

## Modeling Free-style Web Dynpro applications using Visual Composer

### Applies to:

Net Weaver Visual Composer, NW04s SP8 onwards.

### Summary

This document provides a How-to procedure that walks you through using web services in Visual Composer to create an application and running the same on WD4VC as runtime.

**Author(s):** Padmashree B

**Company:** SAP Labs India Private Ltd.

**Created on:** 31 August 2006

### Author Bio



Padmashree B works as a development lead within the Web Dynpro for Java team. She has been associated with SAP Labs since Feb 2003 and is currently involved in the WD4VC project.

## Table of Contents

Introduction .....	3
Getting Started .....	3
Creating a Model .....	3
Configuring Web Service .....	3
Building Content .....	4
Country Look up.....	4
Country Details.....	5
Enhancing user experience.....	6
Deploy and Run .....	7
Screen 1 .....	9
Screen 2 .....	9
Summary .....	9
Copyright .....	10

## Introduction

SAP NetWeaver Visual Composer is a tool for model-driven content development that enable business experts to easily create applications which can be used in SAP NetWeaver Enterprise Portal and be based on service calls to back-end systems.

Models or the applications designed in Visual Composer can be deployed to run in one or more UI technology engines. Other than Flex, Web Dynpro For Visual Composer (WD4VC), is one such runtime available for realizing the applications modeled in Visual Composer. As the name is self-explanatory WD4VC is a Web Dynpro based runtime leveraging the rich features for creating the professional Web user interfaces for business applications.

## Getting Started

This section provides an overview of the application scenario and the steps involved for creating such an application.

This covers building an application using a public web service. The scenario would be to fetch the details of the country like capital city, continent, currency, etc. for a given ISO code of the country.

**WSDL:** <http://www.oorsprong.org/websamples.countryinfo/CountryInfoService.wso?WSDL>

The steps involved in building such an application can be listed as

1. Creating a model
2. Configuring the web service in the system
3. Building the content for the scenario
4. Deploying and running the application

The remaining sections of the document would explain each of the above steps in detail.

## Creating a Model

1. Log on to the Visual Composer client with the authorized user. The link to the client would be of the format **http://<host>:<port>/VC**.
2. Create a new model with the name "*CountryInfo*".
3. Drag and drop an iView onto the story board and change the title to "*Country Information*".

## Configuring Web Service

In order to use the web service in the application, it is required to define this service within the system. This can be done in 2 ways, either in portal landscape or the Visual Composer Story Board itself provides a tool to define the web service.

In this scenario the system would be configured using the Visual Composer Story Board.

1. Go to *Tools* menu.
2. Select the *Define Web service system...*

This will open up a small wizard. Provide the required details as below:

The above web service has several operations like

ListOfCountryNamesByCode

ListOfLanguagesByCode

FullCountryInfo

and more.

## Building Content

This section explains the various steps involved in building the content for the defined scenario.

It can be broadly divided into 2 sections:

1. Country Look Up – Find the list of available countries
2. Country Details – Find the details of a selected country

First of all, select the system and the services available. To do this:

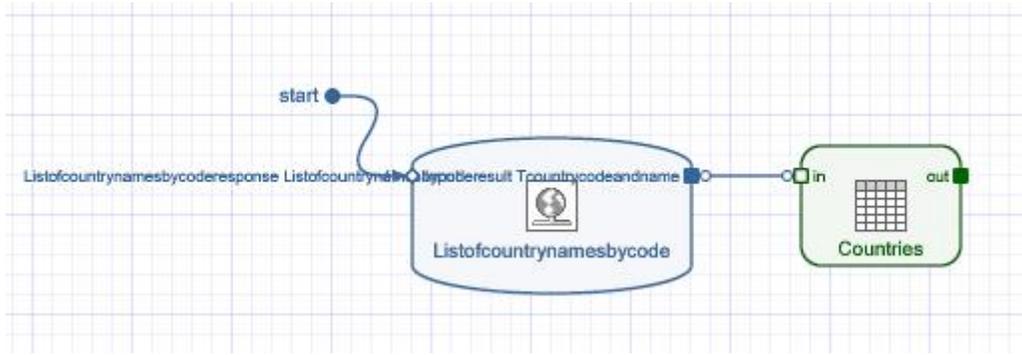
1. Select the *Find Data Services* panel in the Story Board.
2. Select the system “*CountryInfo*”.
3. Search for all the operations supported by this web service.

