Dynamic OVS Help for Table Fields in Web Dynpro for ABAP

Applies to:

SAP ECC6.0 and above. For more information, visit the Web Dynpro ABAP homepage.

Summary

This document explains the dynamic displaying of the regions based on the country selected. It also explains the case as not displaying any regions for particular country by using the dynamic OVS help.

Author: Suma Vamsee Latha Matta

Company: Intelligroup Asia Pvt. Ltd.

Created on: 04 November 2009

Author Bio

Suma Vamsee Latha Matta is working as a Senior Associate Consultant in Intelligroup. She has an overall experience of 4 years in SAP Netweaver ABAP.

Table of Contents

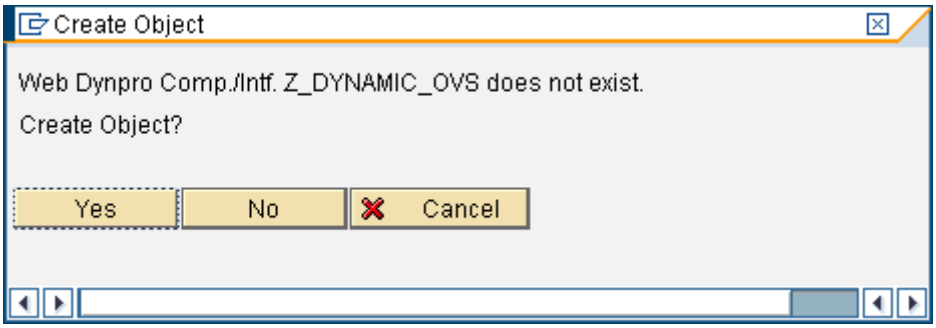
Introduction about Object Value Selector

OVS provide us with the selection screen that can contain multiple input fields to which our selection criteria can be restricted and displaying results values that can be used to populate different input fields.

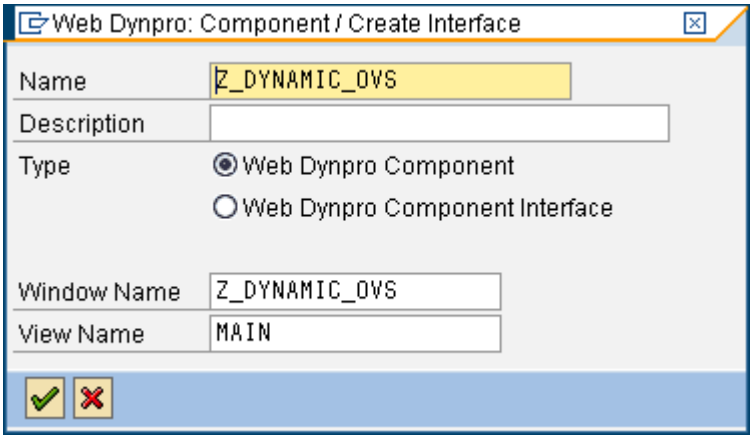
OVS makes use of the PHASE MODEL. We have two components in picture: OVS component and Consumer component. At certain point of time, OVS Component requires some information from consumer component. For this, OVS component fires OVS event. Please check the reference link for information about phases in OVS.

Scenario

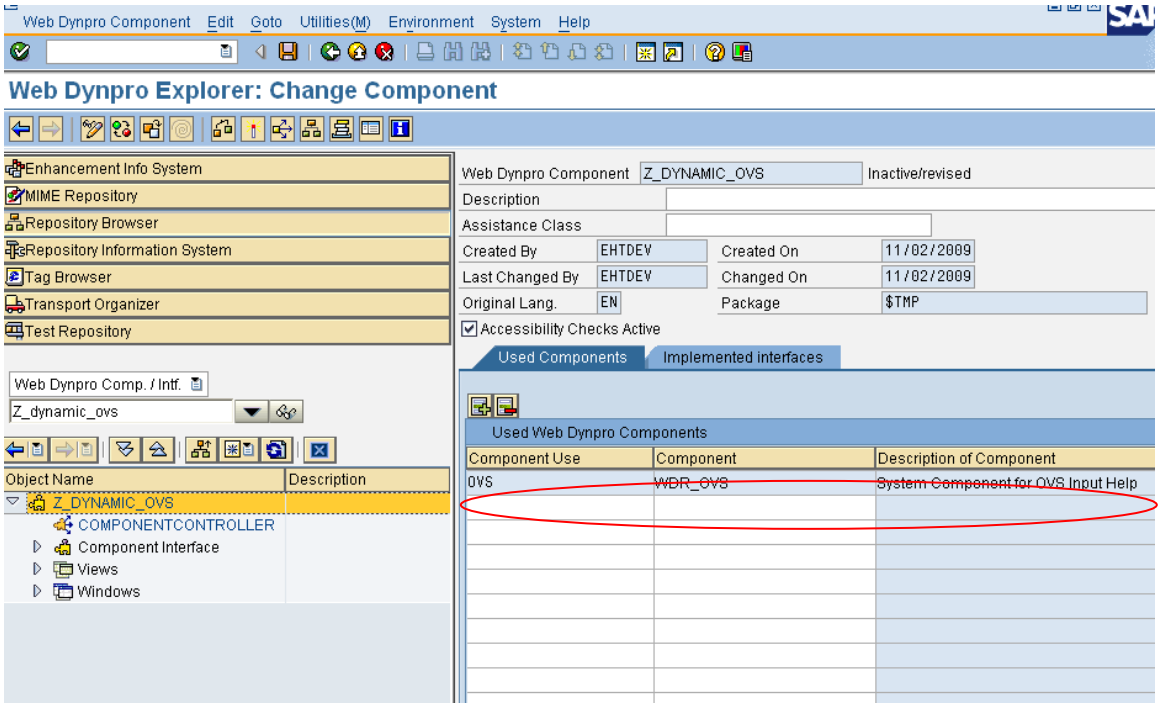
Create one Web Dynpro Component with name Z_DYNAMIC_OVS.



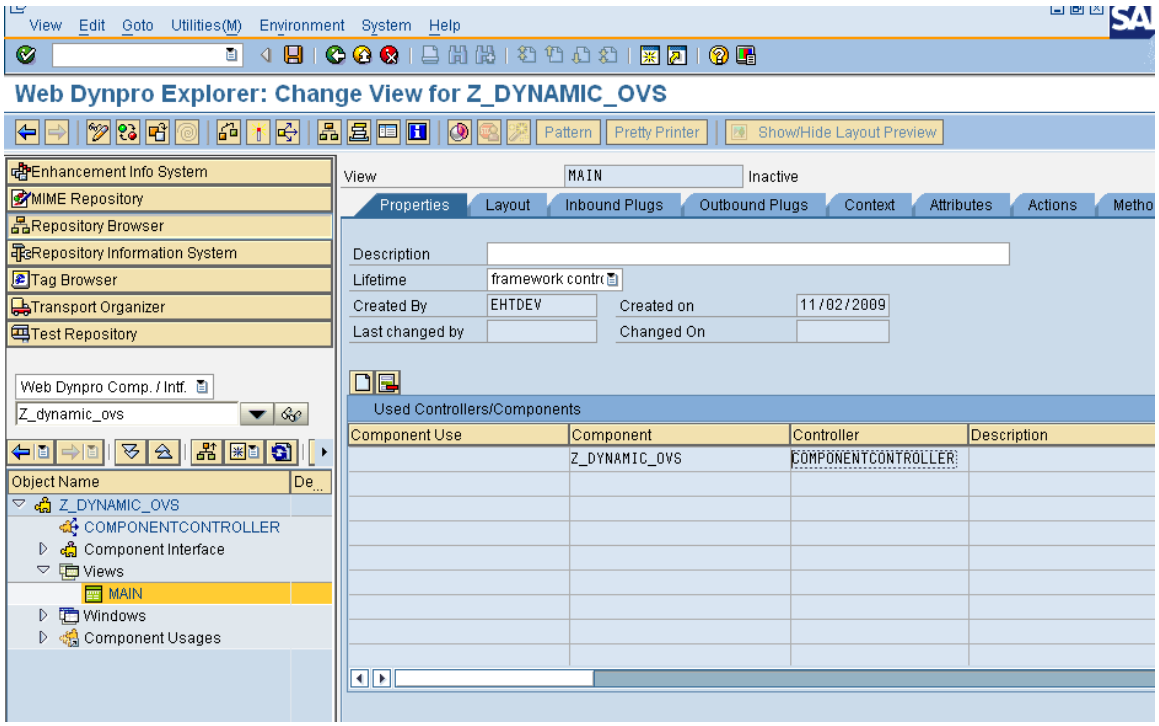
Press Enter.



