

# Consumption of WebDynpro Component as blackbox in Visual Composer Application



## Applies to:

Web Dynpro for Java applications for the SAP NetWeaver Composition Environment 7.1

## Summary

This article explains the basic steps required for creating a WebDynpro Development Component and to consume it as blackbox in the Visual Composer Application.

**Author:** Saranyaa VB

**Company:** SAP Labs India

**Created on:** 23 December 2008

## Author Bio



Saranyaa works as a Quality Engineer at SAP Labs for the last six months and is part of the WebDynpro for Visual Composer Quality team.

## Table of Contents

Prerequisite.....	3
Importing a Project Template.....	3
Initial Project Structure.....	3
Creating the Component Controller Context .....	5
Creating the Interface Controller Context: .....	7
Creating the methods and events for Component Controller .....	8
Creating methods and events for Interface Controller.....	9
Context Mapping.....	10
Designing View .....	11
Binding the UI element to FormView .....	11
Add InteractiveForm UI Element.....	12
Define Form Layout with Adobe Lifecycle Designer .....	13
Consuming the WebDynpro Component in Visual Composer .....	14
Running the VC application .....	14
Related Content.....	16

## Introduction

The objective of this article is to create Webdynpro development Component and consume it as a blackbox in Visual Composer. Since Visual Composer does not provide the facility for saving a form in PDF format, we can add a WebDynpro Component with the required functionality as a blackbox and connect it to our visual composer model.

This tutorial describes how to create a webdynpro form which accepts employee personal details as input and displays it as Adobe interactive form in Visual Composer Application.

## Prerequisite

The SAP NetWeaver Developer Studio including Adobe Lifecycle Designer (forms design tool) is installed on your computer.

1. You have access to the SAP J2EE Engine
2. Adobe Reader 8.1 is installed on your computer.
3. The Adobe document services are configured on the SAP J2EE Engine you are using.
4. Basic knowledge of developing Web Dynpro applications
5. Ensure SLD is configured








## Importing a Project Template

[Download](#) the project template and save the zip file to any directory on your local hard disk or directly in the work area of the SAP NetWeaver Developer Studio.

1. Unpack the content of the ZIP file TutWD\_Blackbox\_Init.zip.
2. Start the SAP NetWeaver Developer Studio.
3. Import the Web Dynpro project TutWD\_Blackbox\_Init
  - a. In the menu, choose File → Import...
  - b. In the next window, choose Existing Project into Workspace and click Next to confirm.
  - c. Choose Browse, open the folder in which you unpacked the project TutWD\_Blackbox\_Init, and select the project.
  - d. Confirm by choosing Finish

