

# Working with Dynamic Tables in Interactive Adobe Forms and WebDynpro ABAP



## Applies to:

Adobe Live Cycle Designer 8.0- Web Dynpro ABAP

## Summary

This article would help ABAP developers, who are faced with a scenario of having a dynamic table within the Interactive Adobe form in their Web Dynpro Component. The scenario discussed here will help the reader to capture data from a table as it grows or shrinks based on users discretion. The paper assumes that the reader already has the basics of PDF based form development.

**Author:** Runal Singh

**Company:** Satyam Computer Services Ltd.

**Created on:** 09 February 2009

## Author Bio

Runal Singh is a certified ABAP application developer with Satyam Computer Services Ltd. He is involved in SAP implementations as an ABAP developer from the past 3 years. He was actively involved in customer implementations in Web Dynpro ABAP and Interactive Adobe forms.

## Table of Contents

Preparing the PDF form.....	3
Step 1:.....	3
Step 2:.....	4
Step 3:.....	4
Step 4:.....	5
Step 5:.....	6
Step 6:.....	7
Step 7:.....	7
Step 8:.....	8
Create Web-Dynpro Component.....	10
Related Content.....	17
Disclaimer and Liability Notice.....	18

## Preparing the PDF form

The first step in this scenario would be to design a form. The steps involved can be broadly categorized as:

Start transaction SFP

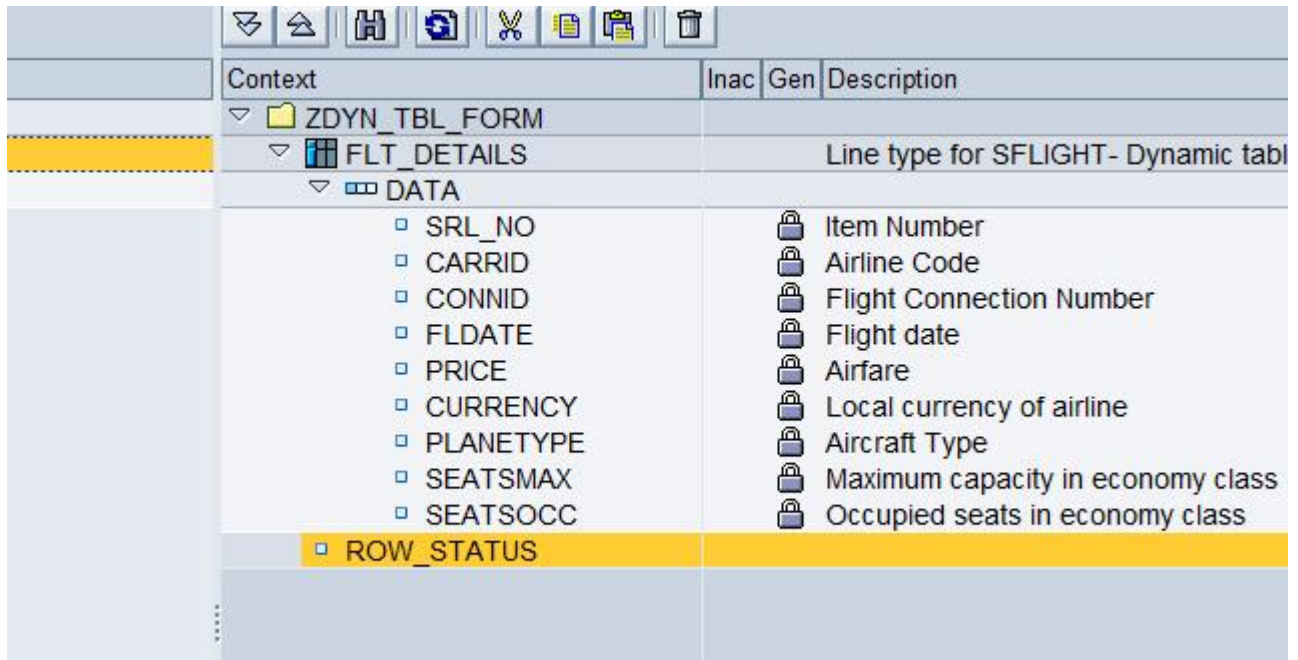
Create an interface

Create a form object

In the context link the required parameters from the interface

Finally create the layout of the form and activate the form.

Assuming that the interface is created with the table structure of S\_FLIGHT, and with an additional string type parameter "ROW\_STATUS". Let us first go through the pre-requisite steps we need to carry out in the form so that it behaves as interactive form when integrated in Web Dynpro. This is how the context would look like.

