

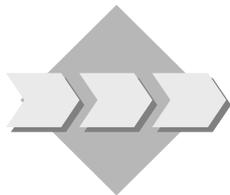


## **Unit: Interactive Forms Integration into Web Dynpro for Java**

### **Topic: Dynamically generated forms**

At the conclusion of this exercise, you will be able to:

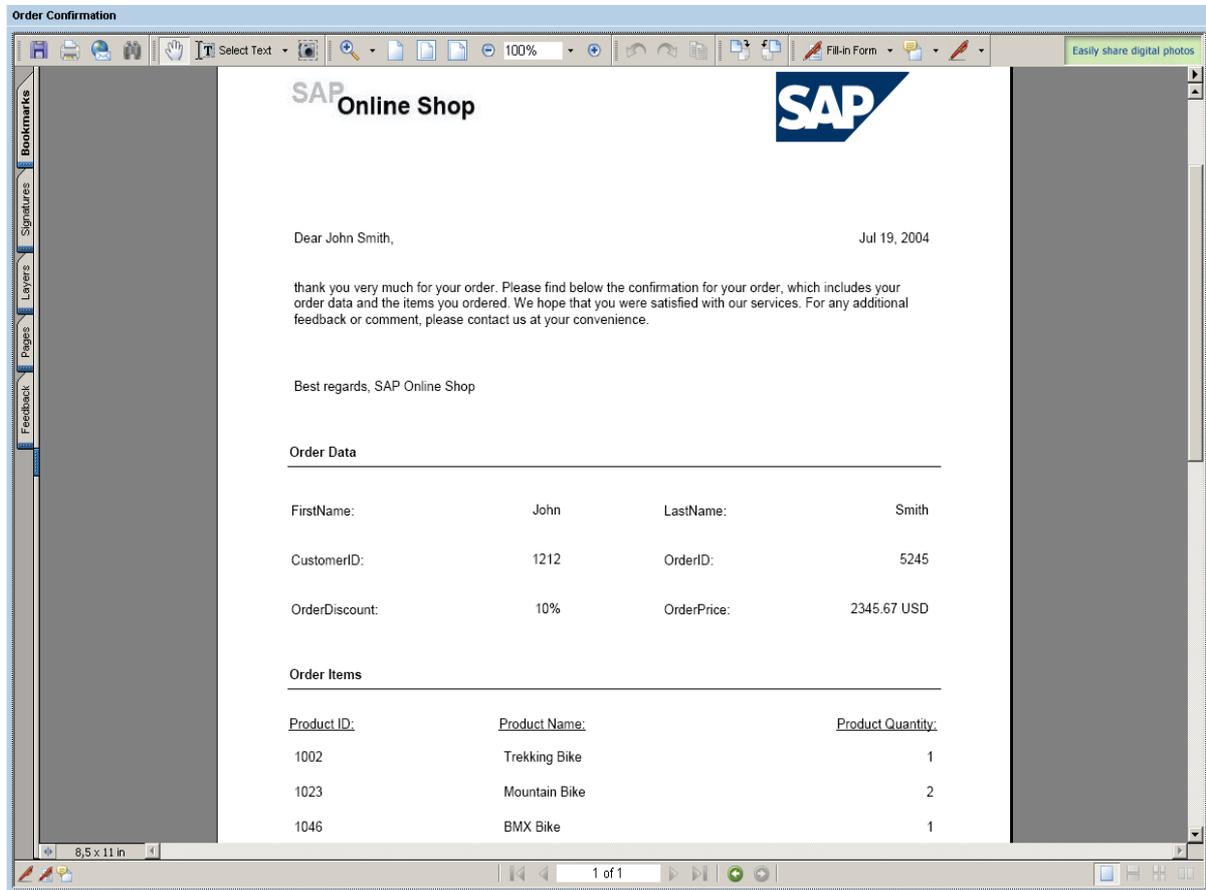
- Generate a dynamic form within a Web Dynpro application
- Use the InteractiveForm UI element
- Layout a PDF form
- Integrate tables into PDF forms
- Integrate images into PDF forms



Think of an Online Shop that presents a confirmation containing all the items ordered to the customer after he placed his order. You can use the Adobe integration within Web Dynpro for this purpose. This exercise shows which steps are necessary to generate a confirmation dynamically. This means that the sample Online Shops application first determines the data before displaying it in the form.

# 1 Overview

The following screenshot shows the Online Shop confirmation you will develop in this exercise. The confirmation will be dynamically generated by a Web Dynpro application.



