

How-to Guide  
SAP NetWeaver '04



# How To... Create Online and Offline Forms in Web Dynpro for Java

(Tips and Tricks on Interactive  
Forms Development Based on  
Adobe Software - Part 1)

Version 1.01 - January 2006

Applicable Releases:  
SAP NetWeaver '04  
SAP Web Application Server 6.40

© Copyright 2006 SAP AG. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP AG. The information contained herein may be changed without prior notice.

Some software products marketed by SAP AG and its distributors contain proprietary software components of other software vendors.

Microsoft, Windows, Outlook, and PowerPoint are registered trademarks of Microsoft Corporation.

IBM, DB2, DB2 Universal Database, OS/2, Parallel Sysplex, MVS/ESA, AIX, S/390, AS/400, OS/390, OS/400, iSeries, pSeries, xSeries, zSeries, z/OS, AFP, Intelligent Miner, WebSphere, Netfinity, Tivoli, and Informix are trademarks or registered trademarks of IBM Corporation in the United States and/or other countries.

Oracle is a registered trademark of Oracle Corporation.

UNIX, X/Open, OSF/1, and Motif are registered trademarks of the Open Group.

Citrix, ICA, Program Neighborhood, MetaFrame, WinFrame, VideoFrame, and MultiWin are trademarks or registered trademarks of Citrix Systems, Inc.

HTML, XML, XHTML and W3C are trademarks or registered trademarks of W3C®, World Wide Web Consortium, Massachusetts Institute of Technology.

Java is a registered trademark of Sun Microsystems, Inc.

JavaScript is a registered trademark of Sun Microsystems, Inc., used under license for technology invented and implemented by Netscape.

MaxDB is a trademark of MySQL AB, Sweden.

SAP, R/3, mySAP, mySAP.com, xApps, xApp, SAP NetWeaver, and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP AG in Germany and in several other countries all over the world. All other product and service names mentioned are the trademarks of their respective companies. Data

contained in this document serves informational purposes only. National product specifications may vary.

These materials are subject to change without notice. These materials are provided by SAP AG and its affiliated companies ("SAP Group") for informational purposes only, without representation or warranty of any kind, and SAP Group shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP Group products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.

These materials are provided "as is" without a warranty of any kind, either express or implied, including but not limited to, the implied warranties of merchantability, fitness for a particular purpose, or non-infringement. SAP shall not be liable for damages of any kind including without limitation direct, special, indirect, or consequential damages that may result from the use of these materials.

SAP does not warrant the accuracy or completeness of the information, text, graphics, links or other items contained within these materials. SAP has no control over the information that you may access through the use of hot links contained in these materials and does not endorse your use of third party web pages nor provide any warranty whatsoever relating to third party web pages.

SAP NetWeaver "How-to" Guides are intended to simplify the product implementation. While specific product features and procedures typically are explained in a practical business context, it is not implied that those features and procedures are the only approach in solving a specific business problem using SAP NetWeaver. Should you wish to receive additional information, clarification or support, please refer to SAP Consulting.

Any software coding and/or code lines / strings ("Code") included in this documentation are only examples and are not intended to be used in a productive system environment. The Code is only intended better explain and visualize the syntax and phrasing rules of certain coding. SAP does not warrant the correctness and completeness of the Code given herein, and SAP shall not be liable for errors or damages caused by the usage of the Code, except if such damages were caused by SAP intentionally or grossly negligent.

# 1 (Business) Scenario

Have you ever think that you can have your users fill out a PDF form and have the inputs automatically kick-off some business processes in your backend? In NetWeaver 04, SAP provides a new technology called Adobe Interactive Forms which allows such business scenario. In this series of how-to paper, we will show you a step-by-step guideline on developing a custom application using the Adobe Interactive Forms technology.

In this paper, in particular, we will focus on the creation of online interactive forms and offline form.

## 2 Introduction

In SAP Web Application Server 6.40 (SAP NetWeaver 04), it provides a new service called Adobe Document Service for rendering and generating Adobe Forms. In this how-to paper, we will show you the basic steps you need for creating an Adobe Interactive Form for online and offline scenarios.

*Remarks: the info in this document is based on the NetWeaver 04 release of SAP Web Application Server 6.40.*

### **Prerequisites**

This paper assumes the readers have already had the basic concepts on Web Dynpro development. The details of Web Dynpro's terminologies and concepts will not be explained in this paper. Users can refer to the corresponding Web Dynpro documents for details.

Users should have had SAP Web Application Server 6.40 – NetWeaver Developers Workplace installed on his/her development machine. Details on installing NetWeaver Developers Workplace can be found in the installation guide of SAP Web Application Server 6.40.

Adobe Document Service should have been installed and ready for use. The detail on the installation and configuration of Adobe Document Service can be found in the installation guide. On the developer's machine, you need to have Adobe Reader (>= 6.0.2) and ActiveX Control Framework installed. For details, please see the installation guideline.