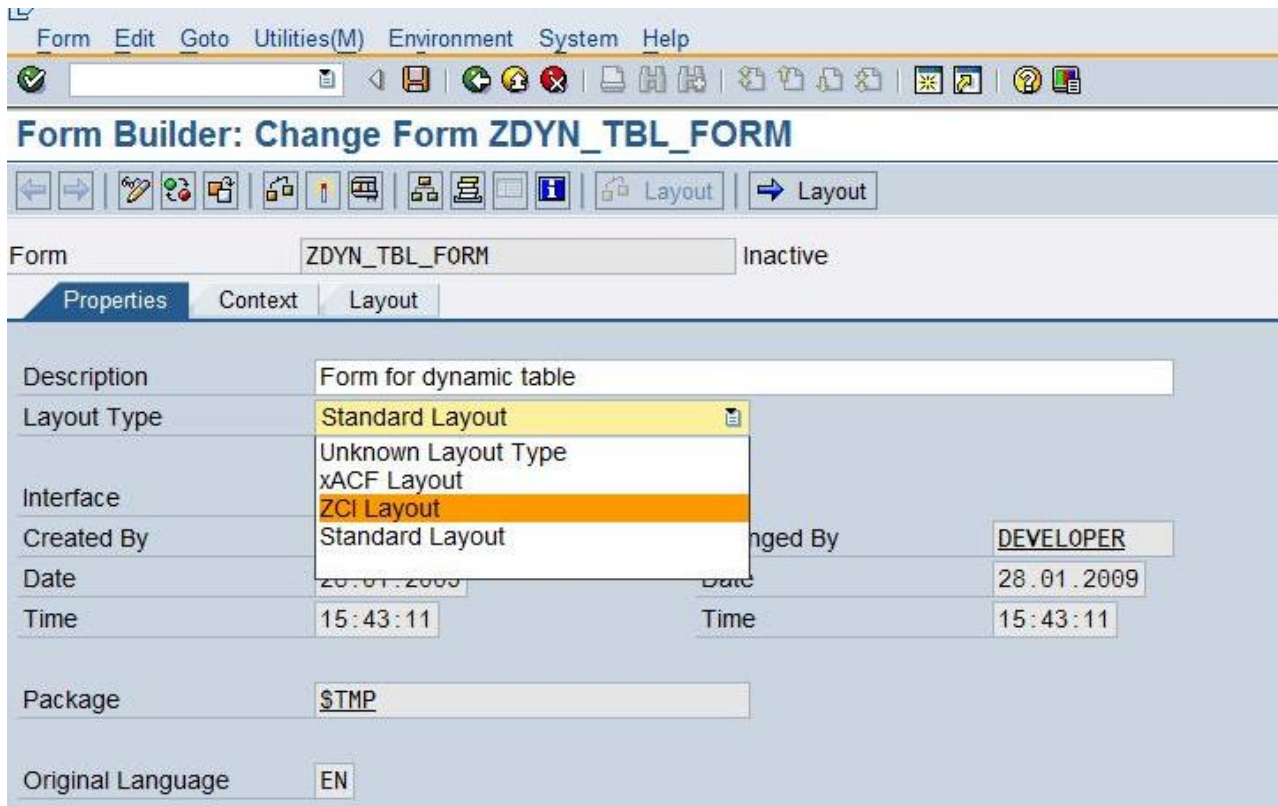


Context	Inac	Gen	Description
<ul style="list-style-type: none"> <li>▼ ZDYN_TBL_FORM           <ul style="list-style-type: none"> <li>▼ FLT_DETAILS               <ul style="list-style-type: none"> <li>DATA                   <ul style="list-style-type: none"> <li>▣ SRL_NO</li> <li>▣ CARRID</li> <li>▣ CONNID</li> <li>▣ FLDATE</li> <li>▣ PRICE</li> <li>▣ CURRENCY</li> <li>▣ PLANETYPE</li> <li>▣ SEATSMAX</li> <li>▣ SEATSOCC</li> <li>▣ ROW_STATUS</li> </ul> </li> </ul> </li> </ul> </li> </ul>			Line type for SFLIGHT- Dynamic tabl Item Number Airline Code Flight Connection Number Flight date Airfare Local currency of airline Aircraft Type Maximum capacity in economy class Occupied seats in economy class

### Step 1:

Open the respective form in T-code SFP, and navigate to properties tab. By default Standard layout is selected, we need to change it to ZCI Layout.

ZCI stands for Zero Client Installation and uses JavaScript in the client to perform interaction between the form and Web-Dynpro. ACF stands for Active Control Framework and uses an ActiveX control for the integration.



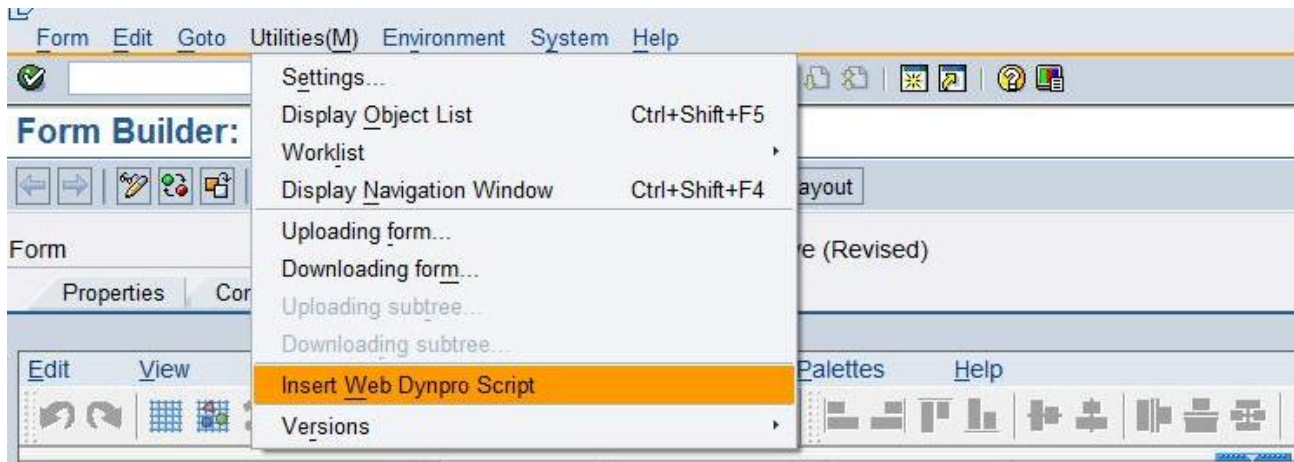
### Step 2:

Once changed the system gives a warning message "Check the Web Dynpro script of the layout", which can be ignored for the time being by pressing enter on the layout type. And we shall now proceed to the Layout tab for further settings.



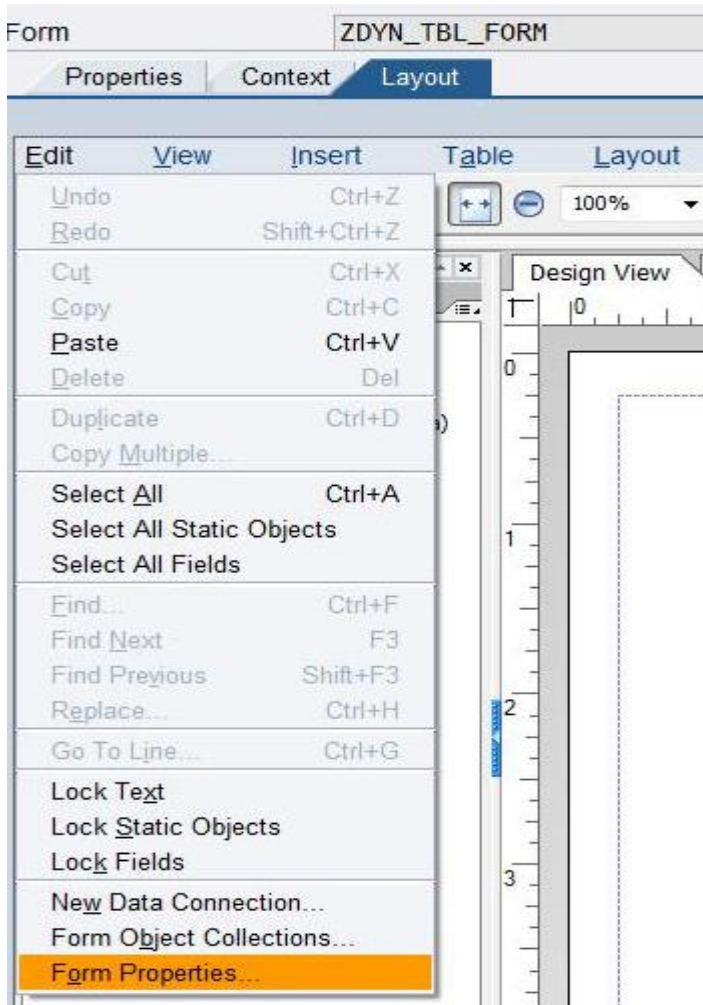
### Step 3:

If we chose Native/ZCI then we need to insert the JavaScript header into our interactive form. That can be done by navigating to layout tab, and then accessing the menu option Utilities→Insert Web Dynpro Script as shown in the screen shot below.