## 2 Prerequisites

To be able to use PDF forms in Web Dynpro applications, the following prerequisites apply:

- 2-1** The SAP NetWeaver Developer Studio (Support Package Stack 11) including Adobe LiveCycle Designer (forms design tool) is installed on your computer. Use the SAP NetWeaver 04 installation CDs/DVDs if it is not yet installed.
- 2-2** You have access to the SAP J2EE Engine (Release 6.40).  
(Note that you can download an evaluation version (Sneak Preview SAP Web Application Server 6.40 Java) from the SDN Download Area at <https://www.sdn.sap.com/sdn/downloadarea.sdn>.)

- 2-3 Adobe Reader 7.0.1 is installed on your computer. If this is not the case, download Adobe Reader 7.0.1 from the Adobe homepage at <http://www.adobe.com/products/acrobat/readstep2.html>).
- 2-4 The Active Component Framework (ACF) of the Interactive Forms integration is installed on your computer.  
With SP Stack 11, calling a Web Dynpro application that includes a PDF form for the first time should automatically install the ACF in the background. If you experience difficulties that may be related to the frontend installation, please see SAP Note 766191 on the SAP Service Marketplace for manual installation.
- 2-5 The Adobe document services are configured on the SAP J2EE Engine you are using. To access the Installation and Configuration Guides for Adobe document services, go to <http://service.sap.com/nw04installation> → SAP Web AS → [SAP Web AS 6.40 SR1 and Related Documentation](#) → [Adobe Document Services](#).
- 2-6 Basic knowledge of developing Web Dynpro applications. For helpful information about Web Dynpro, go to <https://www.sdn.sap.com/rdn/developerareas/webdynpro.sdn?node=linkDnode6-2>.

### 3 Importing a Project Template

To restrict the development of this sample application to the actual content covered, there is a predefined Web Dynpro project template available in SDN under [Web Dynpro Sample Applications and Tutorials](#). Follow the steps listed below to import the predefined Web Dynpro project:

- 3-1 Unzip the contents of the ZIP file *TutWD\_DynamicForm\_Init.zip* into the work area of the SAP NetWeaver Developer Studio or into a local directory.
- 3-2 Start the SAP NetWeaver Developer Studio.
- 3-3 Import (*File* → *Import* → *Existing Project into Workspace*) the Web Dynpro project *TutWD\_DynamicForm\_Init*.
- 3-4 The Web Dynpro project *TutWD\_DynamicForm\_Init* then appears in the Web Dynpro Explorer for further processing and editing in the context of this tutorial. The information display triggered by the Web Dynpro project *TutWD\_DynamicForm\_Init* can be ignored at this time, since we will extend the Web Dynpro project during the remainder of this exercise and will thus remove this information.

Note: Depending on which Java compiler preferences are set, you may see some warnings in the *Task* view after importing the project. If the severity level for problems of type *Unused imports* (set in *Window* → *Preferences* → *Java* → *Compiler*) has the

value *Warning*, the compiler will issue a warning for unused import references. Ignore these warnings!