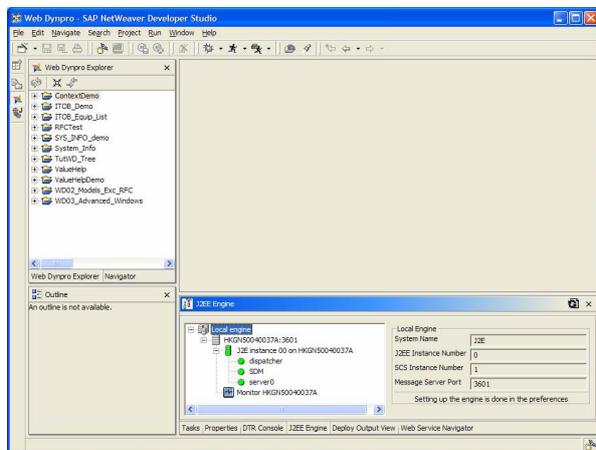
## 3 The Step By Step Solution

### 3.1 PART 1 - Creating an Online Interactive Form

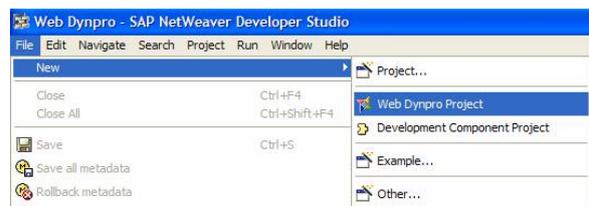
In this part of the paper, it will show the steps for creating an online interactive form in Web Dynpro. As an example, we will build a simple interactive Travel Expense form which takes some user inputs from the form and have Web Dynpro display the info in a second view.

#### 3.1.1 Creating a Web Dynpro application

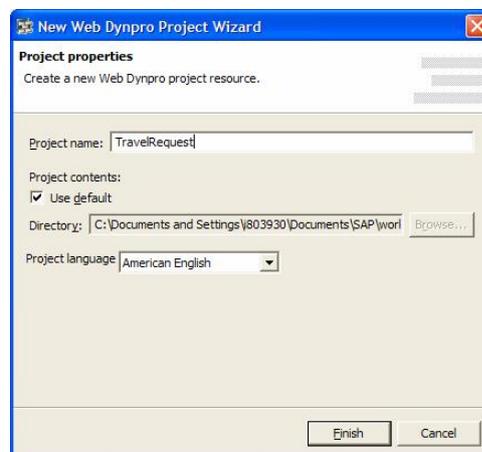
1. Start the SAP Web Application Server and SAP NetWeaver Developer Studio if you haven't done so.



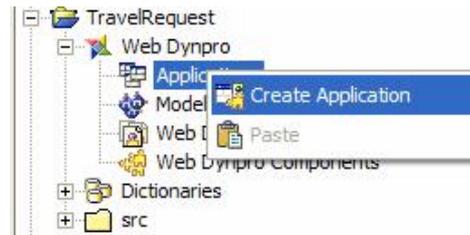
2. In SAP NetWeaver Studio, click on the file menu File -> New -> Web Dynpro Application.



3. In the Dialog Box, enter the TravelRequest as the name of the Web Dynpro project name. Then click on Finish.



- Expand the TravelRequest project inside Web Dynpro Explorer, right click on Applications and choose Create Application.



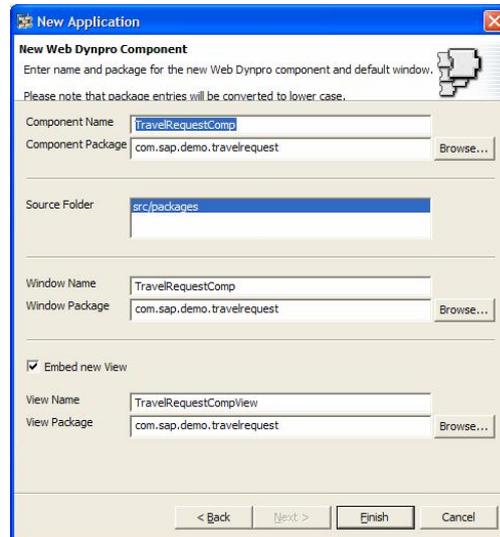
- Enter the following info for creating the application:

- Name:  
TravelRequestApp
- Package:  
com.sap.demo.travelrequest

After that, click on Next.

- Choose to Create a New Component on the following screen. Click on Next.

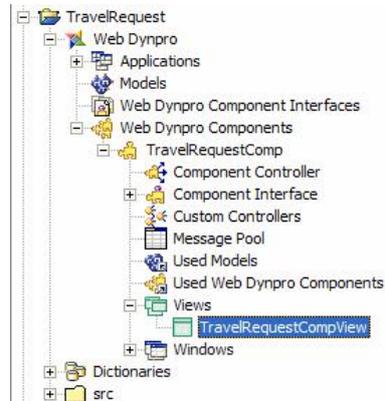
- Change the component name to TravelRequestComp. Click on Finish.



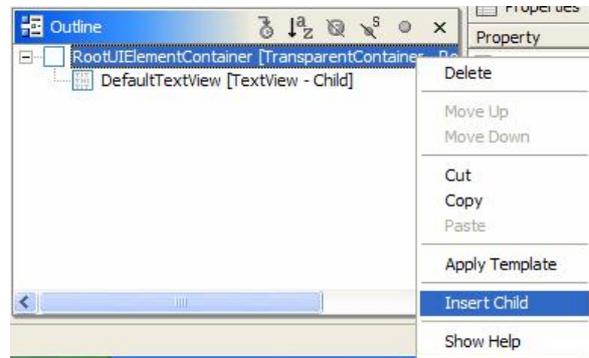
- This will create the Web Dynpro application and a component inside. A default view for the component will also be generated.