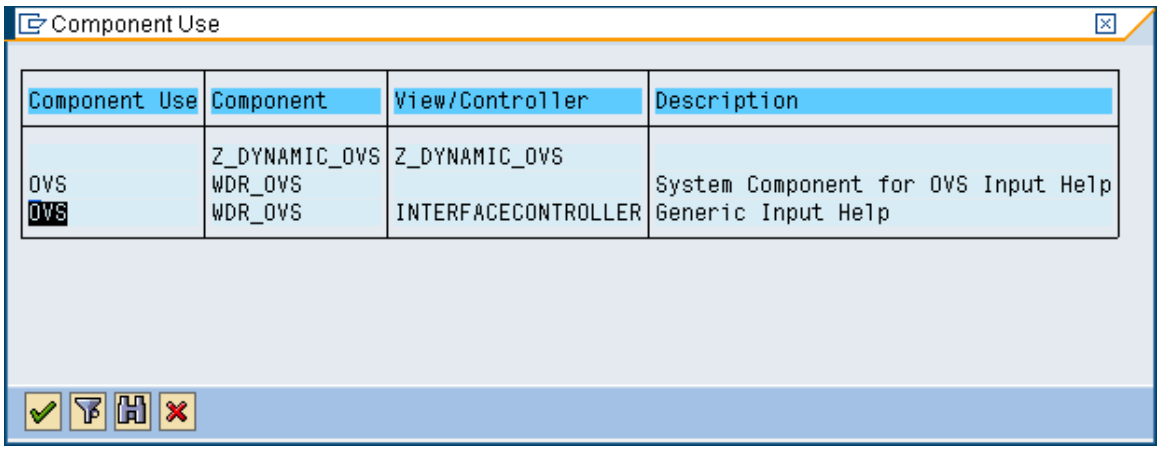
Declare the WDR_OVS Component in the used component list in your WD component as shown below.



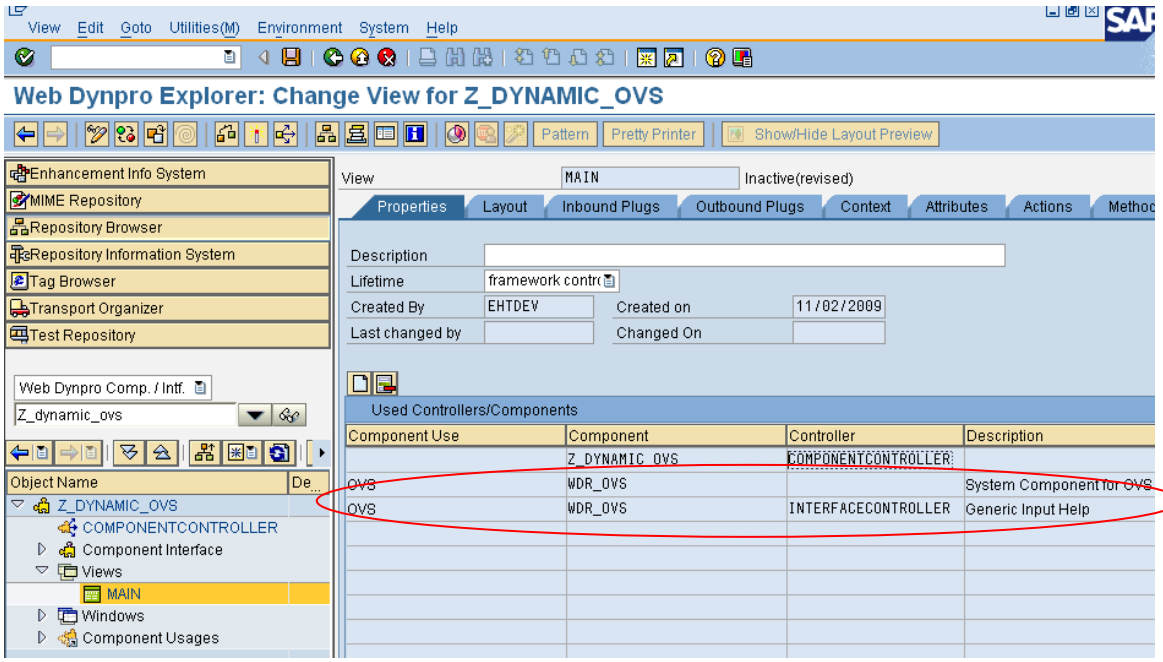
Now go to the View, in the Properties Tab click the Create Controller Usage Button.



It will open a screen with Component Use Entries. There select the Component Use OVS with Interface Controller as shown below. Press Enter.

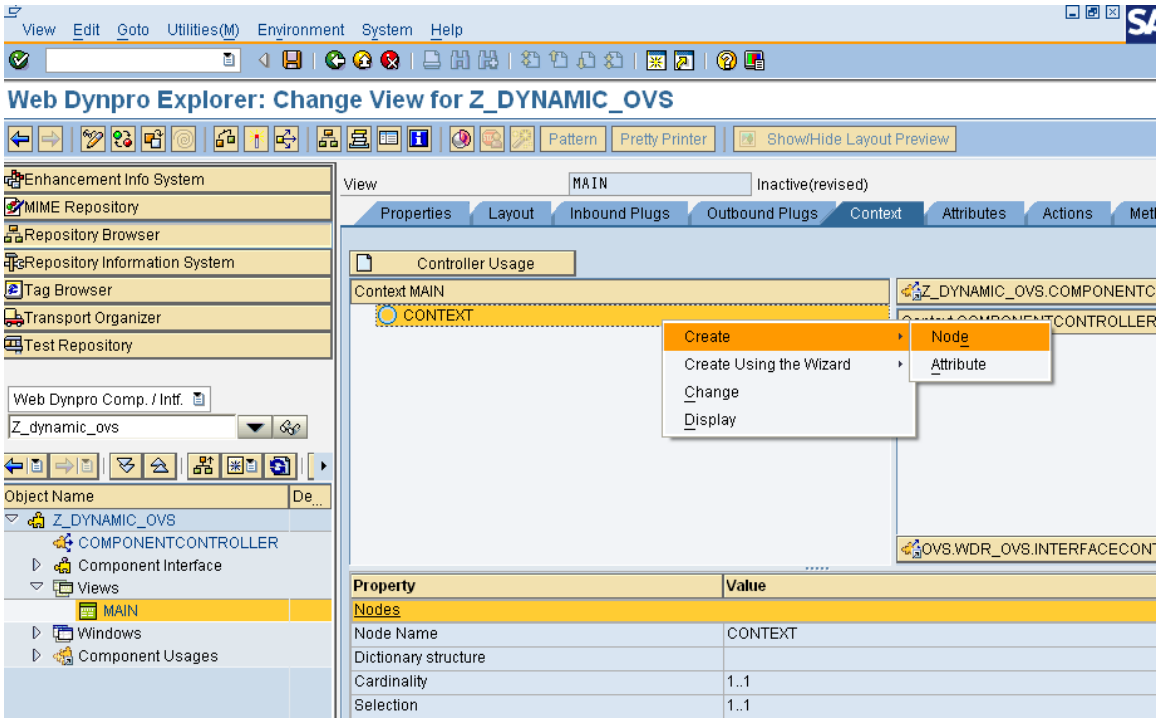


It will display as follows in the View Properties.