### Country Look up

As a first step, identify the list of countries. Below are the steps to be followed:

1. Select the operation “*Listofcountrynamesbycode*”.
2. Drag the service onto the story board.
3. Map the *inport* of the data service to the *Start Point* relay so that the service is triggered on the start of the application and the list of countries is displayed.
4. Drag the *outport* of the data service to a *Table view* and rename the title as “*Countries*”.

At the end of this step the model would look like

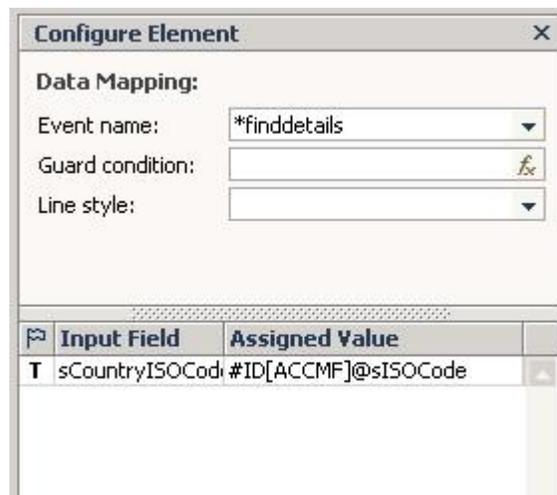


## Country Details

Next step is to enhance this to find the details of a selected country. To do this,

1. Add a *Form view* to the Story Board.
2. Add the control button to this form with the label *"Find Details"*.
3. Assign a custom action *"finddetails"* to the button.
4. Drag the service *"FullCountryInfo"* onto the story board.
5. Map the output of the *"Countries"* view to the inport of the service *"FullCountryInfo"*.
6. Map the input parameter of the service *"sCountryISOCode"* to the column *"sISOCode"* of the *"Countries"* view.
7. Also rename the event on the data mapping link to *"\*finddetails"*

This is as shown below:



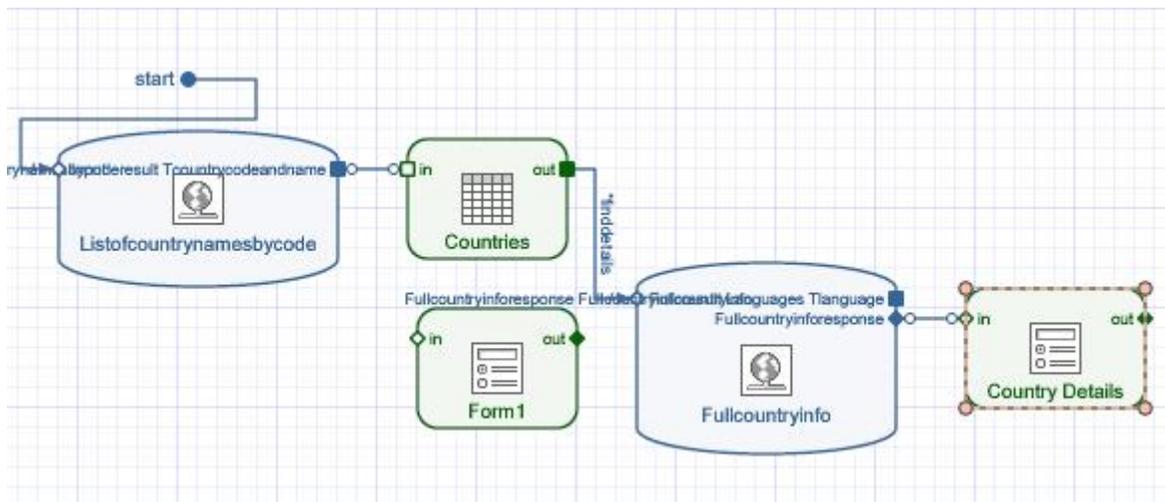
8. Drag the output *"Fullcountryinforesponse"* of the service to the Form view and rename the title as *"Country Details"*.

A history back functionality can be provided to allow the user to return back to the first page to repeat the search process. To achieve this,

1. Add a button to the *"Country Details"* form
2. Assign the system action *"History back"* to it.

3. Also change the label of the button to "Back".

At the end of this step the model would look like...



### Enhancing user experience

Now that the basic scenario is created, the UI of this application can be enhanced with some layering capabilities so that there is some screen transition.