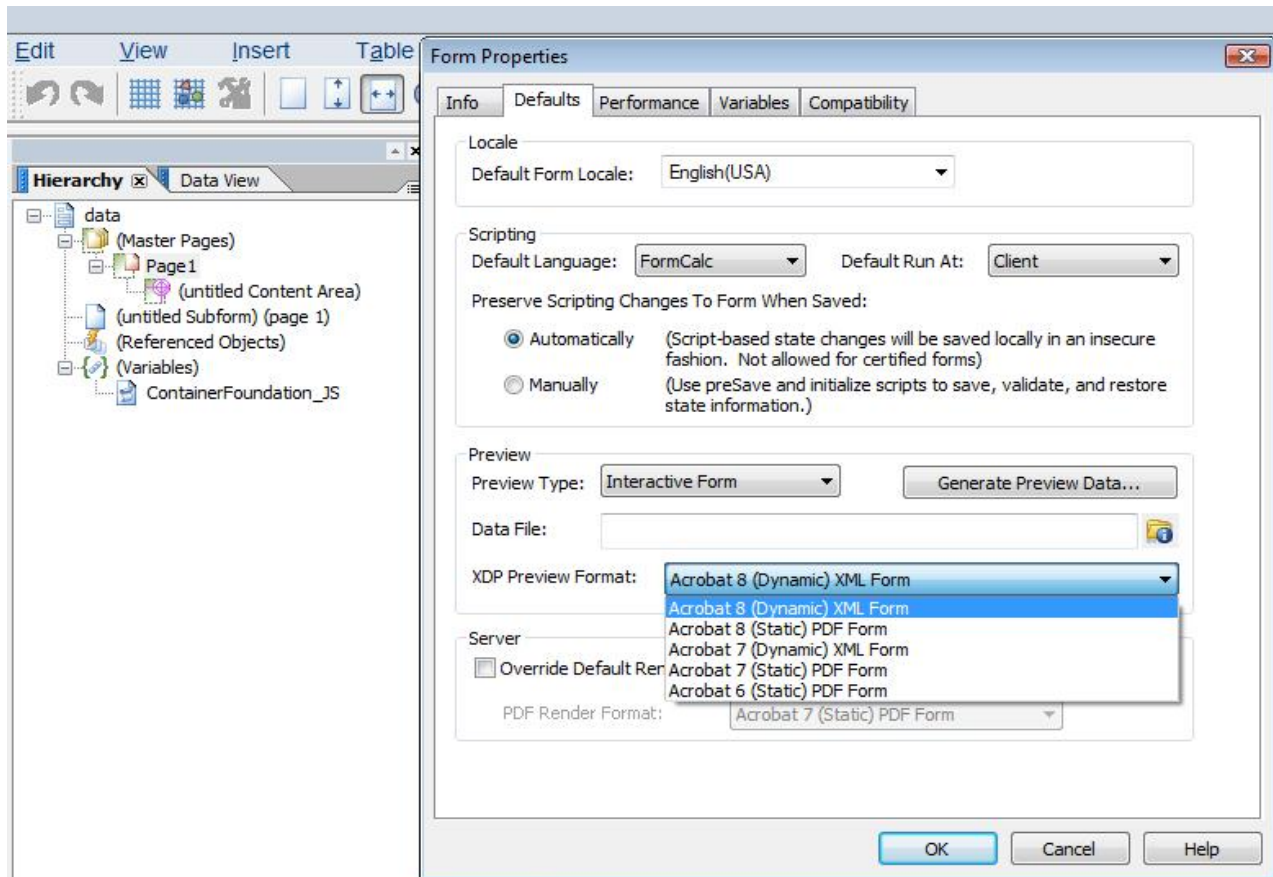


#### Step 4:

We now need to change the form to dynamic type and that can be done by selecting the first menu option within the layout tab as Edit→Form Properties...

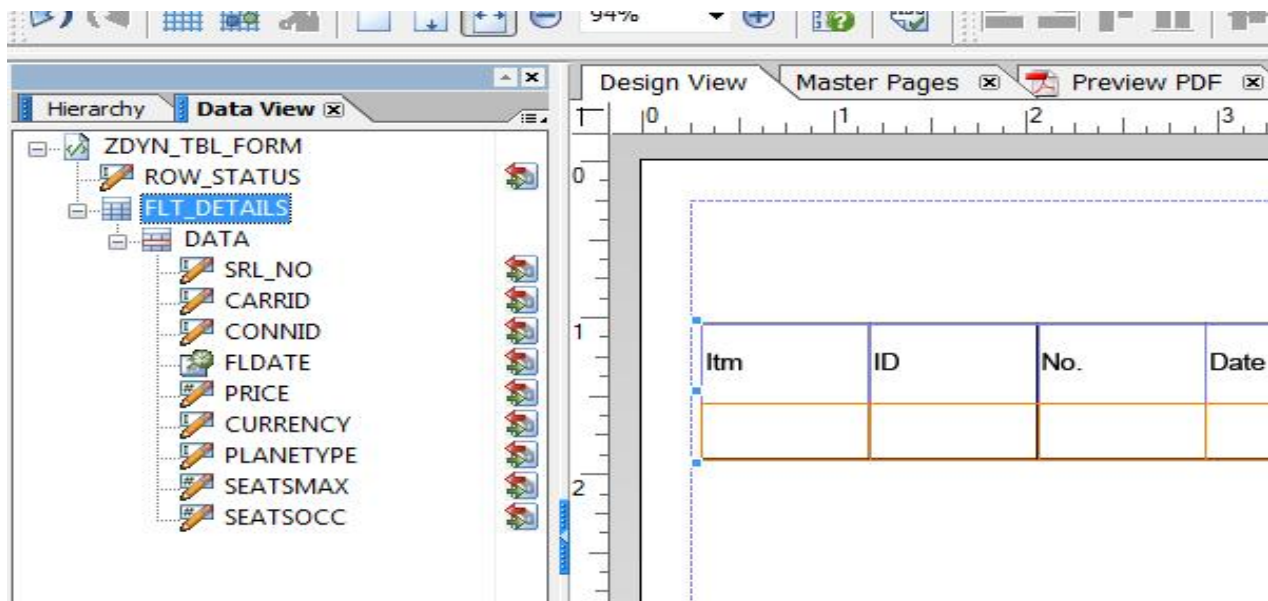


This would display a window, navigate to the defaults tab, and from the drop down for XDP Preview Format change the format to "Acrobat 8 (Dynamic) XML form" if it's not already selected.