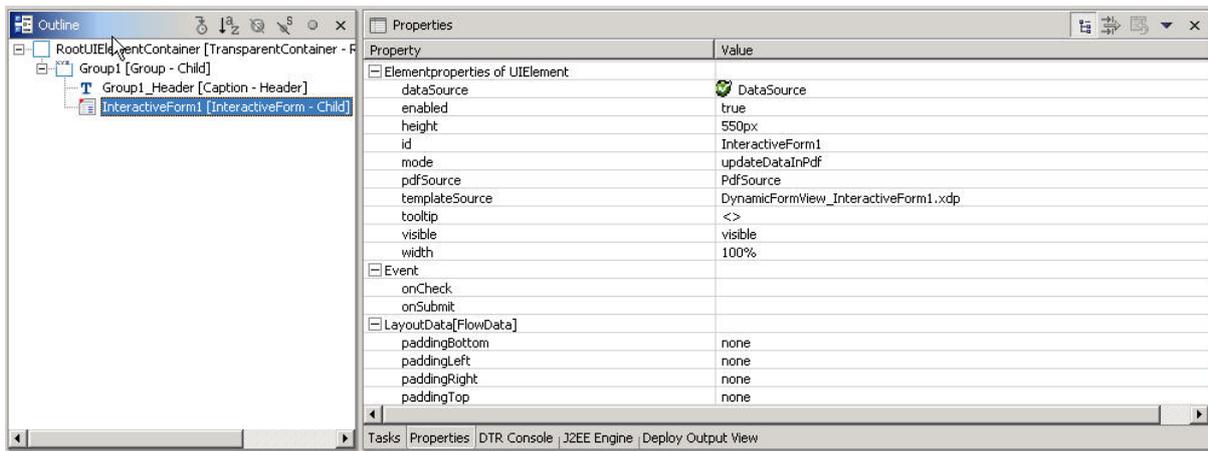
## 4 Initial Project Structure

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

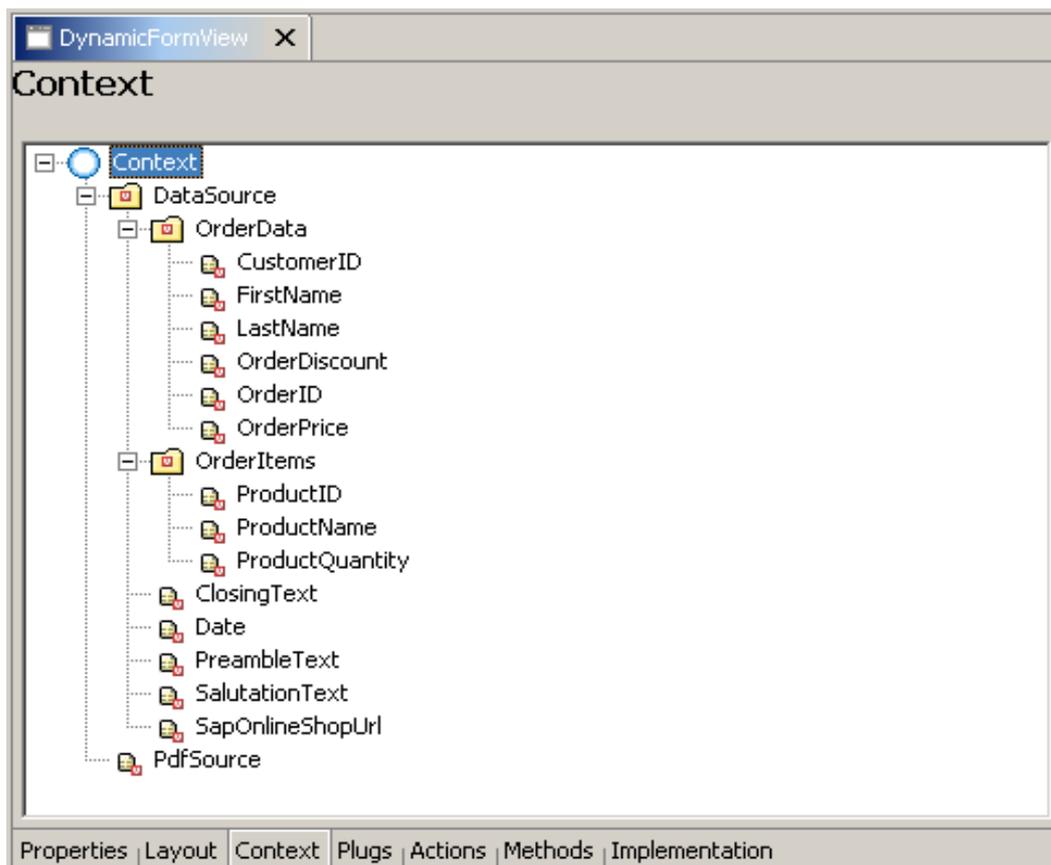
Web Dynpro project structure	
 Web Dynpro project: <b>TutWD_DynamicForm_Init</b>	
 Web Dynpro application: <b>DynamicFormApp</b>	The application <i>DynamicFormApp</i> displays the interface view of the Web Dynpro component <i>DynamicFormComp</i> in the browser window.
 Web Dynpro component: <b>DynamicFormComp</b>	This is the Web Dynpro component that contains our entire application.
 Views: <b>DynamicFormView</b>	In this view, the dynamically generated PDF form will be displayed.
 Window: <b>Window</b>	Initially only contains the <i>DynamicFormView</i> .

## 5 Add InteractiveForm UI Element

- 5-1** In the project structure, double-click the node for the *DynamicFormView* (*Web Dynpro* → *Web Dynpro Components* → *DynamicFormComp* → *Views* → *DynamicFormView*) and choose the *Layout* tab.
- 5-2** In the *Outline* pane, insert a child of Type *InteractiveForm* and Id *InteractiveForm1* in the *Group1*.
- 5-3** Select the *InteractiveForm1* 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*:



The *dataSource* property is used to specify the data source. The data source encapsulates the data you can display in the form at runtime. For the *dataSource* property, you need to specify the path to the context node providing the data. In this step we reference the *DataSource* context node, which is already defined in the context structure of the used template (see *Context* tab of the *DynamicFormView*):



For more information related to context structures refer to the following tutorial, which you can find in the standard SAP Library documentation:

## [Application of Context Programming and Data Binding](#)

The structure of the context node, which is referenced by the *dataSource* property, will also be displayed in the *Data View* tab of Adobe LiveCycle Designer. The corresponding context attributes of this node can be bound to the Designer layout elements defined in the form (see Step 6).