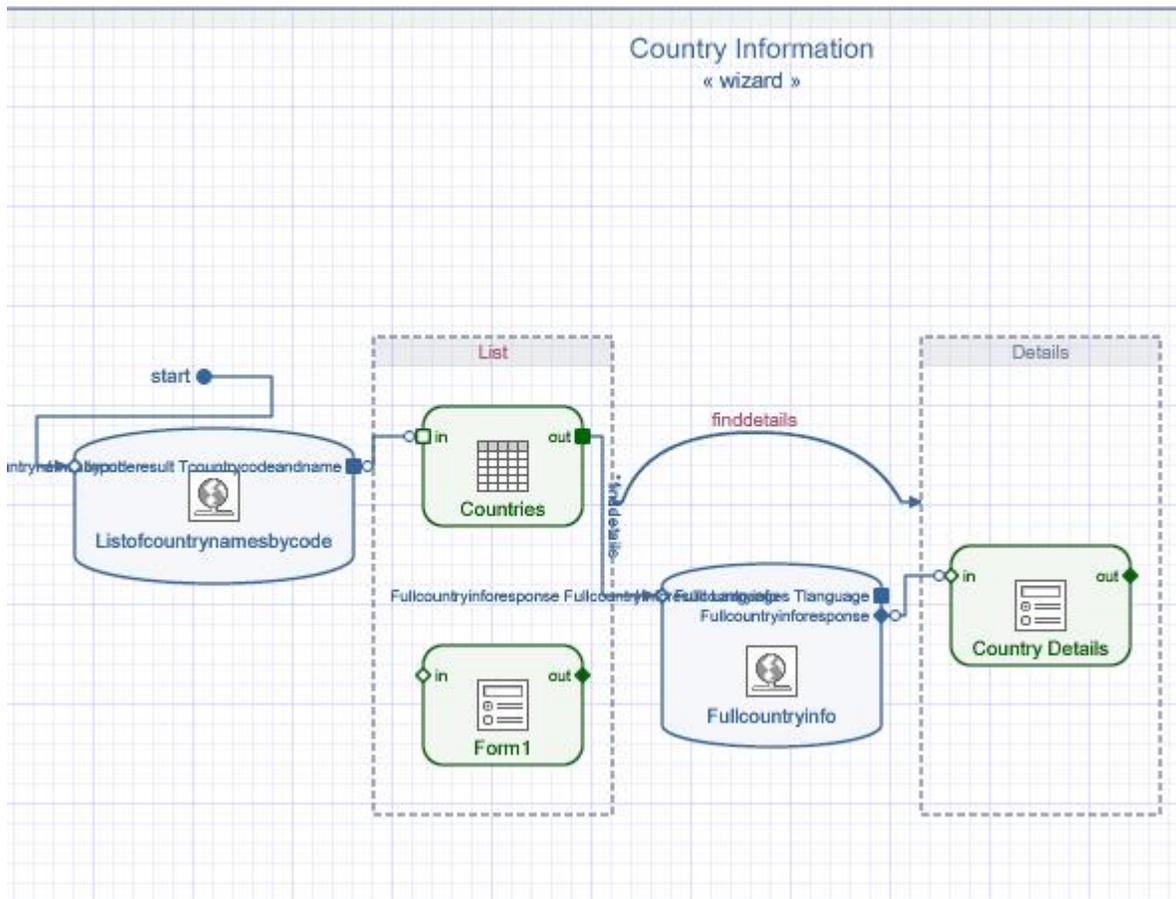
1. Select the *Compose* tab
2. Drag and drop the element *Layer* onto the Story Board. Rename the title as "List". Add the views "Countries" and "Form1" to this layer.
3. Add another layer and title it as "Details". Add the view "Country Details" to this layer.
4. Link the two layers and set the event for this transition as "finddetails".

This means on the trigger of the event "finddetails" there is a transition from layer "List" to layer "Details".

As a last step,

1. Select the iView "Country Information"
2. On the *Configure* tab, change the *Navigation Control* value from *None* to *Wizard*.

At the end of this step the model would look like...