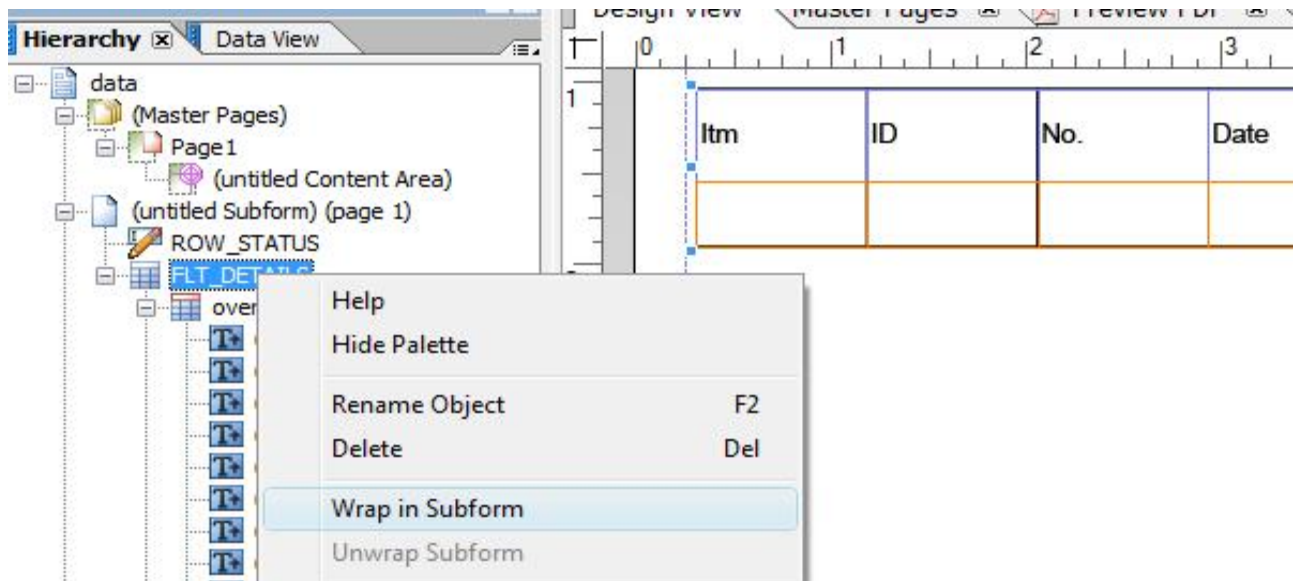
### Step 5:

This completes all the prerequisites. Now we shall drag and drop the table to the layout from the Data View palette as shown below.



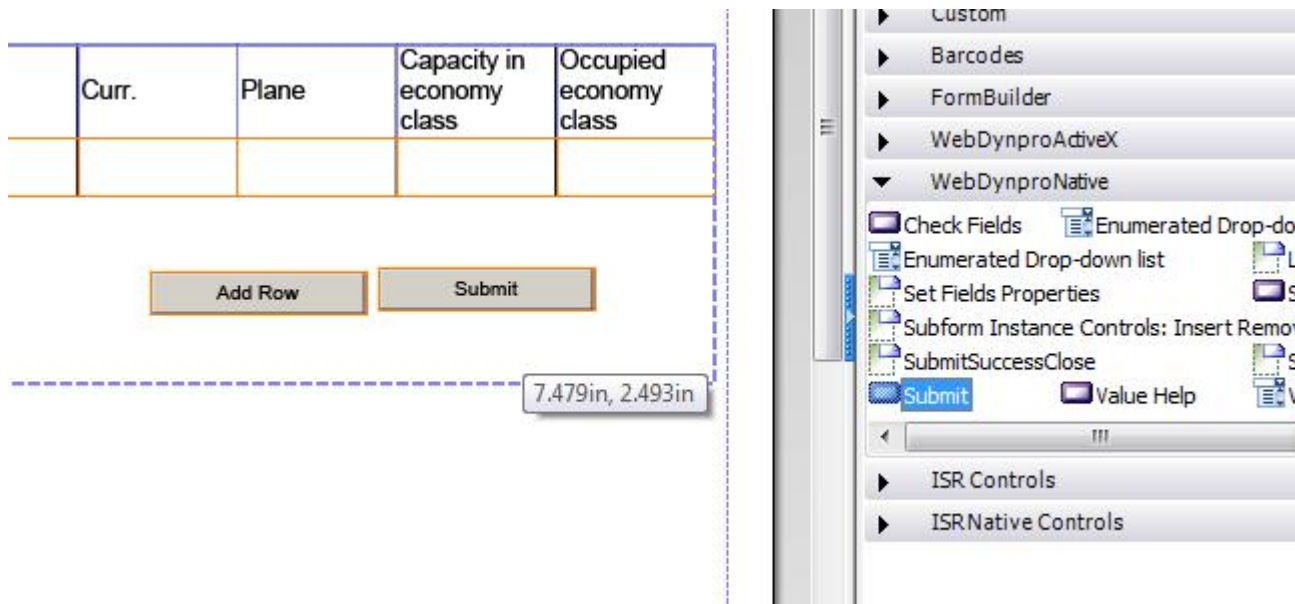
**Step 6:**

Navigate to the Hierarchy palette, and select the table by right click on it, select the option "Wrap in Subform" so that the table gets wrapped in a Subform, which enables the table to grow and shrink in size.