### 3.1.2 Embedding an Interactive Form to a Web Dynpro View

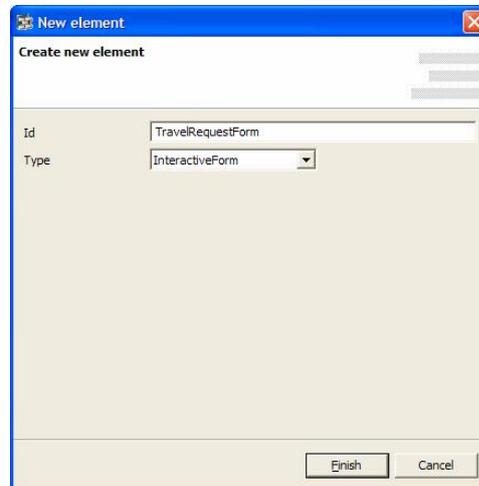
9. Open the view TravelRequestCompView by double clicking on it.



10. In the Layout of the view, right click on the RootUIElementContainer inside the Outline view and select Insert Child.



11. In the Type field, choose InteractiveForm from the drop-down list. Enter TravelRequestForm as the Id. Click on Finish.



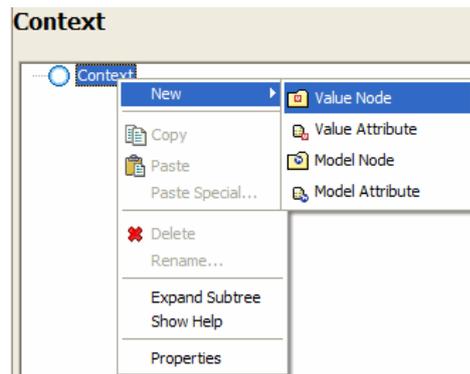
12. Go to the Properties of TravelRequestForm, change the Height to 500px and Width to 600px.

### 3.1.3 Defining context for the interactive form

We now need to define what data we want to show in the interactive form. Just like Web Dynpro, the data are controlled by the Context in Web Dynpro. In this example, we will create several attributes in the Web Dynpro Context and will later on map these attributes to the corresponding fields in the interactive form

13. Click on the Context tab underneath the view.

14. Right click on Context -> New -> Context Node.

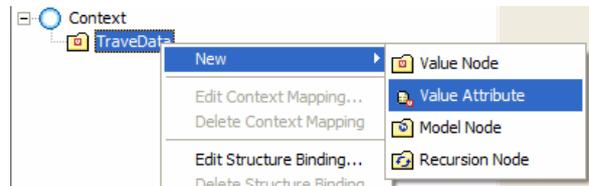


15. Enter the name as TravelData, then click on Finish.

16. Go to the properties of the TravelData node, change the Cardinality to 1..1. This is just to make sure we have at least 1 instance of the data.



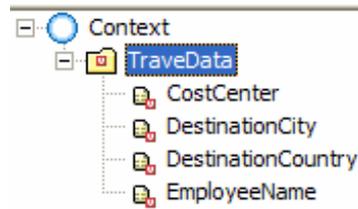
17. Right click on the newly created TravelData node, this time choose New -> Value Attribute.



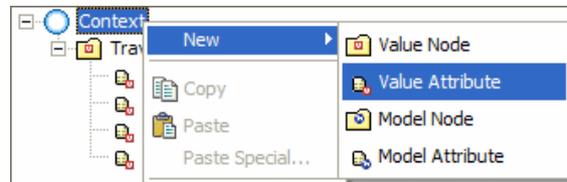
18. Enter the attribute name as EmployeeName. Click on Finish to create the attribute.



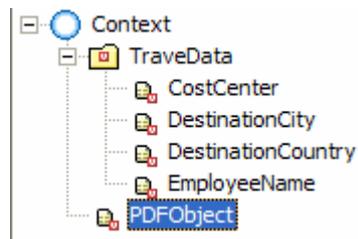
19. Repeat step 17 and 18 to create 3 more fields with names CostCenter, DestinationCity and DestinationCountry. The result Context should look like below.



20. We need to create a PDF binary object field in the Context as well. This will be used for the interactive form as a temporary placeholder for generating the form. To do it, right click on Context again and choose Value Attribute.



21. Enter the name as PDFObject and click on Finish. The attribute will be created as below.



22. Go to the Properties of the PDFObject by right click on it and select Properties. In the Properties, change the Type to binary.

Property	Value
calculated	false
name	PDFObject
readOnly	false
structureElement	
type	binary

23. Click on the Layout tab of the view.

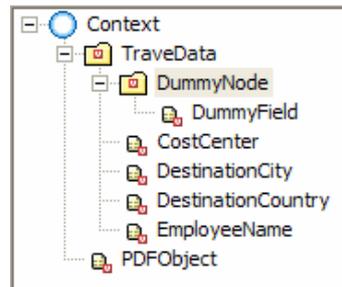
24. Now we need to specify in the Interactive Form Properties to bind to these context. To do it go to the Properties of the TravelRequestForm (right click on it and choose Properties).