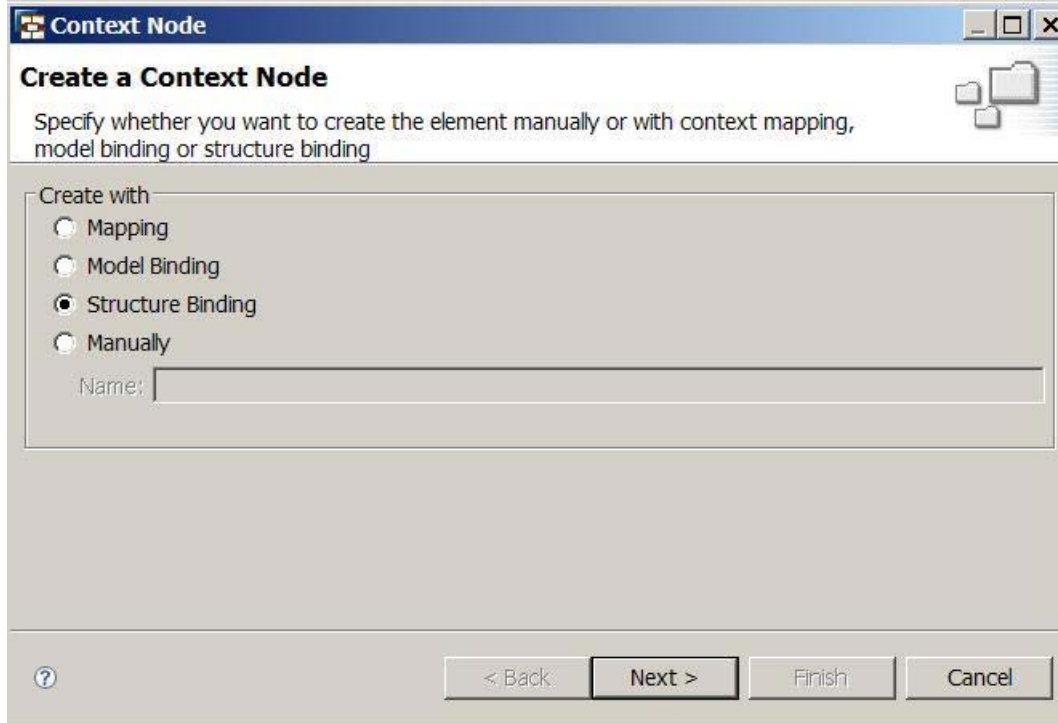
## Initial Project Structure

After you have imported the Web Dynpro project template TutWD\_Blackbox\_Init, the following project structure is displayed in the Web Dynpro Explorer:

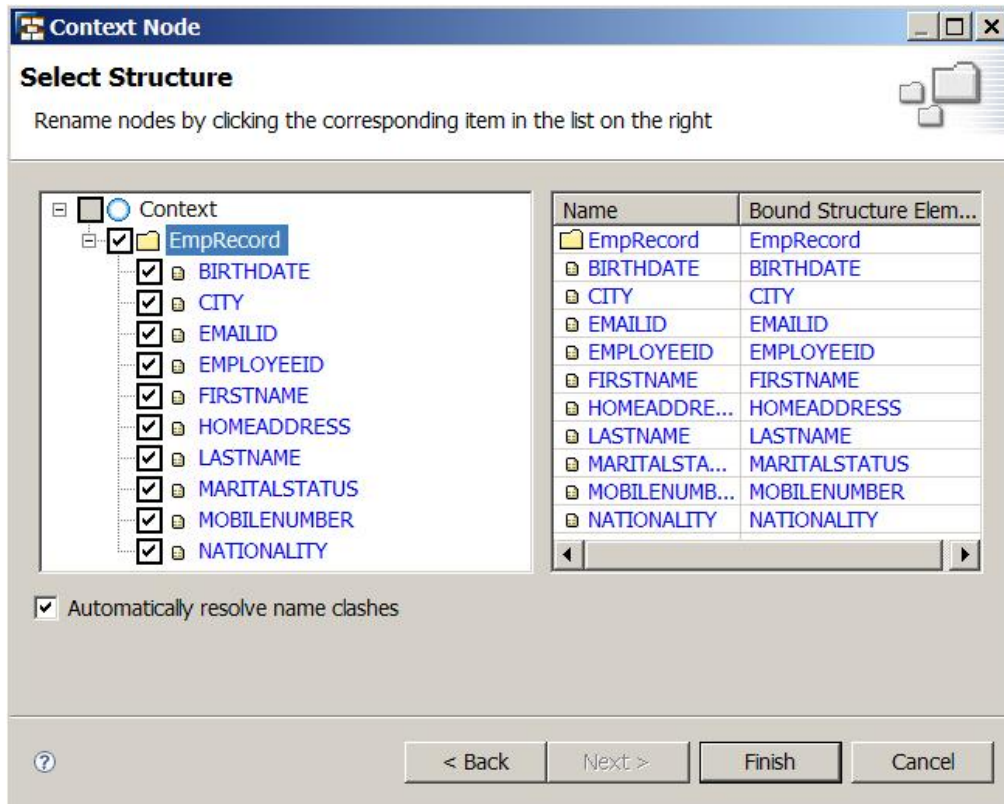
Web Dynpro Project Structure	
	<b>Web Dynpro Project:</b> TutWD_Blackbox_Init
	<b>Web Dynpro Component:</b> EmpRecordComp  This is the Web Dynpro Component that contains our entire application.
	<b>View:</b> EmpRecordView
	This View is provided for displaying employee personal details in a form
	<b>View:</b> PDFView  This view is provided for displaying the employee personal details in PDF form.
	<b>Window:</b> EmpRecordCompWindow  Initially only contains the EmpRecordView.
	<b>Local Dictionary Types</b>
	<b>EmpRecord:</b> Structure Type  Structure type defining EmpRecord with 10 fields: EMPLOYEEID,FIRSTNAME,LASTNAME,NATIONALITY,MARITALSTATUS,BIRTHDATE, HOMEADDRESS,CITY,MOBILENUMBER,EMAILID

## Creating the Component Controller Context

1. Choose TutWD\_Blackbox\_Init -> Web Dynpro -> Web Dynpro Components ->EmpRecordComp-> Component Controller.
2. With the right mouse key, click the Component Controller and choose Controller Editor.
3. Now create the context node for the component controller with the right mouse key, and choose New → Node.
4. Activate Create with structure binding, and click Next.



