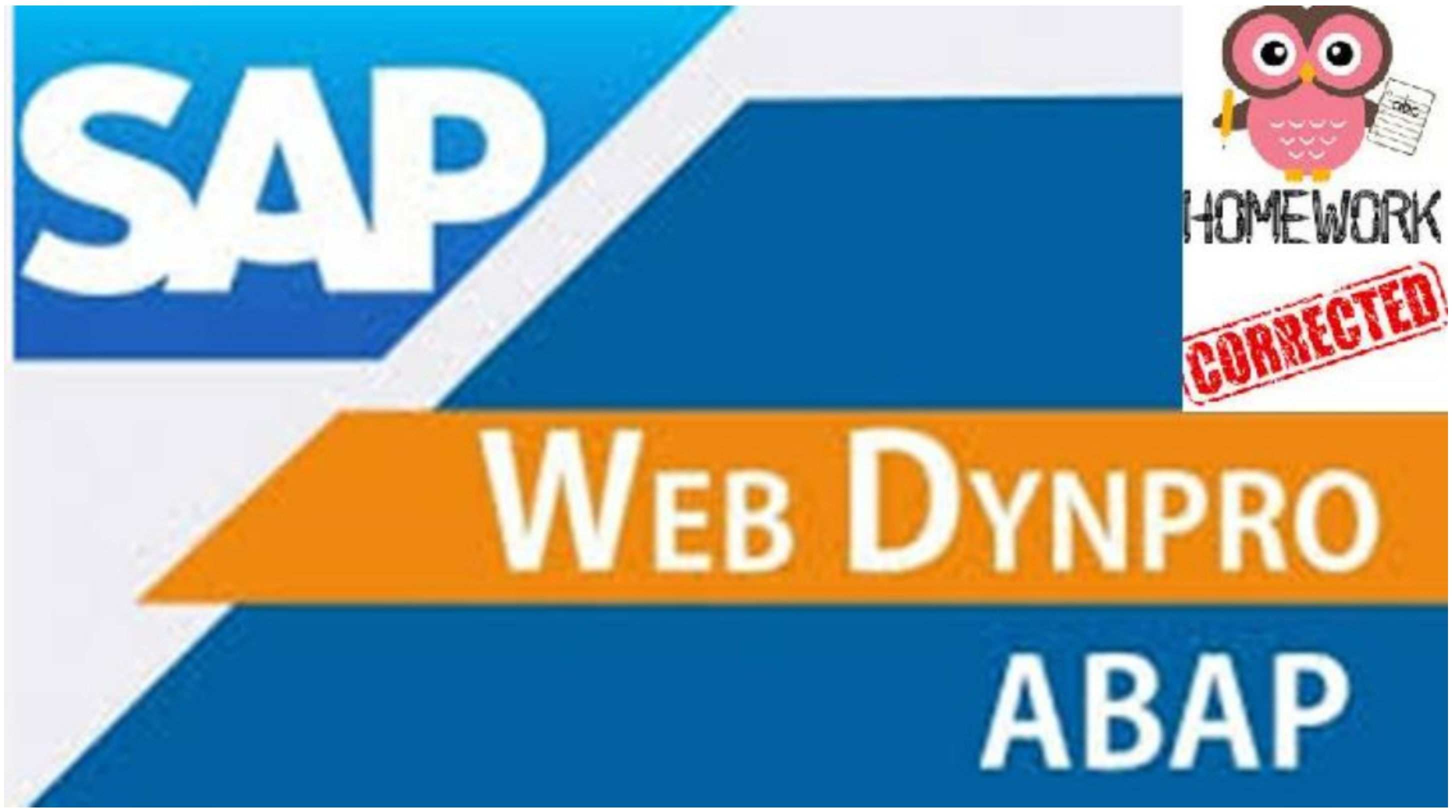
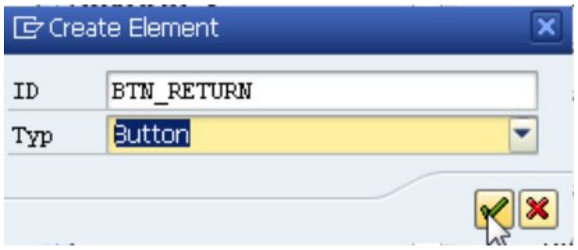


Version 1.0
Date 27.06.2016
Status In Development
Number 1
Initial version by Igor Muntoreanu

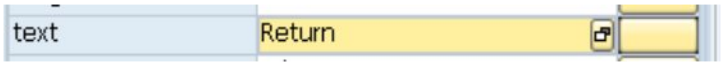
1) Homework (item14 from the Practical Guide):
   a. Create a button in the V_DETAILS to return to the V_SELECT view.
      Processed to Number 2)
   b. Create a validation if the entry data is valid. If not display a POP UP and freeze in the V_SELECT view.
      Processed to Number 3)

2) Return Button to get back to the initial view.
   a. Navigate to the V_DETAILS view
   b. Create a new button in the ROOTUIELEMENTCONTAINER:



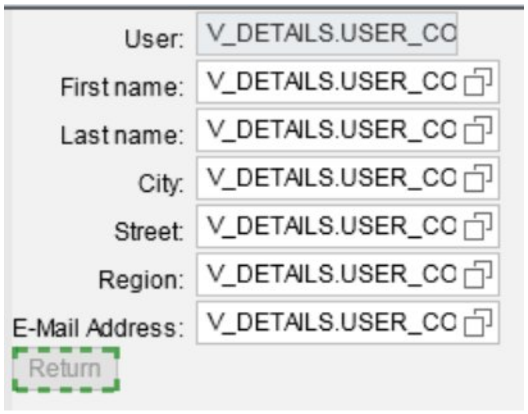
ID	BTN_RETURN
Typ	Button

   c. Name the new button as "RETURN":



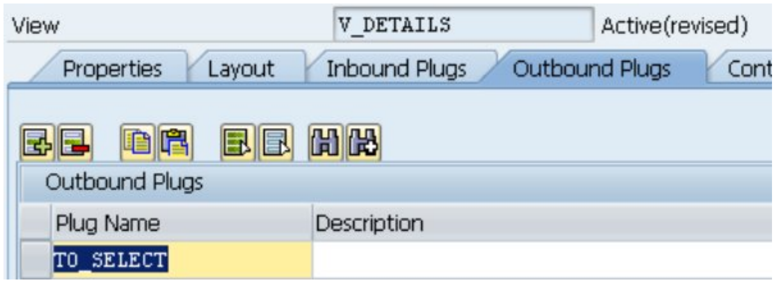
text	Return

   d. You view will be like this now:



User:	V_DETAILS.USER_CO
First name:	V_DETAILS.USER_CO
Last name:	V_DETAILS.USER_CO
City:	V_DETAILS.USER_CO
Street:	V_DETAILS.USER_CO
Region:	V_DETAILS.USER_CO
E-Mail Address:	V_DETAILS.USER_CO

   e. Make sure that you have the Outbound Plug like below:

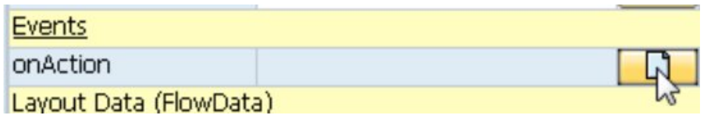


View V_DETAILS Active(revised)

Outbound Plugs

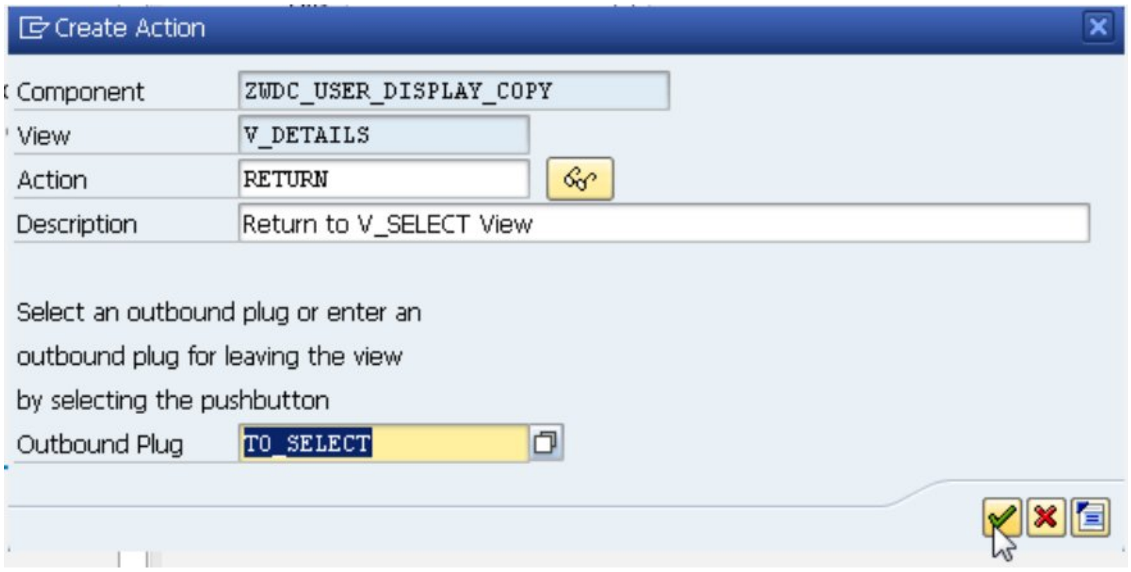
Plug Name	Description
TO_SELECT	

   f. Create an Action to the new button:



Events	
onAction	
Layout Data (FlowData)	

Action: RETURN
Outbound Plug: TO_SELECT
Confirm/Enter

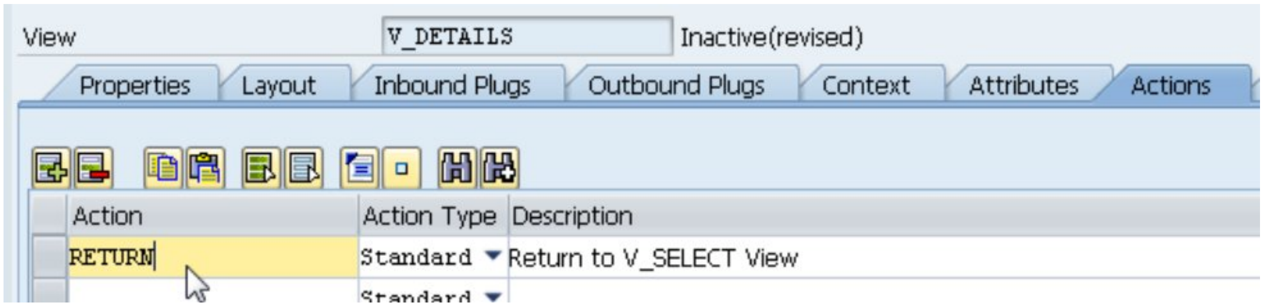


Component	ZWDC_USER_DISPLAY_COPY
View	V_DETAILS
Action	RETURN
Description	Return to V_SELECT View

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

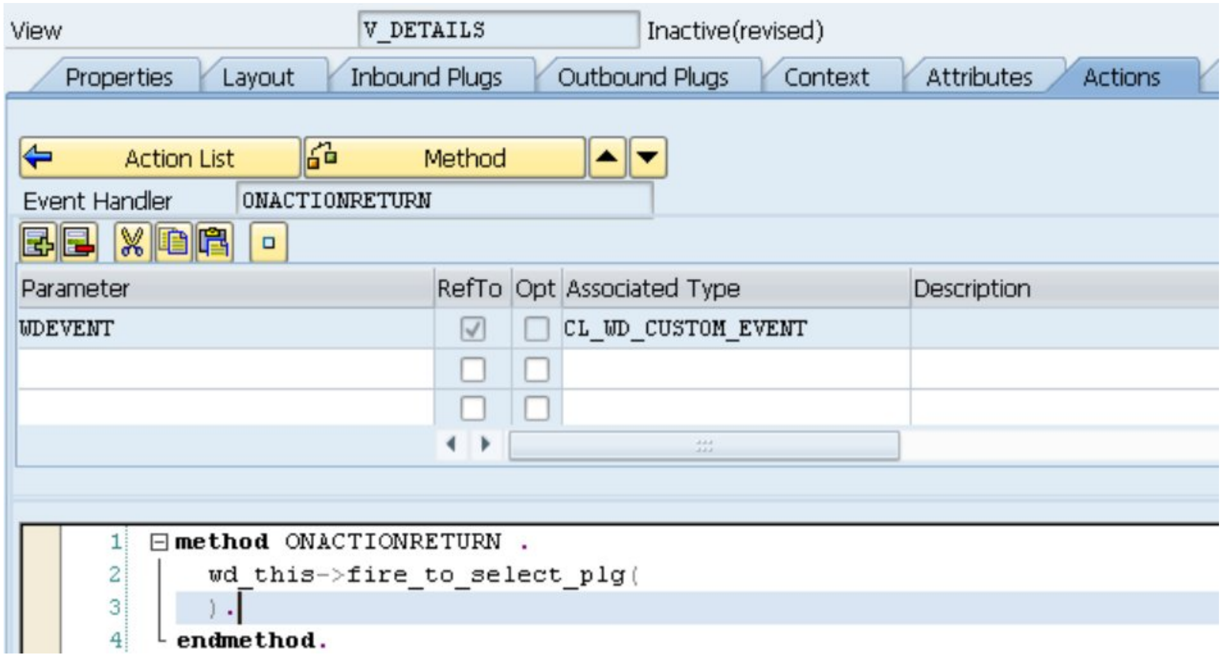
Outbound Plug: TO_SELECT

g. Navigate to the Action tab and click twice in the new action created:



Action	Action Type	Description
RETURN	Standard	Return to V_SELECT View
	Standard	

As you can see the pre-defined code was already generated:

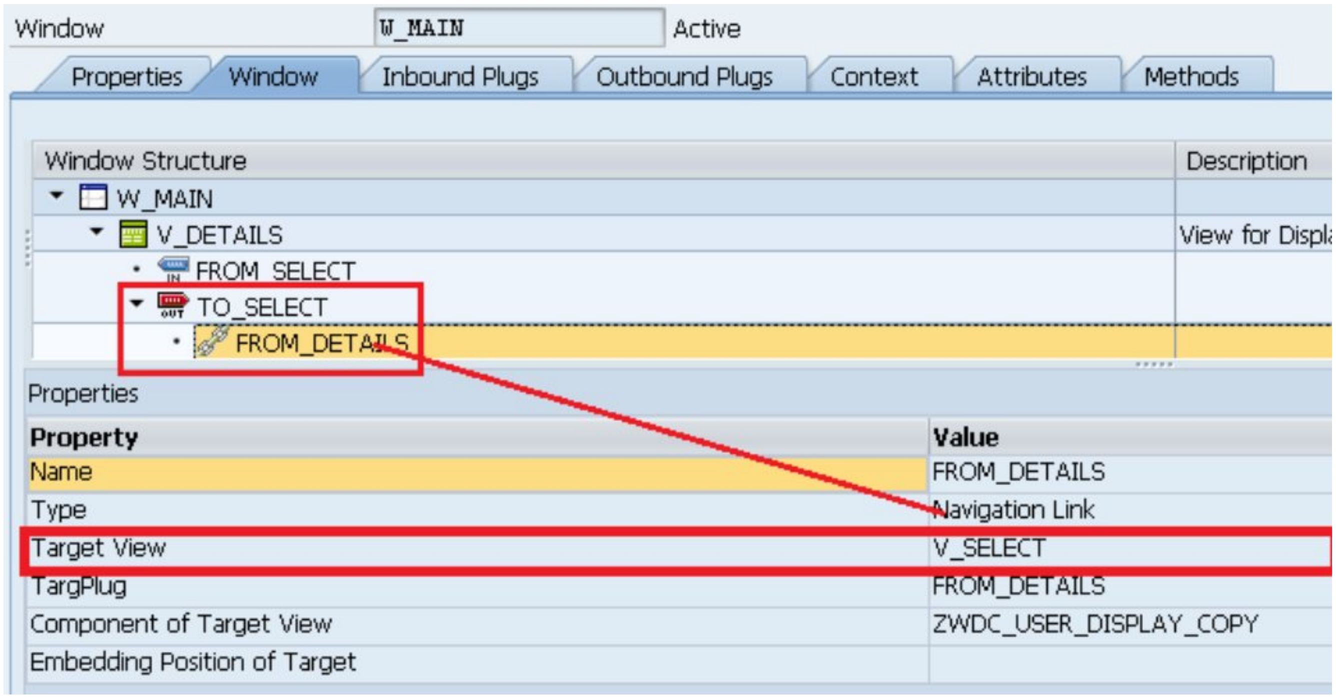


Event Handler: ONACTIONRETURN

Parameter	RefTo	Opt	Associated Type	Description
WDEVENT	✓		CL_WD_CUSTOM_EVENT	

```
method ONACTIONRETURN .
  wd_this->fire_to_select_plg(
  ).
endmethod.
```

h. Go to the window and check the navigation configuration the V_DETAILS view should be like this:



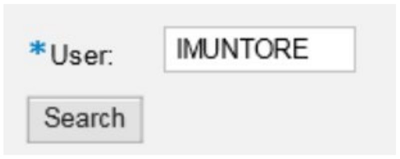
Window W_MAIN Active

Window Structure	Description
W_MAIN	
V_DETAILS	View for Displ
FROM_SELECT	
TO_SELECT	
FROM_DETAILS	

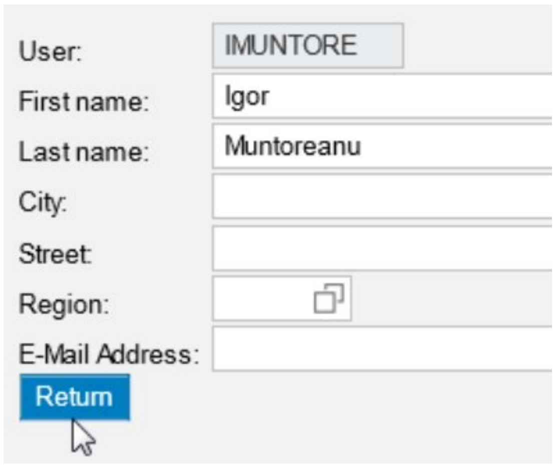
Properties

Property	Value
Name	FROM_DETAILS
Type	Navigation Link
Target View	V_SELECT
TargPlug	FROM_DETAILS
Component of Target View	ZWDC_USER_DISPLAY_COPY
Embedding Position of Target	

i. Activate all the objects and run your WDA application



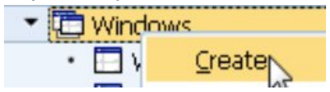
*User:	IMUNTORE



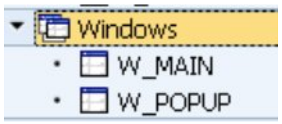
User:	IMUNTORE
First name:	Igor
Last name:	Muntoreanu
City:	
Street:	
Region:	
E-Mail Address:	

3) POPUP to validate the entered user

a. Before anything, if we are going to create a new POPUP it will be necessary to create a WINDOW separately for that POPUP. Right click on the window and create a new one:

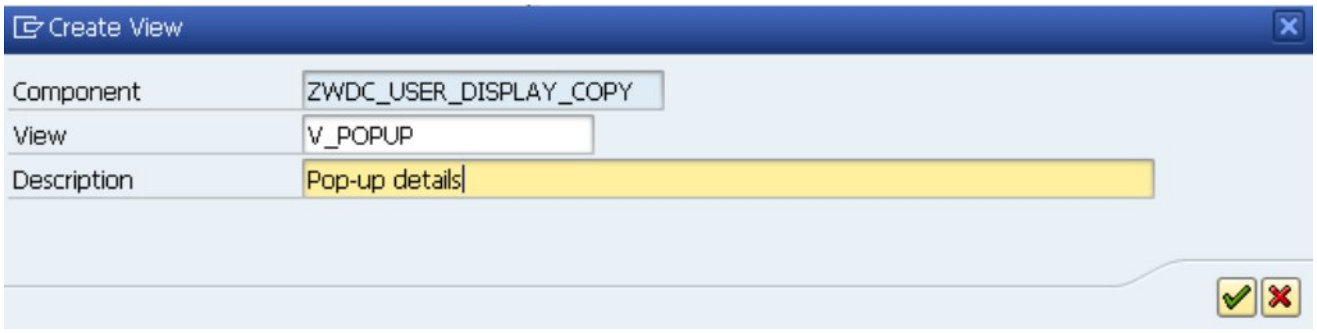


Result:



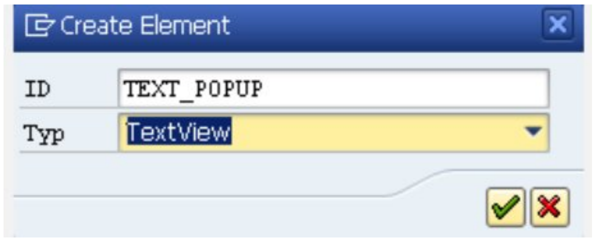
- Windows
  - W_MAIN
  - W_POPUP

b. Also, create a VIEW for this new window:



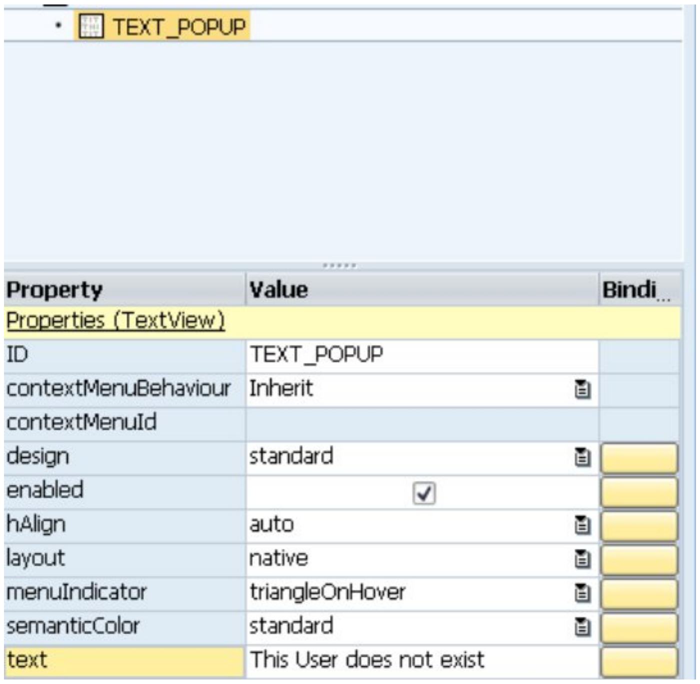
Component	ZWDC_USER_DISPLAY_COPY
View	V_POPUP
Description	Pop-up details

Create the following element in the ROOTUIELEMENTCONTAINER:



ID	TEXT_POPUP
Typ	TextView

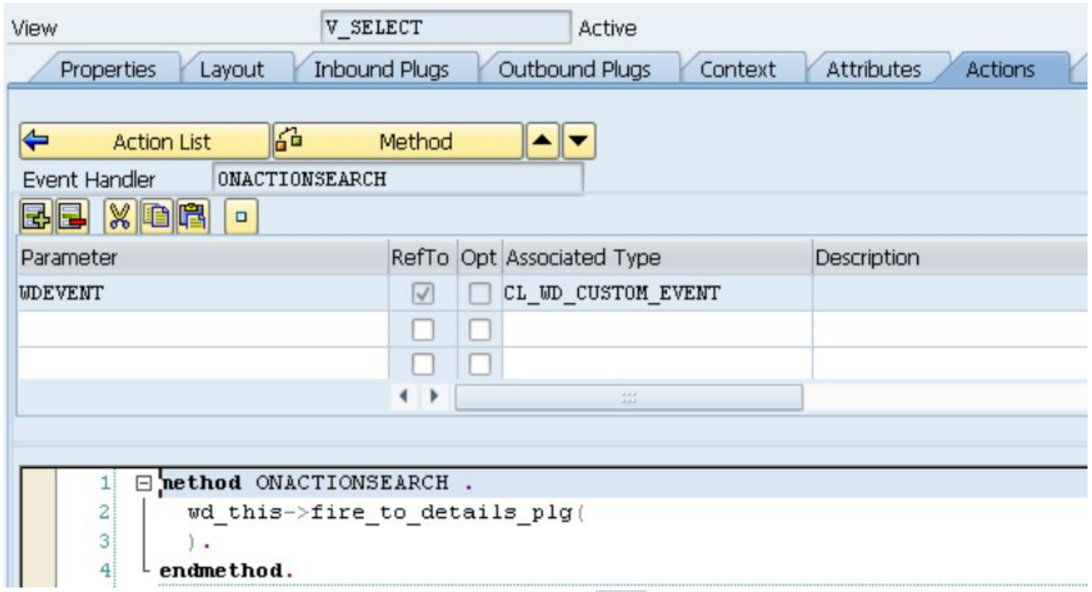
Give a simple text that will be shown in the new POPUP window:



- TEXT_POPUP

Property	Value	Bindi...
Properties (TextView)		
ID	TEXT_POPUP	
contextMenuBehaviour	Inherit	
contextMenuId		
design	standard	
enabled	✓	
hAlign	auto	
layout	native	
menuIndicator	triangleOnHover	
semanticColor	standard	
text	This User does not exist	

c. Navigate to the SEARCH action in the V_SELECT View. (Search action is the action of the V_SELECT Search Button).



View V_SELECT Active

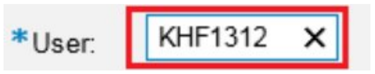
Properties | Layout | Inbound Plugs | Outbound Plugs | Context | Attributes | Actions

Event Handler ONACTIONSEARCH

Parameter	RefTo	Opt	Associated Type	Description
WDEVENT	✓		CL_WD_CUSTOM_EVENT	

```
method ONACTIONSEARCH .
  wd_this->fire_to_details_plg(
  ).
endmethod.
```

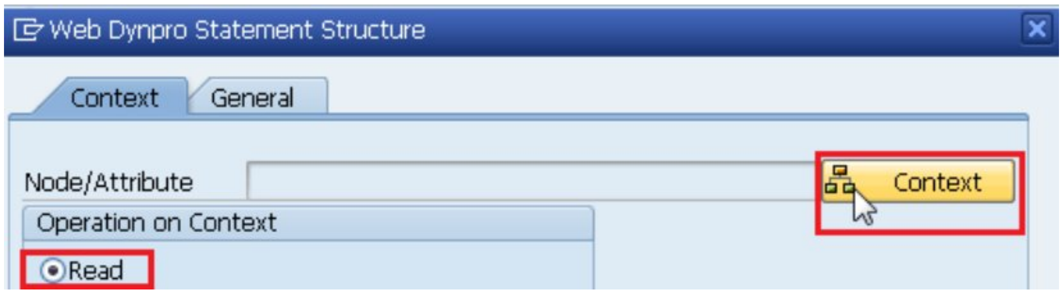
d. Now, let's do some code. Click at the Wizard Toll
The idea here is to check the entered value that the user put in the input field:



*User: KHF1312

If it is Wrong, a popup will be displayed

Getting back to the Wizard, do:



Web Dynpro Statement Structure

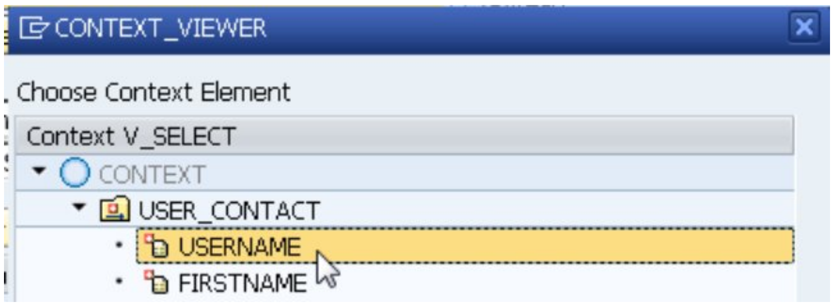
Context | General

Node/Attribute

Operation on Context

[✓ Read]