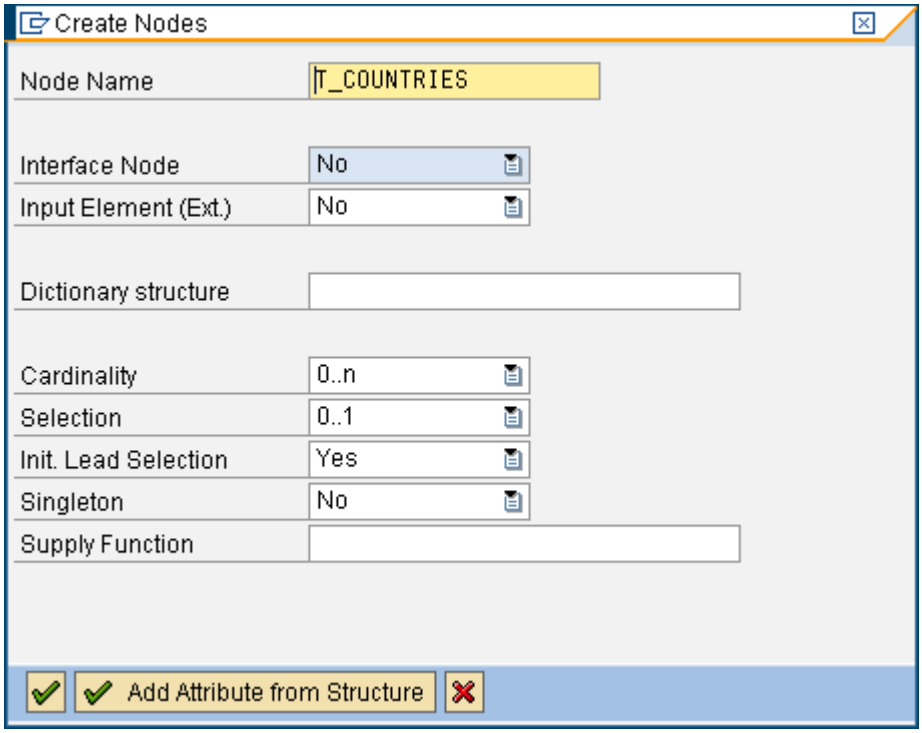


Context

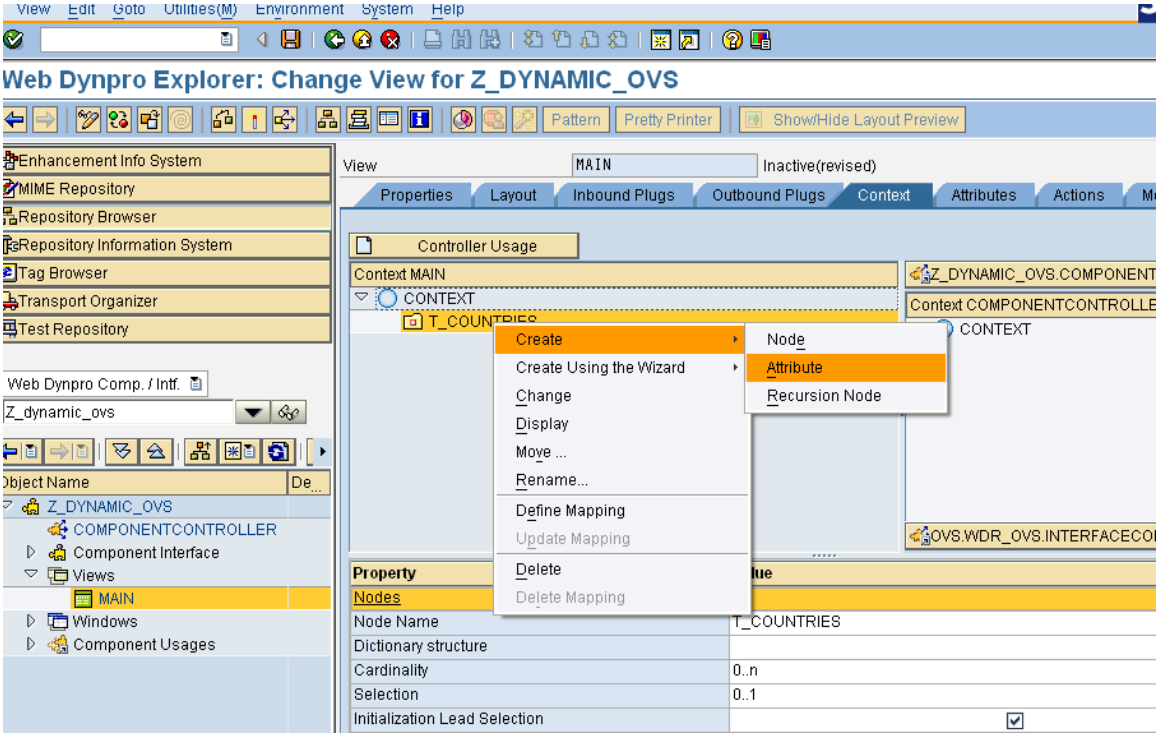
Go to the Context Tab, Right Click the Context and select Create →Node for table.



Give the name for the node as T_COUNTRIES and select 0,n cardinality.

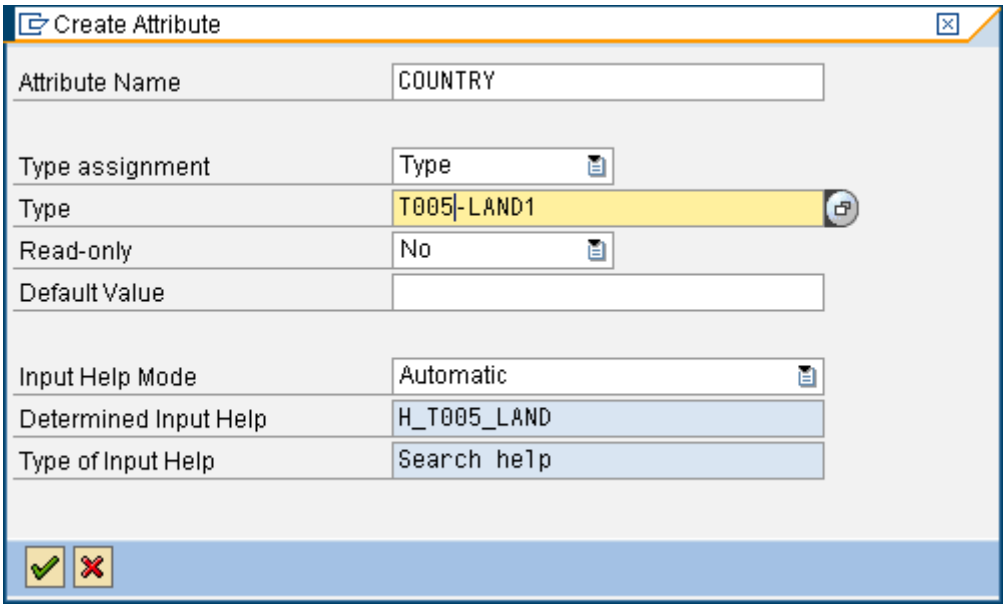


Go to the Context Tab, Right Click the node (T_ COUNTRIES) and select Create → Attribute.