The *updateDataInPdf* value of the *mode* property is used to update a PDF document with the data provided by the data source or to create a new PDF document from the data source and the form template if no PDF document exists.

The *pdfSource* property specifies the path of the context element that contains the PDF document. You must bind this property to a context attribute of the type *binary*. In this tutorial, the context attribute has already been defined in the context structure (see graphic above).

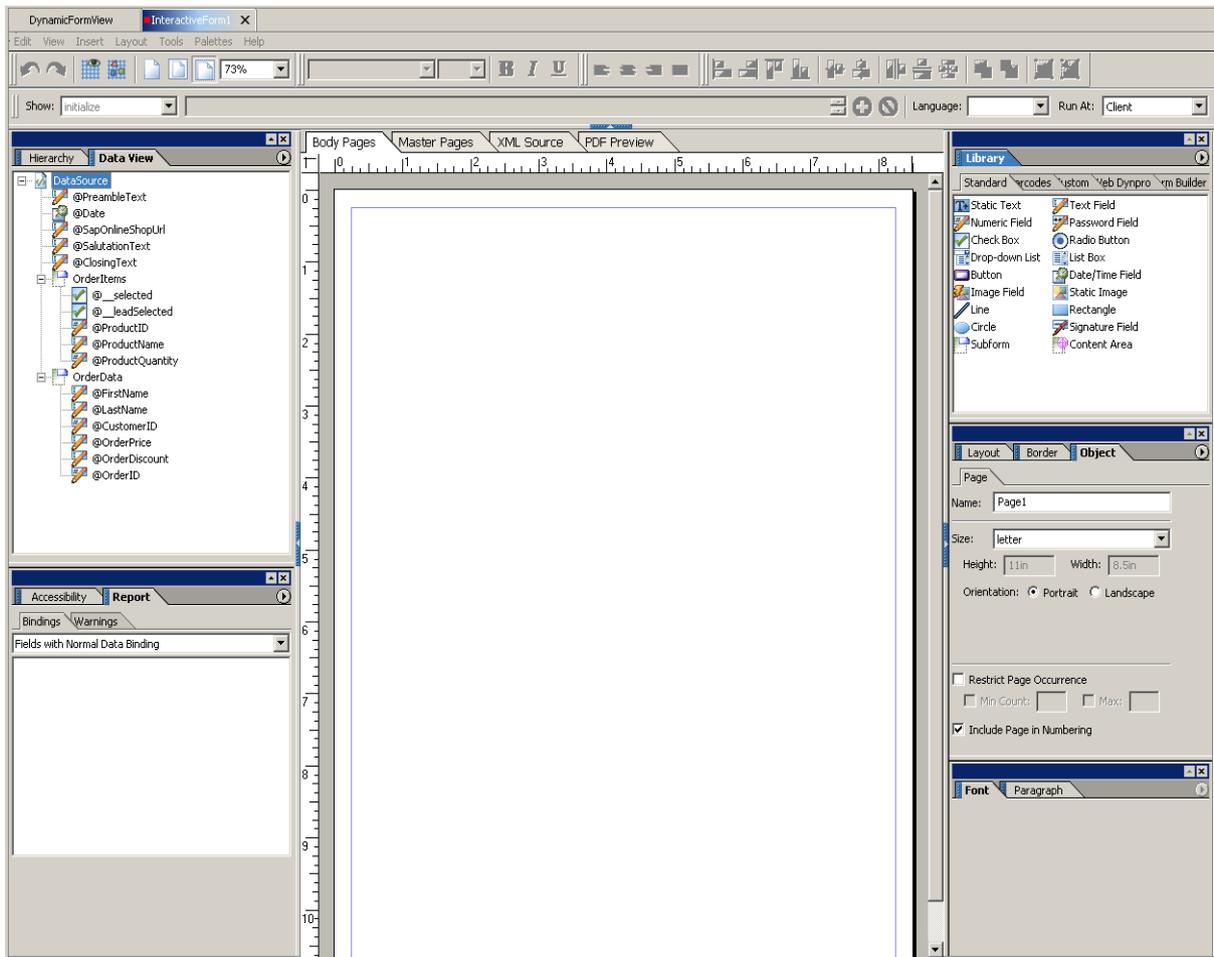
Note: Although not used in this tutorial, this property allows an application developer to access the binary file and download it to the local hard disk or read and send the data to a backend.