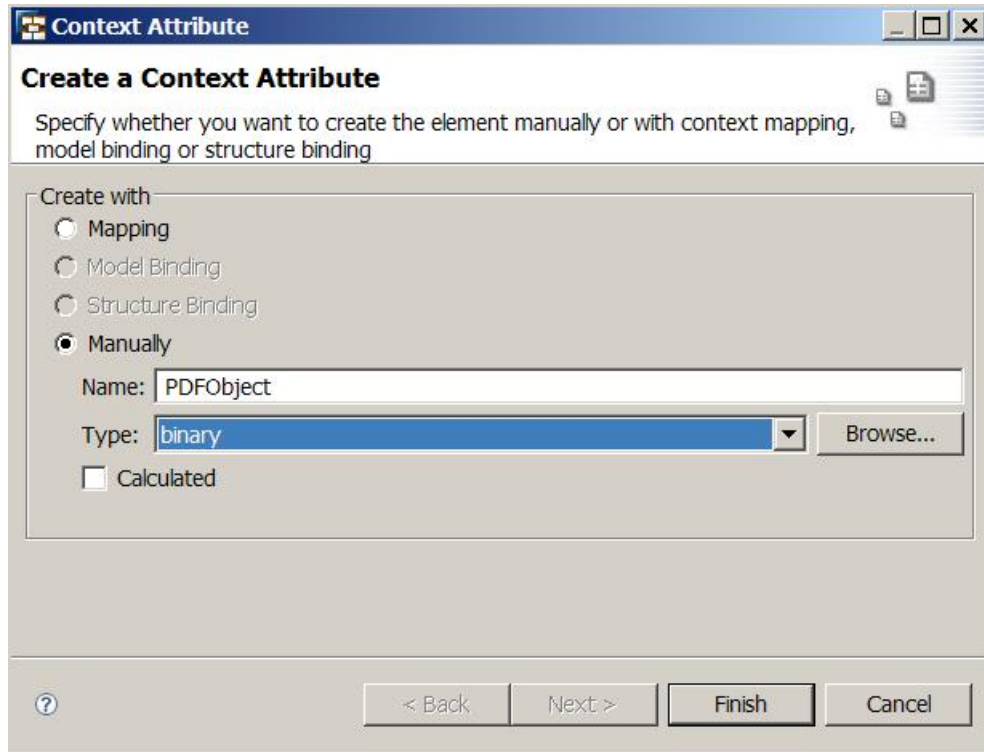
5. Start Dictionaries → Local Dictionary → com.sap.demo.tutwd\_blackbox\_init, choose EmpRecord, and click Next. In the next window you can choose from the node attributes of the context.
6. Select all the available attributes by activating EmpRecord and choose Finish.



7. Select the properties of the EmpRecordNode, set the “Input Element” to true and cardinality from default 0...n to 1...1. Otherwise, the fields for the attributes will be inactive.

Property	Value
Misc	
Name	EmpRecord
Collection Cardinality	1..1
Initialize Lead Selection	true
Input Element	true
Selection Cardinality	1..1
Singleton	true
Structure	com.sap.demo.employeeerecord.EmpRecord

8. Now create attribute of type “binary” with name “PDFObject” under the root node Context. This attribute is used as a placeholder for displaying the interactive form.

