Web Dynpro and Visual Composer Integration in NW CE 7.1 EHP1

Applies to:

NWDS 7.1.1 (NWDS 7.1 with EHP1). For more information, visit the User Interface Technology homepage.

Note: Web Dynpro Components/Applications cannot be accessed using web based VC CE 7.1.1

Summary

Visual Composer can be used to create simple screens without any coding. However, it has few limitations with respect to customization as complex UI elements are not available in it. To overcome the restriction we can avail the componentization property of VC which enables us to consume Web Dynpro Components and pass data between Web Dynpro and VC. This document explains the creation of Web Dynpro component with a Tree UI element, consuming it in VC and passing data from Web Dynpro component to the VC model.

Author: Sathya Gunasekaran

Company: Infosys Technologies Ltd

Created on: 21 May 2010

Author Bio

Sathya Gunasekaran is an ABAP consultant working with Infosys Technologies Limited since December 2007.

Table of Contents

Introduction to Visual Composer and Componentization

Visual Composer is a model driven tool that provides rich user interface by enabling code free composition of UI elements, enterprise services and reusable components.

Componentization is the ability of Visual Composer that enables the reuse of a component in other models. It provides reusability, increases the flexibility; and simplifies design, development and testing.

Visual Composer for CE 7.1.1 is available as part of Net Weaver CE 7.1.1 which enables consumption of Web Dynpro components in Visual Composer perspective and data flow between Web Dynpro component and VC model.

Use cases of Componentization

We may need to go for componentization in any of the following situations.

1. We need to incorporate special UI elements / functionalities which are not present in VC (e.g., Tree UI element in Web Dynpro)
2. A complex model is being split into small components among the team members.
3. We need to reuse a particular functionality in several components / models.

Example Scenario

Our Scenario here is to display the following tree structure to the user.

- **Customers**
  - Northern Region
  - South Region
- **Sales Personnel**
  - Northern Region
  - South Region

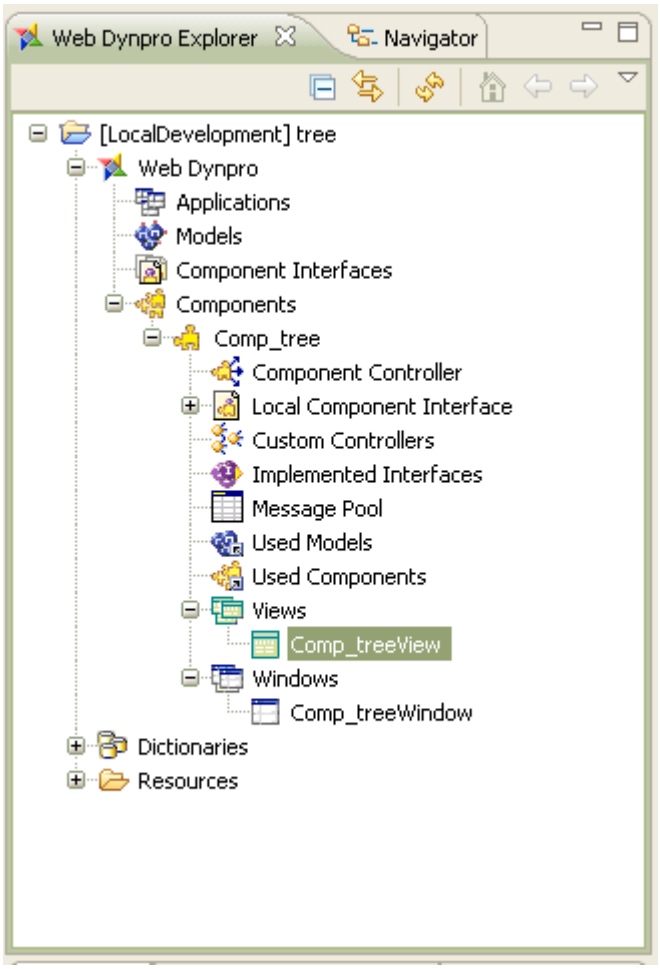
Based on the user selection the detailed list of either the customers or the sales personnel for the selected region needs to be displayed from VC.