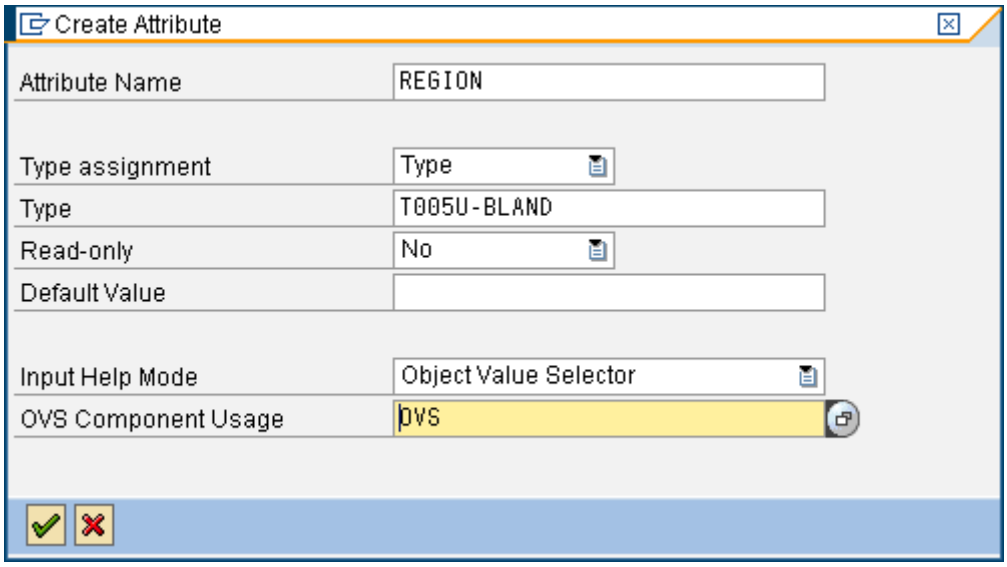
Give the attribute name as 'COUNTRY' with type T005-LAND1.In input help mode select automatic and press enter.

Here select the 'Automatic' input help mode, because given type(T005-LAND1) has already value help at domain level in SAP dictionary. So we are using that help and it will display all countries in the table T005.



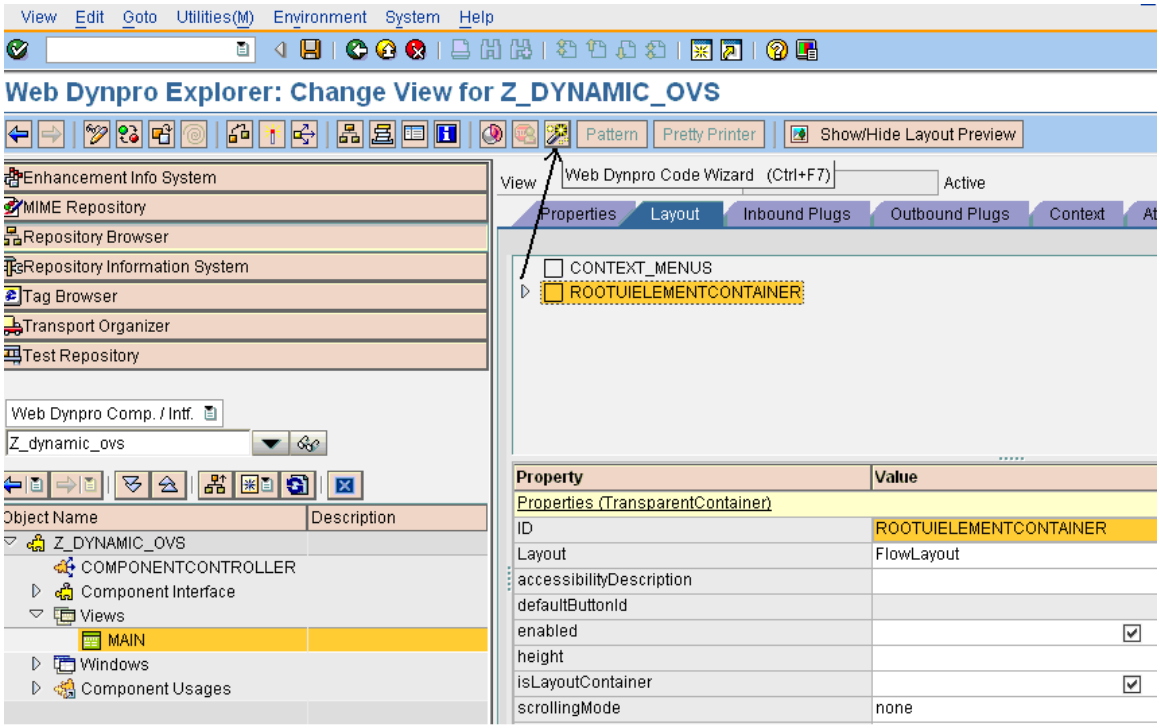
Give the attribute name as 'REGION' with type T005U-BLAND.In input help mode select 'Object Value Selector' and press F4 help for OVS component usage.

Then you will get one popup there select OVS. Recheck the all values as below for attribute 'REGION'.

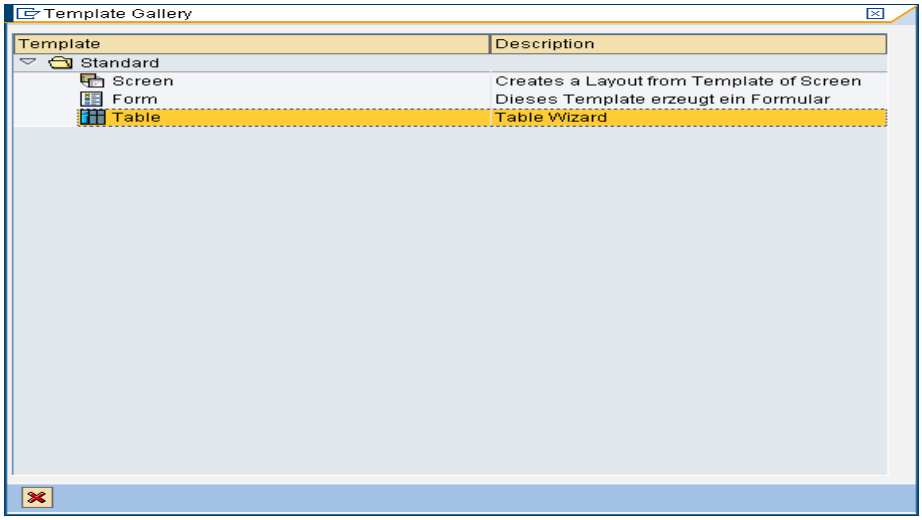


Layout

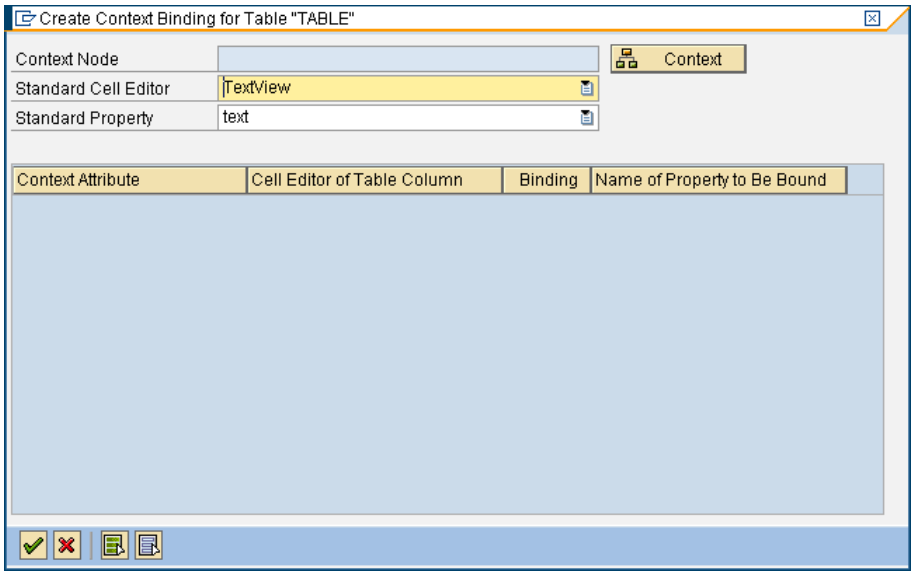
Select ROOTUIELEMENTCONTAINER and then select code wizard in the below screen .