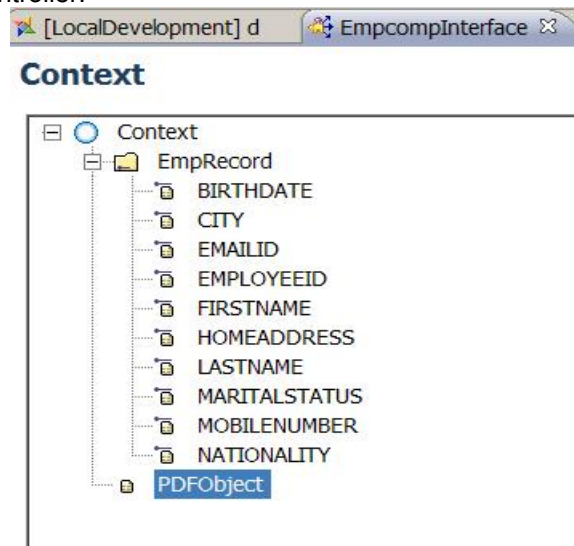


9. Save the current status of your project.

### Creating the Interface Controller Context:

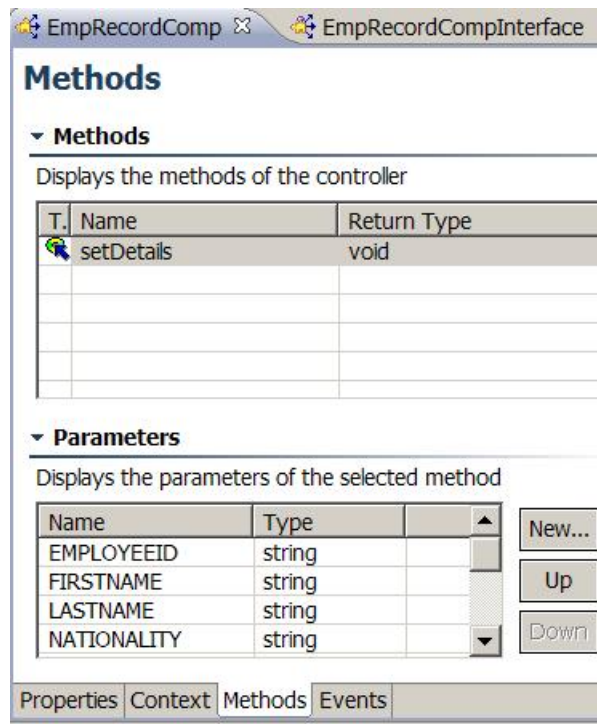
The data exposed as context in the interface controller is shared between the Embedding VC Component and the Web Dynpro Component.

1. Choose TutWD\_Blackbox\_Init -> Web Dynpro -> Web Dynpro Components ->EmpRecordComp ->Local Component Interface->Interface Controller.
2. Create context node for interface controller similar to component controller. Follow steps 2 to 8 of Component Controller.



## Creating the methods and events for Component Controller

1. Now in the component controller of the EmpRecordComp component, create method setDetails by passing attribute name of the node "EmpRecord" as parameters.



2. Add the following code for the method setDetails of the Component controller.

```
wdContext.currentEmpRecordElement().setEMPLOYEEID(EMPLOYEEID);
wdContext.currentEmpRecordElement().setFIRSTNAME(FIRSTNAME);
wdContext.currentEmpRecordElement().setLASTNAME(LASTNAME);
wdContext.currentEmpRecordElement().setNATIONALITY(NATIONALITY);
wdContext.currentEmpRecordElement().setMARITALSTATUS(MARITALSTATUS);
wdContext.currentEmpRecordElement().setHOMEADDRESS(HOMEADDRESS);
wdContext.currentEmpRecordElement().setCITY(CITY);
wdContext.currentEmpRecordElement().setBIRTHDATE(BIRTHDATE);
wdContext.currentEmpRecordElement().setMOBILENUMBER(MOBILENUMBER);
wdContext.currentEmpRecordElement().setEMAILID(EMAILID);
```

3. Now in the component controller of the EmpRecordComp component, create Event ShowMessage (String message).



The screenshot shows a web browser window with three tabs: PDFView, \*EmpRecordComp, and EmpRecordCompInterf. The main content area is titled 'Events' and contains two sections:

- Events:** A section titled 'Events' with a description 'Displays the events of the controller'. Below it is a table with columns 'Name' and 'Type'. One event is listed: 'ShowMessage'.
- Parameters:** A section titled 'Parameters' with a description 'Displays the parameters of the selected event'. Below it is a table with columns 'Name' and 'Type'. One parameter is listed: 'Message' with type 'string'.