The required tree structure cannot be displayed using Visual Composer as complex UI elements are not available with VC and hence we need to develop the hierarchical display in Web Dynpro. By consuming the Web Dynpro development component we can display the tree structure and the user selection can be made available in VC for displaying the detailed list.

Here is the step by step procedure to create a Web Dynpro Component with Tree UI element and consuming it into VC for displaying the details based on the user input.

I. Creating a Web Dynpro Component for Displaying the Tree

1. Create a Web Dynpro development component with default view and window.



2. In the view context add the following context elements.

Context Element	Type
vn_Root	Value Node
vn_Leaf	Value Node
va_Item	String
va_HasChildren	Boolean
va_IsExpanded	Boolean
va_IgnoreAction	Boolean
va_Text	String

3. Set the Singleton property of the vn_Leaf node to false.

4. In the context of the component controller copy the same context elements and bind them with the respective view context elements.

5. In the layout of the view add a Tree UI element to the root Element.

6. Insert two TreeNodeType to the Tree UI (Root_TreeNodeType and Leaf_TreeNodeType).

7. Create an Action named select in the view.

8. Bind the dataSource property of the Tree UI element to the vn_Root node.

9. Bind the properties of Root_TreeNodeType and Leaf_TreeNodeType as below.

Property	Value
Properties [TreeNodeType]	
id	Root_TreeNodeType
dataSource	vn_Root
design	standard*
expanded	vn_Root.va_IsExpanded
hasChildren	vn_Root.va_HasChildren
iconSource	
ignoreAction	vn_Root.va_IgnoreAction
text	vn_Root.va_Text
textDirection	inherit*
tooltip	
Events	
onAction	
onDrop	
onLoadChildren	

Property	Value
Properties [TreeNodeType]	
id	Leaf
dataSource	vn_Root.vn_Leaf
design	standard*
expanded	false
hasChildren	false
iconSource	
ignoreAction	false
text	vn_Root.vn_Leaf.va_Item
textDirection	inherit*
tooltip	
Events	
onAction	select
onDrop	
onLoadChildren	

10. To initialize the text to be displayed in the tree UI, write the following code in wdDoInit method of the view.

```
// @@begin wdDoInit()

// creating elements for the root node.

IPrivateTreeView.IVn_RootElement rootElement1 =
wdContext.nodeVn_Root().createAndAddVn_RootElement();


IPrivateTreeView.IVn_RootElement rootElement2 =
wdContext.nodeVn_Root().createAndAddVn_RootElement();


// setting attributes for the root node elements.

rootElement1.setVa_Text("Customers");
```

```
rootElement2.setVa_Text("Sales Personnel");
          rootElement2.setVa_IsExpandable(true);

// creating elements for the leaf node.

IPrivateTreeView.IVn_LeafElement leafElement1a = rootElement1
          .nodeVn_Leaf().createAndAddVn_LeafElement();

          IPrivateTreeView.IVn_LeafElement leafElement1b = rootElement1
          .nodeVn_Leaf().createAndAddVn_LeafElement();

IPrivateTreeView.IVn_LeafElement leafElement2a = rootElement2
          .nodeVn_Leaf().createAndAddVn_LeafElement();

          IPrivateTreeView.IVn_LeafElement leafElement2b = rootElement2
          .nodeVn_Leaf().createAndAddVn_LeafElement();

// setting attributes for the leaf node elements.

leafElement1a.setVa_Item("Customer details - North");

          leafElement1b.setVa_Item("Customer details - South");

          leafElement2a.setVa_Item("Sales Personnel details - North");

          leafElement2b.setVa_Item("Sales Personnel details - South");
// @@end
```

11. Now our Web Dynpro component is ready with the tree UI element that needs to be displayed.

12. As the user selection needs to be passed from our Web Dynpro component to VC, we need to create an event and a method in the interface controller.

13. Create an event (dataToVC) with a parameter (String) in the component controller which acts as an output port. Create a method (triggerEvent) to trigger the event. Copy both the event and the method to the interface controller as well.