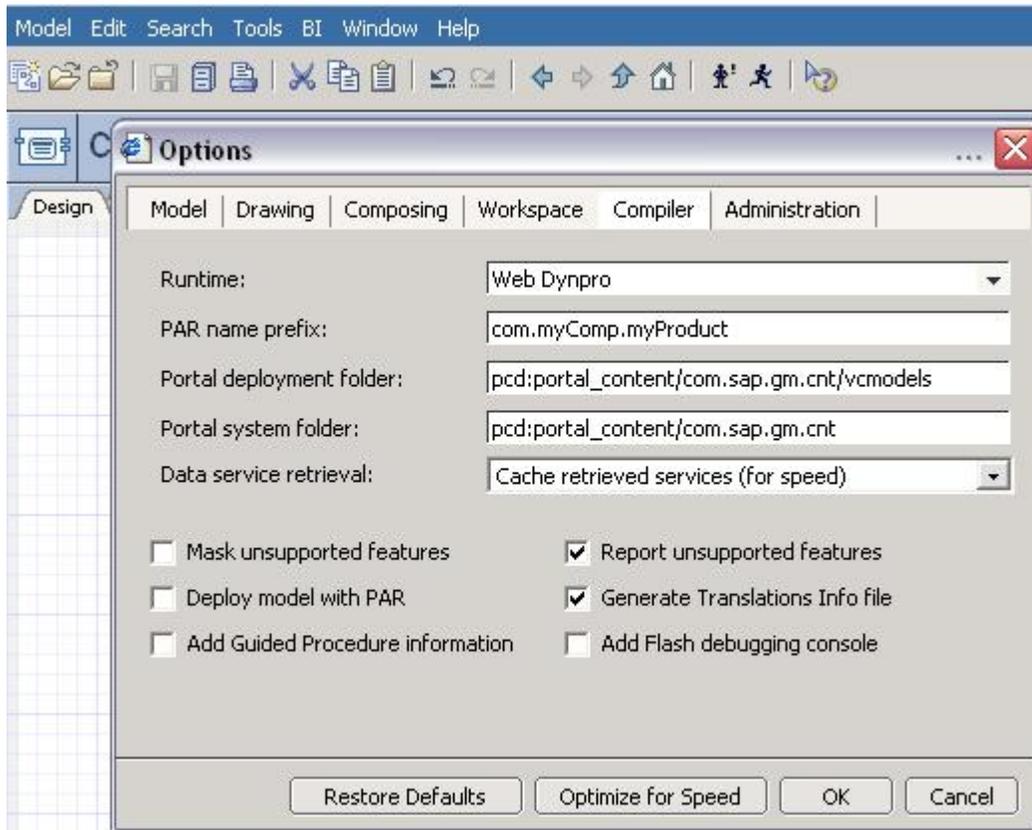


With this the application modeling is complete.

## Deploy and Run

This section explains how the modeled application can be deployed onto the WD4VC runtime and how it can be run in the portal. For this the runtime has to be set as “*Web Dynpro*” in the Visual Composer Story Board. Below are the steps to be followed.

1. Go to *Tools* → *Options* on the menu
2. Select the tab *Compiler*
3. Set the Runtime to *Web Dynpro*
4. Save this by clicking *OK*



Now deploy the application. On the *Deploy* tab, the links for the application are generated after deployment. On clicking the available link the application is hosted in portal. It will be rendered as shown below.

## Screen 1

Screen 1 shows a navigation bar with two steps: '1 List' (highlighted) and '2 Details'. Below the navigation bar is a table titled 'Countries' with two columns: 'Sisocode' and 'Sname'. The table contains 14 rows of country data. The row for 'India' (Sisocode: IN, Sname: India) is highlighted in orange. Below the table is a row of navigation controls including arrows and the text 'Row 97 of 240'. At the bottom is a 'Find Details' button.

Sisocode	Sname
HT	Haiti
HU	Hungary
ID	Indonesia
IE	Ireland
IL	Israel
IN	India
IO	British Indian Ocean Territory
IQ	Iraq
IR	Iran
IS	Iceland
IT	Italy
JM	Jamaica

Select any row. On clicking the button “*Find Details*” the details of the selected country will be shown on the next screen like

## Screen 2

Screen 2 shows a navigation bar with two steps: '1 List' and '2 Details' (highlighted). Below the navigation bar is a form titled 'Country Details' with the following fields:

- ISO Code: IN
- Name: India
- Capital City: New Delhi
- Phone Code: 91
- Continent Code: AS
- Currency ISOCode: INR

At the bottom of the form is a 'Back' button.

## Summary

With this you have created your first Web Dynpro application using Visual Composer. This gets you started with WD4VC.

## Copyright

© 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, Informix, i5/OS, POWER, 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.