25. Specify the followings in the properties.

- dataSource: TravelData
- pdfSource: PDFObject

Property	Value
ElementProperties of UIElement	
dataSource	TraveData
enabled	true
height	300px
id	TravelRequestForm
mode	updateDataInPdf
pdfSource	PDFObject
templateSource	TravelRequestCompView_TravelRequestForm.xdp

26. If you are running Web AS 6.40 (< SP8), there is a limitation that Adobe Form cannot deal with a simple flat context structure like above. To workaround this problem, you can create a dummy node (with cardinality 1..1) underneath the TravelData node with a dummy attribute with it. The context will look like below. This problem is fixed in SP8.

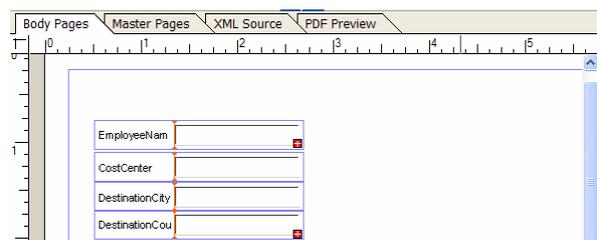
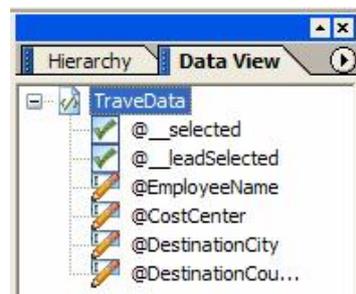


### 3.1.4 Editing the form layout

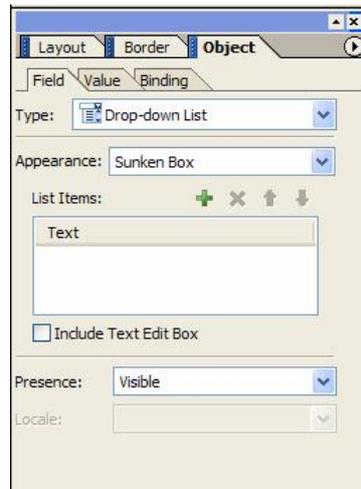
27. Now we can define and edit the UI elements to be on the form. To do it, right click on TravelRequestForm and choose Edit. This will open the form with the Adobe Designer inside the NetWeaver Developer Studio
28. Expand the Data View inside the Adobe Designer, you will see the attributes created from the Web Dynpro context.



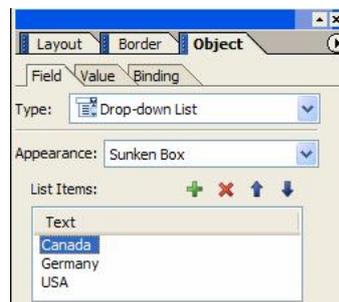
29. To create Text Fields on the form for these attributes, drag these attributes from the Data View onto the form one after one. The resulting form should look like below.



30. Instead of just having the data display/input from a text field, we can change the properties to have another UI element associate with it. For example, we now want the DestinationCountry field to be inputted from a Drop-down List instead of from a Text Field. To do it, select the DestinationCountry field by clicking on it. On the right hand column under the Object tab -> Field tab, change the Type to Drop-down List.



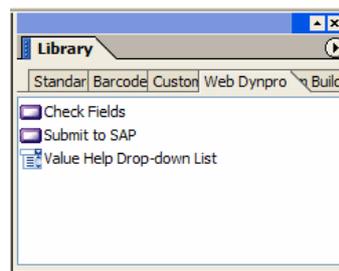
31. We can also predefine certain values for the Drop-down List. To do it, under the List Items in the Field tab, click on the + button and you can define value for the list. Try to create 3 countries like below.



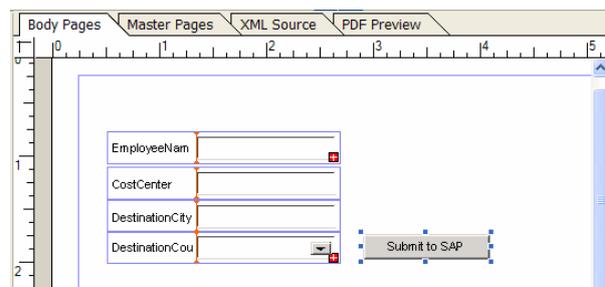
### 3.1.5 Defining actions for the form

We can put buttons on the Interactive Form and specify actions to it on user's clicking on the button. In this example, we will create a Submit button on the form for form submission

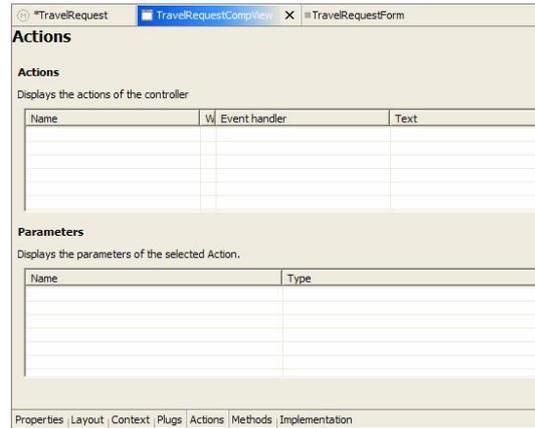
32. Still inside the Adobe Designer, go to the Web Dynpro tab under the Library tab window.



