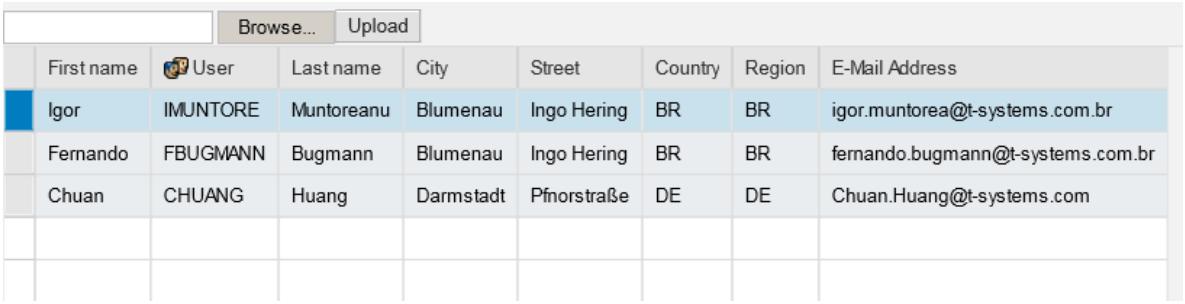
Version 1.0
Date 11.07.2016
Status In Development
Number 4
Initial version by Igor Muntoreanu

Exercise Review:

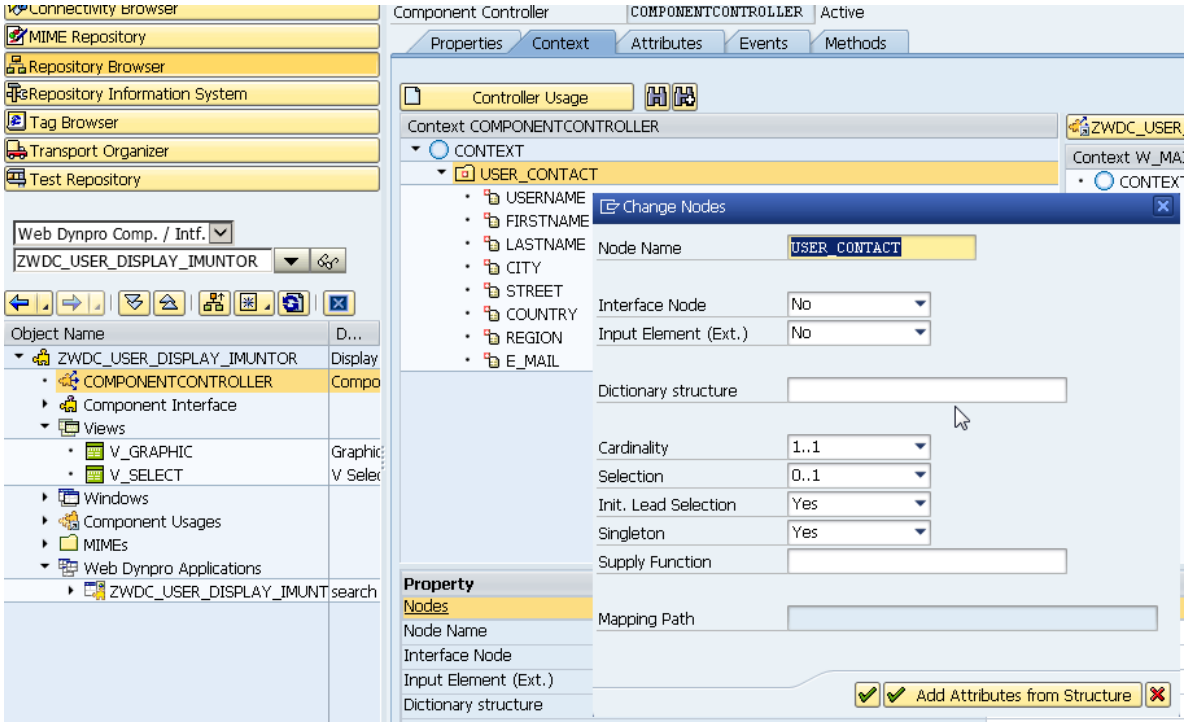
It will be an Application with excel upload functionality:



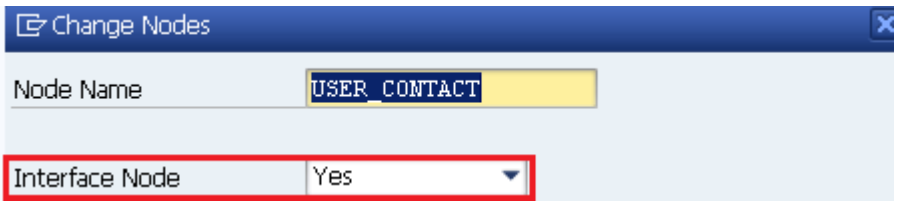
Practical Guide:

1) Go to our ALV WD Component and navigate to the NODE in the COMPONENTCONTROLLER:

Change the node by right clicking on this:

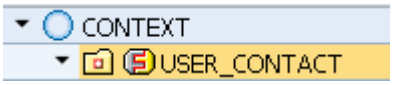


Set the Interface NODE as “Yes”:

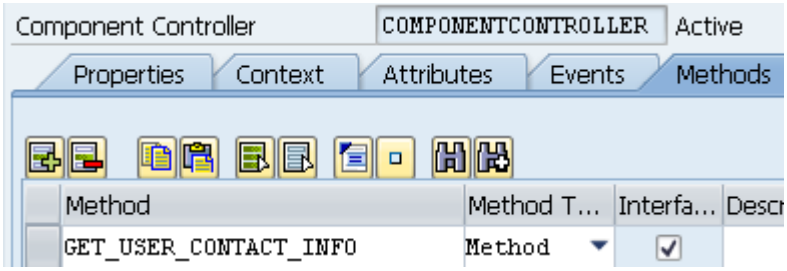


Save and activate

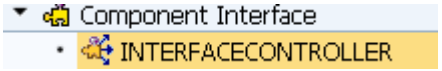
Your node will look like this



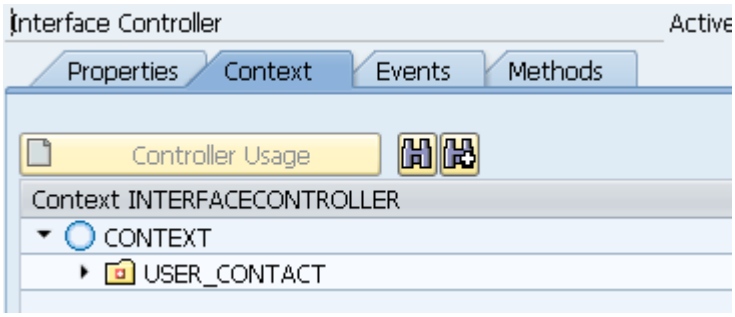
Go to the methods in and mark the “Interface” column of the method GET_USER_CONTACT_INFO

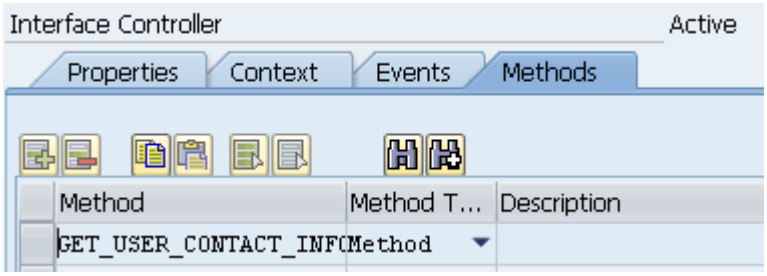


Navigate to the INTERFACECONTROLLER:



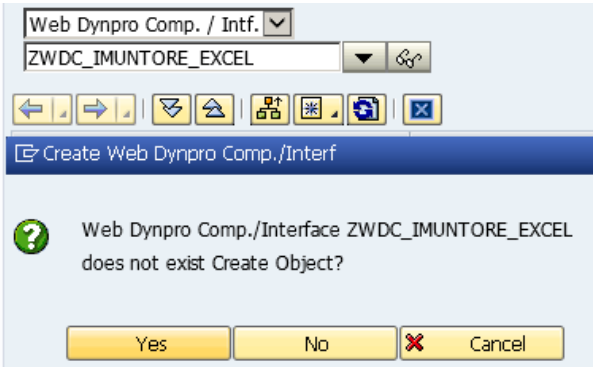
Now we have the context and the method there:

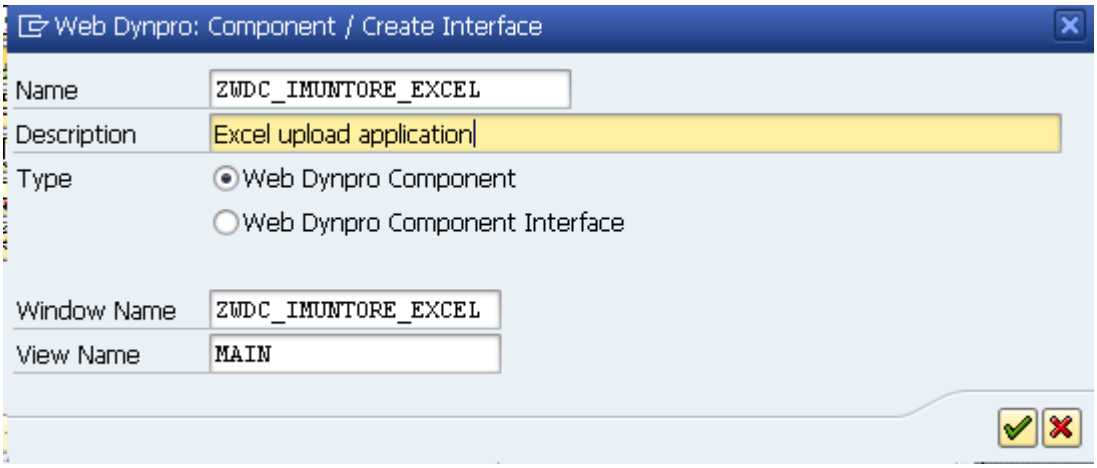




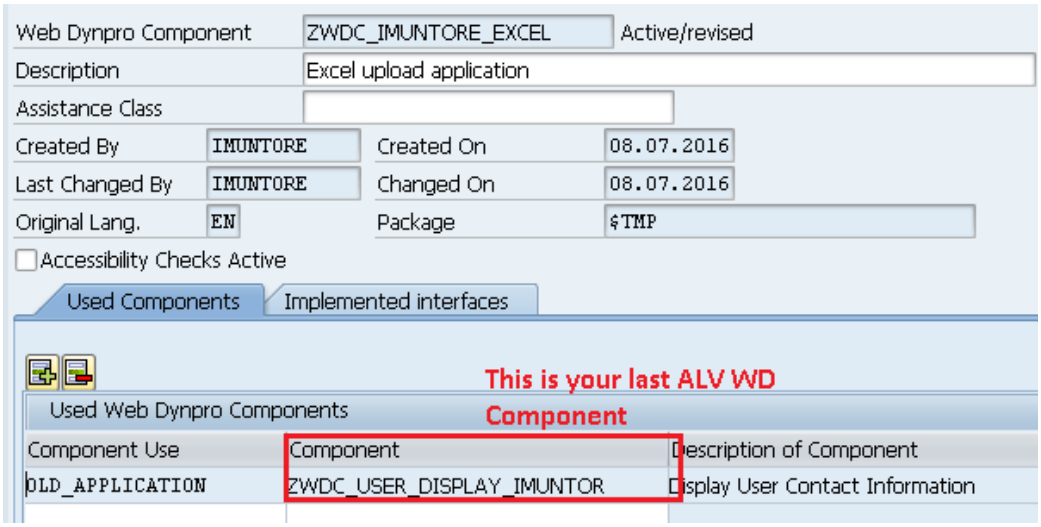
It means that we can use these elements in another WD Components.

2) Create a new Component:

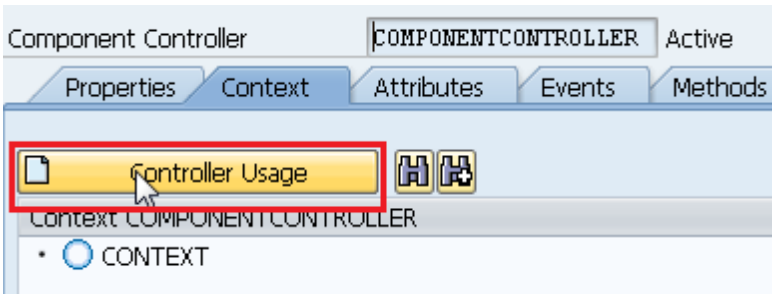


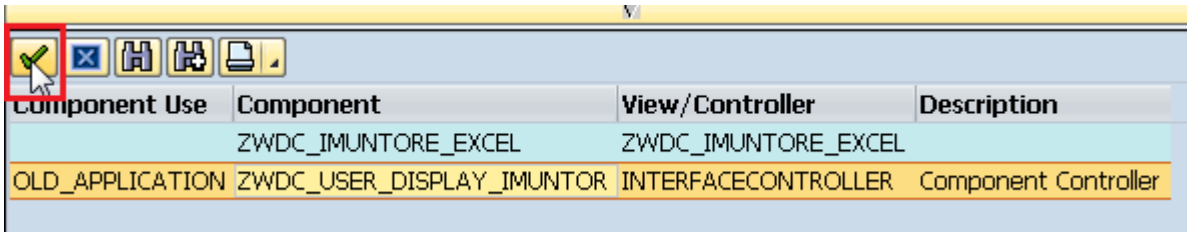


3) Go to the WebDynpro Component on the Used Components, put:

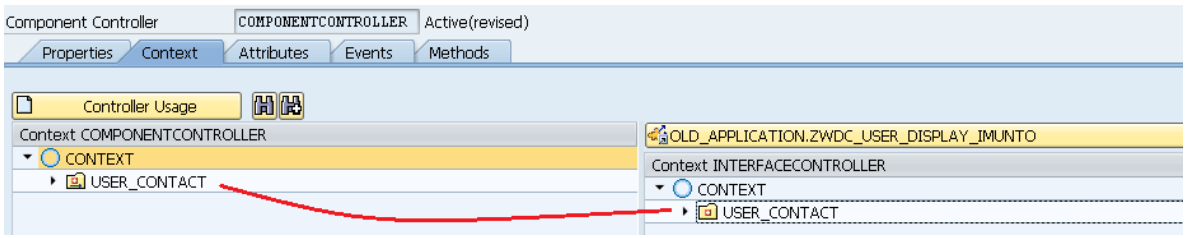


4) Go to the COMPONENTCONTROLLER:

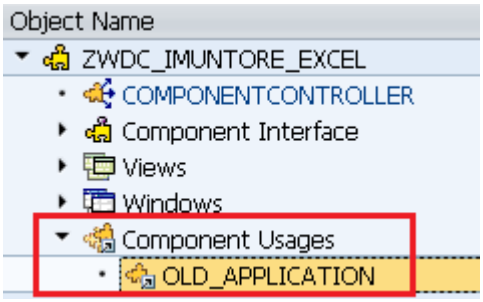




Bind the NODE:

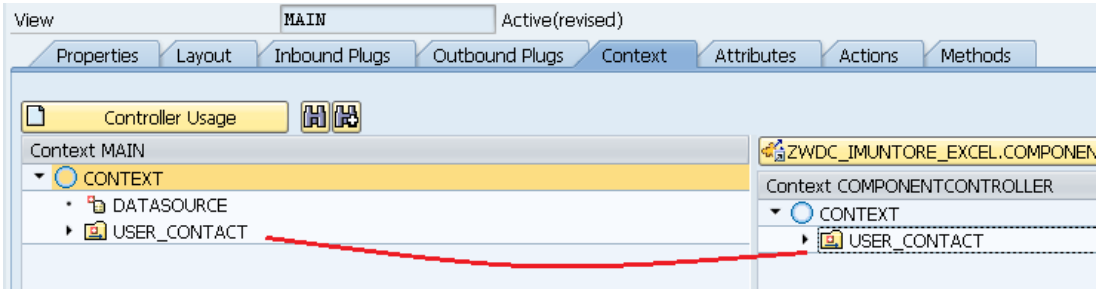


5) The Component Usage will also in the Tree appears:

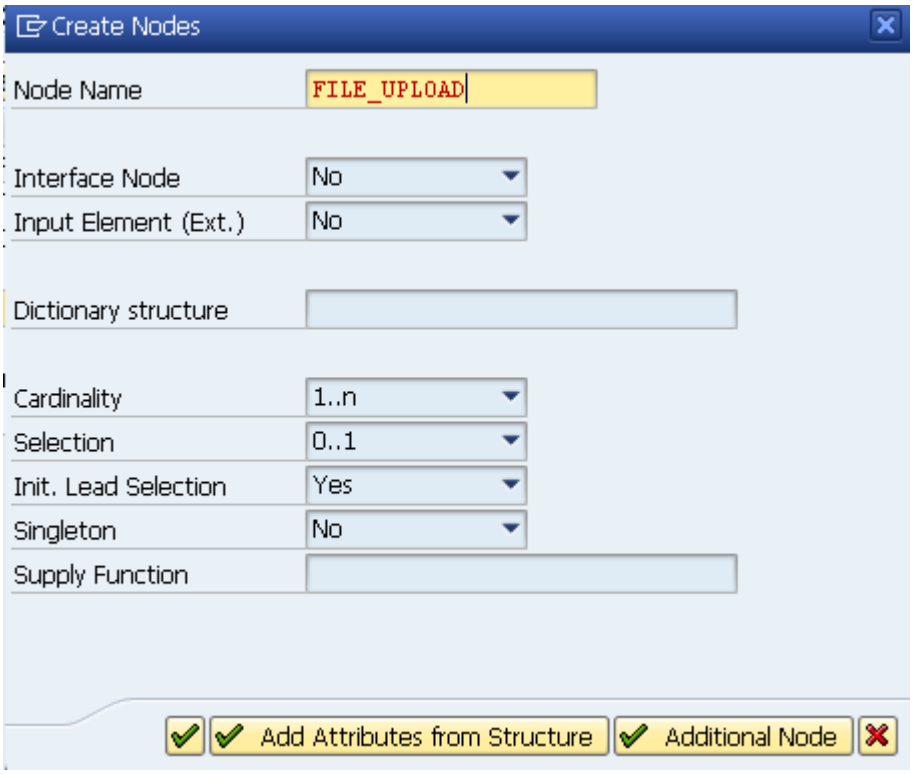


6) Go to the MAIN View

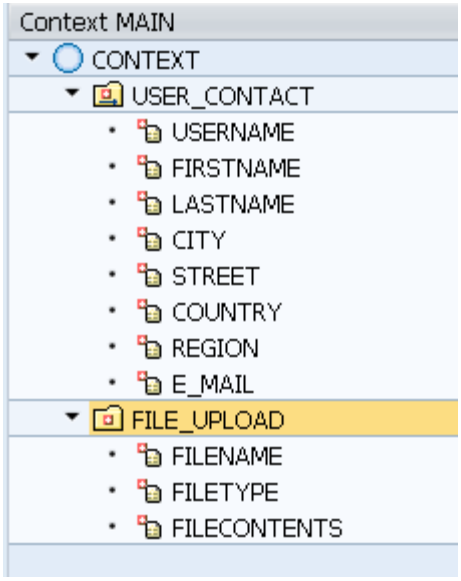
Bind the NODE USER_CONTACT from the CONTROLLER context to the View context:



Create a new node:

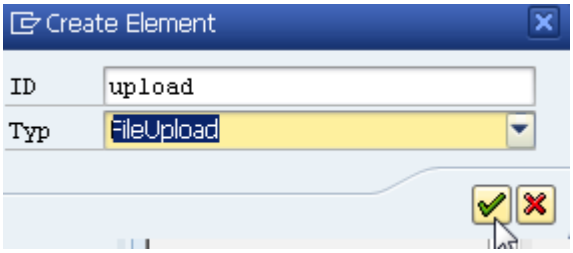


Inside the node create 3 attributes: FILENAME (string), FILETYPE (string), FILECONTENTS (xstring).
You context should look like this:

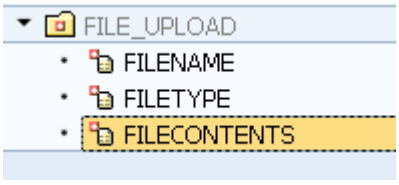


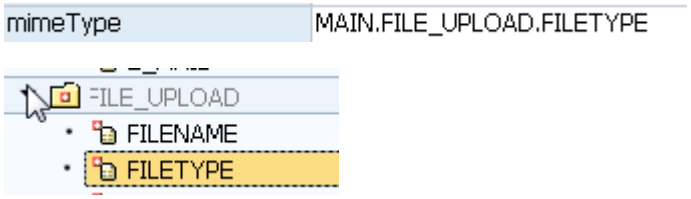
Save and Activate.

Navigate to the Layout tab and create a new element in the Root:



Make its properties exactly like this:

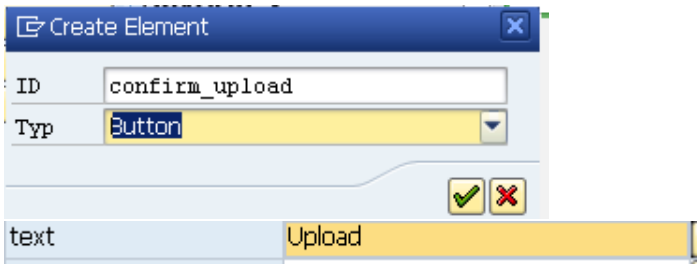




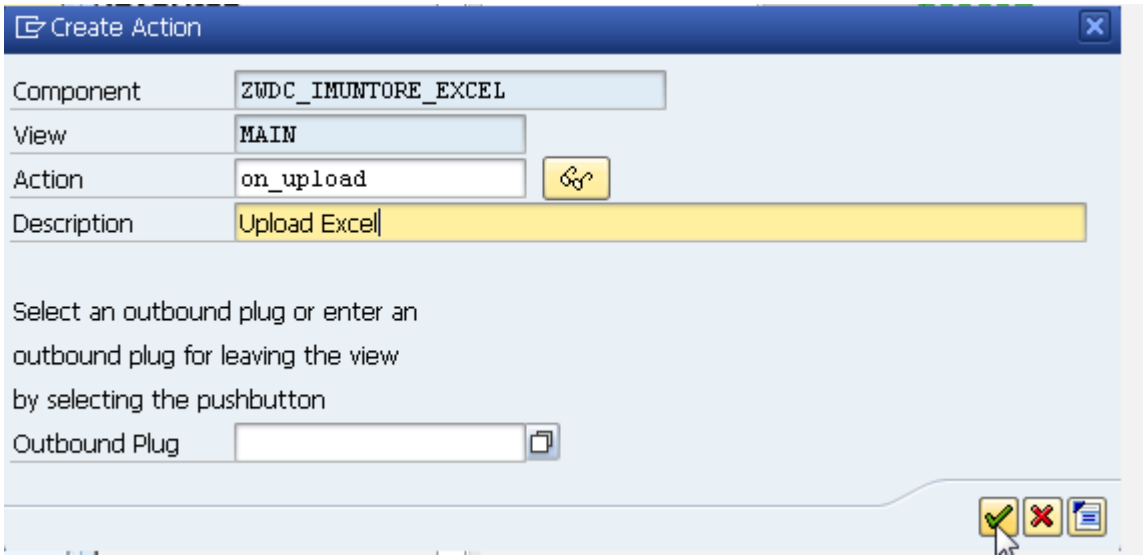
Result:

Property	Value	Binding
Properties (FileUpload)		
ID	UPLOAD	
activateAccessKey	☐	
contextMenuBehaviour	Inherit	
contextMenuId		
data	MAIN.FILE_UPLOAD.FILECONTENTS	
enabled	☑	
explanation		
fileName	MAIN.FILE_UPLOAD.FILENAME	
mimeType	MAIN.FILE_UPLOAD.FILETYPE	
state	Normal Item	
textDirection	Inherit	
tooltip		
virusScanProfile		
visible	Visible	
width		
Layout Data (FlowData)		
cellDesign	padless	
vGutter	None	

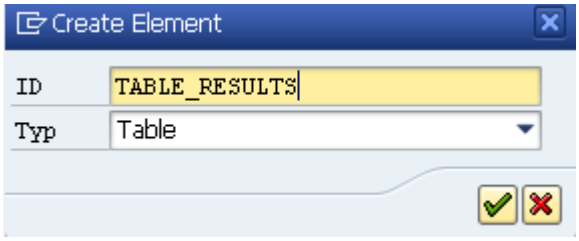
Create also a Button:



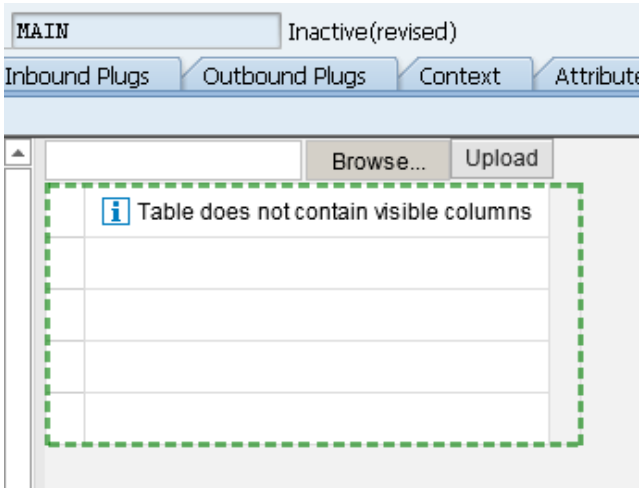
Create an Action to the Button:



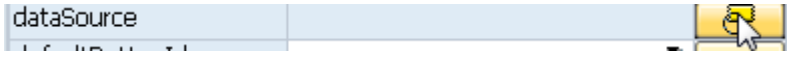
Create another element:

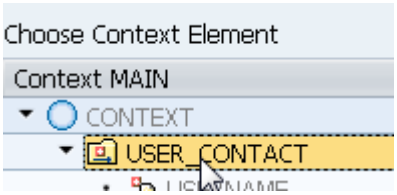


Your screen will look something like this:

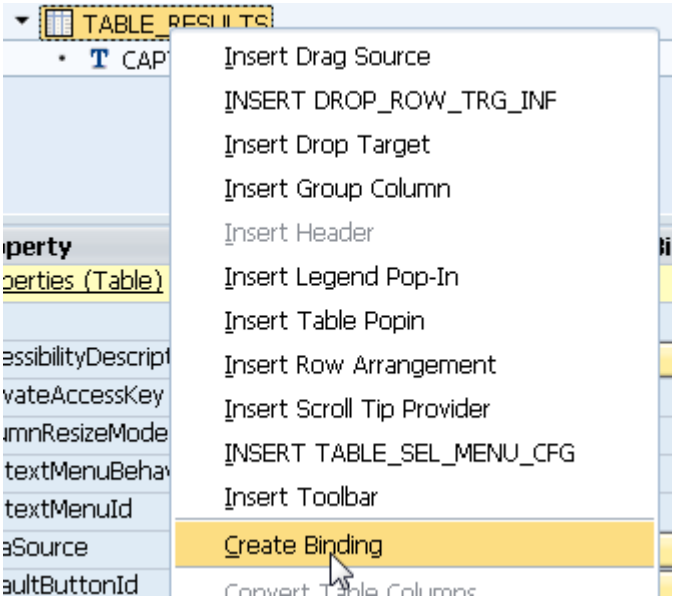


Create a datasource to the table

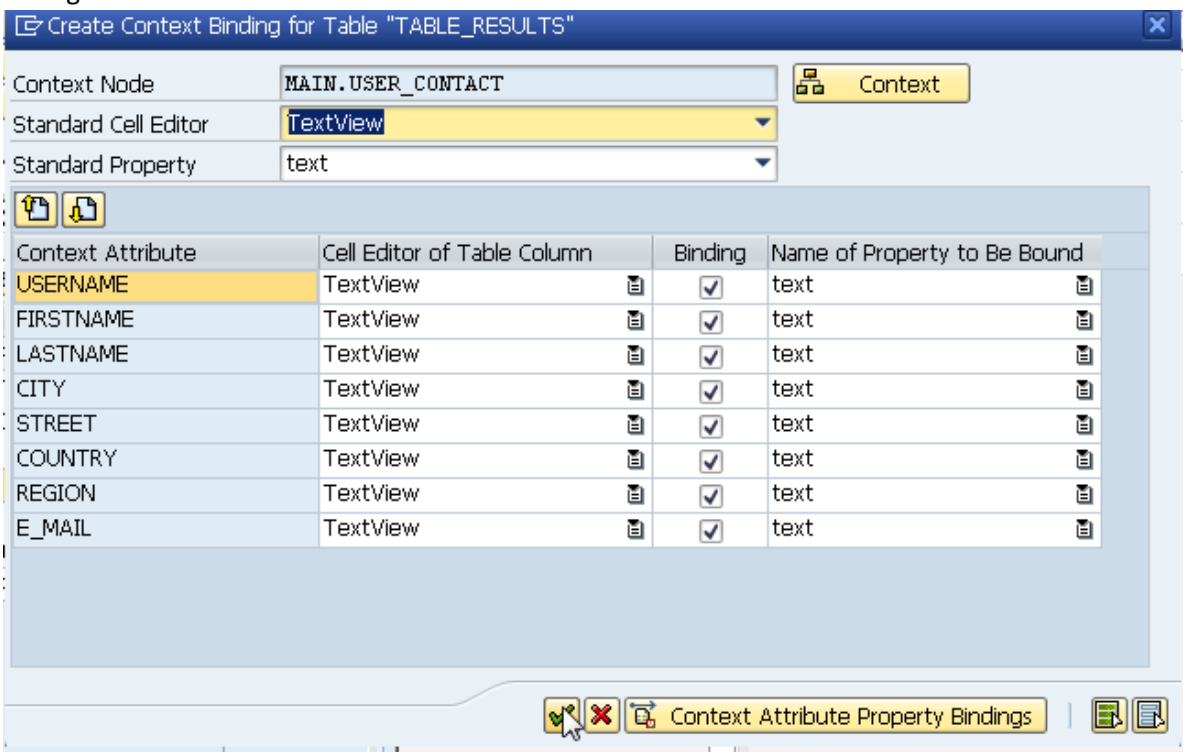




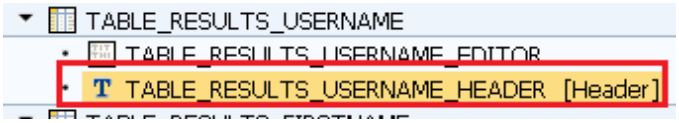
Create the Binding:

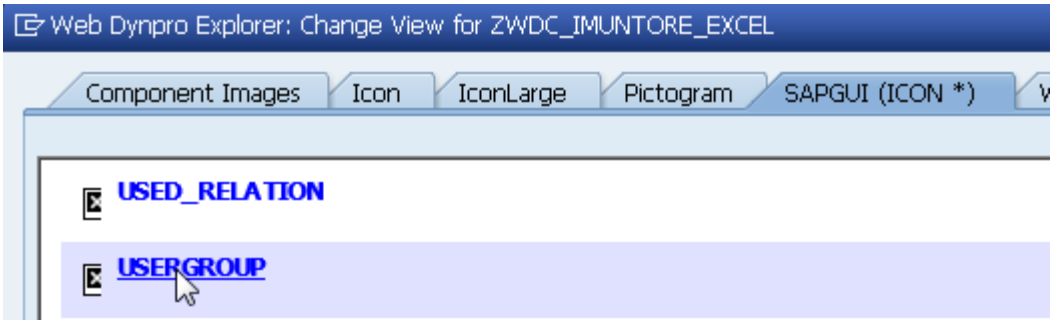


Configure like shown below:

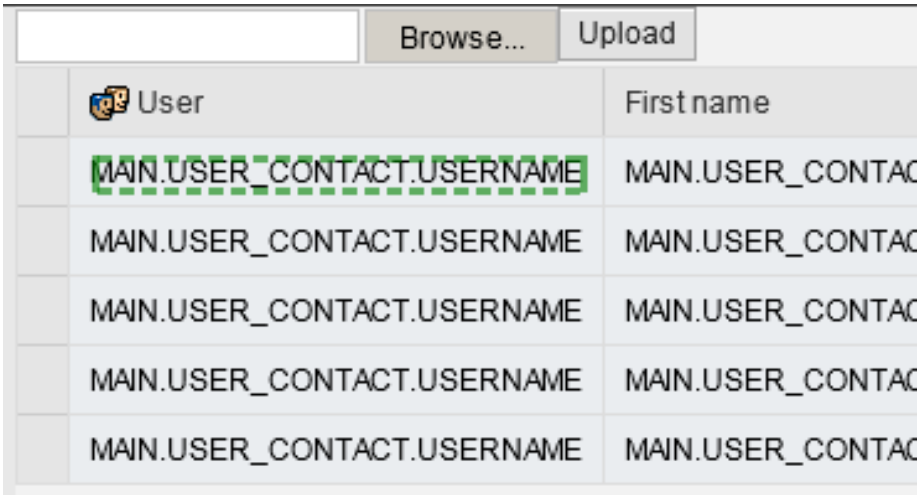


Select the following element and do:

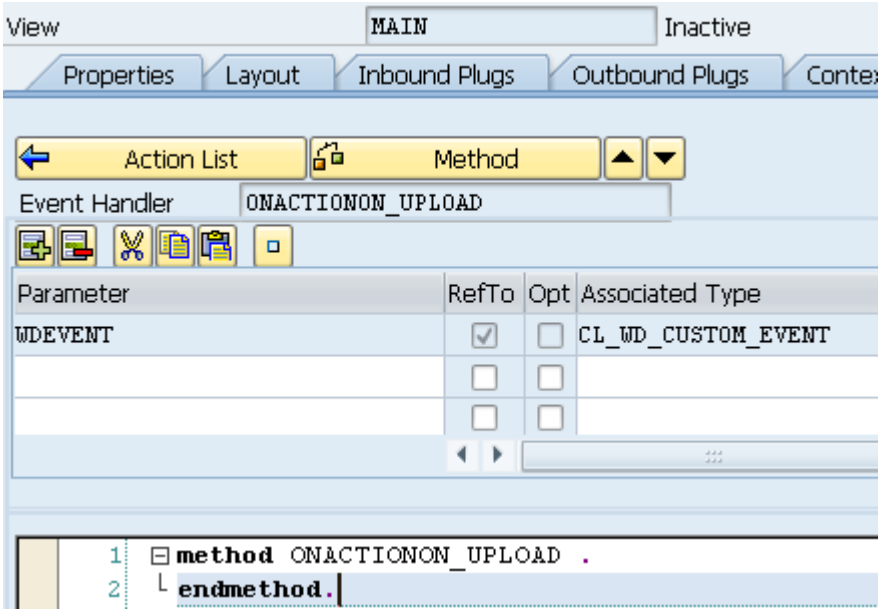




Now your column will be a little more friendly:



7) Time to Code!