The *templateSource* property specifies the unique name of the form template. The name is automatically generated when you insert the *InteractiveForm* UI element into the view.

**5-4** Save the new metadata by choosing the  (*Save All Metadata*) icon from the toolbar.

## **6 Define Form Layout with Adobe LiveCycle Designer**

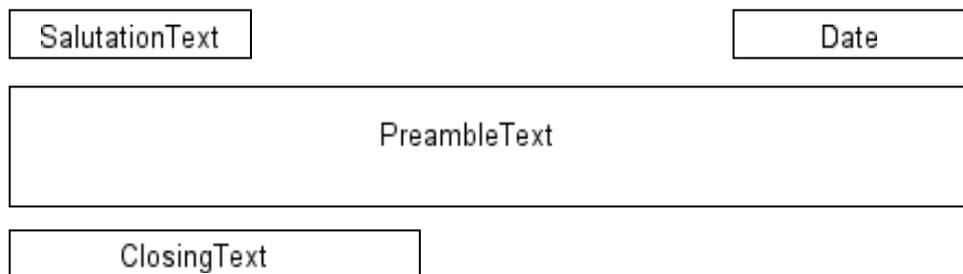
You use the Adobe LiveCycle Designer to set up the layout of your PDF form. Start this tool by choosing *Edit* in the context menu of the *InteractiveForm* UI element in the *Outline* pane. You can enlarge the Designer window by double-clicking on the *InteractiveForm1* tab.



Note: Before you begin to layout your form, click on the *Show Grid* and *Snap to Grid* buttons in the toolbar of Adobe LiveCycle Designer to facilitate your work.

Also ensure that *Object Boundaries* in the *View* menu is checked in order to show the boundaries of form objects during design time. (Depending on your formatting, these boundaries may not be visible at run time.)

The following graphic shows the layout you will develop in this section:



- 6-1** Drag and drop the *@SalutationText* from the *Data View* tab to the *Body Pages* pane of the Adobe LiveCycle Designer. A text field representation of the *SalutationText* appears on the *Body Pages* pane. Select this text field and edit the following properties:
- Click on the *Layout* tab and choose *None* for the *Caption* position.
  - Click on the *Object* and then on the *Field* tab and choose *None* for *Appearance*.
  - Click on the *Value* tab and choose *Read Only* for the *Type* field.
- 6-2** Repeat these steps for the *PreambleText* and *ClosingText*. Enlarge the text field for the *PreambleText* and choose *Allow Multiple Lines* on the *Field* tab. To ensure that the data of the *SalutationText* and *ClosingText* will be displayed completely, specify the width of these elements generously.
- 6-3** To be able to display the current date on the form, drag a Text Field from the Standard Library tab onto the *Body Pages* pane.  
To bind the data against the corresponding backend field, drag the *@Date* node from the *Data View* tab onto the text field.  
On the *Paragraph* tab, select *Align right* for the text field.
- 6-4** Save the new metadata by choosing the  (*Save All Metadata*) icon from the toolbar.
- 6-5** Switch to the *DynamicFormView* and select the *Implementation* page. The following coding lines initialize the context elements, which are bound to the corresponding text fields and visible in the form. Add the source code to the *wdDoInit* method:

```

wdContext.currentDataSourceElement().setDate(
new Date(System.currentTimeMillis()));

wdContext.currentOrderDataElement().setFirstName("John");
wdContext.currentOrderDataElement().setLastName("Smith");

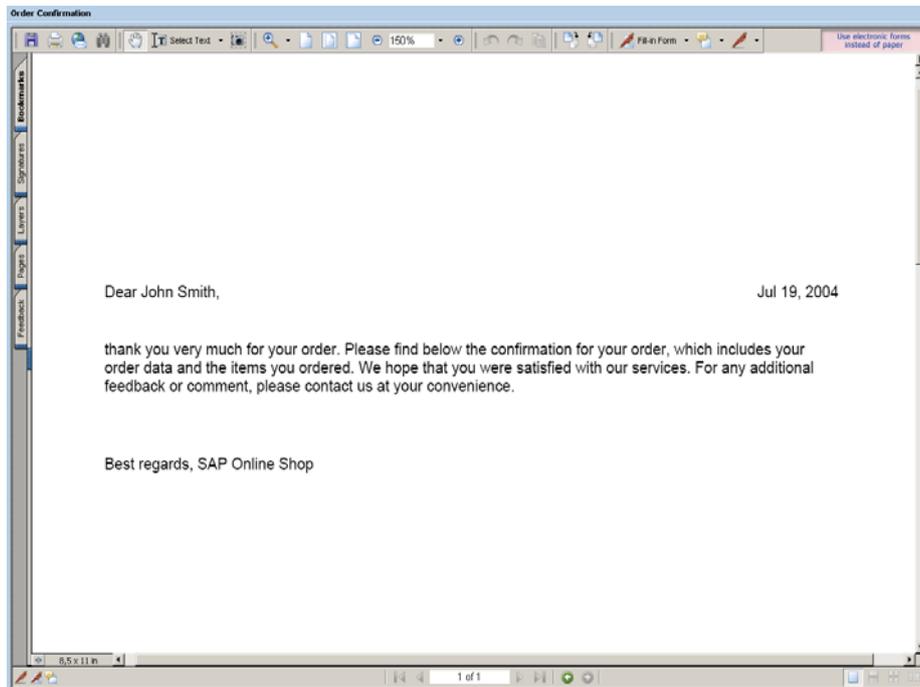
String salutationText = "Dear " +
    wdContext.currentOrderDataElement().getFirstName() + " " +
    wdContext.currentOrderDataElement().getLastName() + ",";
String preambleText = "thank you very much for your order. Please find "
+
    "below the confirmation for your order, which includes your order
" + "data and the items you ordered. We hope that you were
satisfied " +
    "with our services. For any additional feedback or comment, " +
    "please contact us at your convenience.";
String closingText = "Best regards, SAP Online Shop";

wdContext.currentDataSourceElement().setSalutationText(salutationText);
wdContext.currentDataSourceElement().setPreambleText(preambleText);
wdContext.currentDataSourceElement().setClosingText(closingText);