33. Drag the Submit to SAP element onto the Form like below.

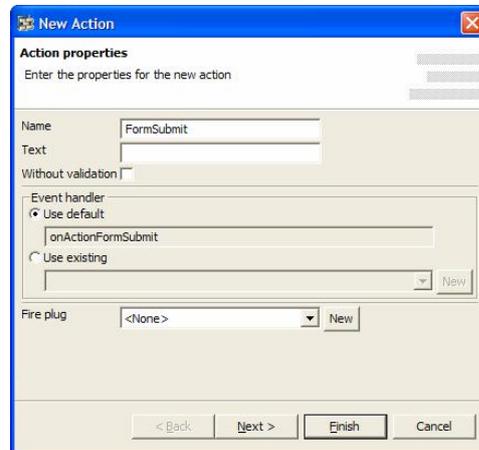


34. Now we need to action to this button. To do it, go back to the TravelRequestCompView and click on the Actions tab.

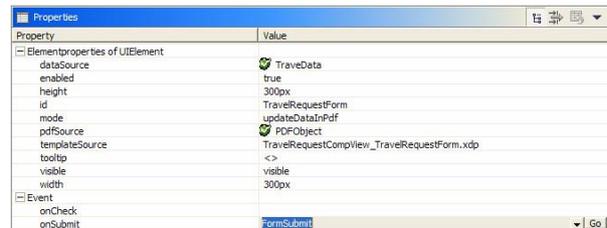


35. Under Actions, click on the New button.

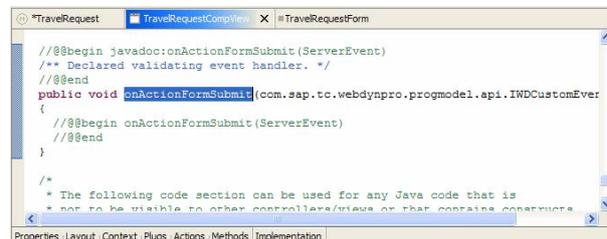
36. Create an Action called FormSubmit. Click on Finish when done.



37. Click on the Layout tab again and go to the Properties of the TravelRequestForm (by right click on it and choose properties). Under Event -> onSubmit, choose FormSubmit.



38. To do the code implementation of the action, click on the Implementation tab. Then go to the onActionFormSubmit() method.



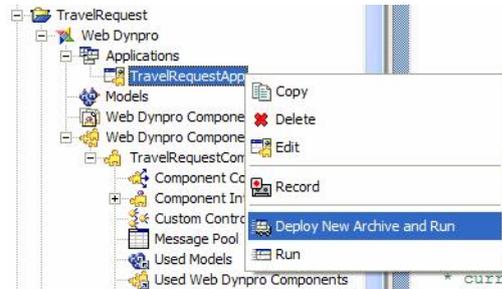
39. As an example, we will just display the inputted data as a message from Web Dynpro. Enter the following codes in the onActionFormSubmit() method.

*<Please refer to Appendix 1 for the sample codes>*

40. Click on Save All Meta-data.

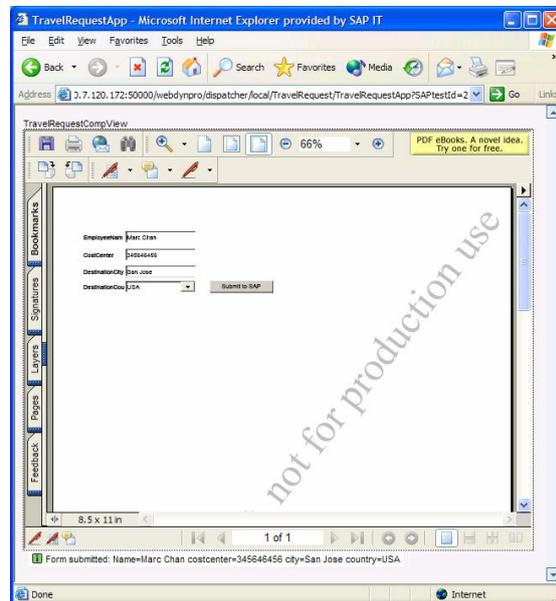
### 3.1.6 Deploying and testing the application

41. We can now deploy and test runs the application. Right click on TravelRequestApp and choose Deploy New Archive and Run. Provide your SDM password when prompt. This will rebuild the project, generate the archive and deploy the application.



42. Your default Web Browser will be pop up with the application and form displayed. Enter some testing data and click on the Submit to SAP button for testing.