14. In onActionSelect of the view call the method (triggerEvent) by passing the selected text.

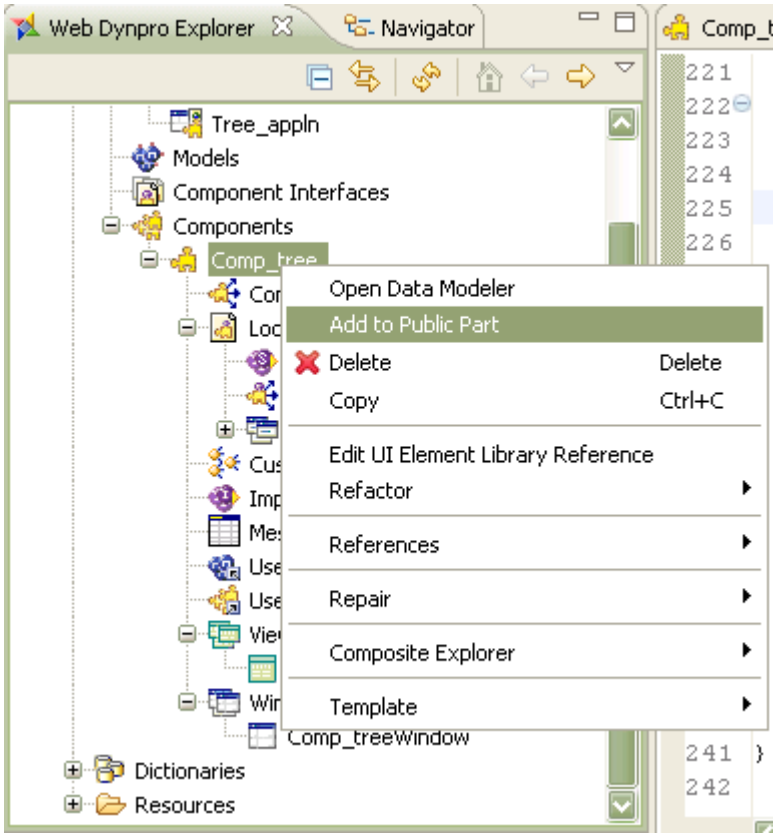
```
  //@@begin onActionSelect(ServerEvent)

    wdThis.wdGetComp_treeController().triggerEvent();

  //@@end
```

15. In the triggerEvent method of the component controller get the selected text and fire the event (dataToVC) by passing the selected text.

```
// @@begin triggerEvent()

String selectedString =
wdContext.nodeVn_Root().currentVn_RootElement().nodeVn_Leaf().currentVn_LeafElement()
.getVa_Item();

    wdThis.wdFireEventDataToVC(selectedString)

// @@end
```