4. Add a method in the Component controller void raiseEvent() and add the following code:-

```
public void raiseEvent( ) {
  @@begin raiseEvent()
    IWDMessageManager msgmgr =
      wdComponentAPI.getMessageManager();
    msgmgr.reportSuccess ( "Data Submitted Successfully" );
    wdThis.wdFireEventShowMessage( "Data Submitted" );
  @@end
}
```

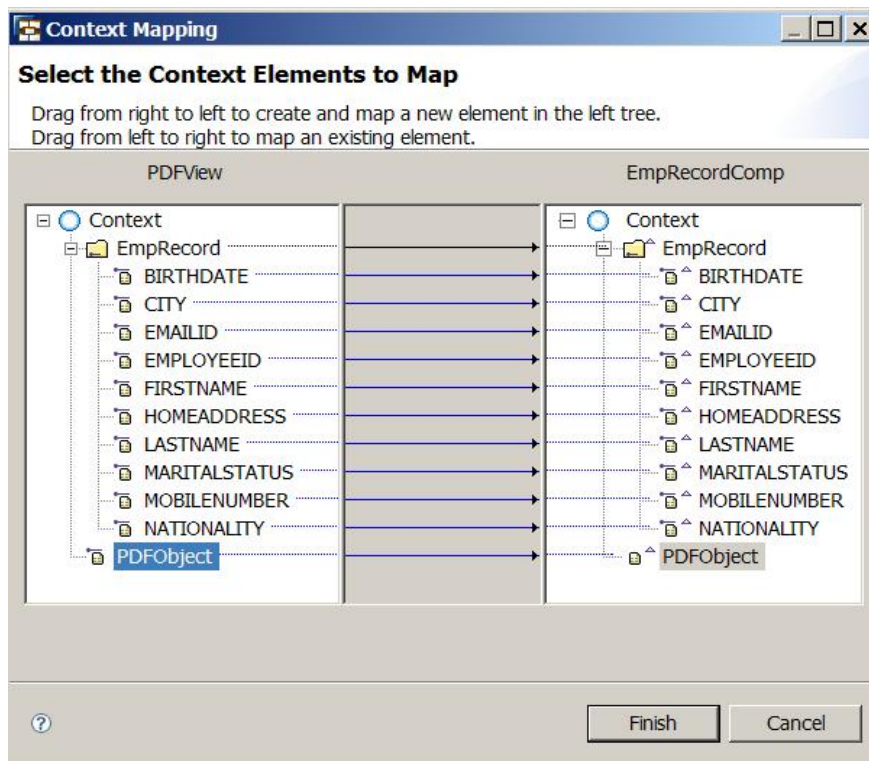
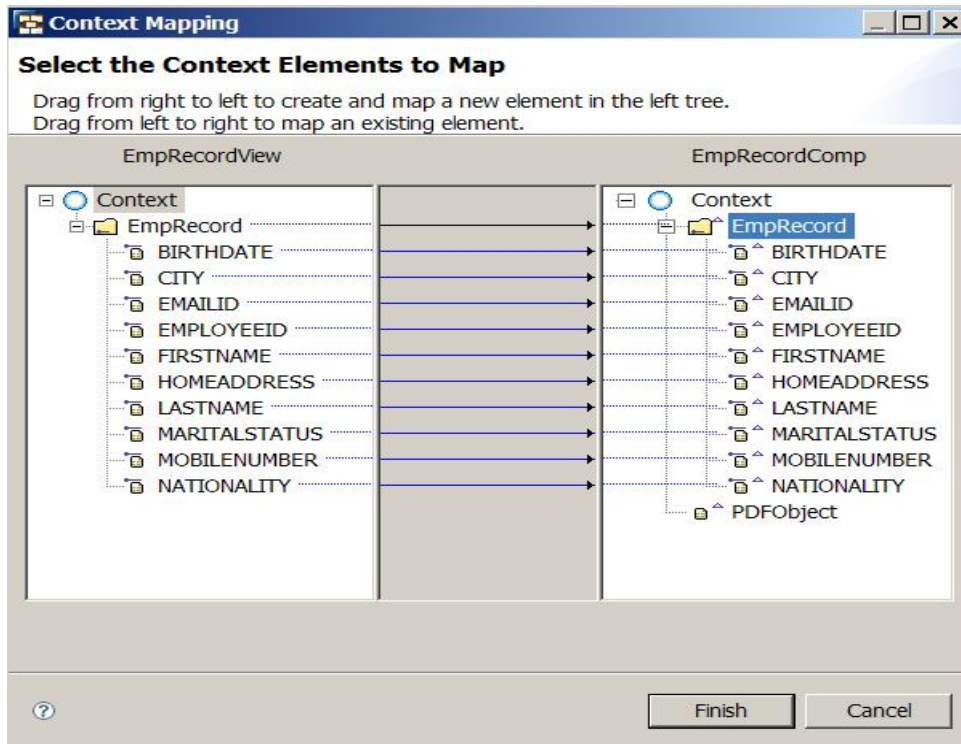
## Creating methods and events for Interface Controller