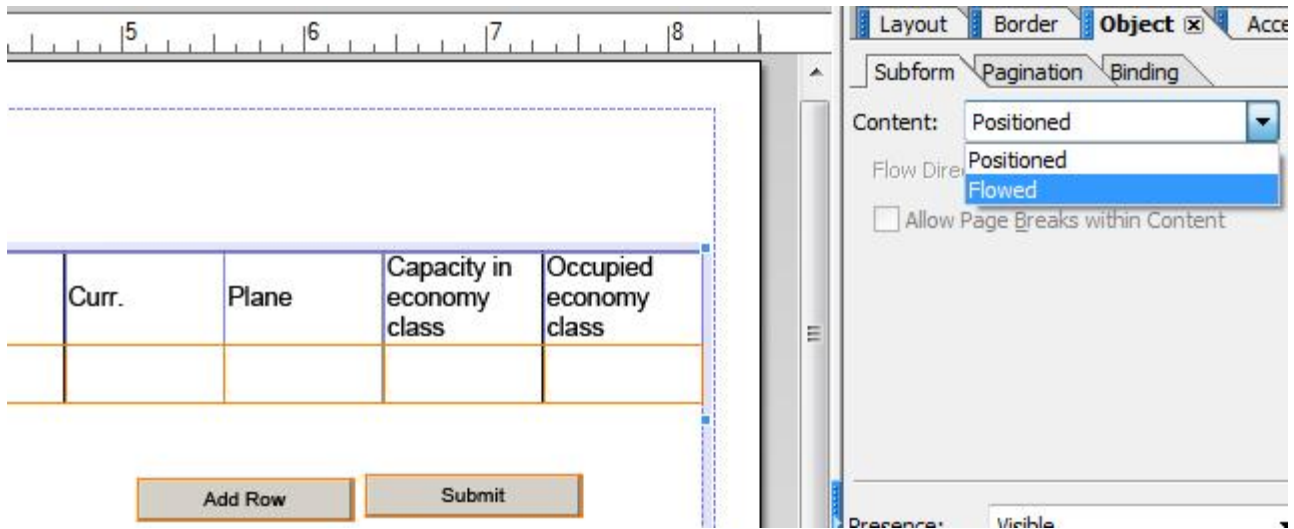
**Step 7:**

Add two buttons inside the Subform, which must be dragged from the Web Dynpro Native library, rename the button captions to "Add Row" and "Remove Row". The buttons must be Submit buttons.

**Note:** We specifically choose the Submit button here because it would trigger an OnSubmit event in Web Dynpro Component, which we will handle by writing our code. We will update the internal table bound to this table as and when the buttons are clicked for addition or removal of rows.



Now, we will change the content type of the Subform to "Flowed", which can be done in the Object palette, within that in the "Subform Tab". As shown in the image below.



### Step 8:

We shall now add code to the added buttons, one for adding a new row and one for removing a row from it. We can add or remove tables, body rows, header rows, and footer rows by adding instance manager scripting expressions to objects on our form

There are two things we need to be sure before we add the code:

We have to ensure that the table that we want to add instances to is contained within a flowed Subform so that we can add new instances of either the table or the rows within the table.

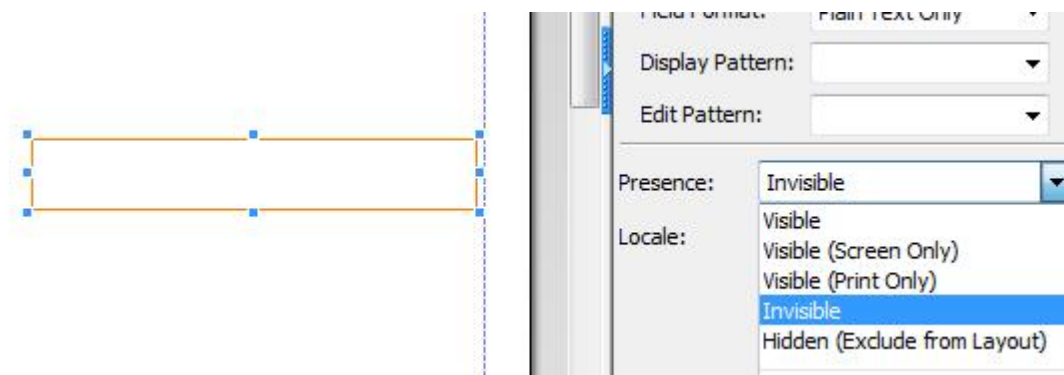
If required, we should set any maximum, minimum, and initial count occurrence values by using either the Binding tab for the table or the table

**Note:** The generic JavaScript code for adding a row instance to the table is  
`Table1.Row1.instanceManager.addInstance(1);`  
 And for removing a row instance from the table would be  
`Table1.Row1.instanceManager.removeInstance(1);`

We have to add this code for addition and deletion of row in the Click event of the respective buttons.

Now that we have added buttons to add and remove a row from the table, the question is how do we achieve the same in our internal table? How do we identify whether to add a row or delete a row as both these buttons invoke the OnSubmit event. For this we had taken a string parameter in interface, which will store the values once the buttons are clicked.

To achieve this we drag the variable to the layout tab and make it as invisible, as its presence is not required in the form, only the value it holds is good enough for us. As shown in the screen shot below.





So we modify the code for addition/deletion of rows, by assigning a value to the string variable which will be read in the Web-Dynpro Component to make out if the user has clicked to add a row or remove a row.

```

Show: click*
----- data.#subform[0].TBL.submitToSAP2::click: - (JavaScript, client) -----
//Code for removing a row
FLT_DETAILS.DATA.instanceManager.removeInstance(1);
ROW_STATUS.rawValue = "REMOVE";

// DO NOT MODIFY THE CODE BEYOND THIS POINT - 800.2007041
ContainerFoundation_JS.SendMessageToContainer(event.target
// END OF DO NOT MODIFY

For Help, press F1
Hierarchy x Data View
  (untitled Text)
  (untitled Text)
  (untitled Text)
  DATA
Design View Master Pages x Preview PDF x
0 1 2 3
0

```

So, the JavaScript code for addition and deletion gets modified as:

**Note:** The generic JavaScript code for adding a row instance to the table is

```
Table1.Row1.instanceManager.addInstance(1);
ROW_STATUS.rawValue = "ADD";
```

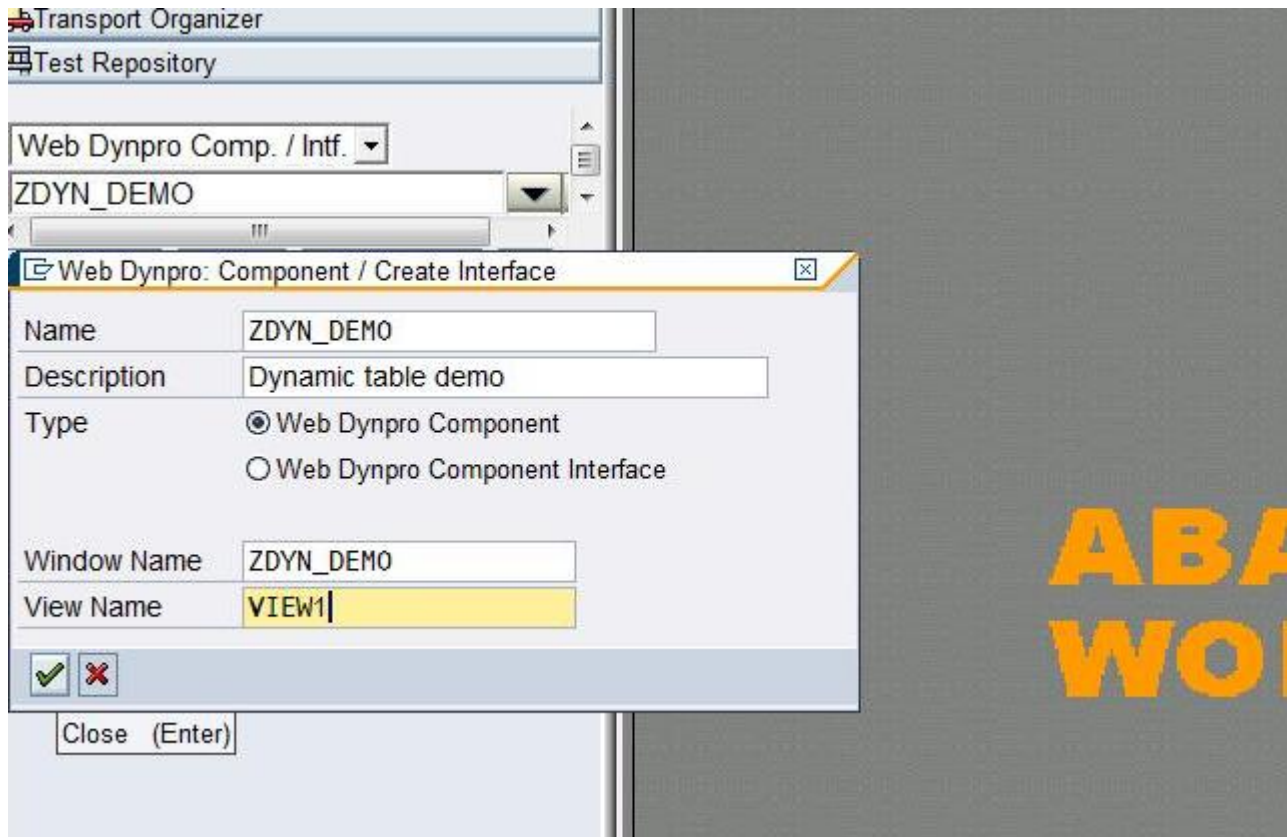
And for removing a row instance from the table would be

```
Table1.Row1.instanceManager.removeInstance(1);
ROW_STATUS.rawValue = "REMOVE";
```