16. Now the Web Dynpro component can pass the selected string to VC.

17. Add the component to public part, create an application and deploy it once to be able to consume it in the VC.

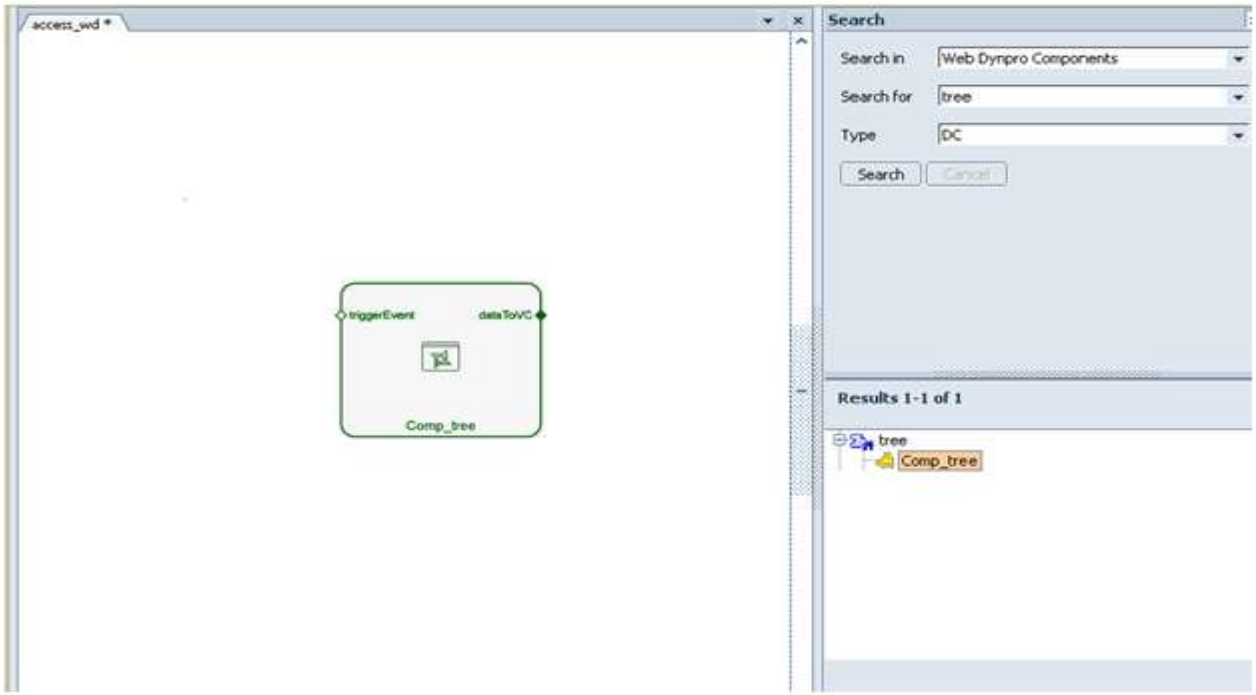


II. Consuming the Web Dynpro component in VC.

1. Switch to Visual Composer Perspective in NWDS 7.1.1 and in Window>Preferences>Visual Composer> Portal Content Modeling > Portal connection give your user name and password.

2. Create a new development component and create a model in it.

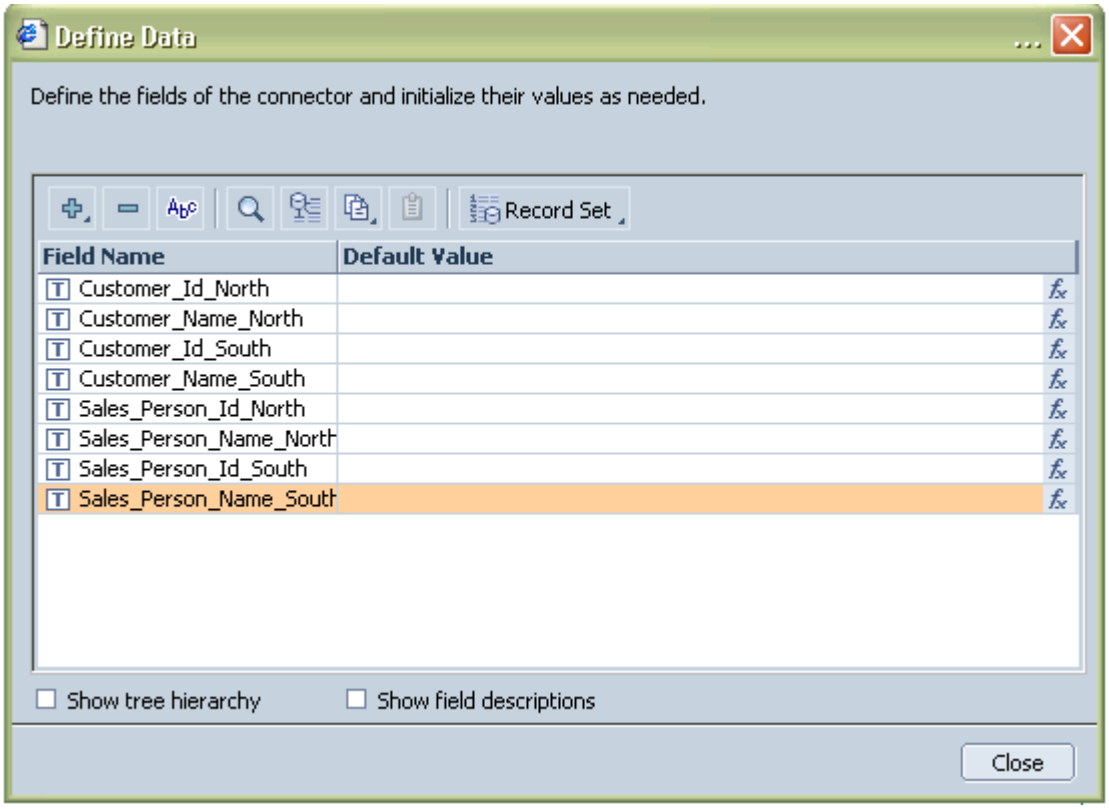
3. Search for the Web Dynpro DC and drag the component (Comp_tree) into the story board.