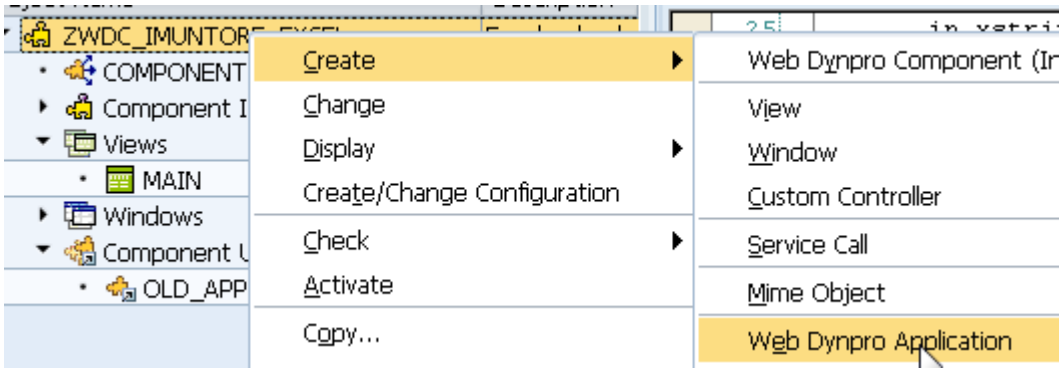
Copy and Past the following:
It will be explained later by debugging:

```
METHOD onactionon_upload .

  DATA lo_nd_file_upload TYPE REF TO if_wd_context_node.
  DATA lo_el_file_upload TYPE REF TO if_wd_context_element.
  DATA ls_file_upload TYPE wd_this->element_file_upload.

  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.
```

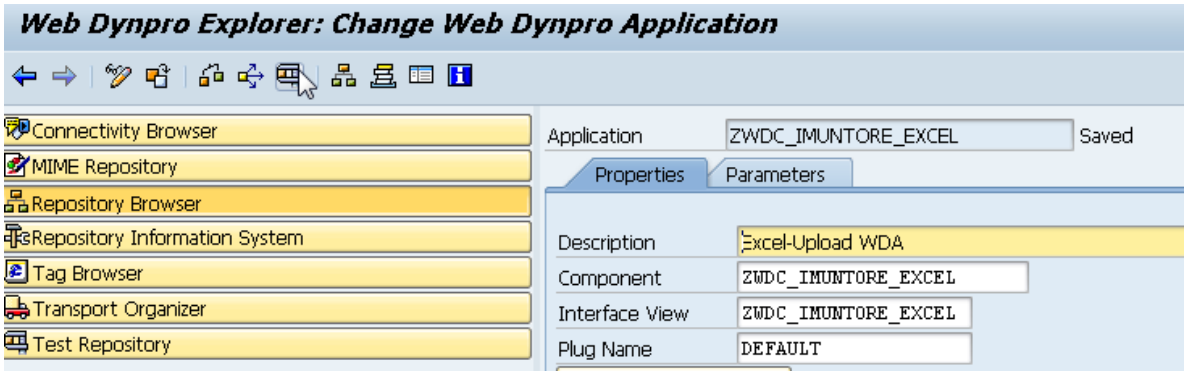
```
  DATA lt_user_contact TYPE TABLE OF wd_this->element_user_contact.

  lo_nd_file_upload = wd_context->get_child_node( name = wd_this-
>wdctx_file_upload ).
  lo_el_file_upload = lo_nd_file_upload->get_element( ).
* Get all declared attributes
  lo_el_file_upload->get_static_attributes(
    IMPORTING
      static_attributes = ls_file_upload ).

* Create object of class to read .xlsx file contents
  DATA(lref_excel) = NEW cl_fdt_xl_spreadsheet( document_name = ls_file_u
pload-filename
                                                xdocument     = ls_file_u
pload-filecontents ).

  lref_excel-
>if_fdt_doc_spreadsheet~get_worksheet_names( IMPORTING worksheet_names =
DATA(lt_worksheets) ).
* Condition to check whether .xlsx file has any active worksheets
  READ TABLE lt_worksheets INDEX 1 INTO DATA(lv_name).
  IF sy-subrc = 0.
* Get reference of .xlsx file contents in the active worksheet
    DATA(lref_data) = lref_excel-
>if_fdt_doc_spreadsheet~get_itab_from_worksheet( lv_name ).
  ENDIF.

  FIELD-SYMBOLS: <fs_table> TYPE ANY TABLE.
* Fetch all records in the active worksheet
  ASSIGN lref_data->* TO <fs_table>.
  IF <fs_table> IS ASSIGNED.

    LOOP AT <fs_table> ASSIGNING FIELD-SYMBOL(<fs_l_table>).

      ASSIGN COMPONENT 'A' OF STRUCTURE <fs_l_table> TO FIELD-
SYMBOL(<fs_value>).
      IF <fs_value> IS ASSIGNED.
        ls_user_contact-username = <fs_value>.
      ENDIF.
      UNASSIGN <fs_value>.
      ASSIGN COMPONENT 'B' OF STRUCTURE <fs_l_table> TO <fs_value>.
      IF <fs_value> IS ASSIGNED .
        ls_user_contact-firstname = <fs_value>.
      ENDIF.
      UNASSIGN <fs_value>.
      ASSIGN COMPONENT 'C' OF STRUCTURE <fs_l_table> TO <fs_value>.
      IF <fs_value> IS ASSIGNED.
        ls_user_contact-lastname = <fs_value>.
      ENDIF.
      UNASSIGN <fs_value>.
      ASSIGN COMPONENT 'D' OF STRUCTURE <fs_l_table> TO <fs_value>.
      IF <fs_value> IS ASSIGNED.
        ls_user_contact-city = <fs_value>.
```

```
      ENDIF.
      UNASSIGN <fs_value>.
      ASSIGN COMPONENT 'E' OF STRUCTURE <fs_l_table> TO <fs_value>.
      IF <fs_value> IS ASSIGNED.
        ls_user_contact-street = <fs_value>.
      ENDIF.
      UNASSIGN <fs_value>.
      ASSIGN COMPONENT 'F' OF STRUCTURE <fs_l_table> TO <fs_value>.
      IF <fs_value> IS ASSIGNED.
        ls_user_contact-country = <fs_value>.
      ENDIF.
      UNASSIGN <fs_value>.
      ASSIGN COMPONENT 'G' OF STRUCTURE <fs_l_table> TO <fs_value>.
      IF <fs_value> IS ASSIGNED.
        ls_user_contact-region = <fs_value>.
      ENDIF.
      UNASSIGN <fs_value>.
      ASSIGN COMPONENT 'H' OF STRUCTURE <fs_l_table> TO <fs_value>.
      IF <fs_value> IS ASSIGNED.
        ls_user_contact-e_mail = <fs_value>.
      ENDIF.
      UNASSIGN <fs_value>.
      APPEND ls_user_contact TO lt_user_contact.
      CLEAR ls_user_contact.
    ENDLOOP.

    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( ).
    lo_nd_user_contact->bind_table( new_items = lt_user_contact ).

  ENDIF.

ENDMETHOD.
```