```

**6-6** On the *Implementation* page of the *DynamicFormView*, choose *Source* → *Organize Imports* from the Context Menu, select *java.sql.date* and press *Finish*.

**6-7** In the Web Dynpro Explorer View, trigger *Deploy New Archive and Run* for the *DynamicFormApp*. The result should look like the screenshot below:



## 7 Enhance Form Layout for Order Data

In this section we will layout the order data of the confirmation. The following graphic shows how we want the layout of the order data to appear:

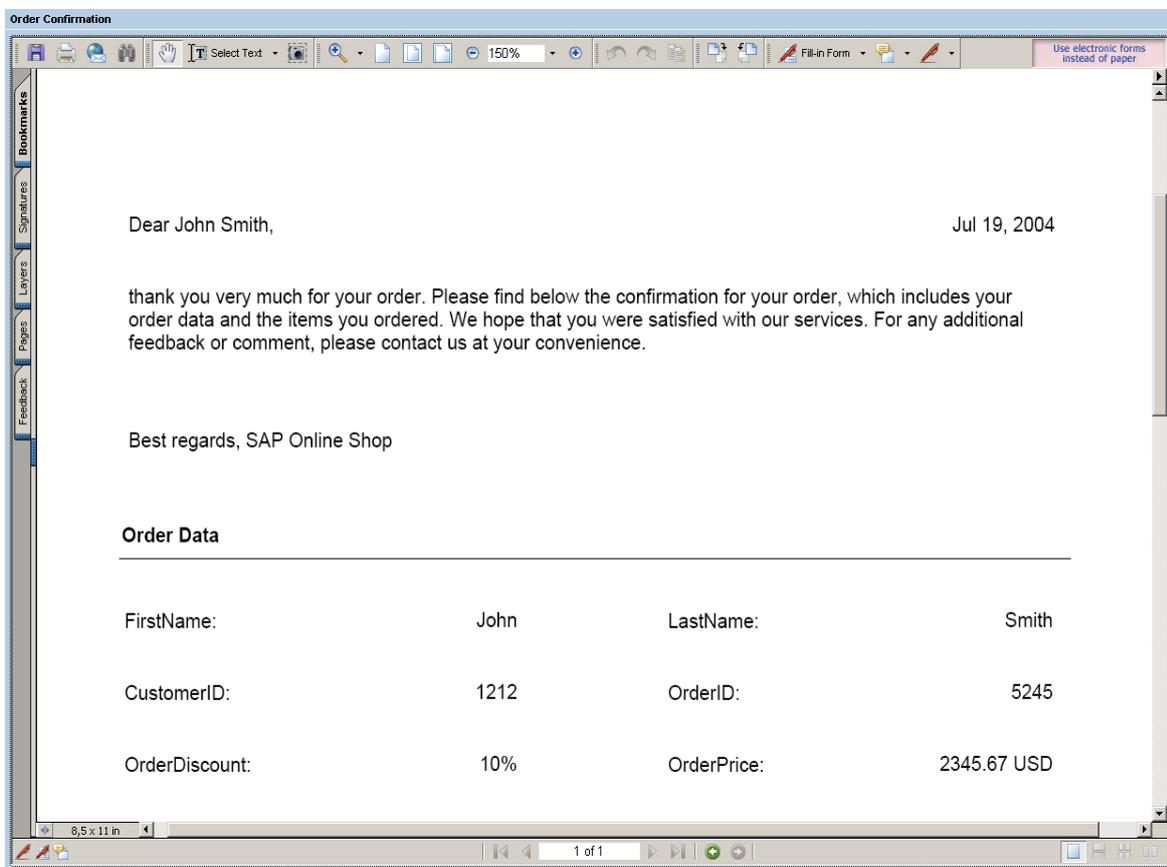
Order Data	
FirstName:	LastName:
CustomerID:	OrderID:
OrderDiscount:	OrderPrice:

- 7-1 In order to enhance the form layout with the order data, switch back to the Adobe LiveCycle Designer view of the *InteractiveForm1* element. Drag and drop the *Static Text* element from the standard *Library* tab to the *Body Pages* pane, select this element and change the text to *Order Data*. Mark the whole text and set the *Style* on the *Font* tab to *Bold*.
- 7-2 Drag and drop the *Line* element from the standard *Library* tab to the *Body Pages* pane and enlarge the line so that it covers the width of the page.
- 7-3 Drag and drop the *@FirstName* node from the *Data View* tab to the *Body Pages* pane of Adobe LiveCycle Designer. A text field representation of *FirstName* appears on the *Body Pages* pane. Select this text field, add a colon at the end of the caption text and edit the following properties:
  - Click on the *Object* tab, then the *Field* tab and choose *None* for *Appearance*.
  - Click on the *Value* tab and choose *Read Only* for the *Type* field.
  - Click on the *Paragraph* tab, choose *Edit value* under *Currently editing Value properties...* and select *Align right*.
- 7-4 Repeat these steps for the *LastName*, *CustomerID*, *OrderID*, *OrderDiscount*, and *OrderPrice* nodes.
- 7-5 Save the new metadata by choosing the  (*Save All Metadata*) icon from the toolbar.
- 7-6 Switch to the *DynamicFormView* and select the *Implementation* page. Add the following source code at the end of the *wdDoInit* method:

```
wdContext.currentOrderDataElement().setCustomerID(1212);  
wdContext.currentOrderDataElement().setOrderID(5245);  
wdContext.currentOrderDataElement().setOrderDiscount("10%");  
wdContext.currentOrderDataElement().setOrderPrice("2345.67 USD");
```

These coding lines initialize the order data context elements, which are bound to the corresponding text fields and visible in the form.

**7-7** In the Web Dynpro Explorer View, trigger *Deploy New Archive and Run* for the *DynamicFormApp*. The result should look like the following screenshot:



## 8 Integrate Tables into PDF Forms

In this section, we will layout the order items of the confirmation using a table. The following graphic shows the intended layout of the order data:

Order Items		
Product ID:	Product Name:	Product Quantity:

- 8-1** In order to enhance the form layout with the order items, switch back to the Adobe LiveCycle Designer view of the *InteractiveForm1* element. Drag and drop the *Static Text* element from the standard *Library* tab to the *Body Pages* pane, select this element and change the text to *Order Items*. Mark the whole text and set the *Style* on the *Font* tab to *Bold*.
- 8-2** Drag and drop the *Line* element from the standard *Library* tab to the *Body Pages* pane and enlarge the line so that it covers the width of the page. Alternatively, you can copy and paste within the Adobe LiveCycle Designer the line used before for the order data, and use the new instance for the order items.
- 8-3** Drag and drop the *Static Text* element from the standard *Library* tab to the *Body Pages* pane, select this element and change the text to *Product ID*. Mark the whole text and set the *Style* on the *Font* tab to *Underline*.
- 8-4** Repeat the last step for static text fields named *Product Name* and *Product Quantity*.
- 8-5** Drag and drop a *Subform* element from the standard *Library* tab to the *Body Pages* pane. Enlarge this subform over the whole width. Drag and drop a second *Subform* element from the standard *Library* tab to the *Body Pages* pane within the first one. Make the two subforms the same size.