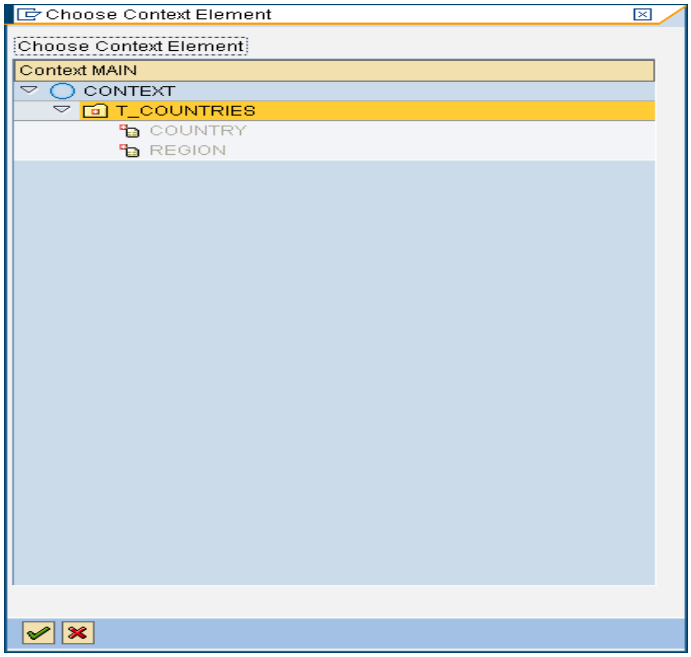
Then it will display the below popup.



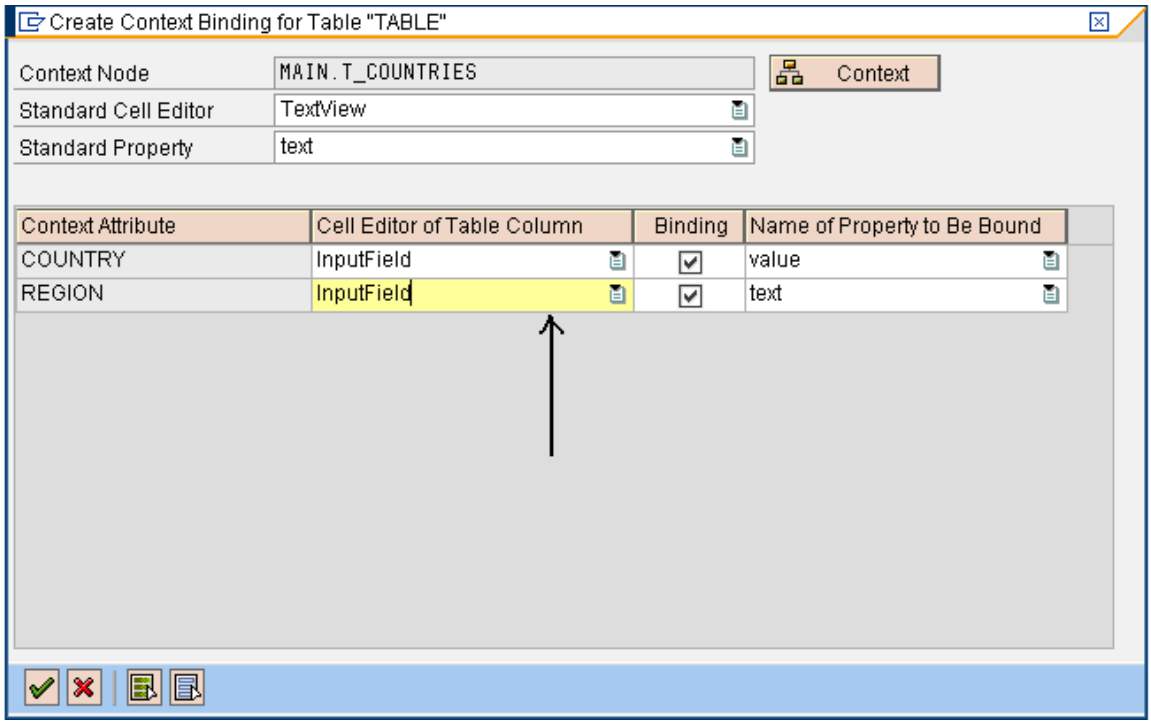
Double click on table, then it will display the below popup



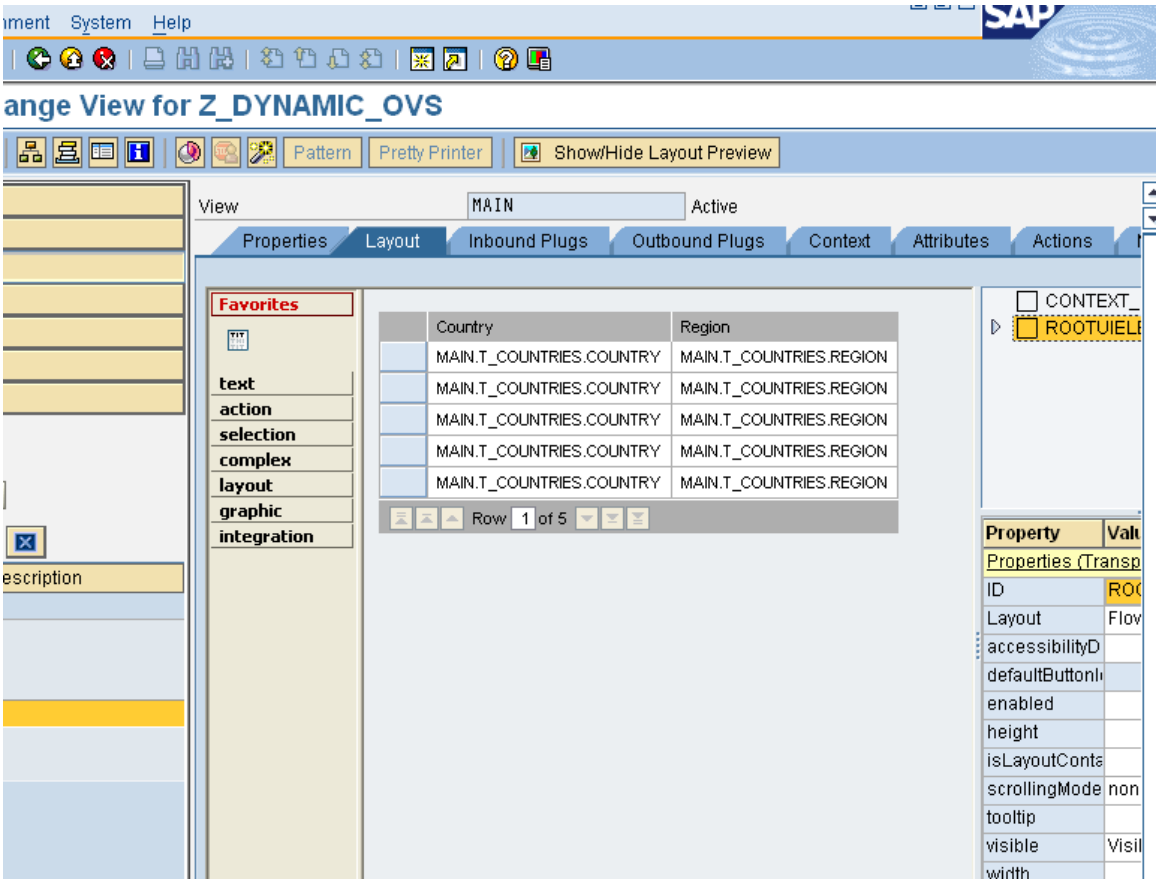
Click the context button and it will display below popup



Select the table node from context and select the 'Input field' in cell editor of table column. And press enter.



It will display table like this in layout.