Methods defined in the component interface appear as SignalIn relays in the embedding VC Component. Events defined in the component interface appear as SignalOut relays in the embedding VC Component.

1. Create the method "setDetails" for the interface controller with required parameter similar to Component controller.
2. Similarly create Event showMessage(String message)

## Context Mapping

1. Open the data modeler of the component “EmpRecordComp”
2. Map the node “EmpRecord” in the component controller to both the views.



## Designing View

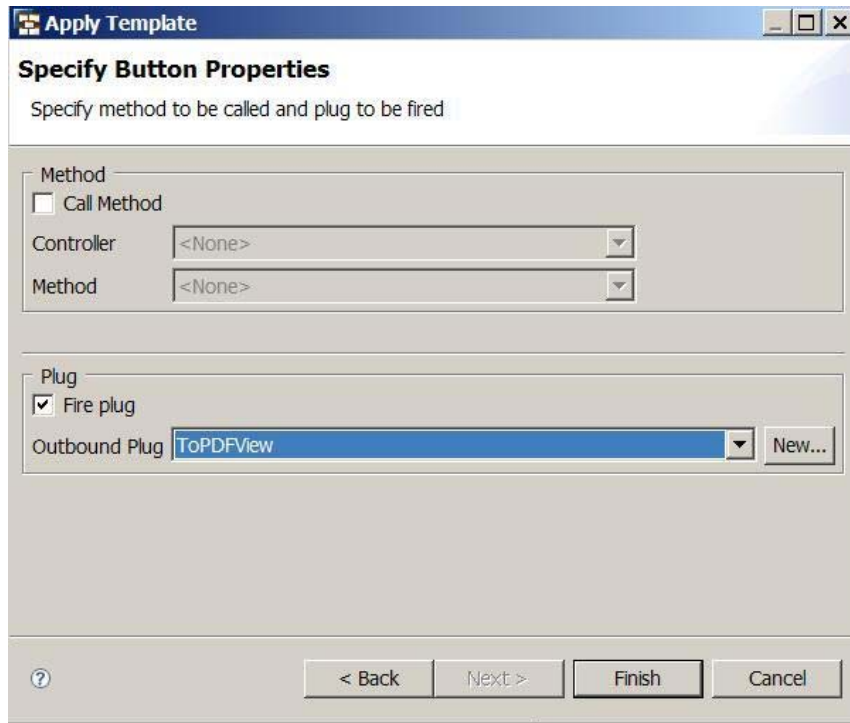
### Binding the UI element to FormView

Currently, the contexts of the views and the component controller are bound with one another. To be able to display the data in the user interface, you have to bind the form with the context of its view.

1. Double click the EmpRecordView for editing
2. In the Outline, click with the right mouse key on RootUIElementContainer and choose Apply Template. The Template wizard is started.
3. Choose Form, enter Employee Form as name, and confirm with Next ->.
4. In the EmpRecord context node, select all the value attributes, and confirm with Next >. Then confirm by pressing Finish.
5. Change the layout property from flowlayout to matrix layout.
6. For the RootUIElementContainer, set the property StretchedHorizontally to false

Employee Details	
EmployeeID:	EmpRecord/@EMPLOYEID
FirstName:	EmpRecord/@FIRSTNAME
LastName:	EmpRecord/@LASTNAME
Nationality:	EmpRecord/@NATIONALITY
MaritalStatus:	<input type="checkbox"/> Married
BirthDate:	EmpRecord/@BIRTHDATE
HomeAddress:	EmpRecord/@HOMEADDRESS
City:	EmpRecord/@CITY
MobileNumber:	EmpRecord/@MOBILENUMBER
EmailID:	EmpRecord/@EMAILID

7. Create a Button Display PDF with the onAction event "submit" and specify the plug to be fired.
8. In the Fire Plug drop down, select the plug ToPDFView and press finish. Already in the "EmpRecordCompWindow", the navigation between the view is defined