*Remarks: if you get a Render / Timeout error on this page, you should check your J2EE Engine settings to the Adobe Document Service. Also, if the ADS cannot recognize your machine name from the network, you may need to run the application with its IP address instead of its machine name*



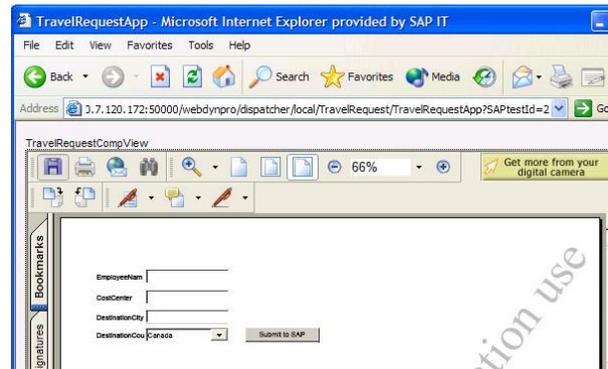
## 3.2 PART 2 - Creating an Offline Form

In this part of the paper, you will modify the interactive form created in part 1 to take care of offline scenario.

### 3.2.1 Obtaining a PDF with Adobe Reader's right

43. Adobe Document Service can generate Adobe Forms with Adobe Reader's Right so that users can use Adobe Reader to render the form.

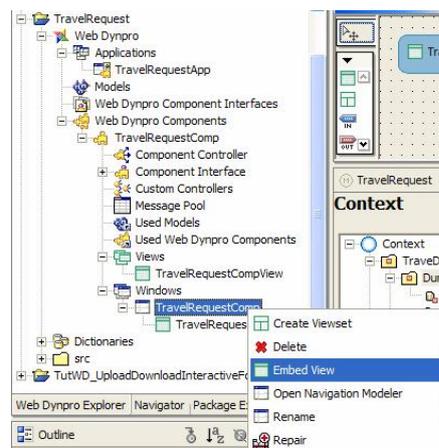
The first thing you need to do for offline scenario is to provide an offline form with Adobe Reader's right. To convert our existing form to an offline form with the right, just run the application again. Then click on the Save button from the Adobe Reader when displaying the form, and save the PDF file for offline download.



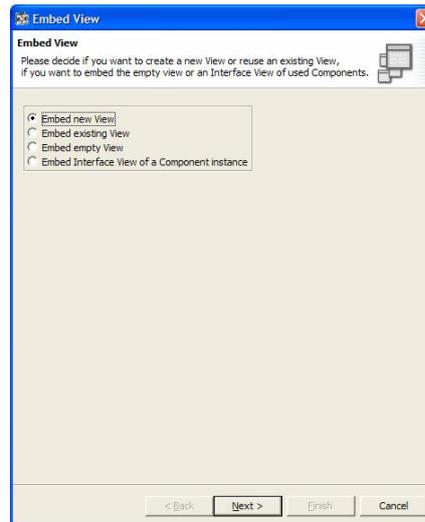
### 3.2.2 Creating a Web Dynpro view for form upload

In offline scenario, users fill out the form offline and upload the form back to the Web Dynpro application for submission. To do it, we will create an UploadView in Web Dynpro to take care of the upload.

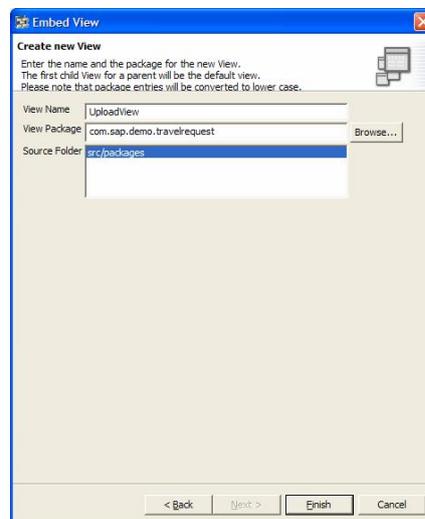
44. Go back to the TravelRequest project, expand until you see TravelRequestComp under Windows. Right click on Windows and select Embed View.



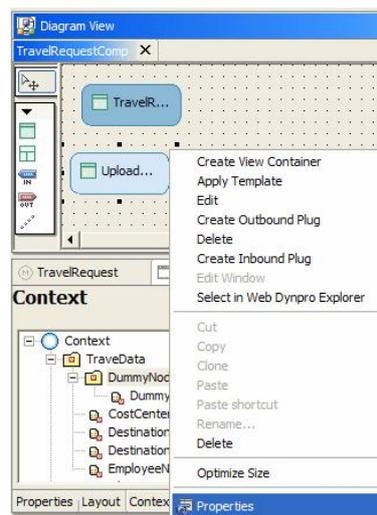
45. Select Embed new View, click on Next.



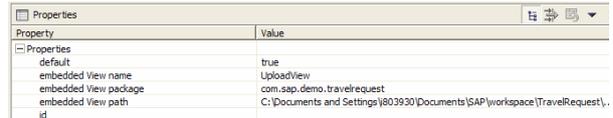
46. Provide View Name as UploadView. Click on Finish.



47. In the Diagram View, right click on the box of UploadView and choose Properties.

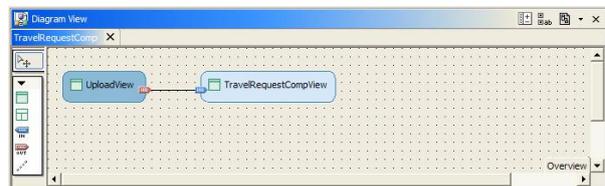


48. In the properties, change the Default field to True.



49. We need to specify the flow that after uploading the form, users can display it as if an online interactive form. To do it, we need to define a navigation flow from the UploadView to the TravelRequestCompView.