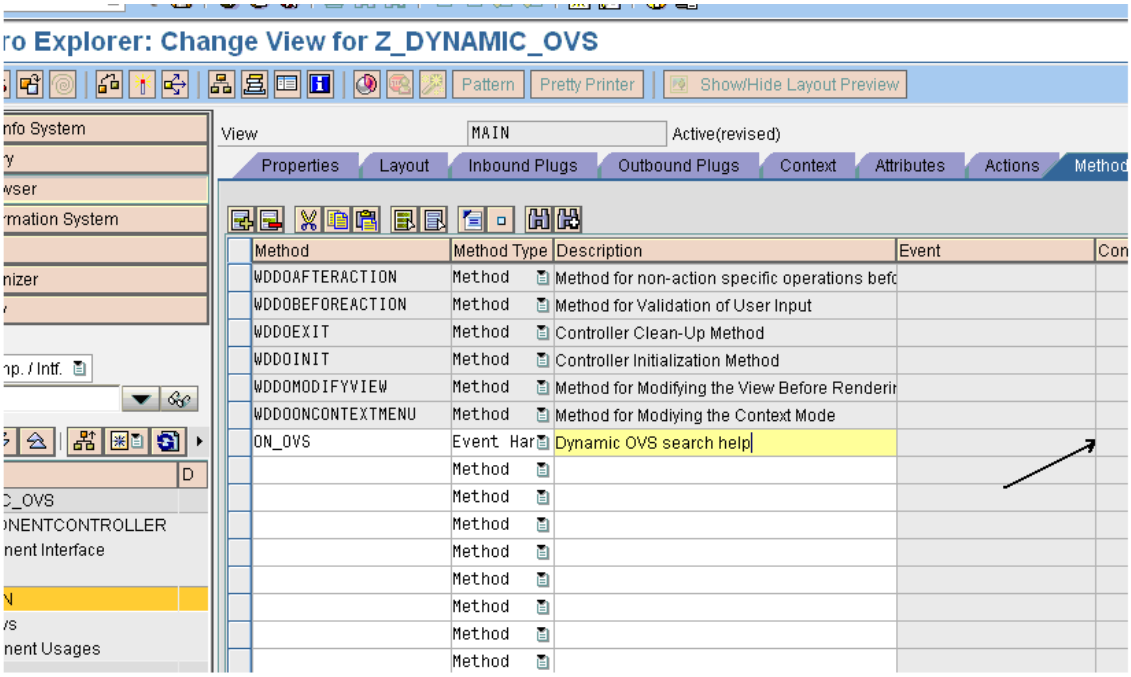


Write below code in the WDDOINIT method of the MAIN view. This code is just appending 10 empty rows as they have to display in editable mode. If we miss the code means table will display but except one row all will display in disable mode.

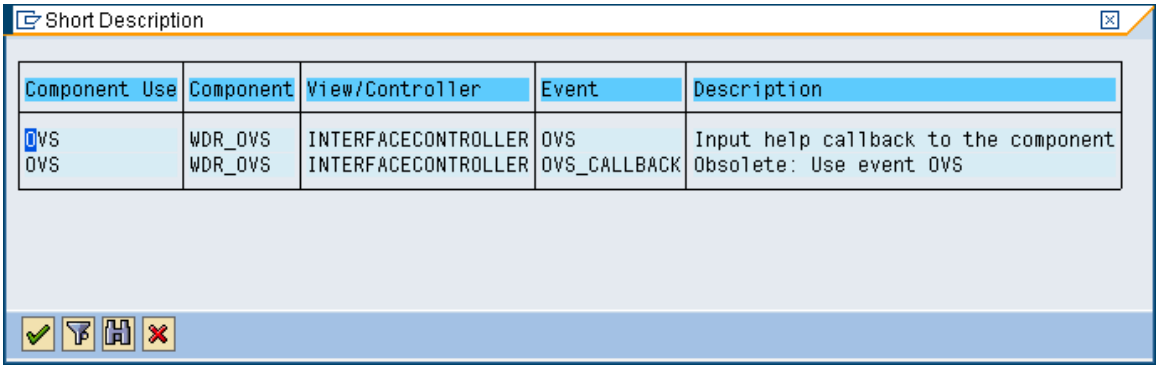
```
method WDDOINIT.
data:
* Internal table for storing empty row
 it_table type if_main=>Elements_t_countries,
* Work area
 wa_table type if_main=>Element_t_countries,
* variable for storing the table node referance
 nd_countries TYPE REF TO if_wd_context_node.

*Appending 10 editable rows
 do 10 times.
  append wa_table to it_table.
 enddo.

* navigate from <CONTEXT> to <N_T005U> via lead selection
  nd_countries = wd_context->get_child_node( name = wd_this->wdctx_t_countries ).
CALL METHOD nd_countries->bind_table
  EXPORTING
    new_items             = it_table.

endmethod.
```

OVS Method

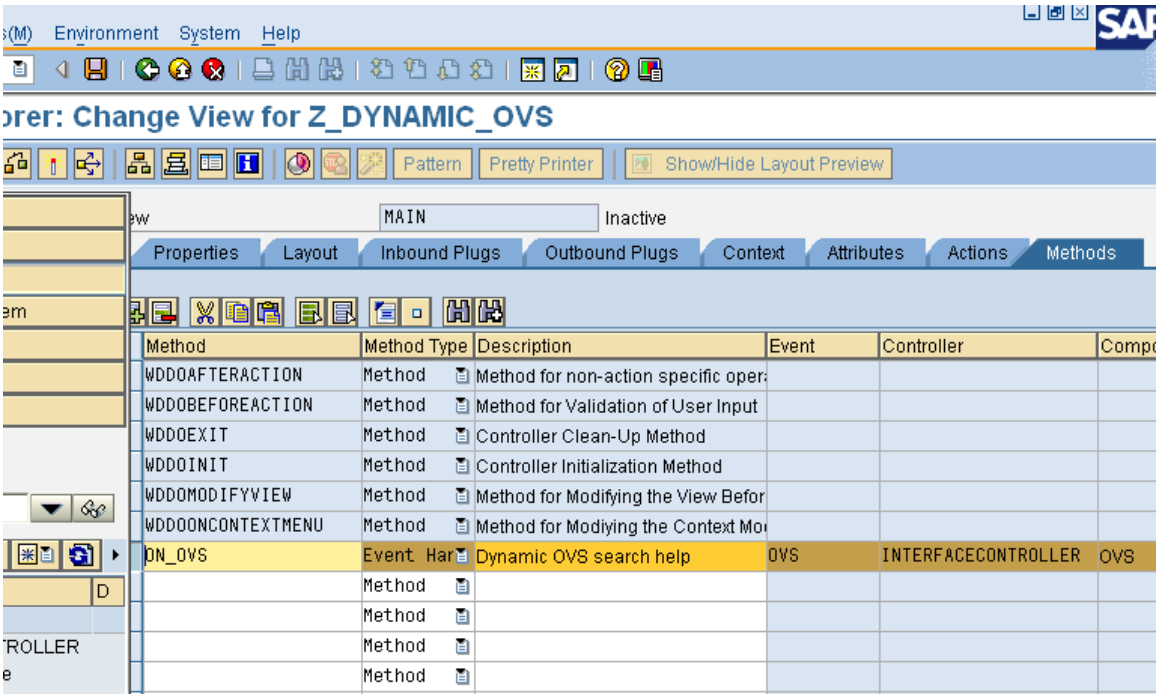
Declare one event handler method with Name ON_OVS in the Method tab of the view. Then Press F4 in the Column 'Event'.



Select the Event OVS as shown below and Press Enter.



It will looks as below screen.



On double click the ON_OVS event, some default code will display. This code has to be use and modify it .Generally code will be in 4 phases

This code contains four phases. Phase0, Phase1, Phase2, Phase3 and their simple descriptions are as below.

Phase0: It is the popup before the values popup to confirm the number of rows, no of columns … etc.

Phase 1: This phase is optional.

In this phase you can set the structure and default values of the search structure. If this phase is omitted, the search fields will not be displayed, but the selection table is displayed directly.