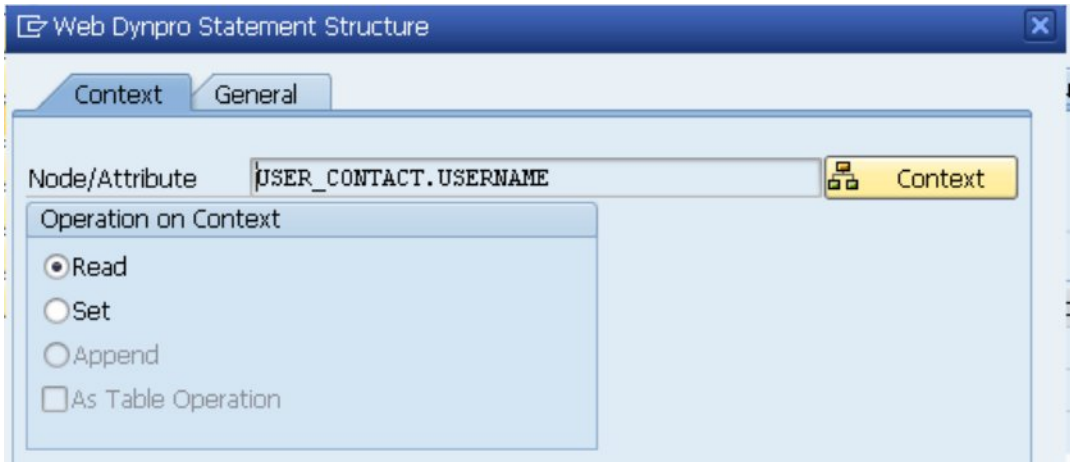
Choose the USERNAME and press ENTER:



CONTEXT_VIEWER

Choose Context Element

Context V_SELECT
- CONTEXT
  - USER_CONTACT
    - USERNAME
    - FIRSTNAME



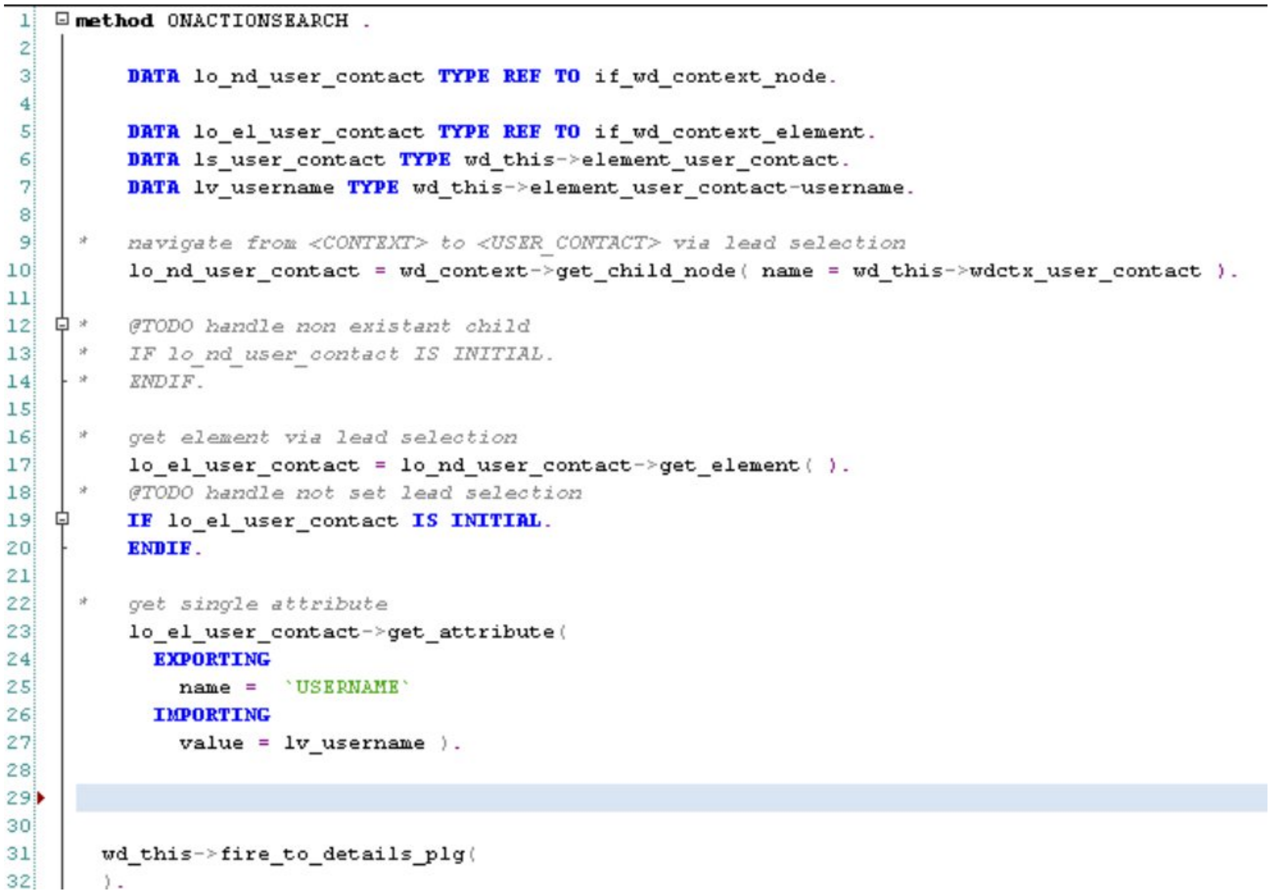
Node/Attribute: USER_CONTACT.USERNAME

Operation on Context

(● Read) ○ Set ○ Append ☐ As Table Operation

Press ENTER

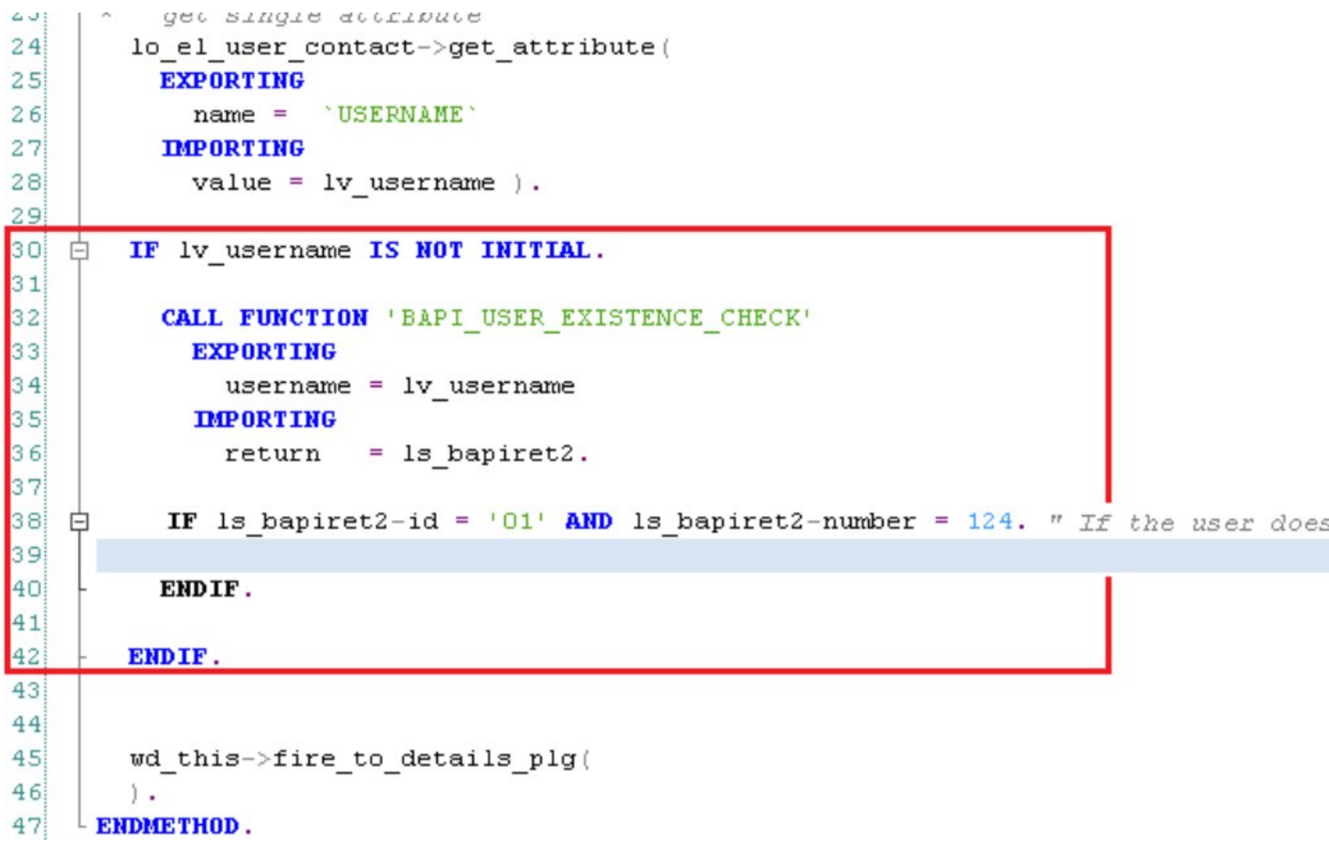
Some CODE will be generated automatically:



```
method ONACTIONSEARCH .

    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.
    DATA lv_username TYPE wd_this->element_user_contact-username.

*   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 single attribute
    lo_el_user_contact->get_attribute(
      EXPORTING
        name =  `USERNAME`
      IMPORTING
        value = lv_username ).



  wd_this->fire_to_details_plg(
  ).
```

By using the variable lv_username, check if the user exits by using the BAPI: “BAPI_USER_EXISTENCE_CHECK”

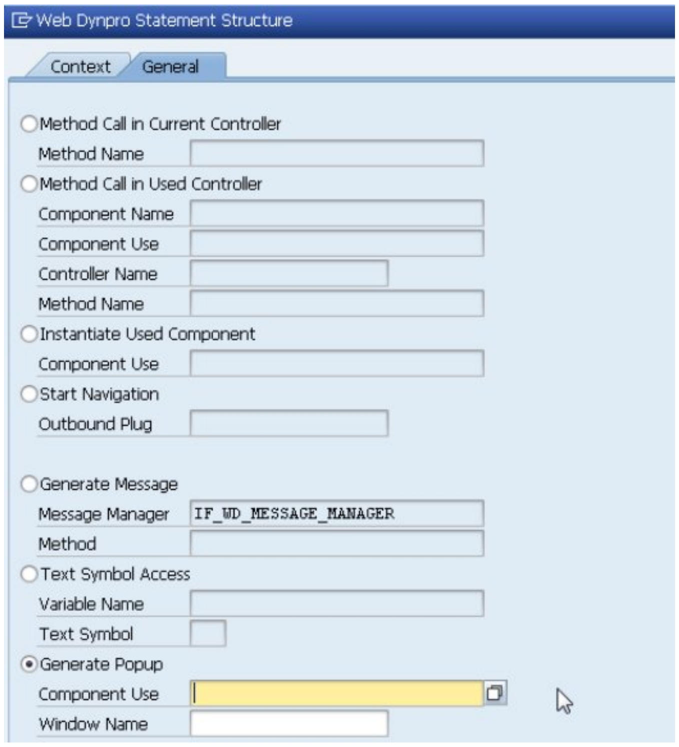
Your code should look like this now:



```
23    * get single attribute
24      lo_el_user_contact->get_attribute(
25        EXPORTING
26          name =  `USERNAME`
27        IMPORTING
28          value = lv_username ).
29
30      IF lv_username IS NOT INITIAL.
31
32        CALL FUNCTION 'BAPI_USER_EXISTENCE_CHECK'
33          EXPORTING
34            username = lv_username
35          IMPORTING
36            return   = ls_bapiret2.
37
38         IF ls_bapiret2-id = '01' AND ls_bapiret2-number = 124. " If the user does
39
40        ENDIF.
41
42      ENDIF.
43
44
45      wd_this->fire_to_details_plg(
46      ).
47    ENDMETHOD.
```

Put the cursor in the line 39 and click at the Wizard Toll .

Navigate until the GENERAL tab und choose the popup option like showed below:

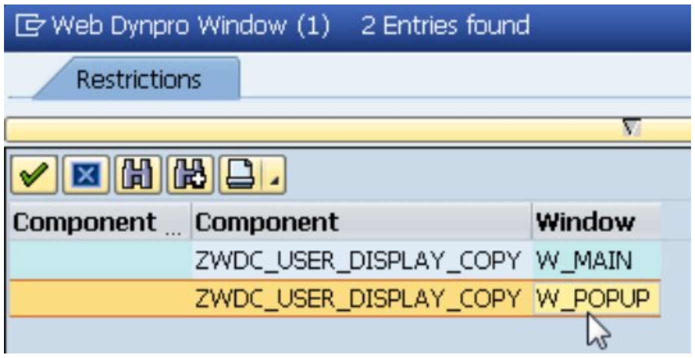


Web Dynpro Statement Structure

Context | General

○ Method Call in Current Controller	
Method Name	
○ Method Call in Used Controller	
Component Name	
Component Use	
Controller Name	
Method Name	
○ Instantiate Used Component	
Component Use	
○ Start Navigation	
Outbound Plug	
○ Generate Message	
Message Manager	IF_WD_MESSAGE_MANAGER
Method	
○ Text Symbol Access	
Variable Name	
Text Symbol	
[✓ Generate Popup]	
Component Use	
Window Name	

