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.

	▽ ☆ 🛗 🗿 🐰 🗎 🛱 着	Ì		
	Context	Inac	Gen	Description
	ZDYN_TBL_FORM			
	FLT_DETAILS			Line type for SFLIGHT- Dynamic tabl
	□ DATA □			
	 SRL_NO CARRID CONNID FLDATE PRICE CURRENCY PLANETYPE SEATSMAX SEATSOCC ROW_STATUS 			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
I				

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.

	avout	
and the other the formation to be a first the second the second of the second s	Layour Cayour	
ZDYN_TBL_FORM	Inactive	
Layout		
Form for dynamic table		
Standard Layout	1	
Unknown Layout Type xACF Layout		
ZCI Layout		
Standard Layout	nged By	DEVELOPER
20.01.2000	Daic	28.01.2009
15:43:11	Time	15:43:11
	ZDYN_TBL_FORM Layout Form for dynamic table Standard Layout Unknown Layout Type xACF Layout ZCF Layout Standard Layout 20.01.2000 15:43:11	ZDYN_TBL_F0RM Inactive Layout Inactive Form for dynamic table Inactive Standard Layout Image: Contract of the second

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.

Check the V	Veb Dynpro script of the	layout	
10		II and	11 mar

Step 3:

If we chose Native/ZCI then we need to insert the JavaScript header into our interactive from. 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.

Ø	Settings		🕰 🎗 🕱 🛛 I 🍘 📑
Form Builder:	Display Object List Ct Worklist	rl+Shift+F5	
	Display Navigation Window Ct	rl+Shift+F4	ayout
Form Properties Cor	Uploading <u>f</u> orm Downloading for <u>m</u>		re (Revised)
	Downloading subtree		
<u>E</u> dit <u>V</u> iew	Insert Web Dynpro Script		Palettes <u>H</u> elp
	Versions		특별 IF <u>1</u> 분수 바늘 준

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 $Edit \rightarrow Form$ Properties...

orm	ZDYN_	TBL_FOR	M
Properties	Context Lay	rout	
Edit View	Insert	T <u>a</u> ble	Layout
<u>U</u> ndo <u>R</u> edo	Ctrl+Z Shift+Ctrl+Z) 100% ▼
Cut <u>C</u> opy <u>P</u> aste Delete	Ctrl+X Ctrl+C Ctrl+V		Design View
Duplicate Copy <u>M</u> ultiple	Ctrl+D))) _	
Select <u>A</u> ll Select All Static Select All Fields	Ctrl+A Objects	1	
Eind Find <u>N</u> ext Find Pre <u>v</u> ious R <u>e</u> place	Ctrl+F F3 Shift+F3 Ctrl+H	-	
Go To Line	Ctrl+G		
Lock Te <u>x</u> t Lock <u>S</u> tatic Objec Loc <u>k</u> Fields	cts	-	
Ne <u>w</u> Data Connec Form O <u>bj</u> ect Colle	ction ections	3	
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.

Image: Second and Second	•
Hierarchy Data View Default Form Locale: English(USA) data Default Form Locale: English(USA) Page 1 Default Language: FormCalc Default Run At: (untitled Subform) (page 1) (untitled Subform) (page 1) Preserve Scripting Changes To Form When Saved: (Variables) @ Automatically (Script-based state changes will be saved locally in an instafashion. Not allowed for certified forms) (Variables) @ Manually (Use preSave and initialize scripts to save, validate, and state information.) Preview	•
Hierarchy Data View Default Form Locale: English(USA) ContainerFoundation_JS Default Content Area Default	•
data Master Pages) Page1 (untitled Content Area) (untitled Subform) (page 1) (Referenced Objects) (Variables) ContainerFoundation_JS Scripting Default Language: FormCalc Default Run At: Client Preserve Scripting Changes To Form When Saved: (Script-based state changes will be saved locally in an instafation. Not allowed for certified forms) Manually (Use preSave and initialize scripts to save, validate, and state information.) Preview 	•
Image: Page1 Default Language: FormCalc Default Run At: Client Image: Image	•
Image: ContainerFoundation_JS Preserve Scripting Changes To Form When Saved: Image: ContainerFoundation_JS Preserve Scripting Changes To Form When Saved: Image: ContainerFoundation_JS Image: ContainerFoundation_JS Preview	00
Preview	i restore
Preview Type: Interactive Form Generate Preview Data	ıta
Data File:	6
XDP Preview Format: Acrobat 8 (Dynamic) XML Form	-
Acrobat 8 (Dynamic) XML Form Acrobat 8 (Static) PDF Form Acrobat 7 (Dynamic) XML Form Override Default Rer Acrobat 7 (Static) PDF Form Acrobat 6 (Static) PDF Form PDF Render Format: Acrobat 7 (Static) PDF Form	

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.