Phase 2: Populate the internal table and bind the table to display as values for selecting.

Phase 3. In this phase the selected particular value is bind to the attribute and display in the view after closing the search help popup.

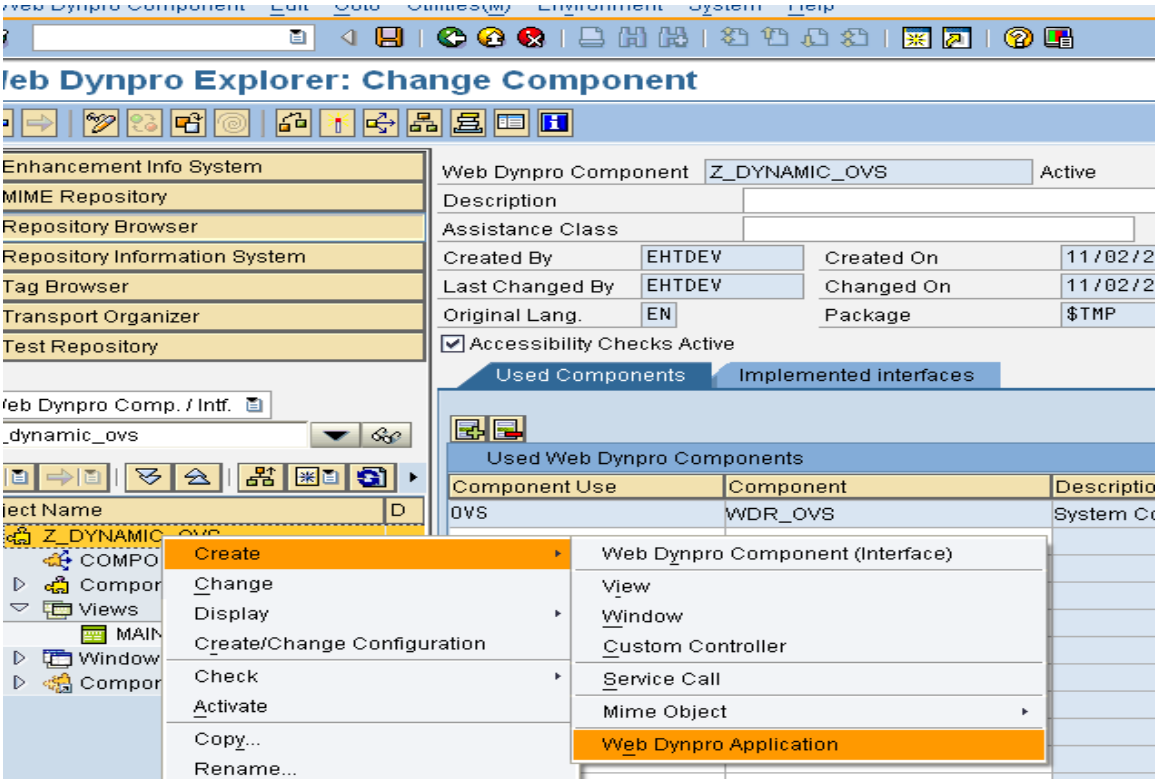
Write the below code in the ON_OVS event.

```
method ON_OVS .
* declare data structures for the fields to be displayed and
* for the table columns of the selection list, if necessary
  types:
    begin of lty_stru_input,
*    add fields for the display of your search input here
      field1 type string,
    end of lty_stru_input.
  types:
    begin of lty_stru_list,
*    add fields for the selection list here
     country type t005-land1,
     bland   type t005u-bland,
     bezei   type t005u-bezei,
    end of lty_stru_list.
 types:
```

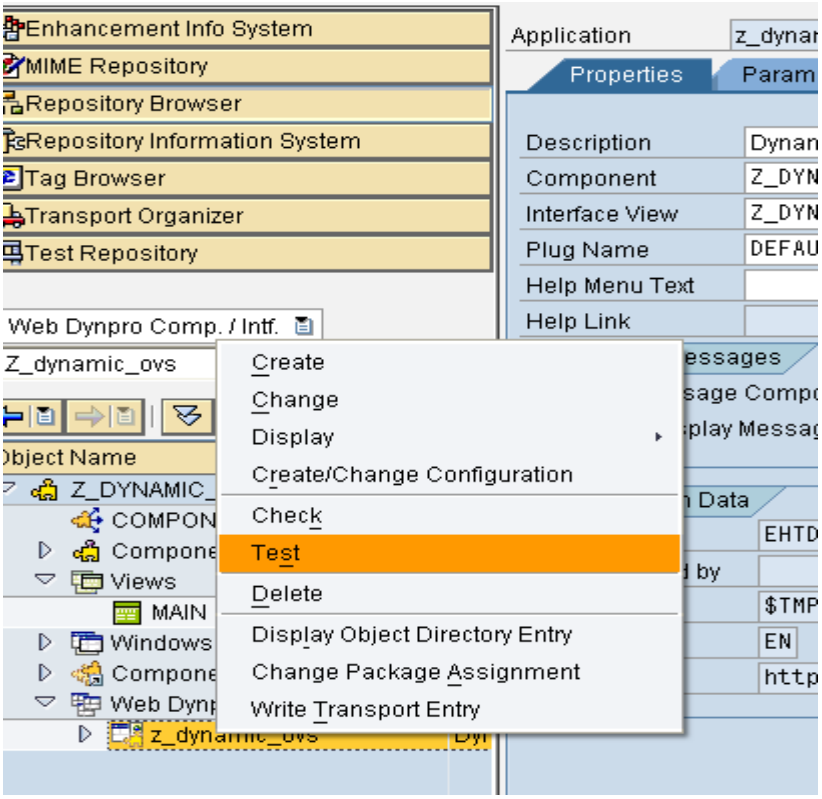
```
    begin of lty_empty,
      text(40) TYPE c,
    end of lty_empty.
  data: ls_search_input  type lty_stru_input,
        lt_select_list   type standard table of lty_stru_list,
        it_empty         type standard table of lty_empty,
        ls_empty         type lty_empty,
        ls_text          type wdr_name_value,
        lt_label_texts   type wdr_name_value_list,
        lt_column_texts  type wdr_name_value_list,
        lv_window_title  type string,
        lv_group_header  type string,
        lv_table_header  type string,
        lv_land1 type t005u-land1.

  field-symbols: <ls_query_params> type lty_stru_input,
                 <ls_selection>    type lty_stru_list.

  case ovs_callback_object->phase_indicator.

    when if_wd_ovs=>co_phase_0.  "configuration phase, may be omitted

      ovs_callback_object->set_configuration(
                label_texts  = lt_label_texts
                column_texts = lt_column_texts
                group_header = lv_group_header
                window_title = lv_window_title
                table_header = lv_table_header
                col_count    = 2
                row_count    = 20 ).

    when if_wd_ovs=>co_phase_1.  "set search structure and defaults

    when if_wd_ovs=>co_phase_2.
      if ovs_callback_object->query_parameters is not bound.
******** TODO exception handling
      endif.
      assign ovs_callback_object->query_parameters->*
                              to <ls_query_params>.
      if not <ls_query_params> is assigned.
******** TODO exception handling
      endif.

* Read the context 'COUNTRY' to get the value of the country
     CALL METHOD ovs_callback_object->context_element->get_attribute
       EXPORTING
         name   = 'COUNTRY'
       IMPORTING
        value  = lv_land1 .
*Dynamic OVS population
    IF lv_land1 = 'AR'.
* no values I am selecting and I am appending the empty row with some text
      ls_empty-text = 'OVS help is N/A for AR Country'.
      append ls_empty to it_empty.
      ovs_callback_object->set_output_table( output = it_empty ).
    ELSE.
```

```
* selecting the all regions corresponding to the selected country
      select land1
             bland
             bezei
           into table lt_select_list
           from  t005u
           where spras = sy-langu
             and land1 = lv_land1.
    ovs_callback_object->set_output_table( output = lt_select_list ).
   ENDIF.

   when if_wd_ovs=>co_phase_3.
*   apply result

   CALL METHOD ovs_callback_object->context_element->get_attribute
     EXPORTING
      name   = 'COUNTRY'
    IMPORTING
      value  = lv_land1 .
* Dynamic OVS value assignment
  IF lv_land1 <> 'AR'.
     assign ovs_callback_object->selection->* to <ls_selection>.
     if <ls_selection> is assigned.
       ovs_callback_object->context_element->set_attribute(
                              name  = `REGION`
                              value = <ls_selection>-bland ).

     endif.
   endif.
 endcase.
endmethod.
```

Create Web Dynpro application as below and save it.