We shall now proceed to create our web-Dynpro component in the next section.

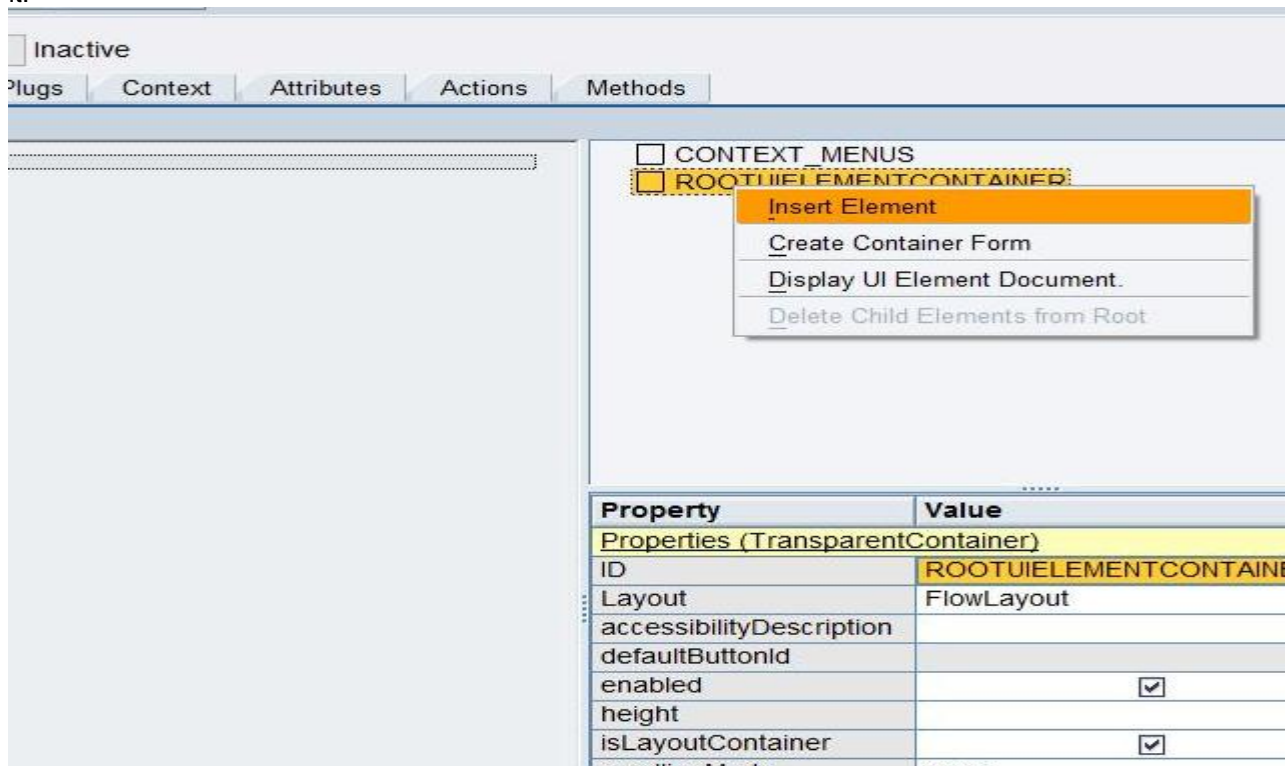
## Create Web-Dynpro Component

To Create a Web-Dynpro component, open transaction SE80; create a window and a view.

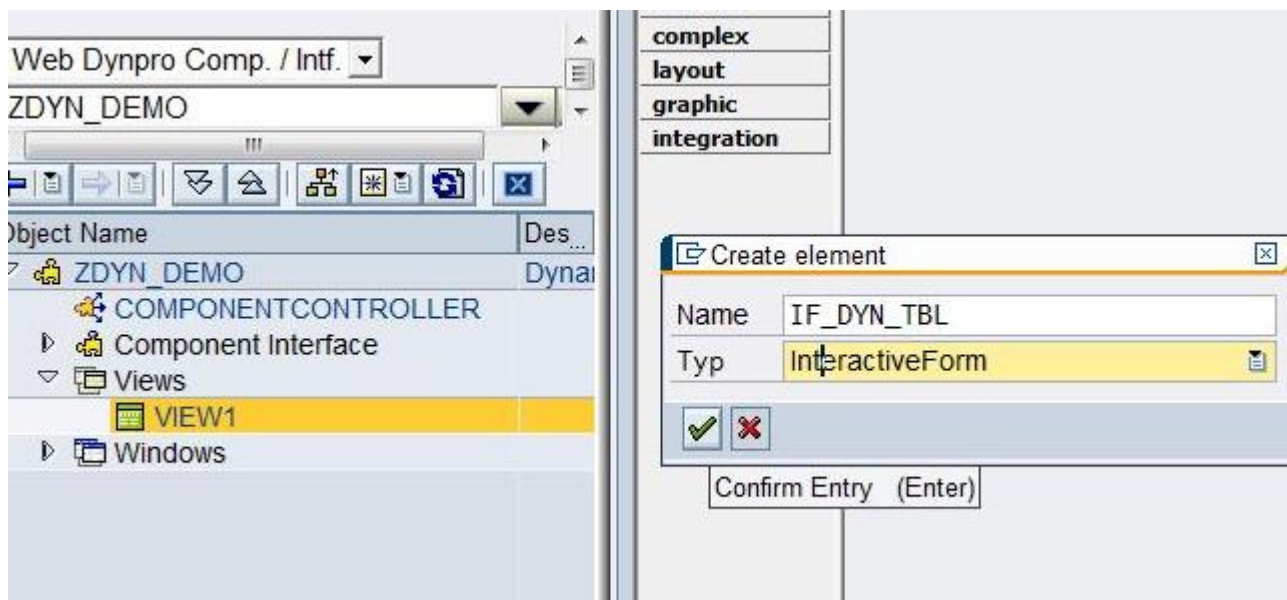


Navigate to the layout tab of the VIEW1. And on the left side right click the ROOTUIELEMENT to add elements to