50. Create an Outbound Plug called OutToDisplay on UploadView. Create an Inbound Plug called InFromUpload on the TravelRequestCompView. Create a Navigation Link between these plugs. The result should look like below.

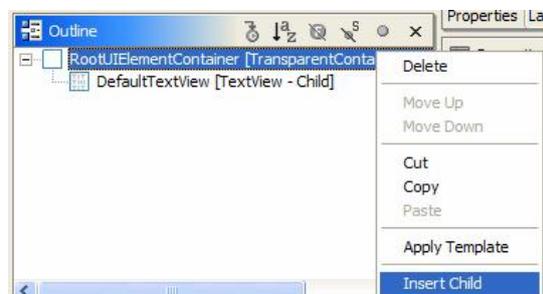


### 3.2.3 Editing the Upload View

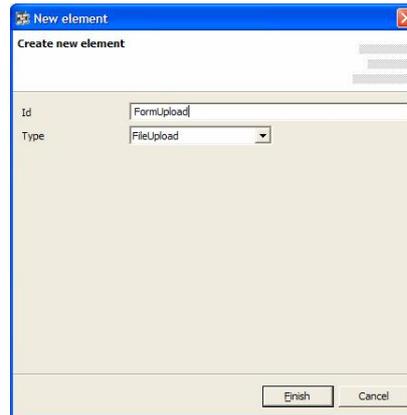
51. In the UploadView, we need to provide the user interface for the users to upload the filled form and display it.

Double click on UploadView to open it. Go to the Layout tab.

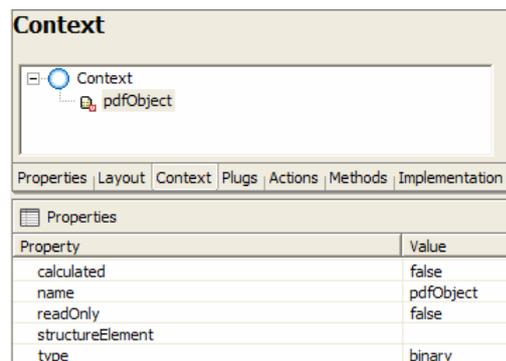
52. Place a FileUpload UI element on the view. To do it, right click on RootUIElementContainer and choose Insert Child.



53. Choose the Type FileUpload. Specify the Id as FormUpload. Click on Finish.



54. Click on the Context tab. Create a Value Attribute called pdfObject under the root with binary type.



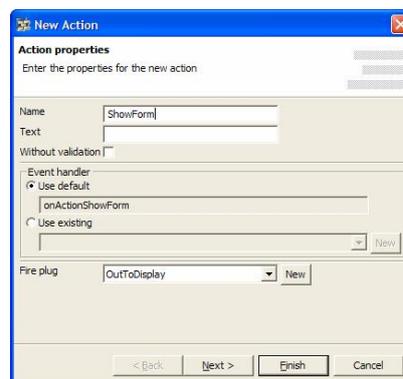
55. Click on the Layout tab again, in the Properties of FormUpload, set the Data field to pdfObject.



56. Create another Button on the UploadView called DisplayForm.

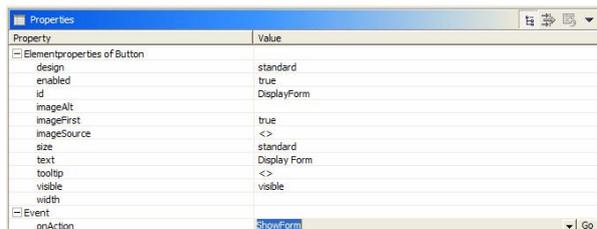


57. Go to the Actions tab, create an action called ShowForm. To do it click on the New button. Input ShowForm as the action name. On the Fire Plug field, choose OutToDisplay. Click on Finish.



58. Go back to the Layout tab, specify the followings on the Properties of the DisplayForm button.

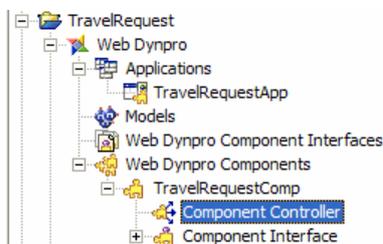
- Text: Display Form
- onAction: ShowForm



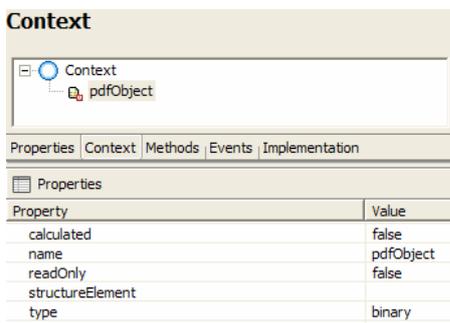
### 3.2.4 Transferring the uploaded form for display

To make the uploaded form be able to display on the TravelRequestCompView, we need to create a Web Dynpro Controller to control the flow of the pdfObject.