					Content:	Positioned Positioned
Curr.	Plane	Capacity in economy class	Occupied economy class	E	Allow	Flowed Page Breaks within Content
	Add Pow	Submit				

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.

	Display Pa Edit Patter	nttern:
4	Presence:	Invisible
	Locale:	Visible Visible (Screen Only) Visible (Print Only)
		Invisible
		Hidden (Exclude from Layout)

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: dick* -	
//Code for removing FLT_DETAILS.DATA.ins	a row tanceManager.removeInstance(1);
_	<pre>// DO NOT MODIFY THE CODE BEYOND THIS POINT - 800.2007041 ContainerFoundation_JS.SendMessageToContainer(event.targe // END OF DO NOT MODIFY</pre>
For Help, press F1	
Hierarchy 🗷 Data View	Design View Master Pages 🗷 😤 Preview PDF 🗷

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.

,⇔Transport Organ	izer III and III	
In the second		
Web Dynpro Co	omp. / Intf. 💌	
ZDYN_DEMO	······································	
🖙 Web Dynpro:	Component / Create Interface	
Name	ZDYN_DEMO	
Description	Dynamic table demo	
Туре	Web Dynpro Component O Web Dynpro Component Interface	
Window Name	ZDYN_DEMO	141-91
View Name	VIEW1	
Close (Enter)	

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

it		
Inactive		
Plugs Context Attributes Actions	Methods	
		3
	insert Elem	ent
	<u>Create Cont</u>	ainer Form
	Display UI E	Element Document.
	Delete Child	Elements from Root
	Property	
	Flopenty	Value
	Properties (Transparent	Value
	Properties (Transparent	Value Container) ROOTUIELEMENTCONTAIN
	Properties (Transparent ID : Lavout	Value Container) ROOTUIELEMENTCONTAIN FlowLayout
	Properties (Transparent ID Layout accessibilityDescription	Value Container) ROOTUIELEMENTCONTAIN FlowLayout
	Properties (Transparent ID Layout accessibilityDescription defaultButtonId	Value Container) ROOTUIELEMENTCONTAIN FlowLayout
	Properties (Transparent ID Layout accessibilityDescription defaultButtonId enabled	Value Container) ROOTUIELEMENTCONTAIN FlowLayout
	Properties (Transparent ID Layout accessibilityDescription defaultButtonId enabled height	Value Container) ROOTUIELEMENTCONTAIN FlowLayout

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		Bir		
	Properties (InteractiveForm)					
	ID	IF_DYN_TBL				
	additionalArchives					
	dataSource	VIEW1.ZDYN_TBL_FORM				
	displayType	native		1		
	enabled	v				
InteractiveForm: IF_DY	height	900px				
	paisource					
	readOnly					
	templateSource	ZDYN_TBL_FORM	8			
	tooltip					
	visible	Visible	1			
	width	900px		1		
	Events					
	onSubmit	ROW_EDIT	1			
	Layout Data (FlowData)					
	cellDesign	padless				
	vGutter	none	1			

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	Din			
		Droportion (Inter	value	DIII			
		Properties (intera	Properties (InteractiveForm)				
Create Action							
Component	ZDYN_TBL						
View	MAIN						
Action	Row_Edit						
Description	Handle the row a						
Transfer III Eve							
Thansier of Lite							
			700px				
InteractiveForm	n: IF_DYN_TBL		700px				
InteractiveForm	n: IF_DYN_TBL		700px				
InteractiveForr	n: IF_DYN_TBL	width Events onSubmit Layout Data (Fie	700px				
InteractiveForm	n: IF_DYN_TBL		700px wData) padless				

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 = 1TO= 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. $ROW_COUNT = ROW_COUNT + 1.$ ENDLOOP. 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.



endmethod.

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

Wah Duppro Comp. / Intf		Plug	Name	DEFAULT		
vveb Dynpro Cor	np. / inu. 🕙 🗐	Help	Link			
ZDYN_DEMO ↓ IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII		Handling of Messages Show Message Component on Demand Always Display Message Component				
Z dia ZDYN_DEM ²	Create <u>C</u> hange Display Create/Change Configuration		Web Dynpro Component (Interface)			
 Image: Solution of the solution			V <u>i</u> ew <u>W</u> indow Custom Controller		l on d o	
	Check <u>A</u> ctivate	•	<u>Service</u> Ca Mime Obje	ll ect	- -	
	Cop <u>y</u> Re <u>n</u> ame Delete		Web Dynp	ro Application	u.o	
	Other Functions	÷				

Show Message Cor Object Name Des Always Display Me 🗢 🚓 ZDYN DEMO Dynai COMPONENTCONTROLLER Component Interface Administration Data Views Created By WIEW1 Last changed by Windows Web Dynpro Applications Package ▶ Z zdyn_d age Create Change Display Þ Create/Change Configuration Check Test Delete Display Object Directory Entry Change Package Assignment Write Transport Entry

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

The result would look like this when executed.



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.