9. Add the event handler code as below.

```

public void onActionSubmit(com.sap.tc.webdynpro.progmodel.api.IWDCustomEvent
wdEvent )
{
    //@@begin onActionSubmit(ServerEvent)
    //$begin Action Button(-696529454)
    wdThis.wdGetEmpRecordCompController().raiseEvent();
    wdThis.wdFirePlugToPDFView();
    //$end
    //@@end
}

```

### Add InteractiveForm UI Element

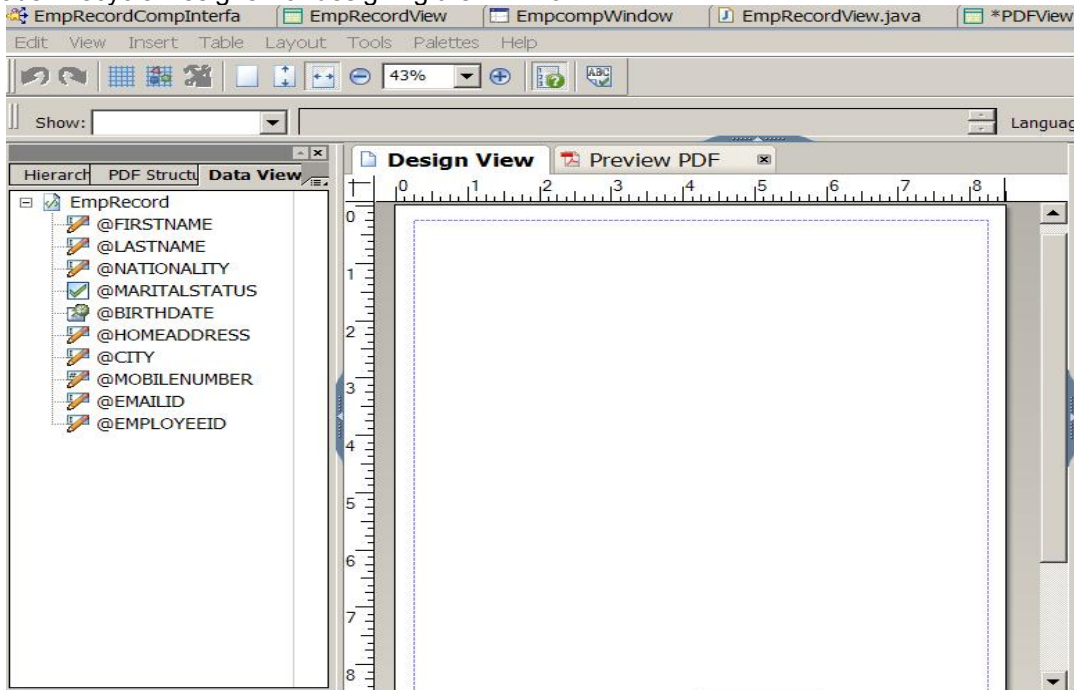
1. In the project structure, double-click the node for the PDFView (Web Dynpro → Web Dynpro Components → EmpRecordComp → Views → PDFView) and choose the Layout tab.
2. In the Outline pane, insert a child of Type InteractiveForm and Id InteractiveForm in the Group.
3. Select the InteractiveForm element in the Outline pane, switch to the corresponding Properties page and change the properties as shown below. Reference the corresponding DataSource and PdfSource from the context and set the width and height and also the display type:

Property	Value
[-] Properties [InteractiveForm]	
id	InteractiveForm
additionalArchives	
dataSource	EmpRecord
displayType	native
enabled	true
height	500px
mode	updateDataInPdf*
pdfResource	
pdfSource	PDFObject
readOnly	false

The dataSource property is used to specify the data source. Assign the EmpRecord context node to dataSource property. And bind the pdfSource property to a context attribute of the type binary.