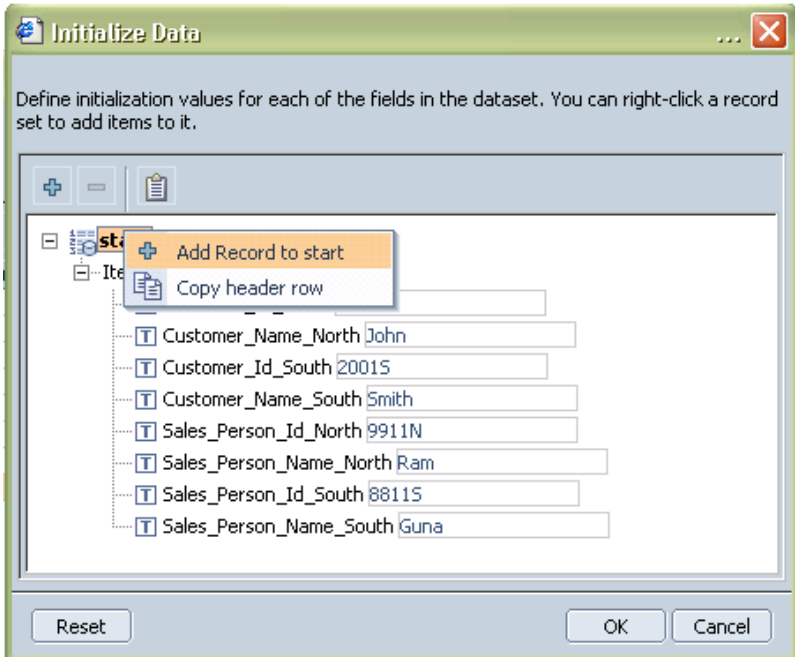
Now the selectedString from the Web Dynpro component will be available for use in VC.