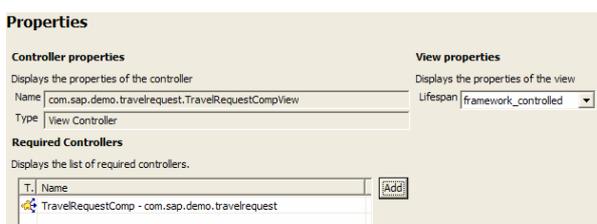
59. Open the Component Controller by double clicking on Component Controller.



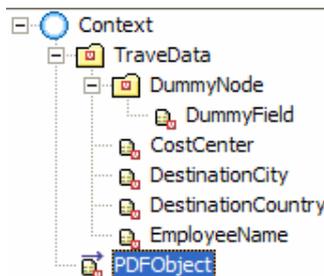
60. Under the Context, create a new Value Attribute called pdfObject with type binary.



61. Open the TravelRequestCompView and go to the Properties tab. Add the TravelRequestComp – com.sap.demo.travelrequest as the controller for this view.

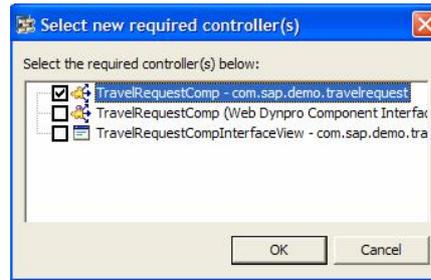


62. Go to the Context tab. Right click on pdfObject on the left and choose Edit Context Mapping. Choose pdfObject on the following dialog box.



63. Go back to UploadView and click on the Properties tab.

64. Under Required Controller, click on Add. Choose TravelRequestComp – com.sap.demo.travelrequest. Click OK.



65. Go to the Implementation tab. Add the following line of code to the method wdDoInit.

```
wdContext.getNodeInfo().getAttribute(
"pdfObject").getModifiableSimpleType();
```

66. Add the following line of code right before wdThis.wdFirePlugOutToDisplay() in the method onActionShowForm().

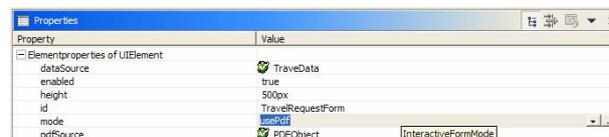
```
wdThis.wdGetTravelRequestCompController().wdGetContext().currentContextElement().setPdfObject(wdContext.currentContextElement().getPdfObject());
```

### 3.2.5 Modifying the online view to display the uploaded form

We now need to specify in the TravelRequestCompView to display the uploaded form and not the online interactive form.

67. Go to the Layout of TravelRequestCompView.

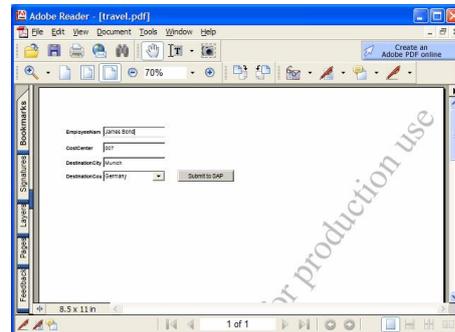
68. Go to the Properties of TravelRequestForm. Change the Mode to usePdf.



69. Click on Save All Meta-data.

### 3.2.6 Deploying and testing the offline scenario

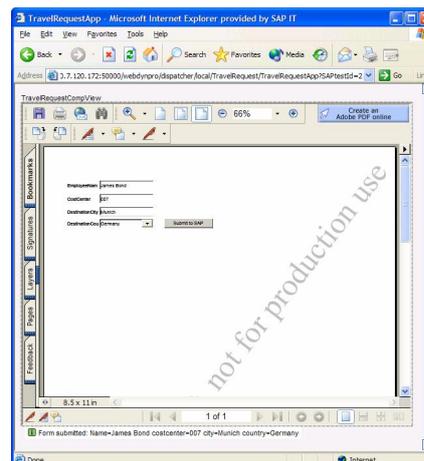
70. Fill out the PDF form from section 2.1 using your Adobe Reader while you are offline. Save the input by clicking on the regular Save button.



71. Deploy and Run the TravelRequestApp.

72. Click on Browse and choose the PDF file you filled out offline. Click on the Display Form button will display the filled form.

73. You can click on the Submit to SAP Button for submission testing.



## 4 Appendix

### 4.1 4.1 Appendix 1 – sample code for form submit

```
public void onActionFormSubmit(com.sap.tc.webdynpro.progmodel.api.IWDCustomEvent
wdEvent )
{
    //@@begin onActionFormSubmit(ServerEvent)
    String name = wdContext.currentTraveDataElement().getEmployeeName();
    String costcenter = wdContext.currentTraveDataElement().getCostCenter();
    String city = wdContext.currentTraveDataElement().getDestinationCity();
    String country =
        wdContext.currentTraveDataElement().getDestinationCountry();
    wdThis.wdGetAPI().getComponent().getMessageManager().reportSuccess(
        "Form submitted: Name="
        + name
        + " costcenter="
        + costcenter
        + " city="
        + city
        + " country="
        + country);
    //@@end
}
```

[www.sdn.sap.com/irj/sdn/howtoguides](http://www.sdn.sap.com/irj/sdn/howtoguides)