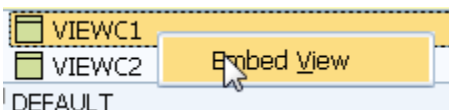


Select Work Center and select any image, my choice was "Company"

Result:



Go to the Window W_MAIN and Assign the view like shown below and choose the V_GRAPHIC.



13) Final Result:

User Details Application

*User:

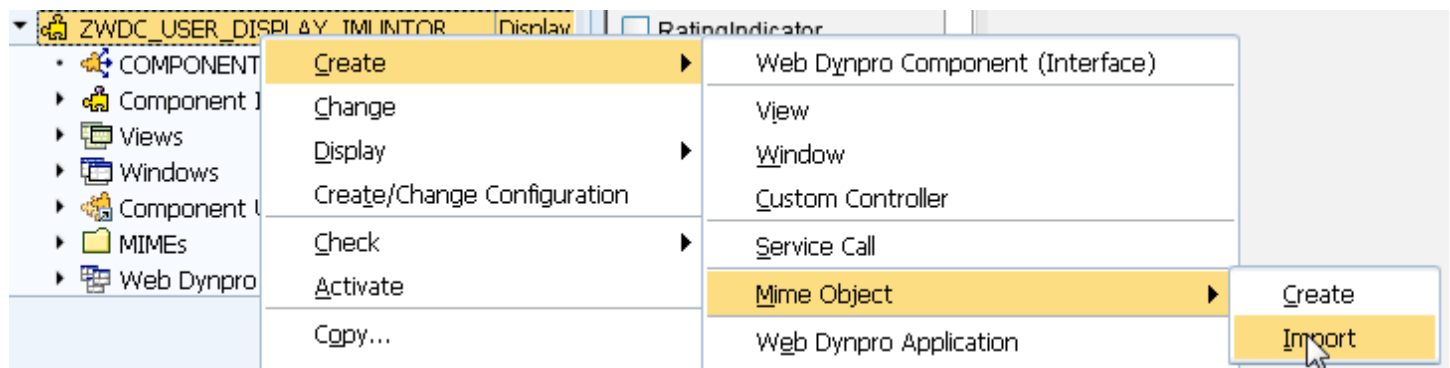
View:

User Name	First name	Last name	City	Street	Rg	E-Mail Address
IMUNTORE	Igor	Muntoreanu				

NOTE: These images are standard. You can add your own image by using the MIME Repository.

MIME: The MIME Repository is used to store all MIME objects (style sheets, graphics, icons and so on) in the SAP System.

Example:



Upload the image you want and save:

Create New Document

Shared Log.Doc.Properties Technical Information

Doc. Class Image Object

Name Systems.gif

Description

File Name C:\Users\f0fp532\Desktop\System.gif

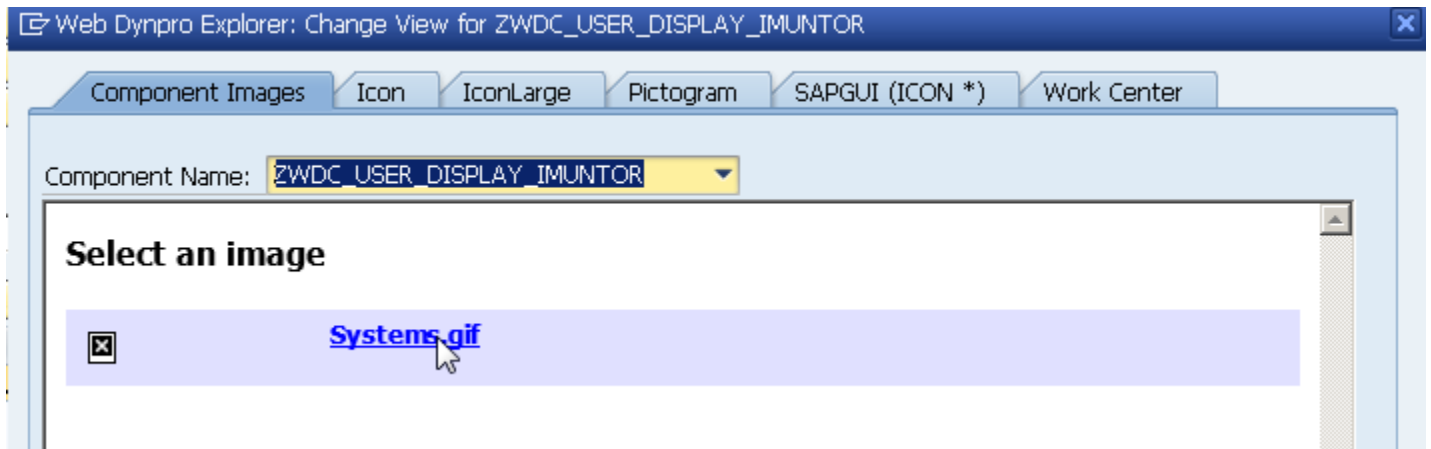
Mime Type image/gif

Save Cancel

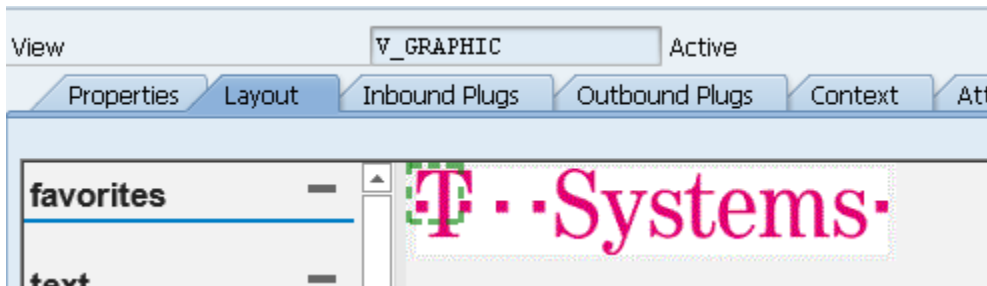
Go to the V_GRAPHIC view

USER_IMAGE	
Property	Value
Properties (Image)	
ID	USER_IMAGE
adjustImageSize	<input type="checkbox"/>
border	0
contextMenuBehaviour	Inherit
contextMenuId	
enabled	<input checked="" type="checkbox"/>
height	
isDecorative	<input type="checkbox"/>
isDragHandle	<input type="checkbox"/>
source	systems.gif

Choose your image:



It is done:



Final Result:

User Details Application

T...Systems

*User:

View: [Standard View] 

	User Name	First name	Last name	City	Street	Rg	E-Mail Address
--	-----------	------------	-----------	------	--------	----	----------------



14) Homework:

- Insert a new column (Country) in the ALV between Last Name and City. Tip: It comes from BAPIADDR3-COUNTRY.

	User Name	First name	Last name	City	Street	Rg	E-Mail Address

Country