it.



Give a name to the Interactive form UI element and select the type from drop down as shown below.



In the properties window, select the PDF form prepared above in the Template source property of this element.

Property	Value	Bind
<b>Properties (InteractiveForm)</b>		
ID	IF_DYN_TBL	
additionalArchives		
dataSource	VIEW1.ZDYN_TBL_FORM	
displayType	native	
enabled	<input checked="" type="checkbox"/>	
height	900px	
templateSource	ZDYN_TBL_FORM	
readOnly	<input type="checkbox"/>	
tooltip		
visible	Visible	
width	900px	
<b>Events</b>		
onSubmit	ROW_EDIT	
<b>Layout Data (FlowData)</b>		
cellDesign	padless	
vGutter	none	

And in the OnSubmit event create an event handler as "Row\_Edit" where we will add our code for addition and deletion of rows. The below screen shot shows these steps

Property	Value	Bindi...
<b>Properties (InteractiveForm)</b>		
<b>Create Action</b>		
Component	ZDYN_TBL	
View	MAIN	
Action	Row_Edit	
Description	Handle the row addition/deletion here	
Transfer UI Event Parameters	<input checked="" type="checkbox"/>	
<b>Events</b>		
onSubmit	Row_Edit	
<b>Layout Data (FlowData)</b>		
cellDesign	padless	
vGutter	none	

Once this event handler is created, navigate to the methods tab and double click the method name to write our code in it.

Put the following code in the event handler method just created.

```

method ONACTIONROW_EDIT .
  DATA LO_ND_ZDYN_TBL_FORM TYPE REF TO IF_WD_CONTEXT_NODE.
  DATA LO_ND_FLT_DETAILS TYPE REF TO IF_WD_CONTEXT_NODE.
  DATA LO_EL_ZDYN_TBL_FORM TYPE REF TO IF_WD_CONTEXT_ELEMENT.
  DATA LS_ZDYN_TBL_FORM TYPE WD_THIS->ELEMENT_ZDYN_TBL_FORM.
  DATA LV_ROW_STATUS LIKE LS_ZDYN_TBL_FORM-ROW_STATUS.

  DATA LS_FLT_DETAILS TYPE WD_THIS->ELEMENT_FLT_DETAILS.
  DATA: IT_DYN_TBL TYPE TABLE OF ZFLT_TBL,
        WA_DYN_TBL TYPE ZFLT_TBL,
        ROW_COUNT TYPE I,
        LAST_ROW TYPE I.
*  navigate from <CONTEXT> to <ZDYN_TBL_FORM> via lead selection
  LO_ND_ZDYN_TBL_FORM = WD_CONTEXT->GET_CHILD_NODE( NAME = WD_THIS-
>WDCTX_ZDYN_TBL_FORM ).

*  navigate from <ZDYN_TBL_FORM> to <FLT_DETAILS> via lead selection
  LO_ND_FLT_DETAILS = LO_ND_ZDYN_TBL_FORM->GET_CHILD_NODE( NAME = WD_THIS-
>WDCTX_FLT_DETAILS ).

* GET THE DATA FROM PDF
  CALL METHOD LO_ND_FLT_DETAILS->GET_STATIC_ATTRIBUTES_TABLE
*  EXPORTING
*  FROM = 1
*  TO = 2147483647
  IMPORTING
  TABLE = IT_DYN_TBL.

*  navigate from <CONTEXT> to <ZDYN_TBL_FORM> via lead selection
  LO_ND_ZDYN_TBL_FORM = WD_CONTEXT->GET_CHILD_NODE( NAME = WD_THIS-
>WDCTX_ZDYN_TBL_FORM ).

*  get element via lead selection
  LO_EL_ZDYN_TBL_FORM = LO_ND_ZDYN_TBL_FORM->GET_ELEMENT( ).

* GET THE STATUS OF ROW ADDITION OR DELETION
  LO_EL_ZDYN_TBL_FORM->GET_ATTRIBUTE(
  EXPORTING
  NAME = `ROW_STATUS`
  IMPORTING
  VALUE = LV_ROW_STATUS ).

* GET THE ROW COUNT AND THE SERIAL NUMBER AT LAST ROW
  ROW_COUNT = 1.
  FIELD-SYMBOLS <WA> TYPE ZFLT_TBL.
  LOOP AT IT_DYN_TBL ASSIGNING <WA>.
    <WA>-SRL_NO = ROW_COUNT.
    ROW_COUNT = ROW_COUNT + 1.
  ENDLLOOP.
  LAST_ROW = <WA>-SRL_NO.

*DEPENDING ON USER SELECTION PERFORM ROW ADDITION/DELETION TO INTERNAL TABLE
  HERE
  IF LV_ROW_STATUS EQ 'ADD'.
    WA_DYN_TBL-SRL_NO = LAST_ROW + 1.
    APPEND WA_DYN_TBL TO IT_DYN_TBL.