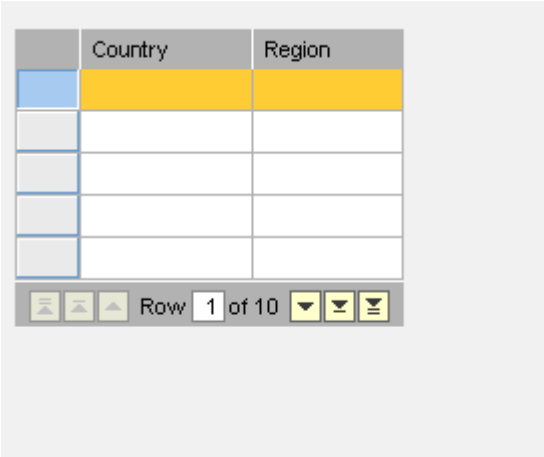


Test the Web Dynpro application as below



Output

Output is like this

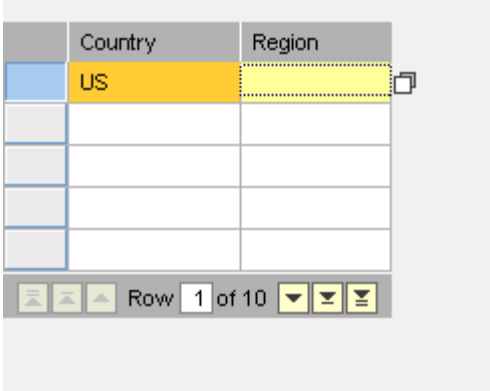


Now I am explaining the dynamic OVS help cases as below.

Case 1: Based on the country entered corresponding regions only display in search help

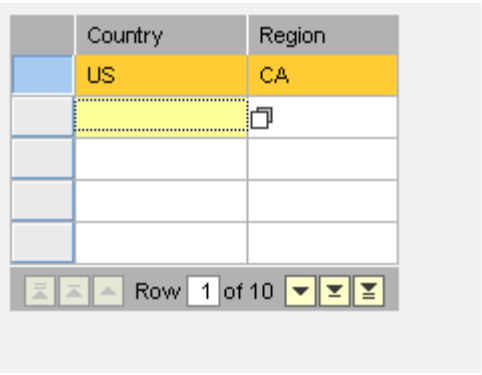
Case 2: For particular country ‘AR’, I want to display a text 'OVS help is N/A for AR Country' instead of regions it has .

Case1 test procedure: For example select the ‘US’ country and then select OVS search help for region .

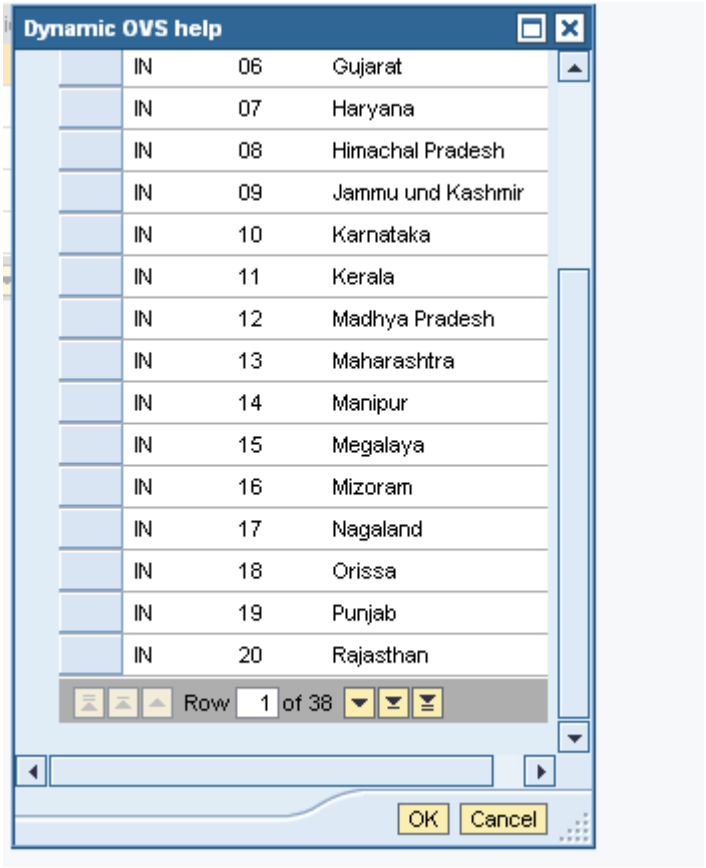


F4 help for the region is as below popup screen with total 56 regions for ‘US’ country.

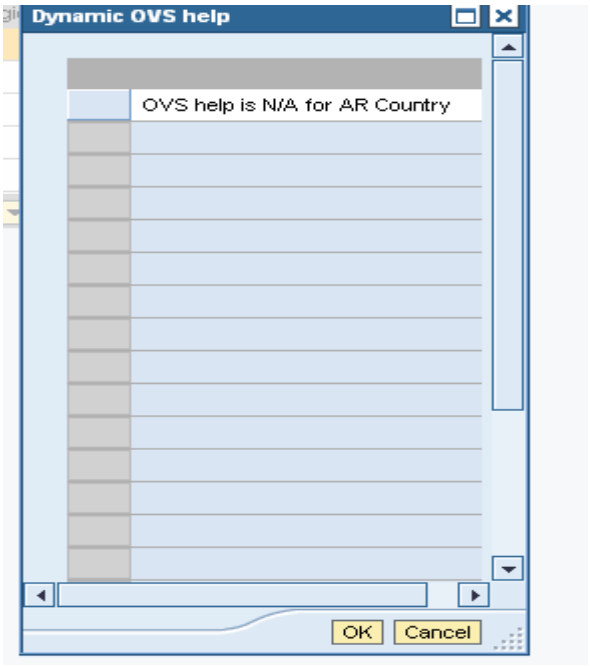
Select one region and press OK. Now o/p is like as below.



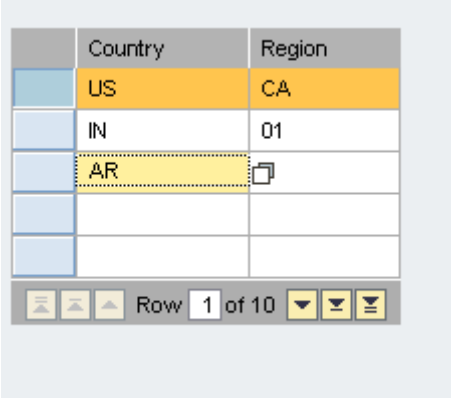
Similarly select ‘IN’ country, it will get 38 regions for India



Case 2 test procedure: Now select the country ‘AR’. And press F4 for the region then below popup screen will display.



Now select that text and press ‘OK’. It will not populate any value. O/p screen is as below with no value in region. Actually for ‘AR’ country some regions are there, but here we are dynamically not populating the regions. And also it is not assigning any value through F4 help for this particular ‘AR’ country.



Related Content

http://help.sap.com/saphelp_erp2005/helpdata/en/30/d7fa41c915da6fe10000000a1550b0/content.htm

For more information, visit the Web Dynpro ABAP homepage

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.