Press the match-code and select our component

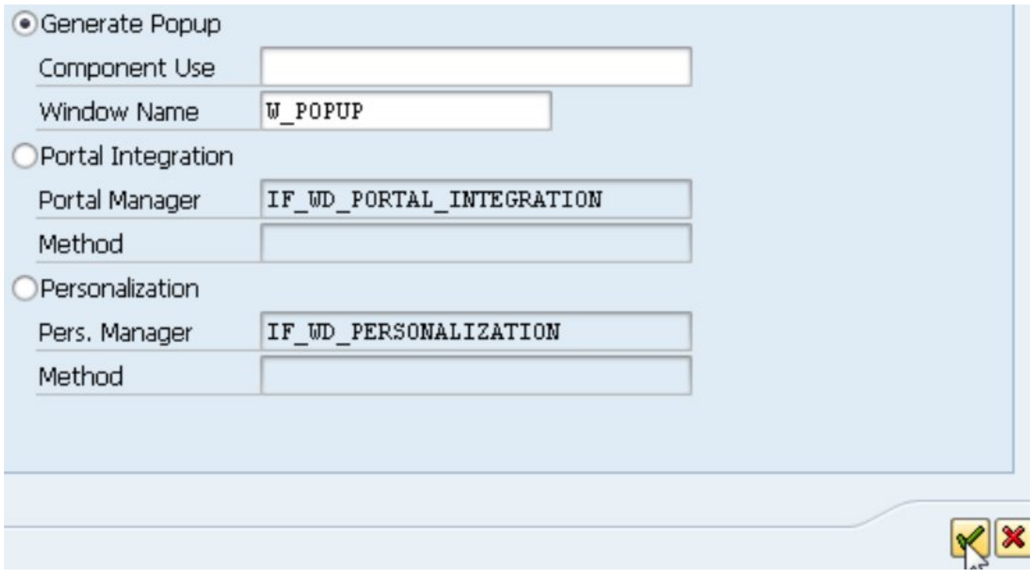


Web Dynpro Window (1)   2 Entries found

Restrictions

Component ...	Component	Window
	ZWDC_USER_DISPLAY_COPY	W_MAIN
	ZWDC_USER_DISPLAY_COPY	W_POPUP

Press OK:



(●) Generate Popup

Component Use:

Window Name: W_POPUP

( ) Portal Integration

Portal Manager: IF_WD_PORTAL_INTEGRATION

Method:

( ) Personalization

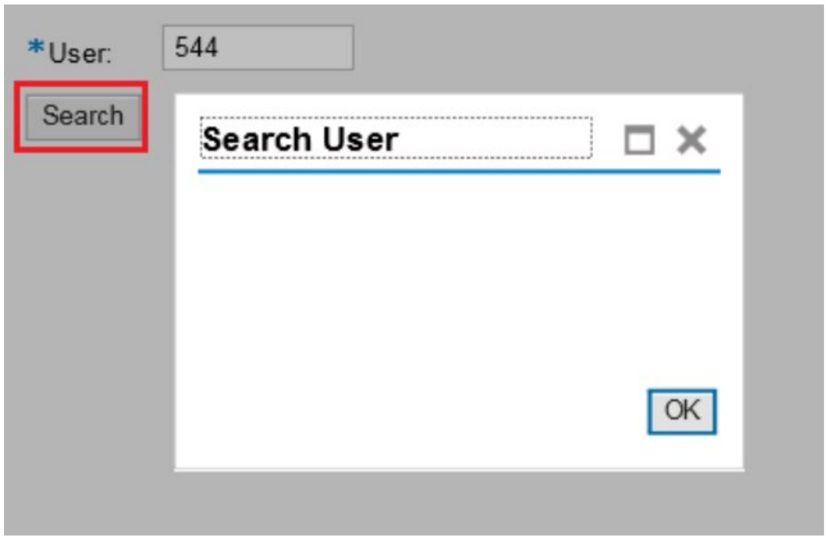
Pers. Manager: IF_WD_PERSONALIZATION

Method:

Now your code should be like shown below:

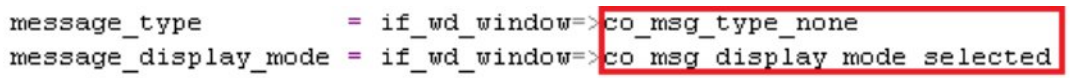
```
METHOD onactionsearch .

  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.
  DATA ls_bapiret2 TYPE bapiret2.
  DATA lv_username TYPE wd_this->element_user_contact-username.

*   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 single attribute
  lo_el_user_contact->get_attribute(
    EXPORTING
      name =  `USERNAME`
    IMPORTING
      value = lv_username ).

  IF lv_username IS NOT INITIAL.

    CALL FUNCTION 'BAPI_USER_EXISTENCE_CHECK'
      EXPORTING
        username = lv_username
      IMPORTING
        return   = ls_bapiret2.

    IF ls_bapiret2-id = '01' AND ls_bapiret2-
number = 124. " If the user does not exist


      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.
      DATA lt_buttons        TYPE wdr_popup_button_list.
      DATA ls_canc_action    TYPE wdr_popup_button_action.

      lo_api_component            = wd_comp_controller->wd_get_api( ).
      lo_window_manager           = lo_api_component->get_window_manager( ).
*       create the cancel icon, but without any action handler
      ls_canc_action-action_name = '*'.
*       Simple example, see docu of method create_and_open_popup for details
      lt_buttons                  = lo_window_manager->get_buttons_ok(
*          default_button       = if_wd_window=>co_button_ok
       ).

      lo_window                   = lo_window_manager->create_and_open_popup(
          window_name           = 'W_POPUP'

*          title                 =
          message_type          = if_wd_window=>co_msg_type_none
          message_display_mode = if_wd_window=>co_msg_display_mode_selected
*          is_resizable          = ABAP_TRUE
          buttons               = lt_buttons
          cancel_action         = ls_canc_action
      ).


  ELSE.
    wd_this->fire_to_details_plg( ).
    ENDIF.
  ELSE.
    wd_this->fire_to_details_plg( ).
  ENDIF.

ENDMETHOD.
```

e. Execute your WDA, put an invalid user and press SEARCH:



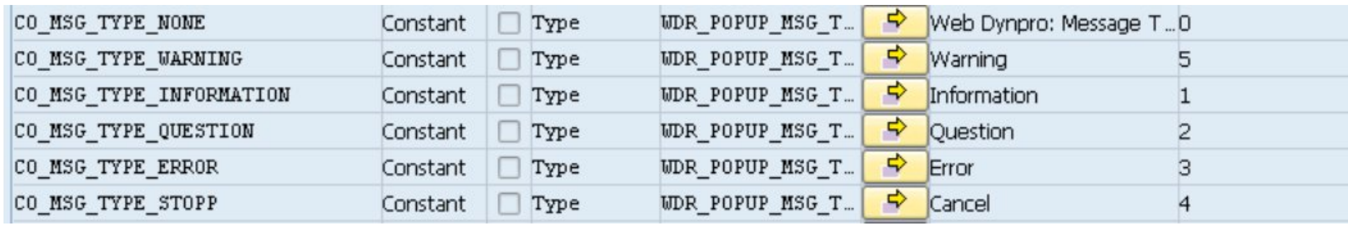
As we can see a new POPUP is appearing when the user is invalid. However this popup is very poor of information, lets improve it a little bit, so return to the ABAP code:

Click 2x in one of the marked attributes:



```
message_type         = if_wd_window=>co_msg_type_none
message_display_mode = if_wd_window=>co_msg_display_mode_selected
```

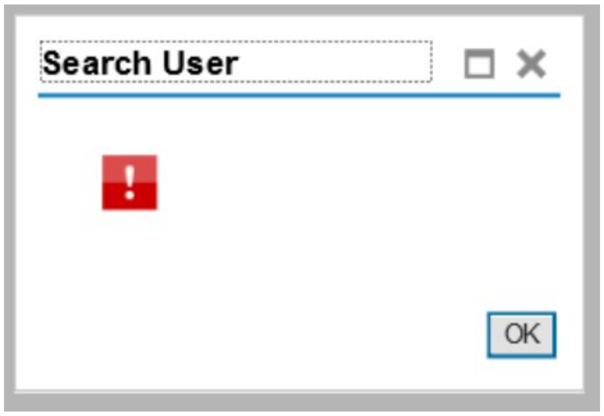
Instead of using the type "none", lets use the "error" type:



CO_MSG_TYPE_NONE	Constant	☐	Type	WDR_POPUP_MSG_T...	Web Dynpro: Message T...	0
CO_MSG_TYPE_WARNING	Constant	☐	Type	WDR_POPUP_MSG_T...	Warning	5
CO_MSG_TYPE_INFORMATION	Constant	☐	Type	WDR_POPUP_MSG_T...	Information	1
CO_MSG_TYPE_QUESTION	Constant	☐	Type	WDR_POPUP_MSG_T...	Question	2
CO_MSG_TYPE_ERROR	Constant	☐	Type	WDR_POPUP_MSG_T...	Error	3
CO_MSG_TYPE_STOPP	Constant	☐	Type	WDR_POPUP_MSG_T...	Cancel	4

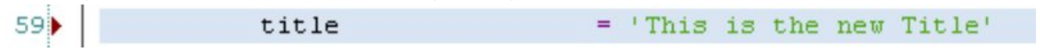
```
message_type         = if_wd_window=>co_msg_type_error
message_display_mode = if_wd_window=>co_msg_type_error
```

Now, your popup will look like this:



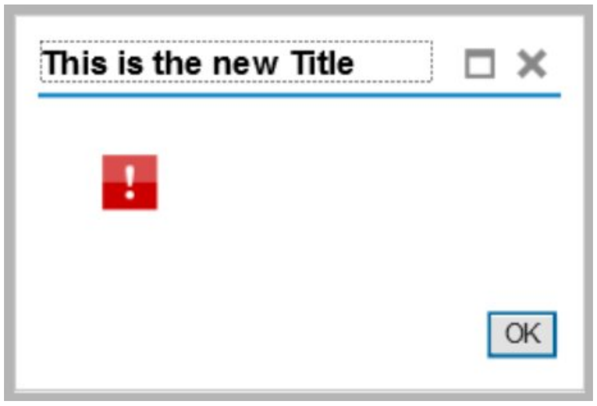
e. Changing the Title and the content:

Backing to the code, do the following change:



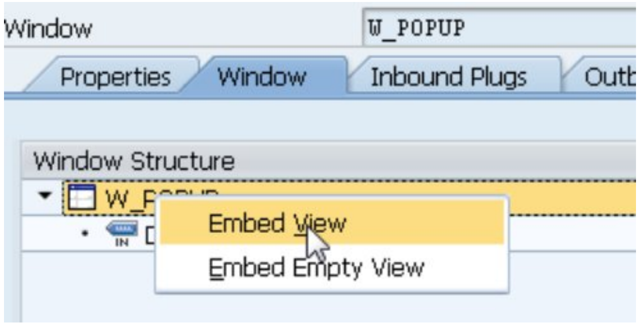
```
title = 'This is the new Title'
```

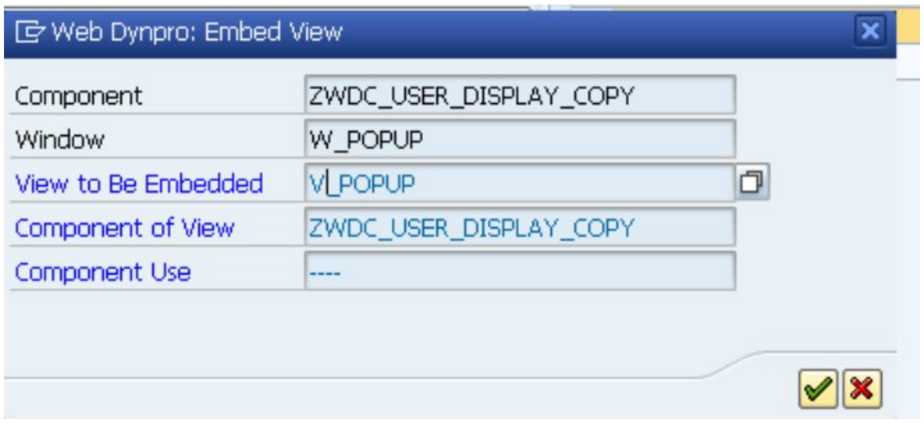
Result:



Adding some content (Help procedure for the user):

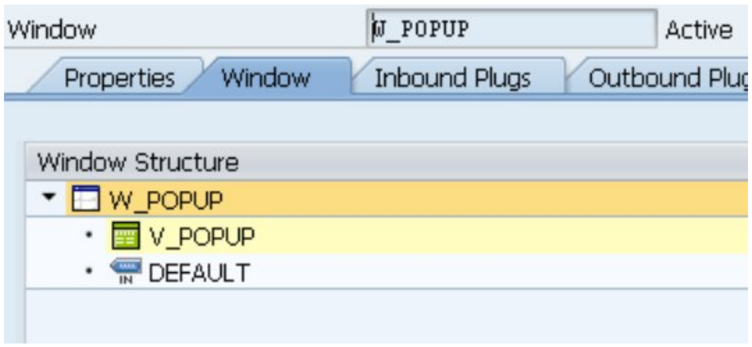
Just Embed the View V_POPUP into the W_POPUP window:



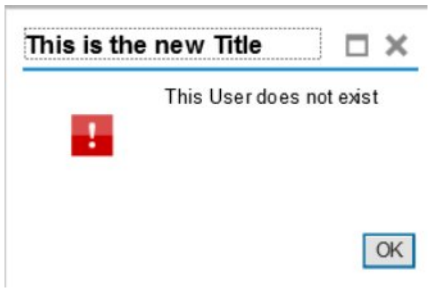


Web Dynpro: Embed View

Component	ZWDC_USER_DISPLAY_COPY
Window	W_POPUP
View to Be Embedded	V_POPUP
Component of View	ZWDC_USER_DISPLAY_COPY
Component Use	----



Final Result:



This is the new Title

This User does not exist