```

```

CLEAR: ROW_COUNT, WA_DYN_TBL.
ELSEIF LV_ROW_STATUS EQ 'REMOVE'.
  DELETE IT_DYN_TBL WHERE SRL_NO EQ LAST_ROW.
ENDIF.

*   navigate from <CONTEXT> to <ZDYN_TBL_FORM> via lead selection
  LO_ND_ZDYN_TBL_FORM = WD_CONTEXT->GET_CHILD_NODE( NAME = WD_THIS-
>WDCTX_ZDYN_TBL_FORM ).

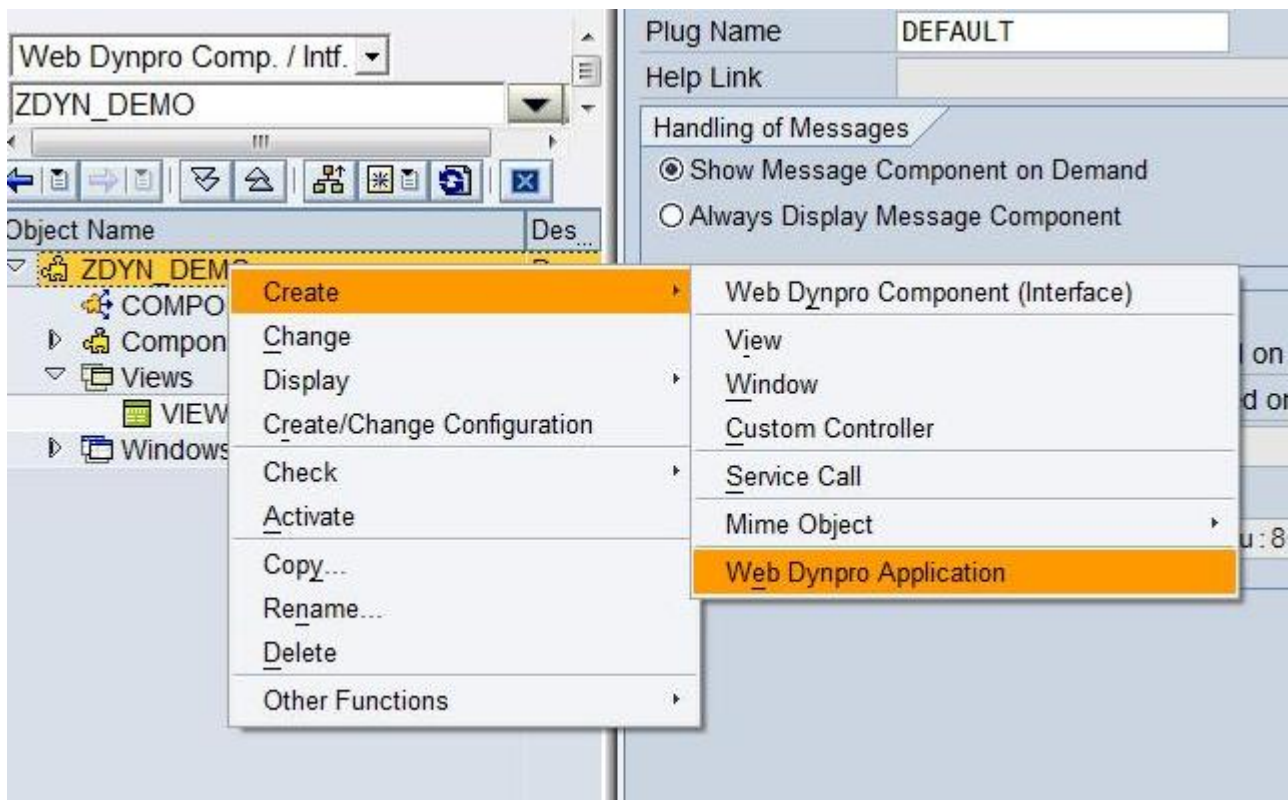
*   navigate from <ZDYN_TBL_FORM> to <FLT_DETAILS> via lead selection
  LO_ND_FLT_DETAILS = LO_ND_ZDYN_TBL_FORM->GET_CHILD_NODE( NAME = WD_THIS-
>WDCTX_FLT_DETAILS ).

* BIND THE MODIFIED INTERNAL TABLE HERE TO THE PDF CONTEXT
  CALL METHOD LO_ND_FLT_DETAILS->BIND_TABLE
    EXPORTING
      NEW_ITEMS          = IT_DYN_TBL.
*   SET_INITIAL_ELEMENTS = ABAP_TRUE
*   INDEX                =

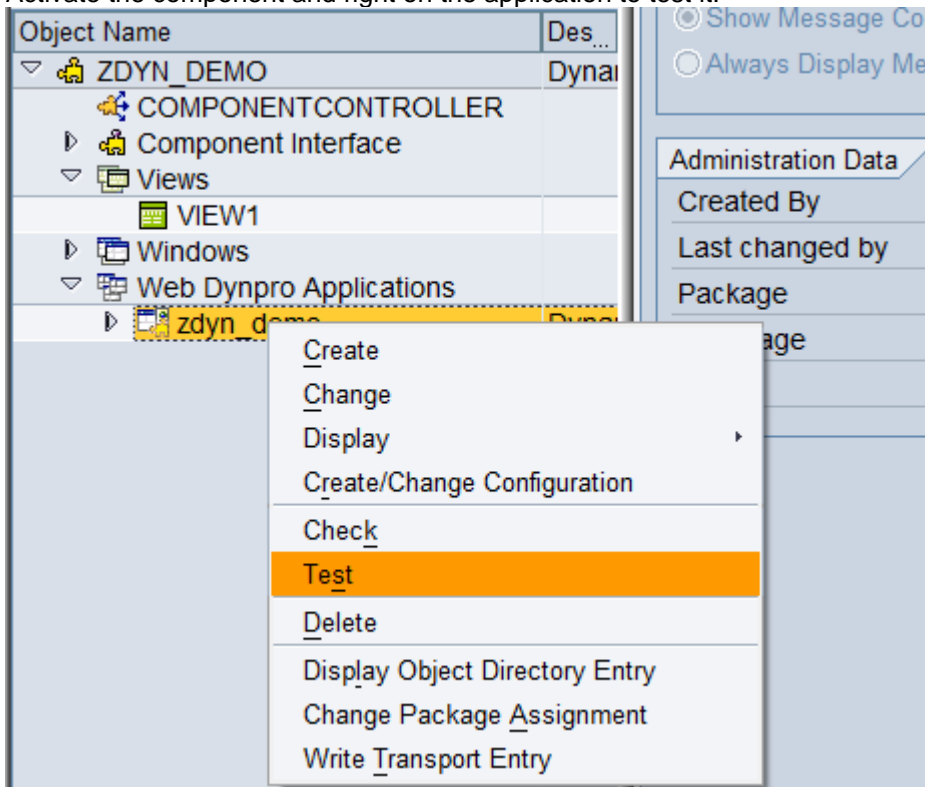
endmethod.

```

Once this code is checked for syntax errors, we would create a URL to this component.



Activate the component and right on the application to test it.



The result would look like this when executed.

Dynamic table demo

1 / 1 89.1% Sign Find

Itm	ID	No.	Date	Airfare	Curr.	Plane
001		0000	0000-00-00	0		
002		0000	0000-00-00	0		
003		0000	0000-00-00	0		
004		0000	0000-00-00	0		
005		0000	0000-00-00	0		
006		0000	0000-00-00	0		

Add Row Remove Row



## Related Content

[E-learning tutorials - Web Dynpro](#)

[Web Dynpro ABAP in SDN](#)

[Web Dynpro Wiki in SDN](#)

## 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.