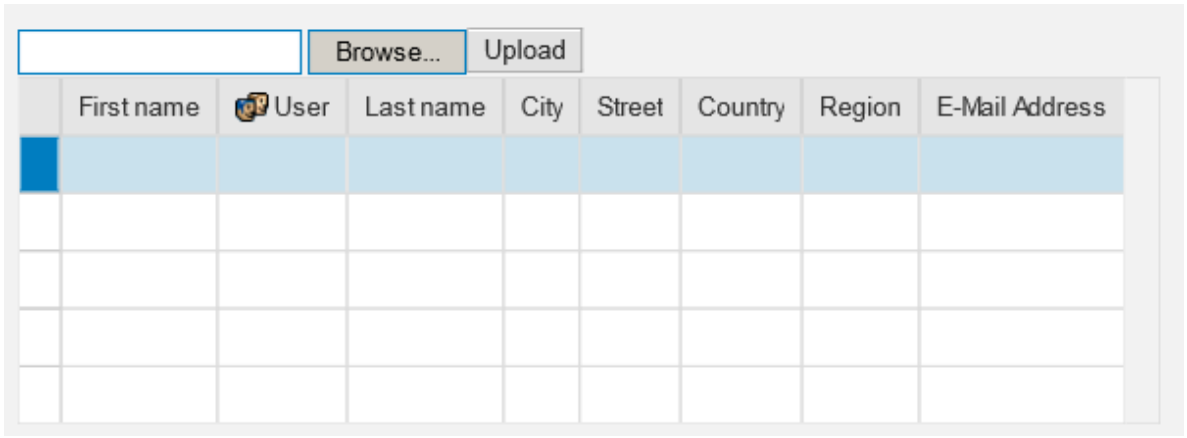
8) Create the WDA:



Execute it.



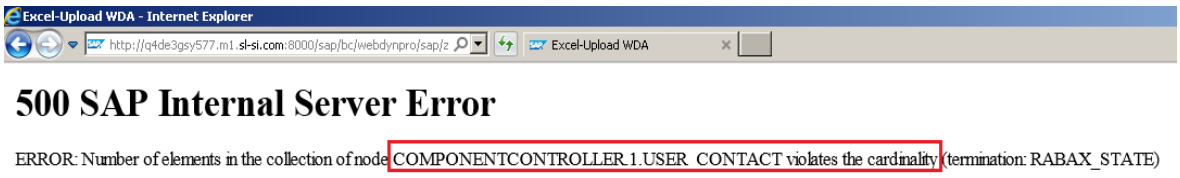
Your application will look like this:



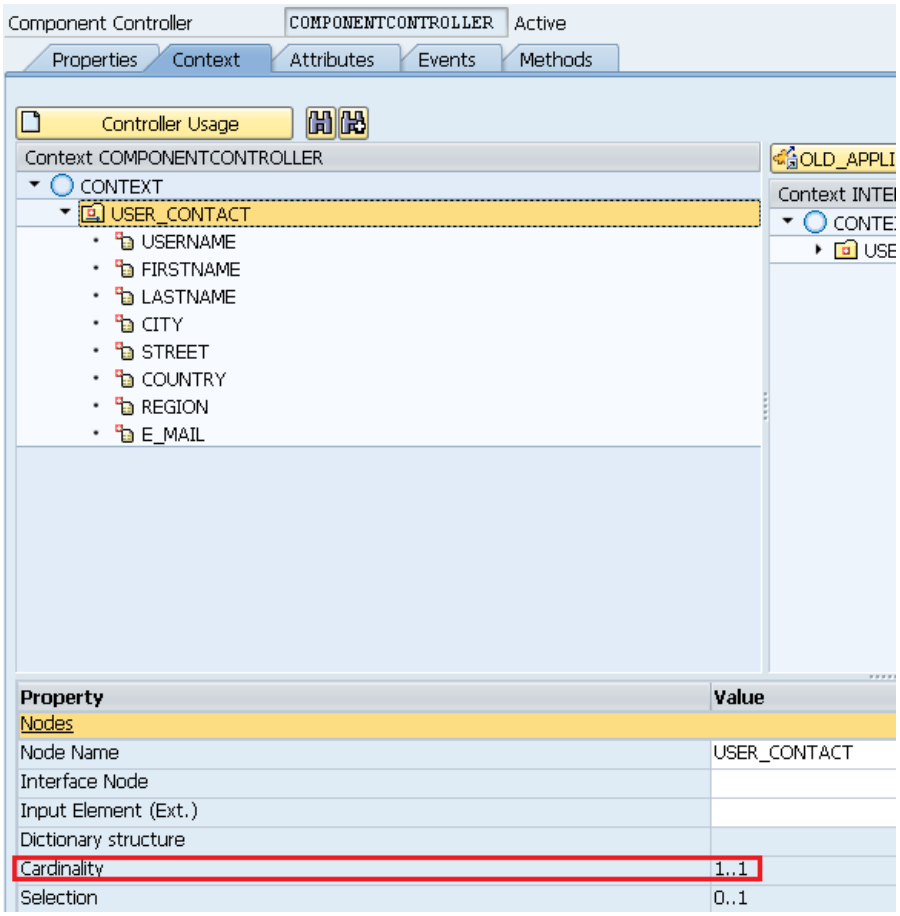
Now upload the following file:

Users_Excel_Contacts_3.xlsx

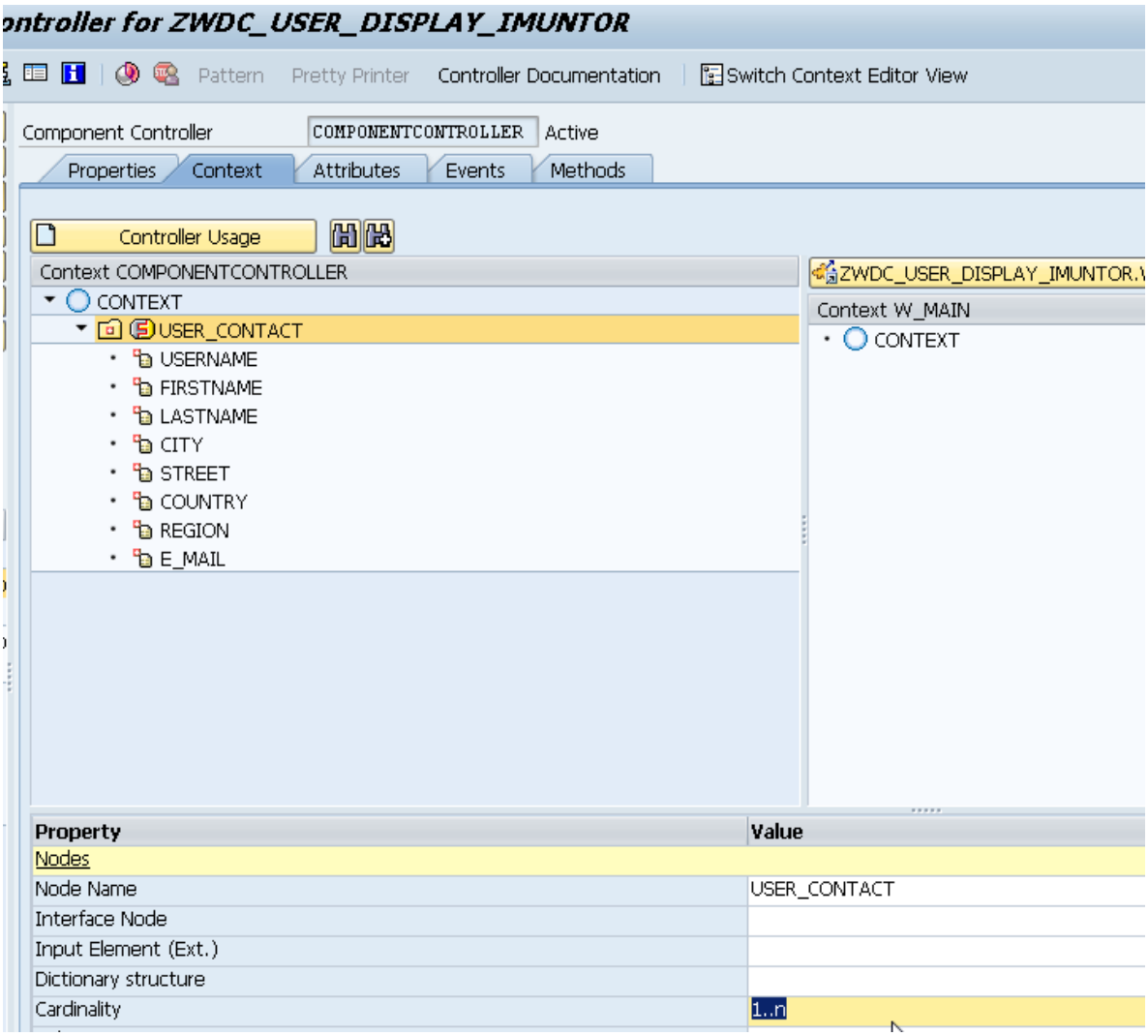
You will get the following error:



And this is the reason:

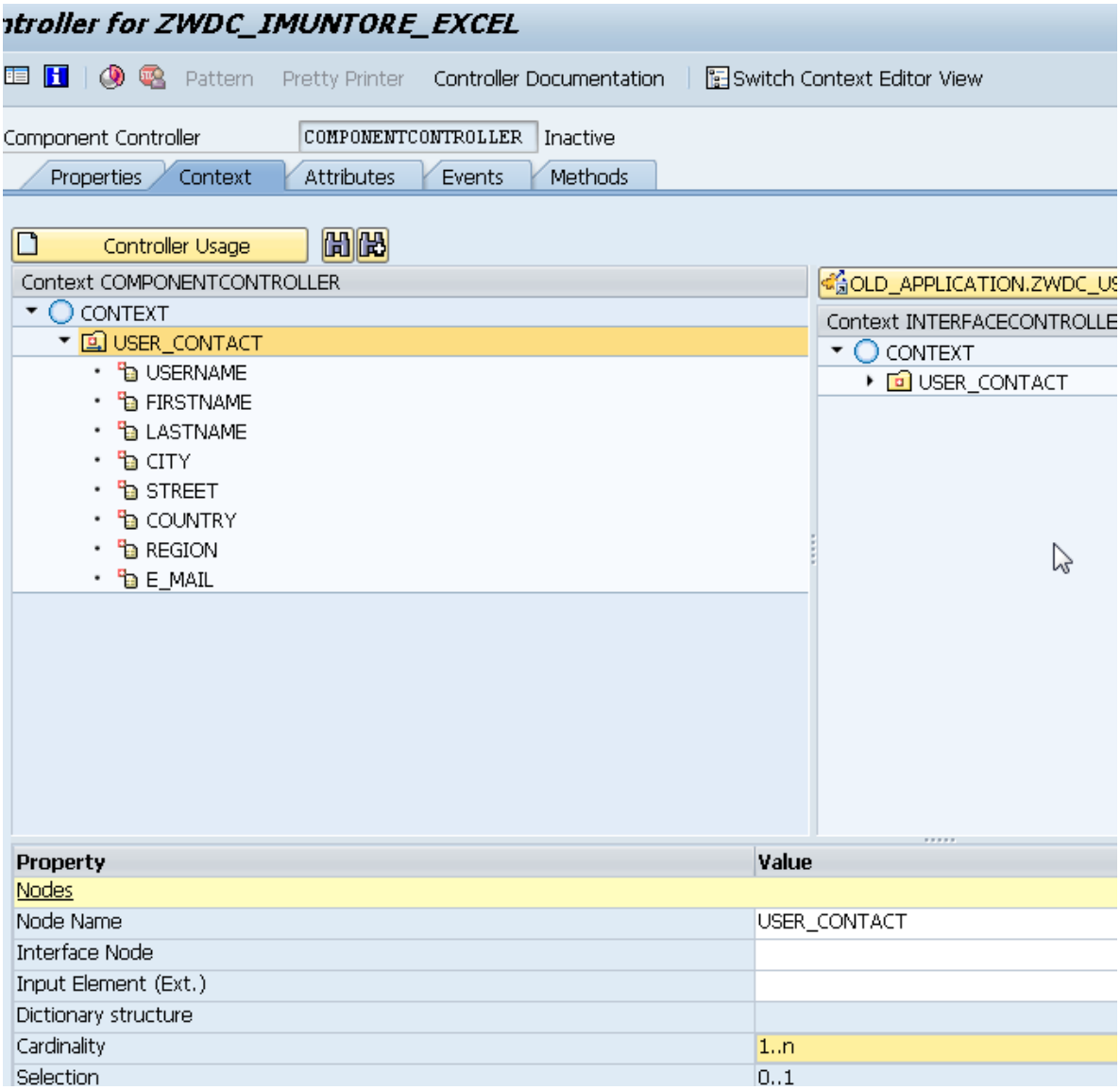


Since it was inherited from the old component. Open the old component and do the following changes:



Save and activate!

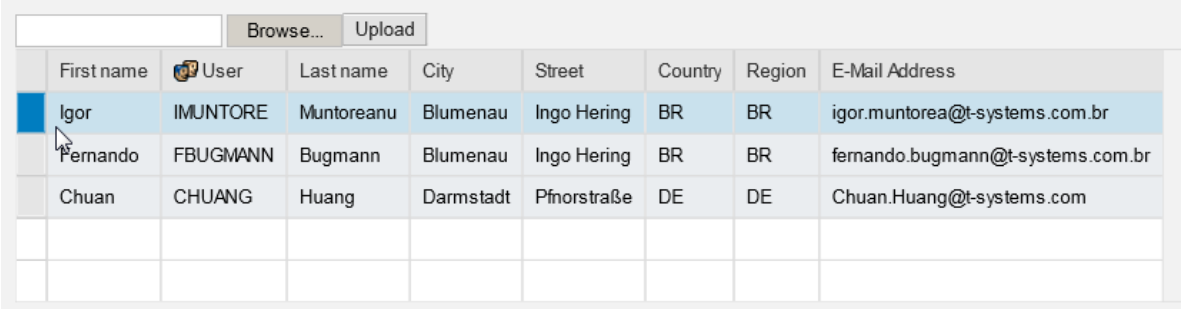
Go to our new Component and check out the results



Save and activate all.

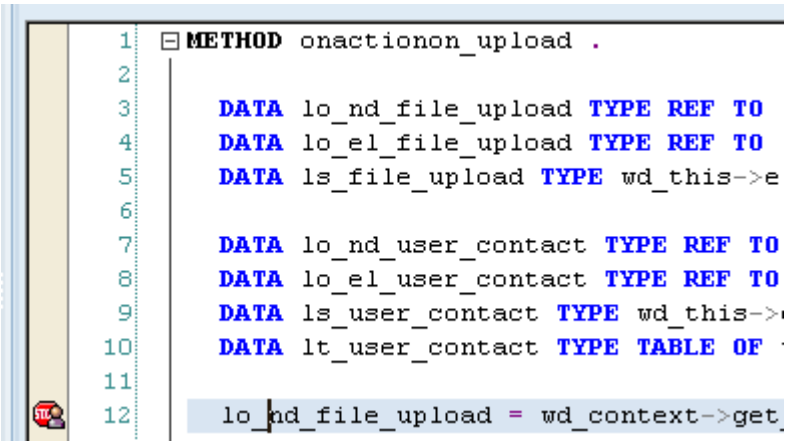
Run the WDA again uploading the file.

Final Result:



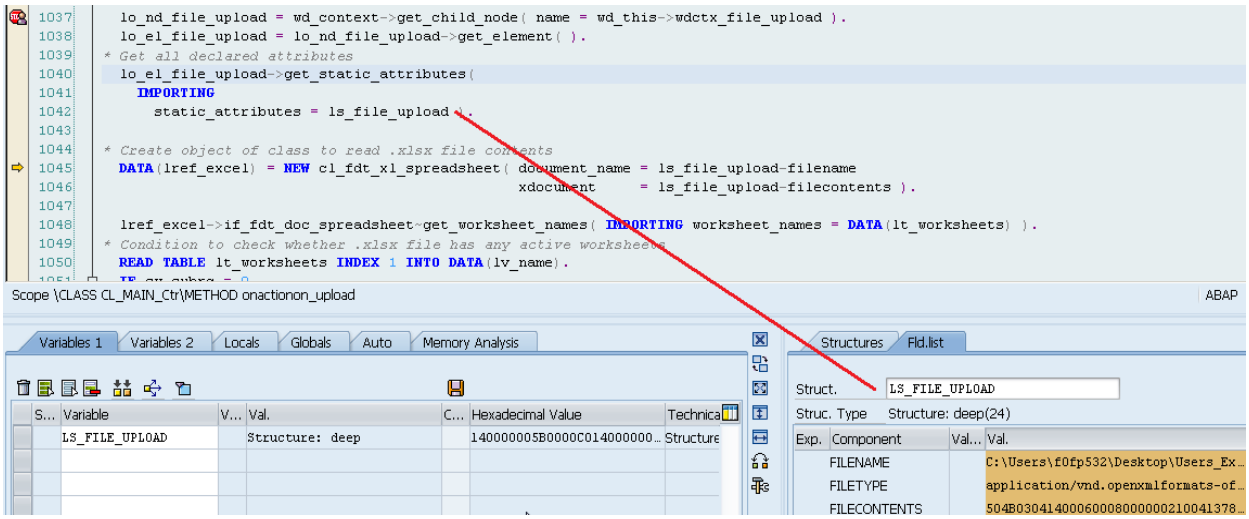
Understanding the code by debugging it:

Put an external break-point on the following line:

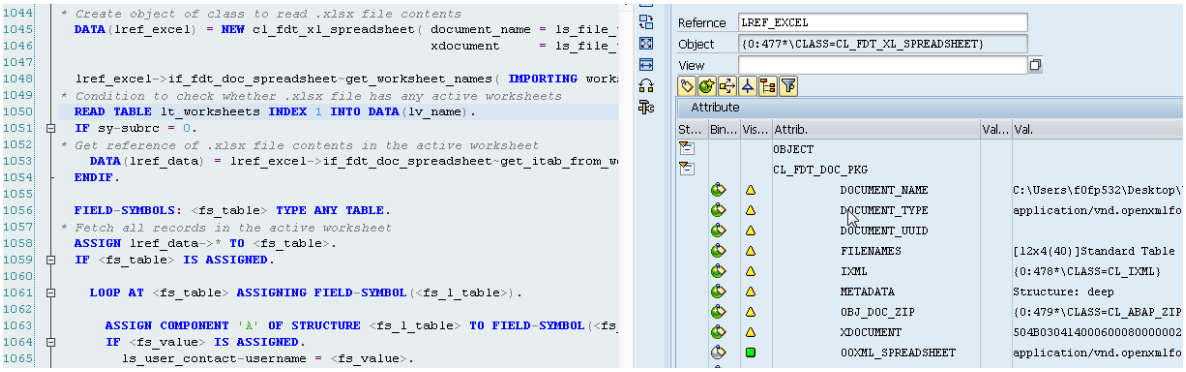


Execute the WDA and upload the file:

Until this point, we are just reading the FILE_UPLOAD node.



Here we read the uploaded excel file:



All the details are inside lref_excel.

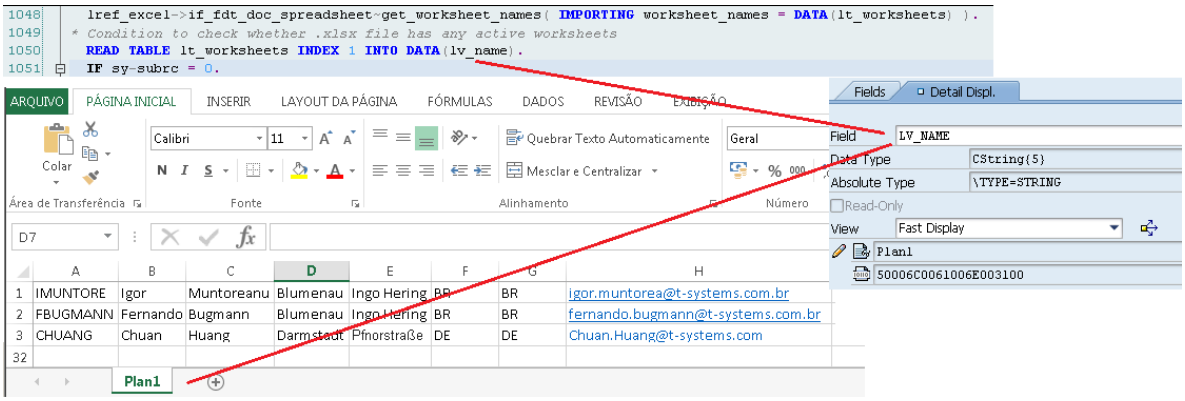
This part is ABAP 7.4 Programing:

```
* Create object of class to read .xlsx file contents
DATA(lref_excel) = NEW cl_fdt_xl_spreadsheet( document_name = ls_file_upload-filename
                                              xdocument     = ls_file_upload-filecontents ).
```

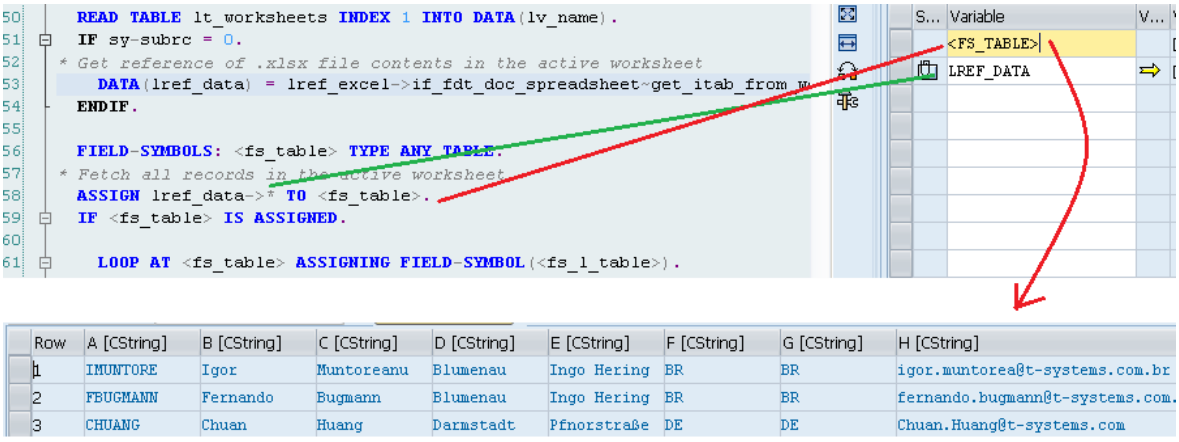
It is just like doing that below on old ABAP versions:

```
CREATE OBJECT lref_excel
  Importing
    document_name = ls_file_upload-filename
    xdocument     = ls_file_upload-filecontents
```

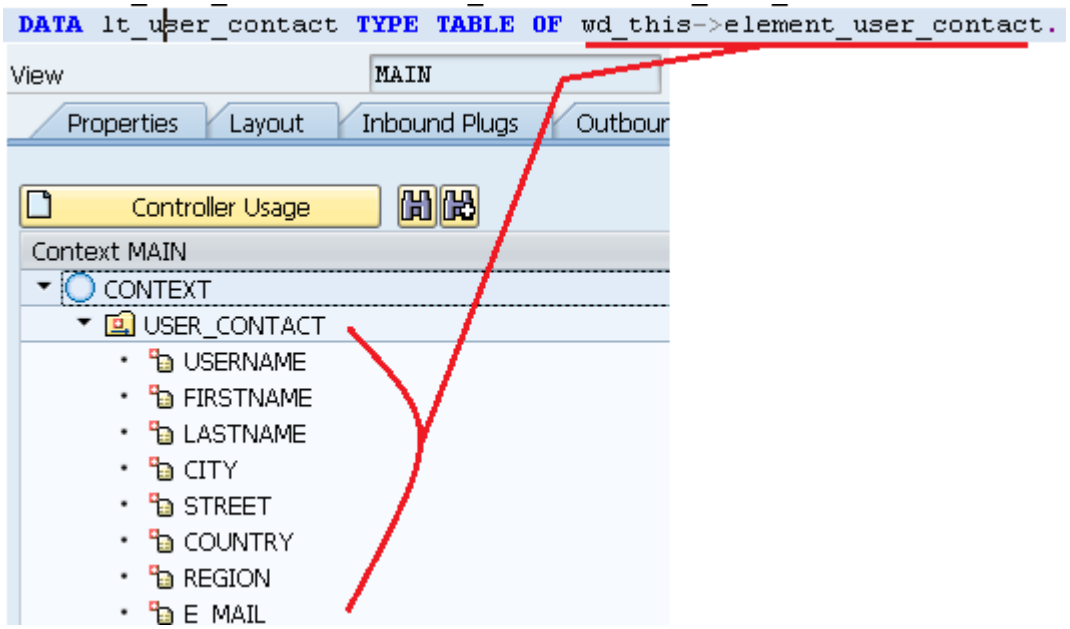
This part of the code means that the program is reading the Plan1 worksheet:



The <fs_table> contains all the contents of the EXCEL file



The following loop is to feed the final internal table lt_user_contact that is the type of the context node that we built:



And finally here we set the final fed internal table to the context and the layout table:

```
1107      lo_nd_user_contact = wd_context->get_child_node( name = wd_this->wdctx_user_contact ).
1108      lo_el_user_contact = lo_nd_user_contact->get_element( ).
1109      lo_nd_user_contact->bind_table( new_items = lt_user_contact ).
```

9) Homework

a. Create a Download button to do the reverse logic:

b. Validate the File entered. Check if it is XLS or XLSX and if now send an error message.