4. Drag and drop start point connector to the story board to store customer and sales personnel data.

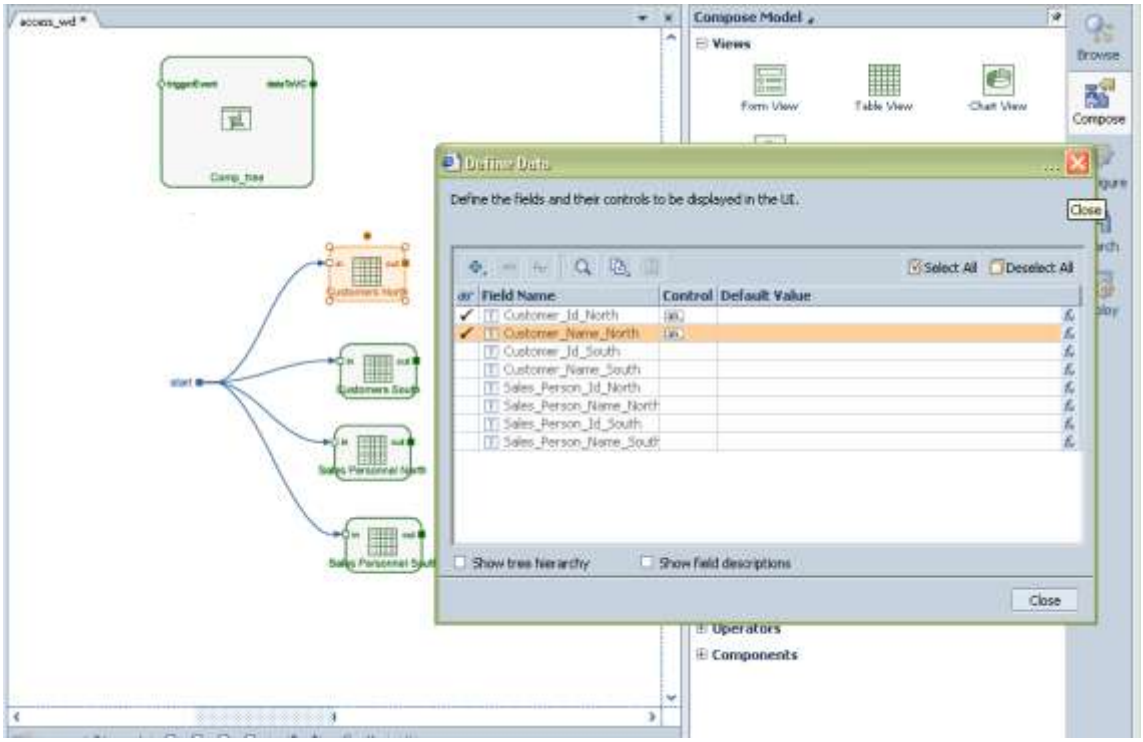
5. Define data for the start point to store customers and sales personnel based on region.



6. Click on Initialize icon and initialize the data



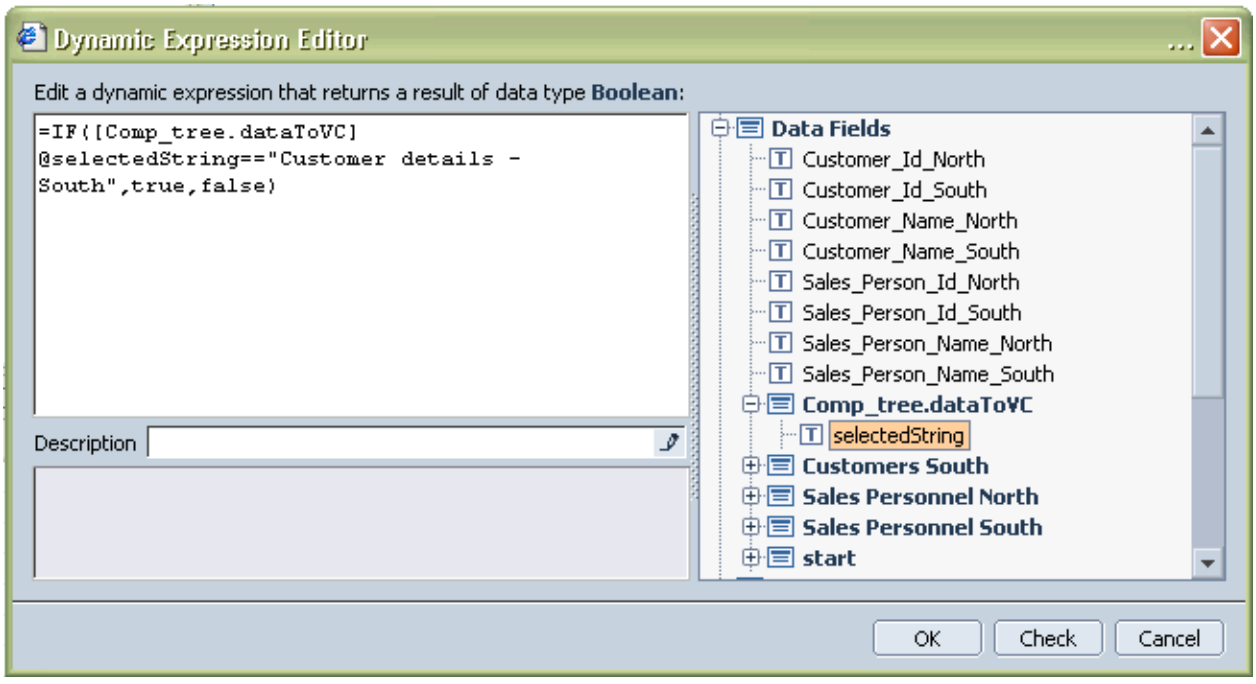
7. Include four table views (Customers North, Customers South, Sales Personnel North and Sales Personnel South) and connect them to the connector start and define data for each of the table view accordingly.