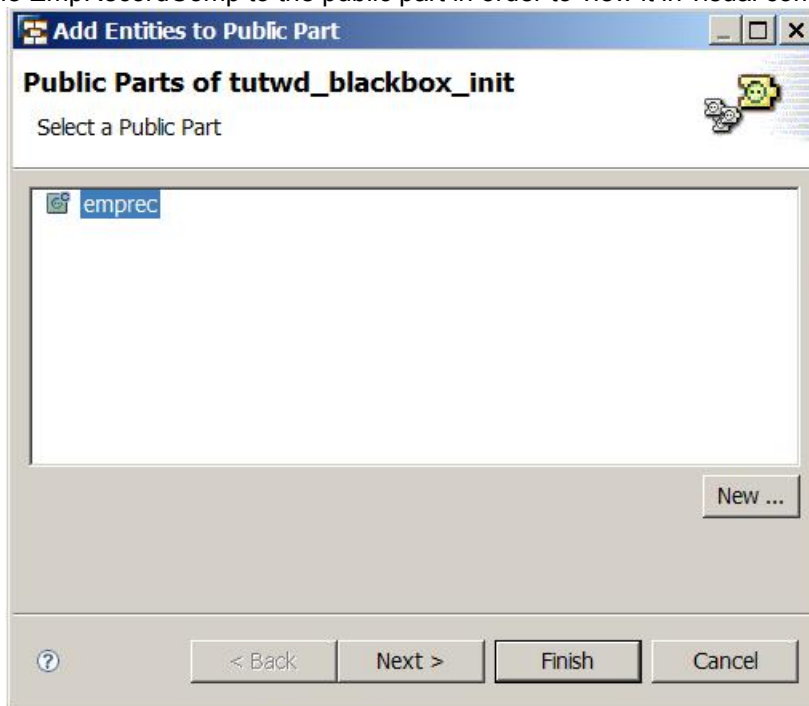
### Define Form Layout with Adobe Lifecycle Designer

Right click on the edit in the context menu of the InteractiveForm UI element in the Outline pane Use the Adobe Lifecycle Designer for designing the PDF form.



From the Data view drag and drop the required fields to the body pages of the PDF. After which you can align or do any adjustments to the look and feel of the Adobe form.

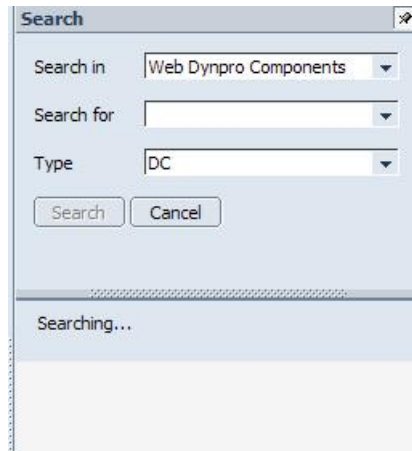
Now add the EmpRecordComp to the public part in order to view it in visual composer.



Now build and deploy the project

## Consuming the WebDynpro Component in Visual Composer

1. Open Visual composer in IDE and create model “Employee”
2. Now search for webdynpro components of the type DC in search panel.



3. Now drag and drop the blackbox component “emprecordcomp” which is inside the created dc onto the storyboard.



## Running the VC application

1. Compile and deploy the application.
2. Open the HTML view of the visual composer application
3. Give input for the form setDetails and click on the submit action. The data will be stored in the context “EmpRecordComp”

### Employee Personal Details

**EnterDetails**


Employeeid:

Firstname:

Lastname:


Nationality:

Married

Birthdate:  

Homeaddress:

City:

Mobilenumber:  

Emailid:

**EmployeeRecord**

**Employee Details**

EmployeeID:

FirstName:

LastName:

Nationality:

MaritalStatus:  Married

BirthDate:

HomeAddress:

City:

MobileNumber:

EmailID:

**ShowMessage**

Message:

4. Now click on the submit button of EmprecordComp in order to view the details in PDF format.

## Employee Details Form

Employeeid

FirstName       LastName

BirthDate

Nationality        Married

HomeAddress

City

EmailID

## Related Content

[WebDynpro Java](#)

[How to Create Online and Offline forms in WebDynpro for Java](#)

For more information, visit the [User Interface Technology homepage](#).



## Copyright

© 2008 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, System i, System i5, System p, System p5, System x, System z, System z9, z/OS, AFP, Intelligent Miner, WebSphere, Netfinity, Tivoli, Informix, i5/OS, POWER, POWER5, POWER5+, OpenPower and PowerPC are trademarks or registered trademarks of IBM Corporation.

Adobe, the Adobe logo, Acrobat, PostScript, and Reader are either trademarks or registered trademarks of Adobe Systems Incorporated 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.

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.