8. Data needs to be displayed based on the user selection.

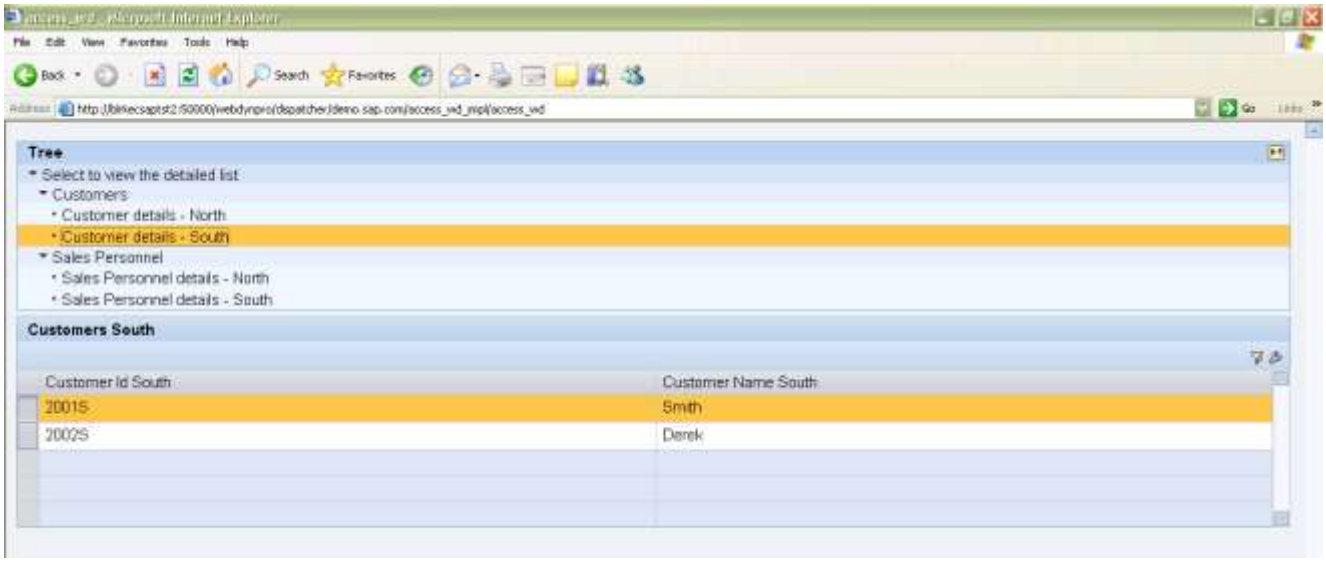
E.g., when the user chooses Customer details – North the corresponding table should alone be visible. Hence provide the following condition in the visible property of the configure table view of the Customers South table

=IF([Comp_tree.dataToVC]@selectedString=="Customer details - South",true,false)



Follow the same procedure for all the four tables.

9. Save the visual composer model, compile it and deploy.



Thus the Web Dynpro component has been consumed in VC to display the tree structure and the selected string has been passed from the Web Dynpro component to the VC model. This way, we will be able to have all the customization that is possible with Web Dynpro in VC as well. This enables us to have the advantages of both Web Dynpro (complex UI elements and flexible customization) and VC (code free development) in the same application.

Related Content

Consuming Web Dynpro components in Visual Composer

Consumption of WebDynpro Component as blackbox in Visual Composer Application

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.