Drag and drop the element *@ProductID* from the *Data View* tab to the *Body Pages* pane of the Adobe LiveCycle Designer within the second subform. A numeric field representation of the *ProductID* appears on the *Body Pages* pane within the second subform. Select this numeric field and edit following properties:

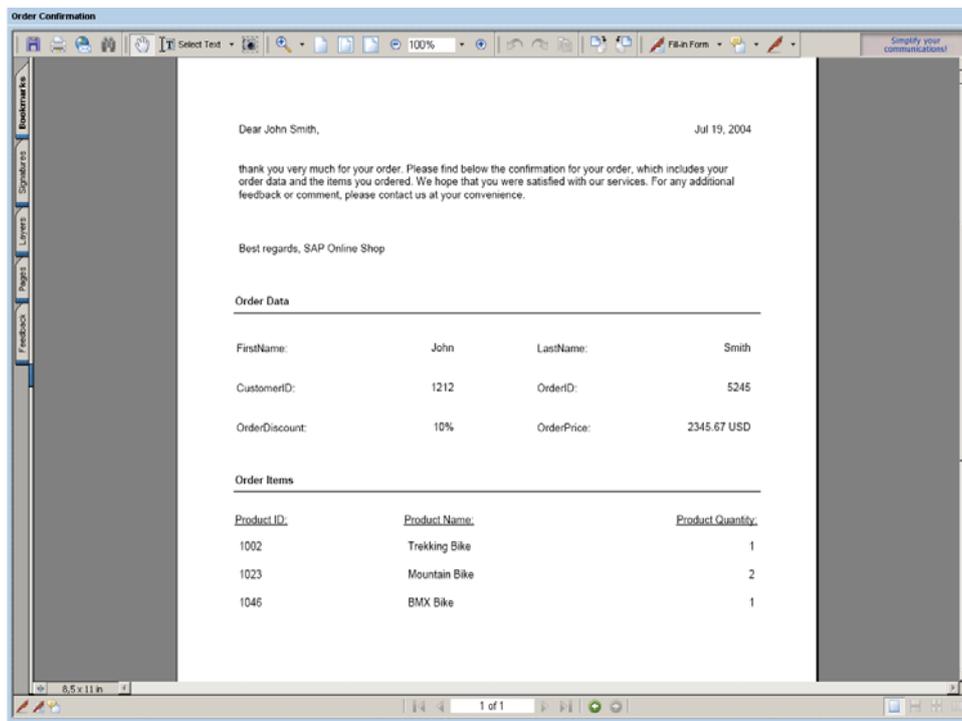
- Click on the *Layout* tab and choose *None* for the *Caption* position.
  - Click on the *Object* tab, then on the *Field* tab and choose *None* for *Appearance*.
  - Click on the *Value* tab and choose *Read Only* for the *Type* field.
- 8-6** Proceed in the same way for *ProductName* and *ProductQuantity*.
- 8-7** Select the numeric field for *ProductQuantity*, click on the *Paragraph* tab, and select *Align right*.

- 8-8** Click on the *Hierarchy* tab, select *Subform1*, and enter on the *Subform* tab of the *Object* tab *Flow Content* for the *Type* and *Table* for the *Flow Direction*.
- 8-9** Select the inner *Subform2* on the *Hierarchy* tab, and enter in the *Subform* tab of the *Object* tab *Flow Content* for the *Type* and *Table Row* for the *Flow Direction*. On the *Binding* tab of the *Object* tab, activate *Repeat Subform for Each Data Item*.
- 8-10** Enlarge the numeric field for *ProductID*, so that it will use the whole space under the static text labeled *ProductID*. Do the same for *ProductName* and *ProductQuantity*.
- 8-11** Save the new metadata by choosing the  (*Save All Metadata*) icon from the toolbar.
- 8-12** Switch to the *DynamicFormView* and select the *Implementation* page. Add the following source code at the end of the *wdDoInit* method:

```
IPrivateDynamicFormView.IOrderItemsElement orderItemsElement =
wdContext.createOrderItemsElement();
orderItemsElement.setProductID(1002);
orderItemsElement.setProductName("Trekking Bike");
orderItemsElement.setProductQuantity(1);
wdContext.nodeOrderItems().addElement(orderItemsElement);
orderItemsElement = wdContext.createOrderItemsElement();
orderItemsElement.setProductID(1023);
orderItemsElement.setProductName("Mountain Bike");
orderItemsElement.setProductQuantity(2);
wdContext.nodeOrderItems().addElement(orderItemsElement);
orderItemsElement = wdContext.createOrderItemsElement();
orderItemsElement.setProductID(1046);
orderItemsElement.setProductName("BMX Bike");
orderItemsElement.setProductQuantity(1);
wdContext.nodeOrderItems().addElement(orderItemsElement);
```

These coding lines initialize the order items context elements, which are bound to the corresponding fields and visible in the form.

- 8-13** On the *Implementation* page of the *DynamicFormView*, choose *Source* → *Organize Imports*.
- 8-14** In the Web Dynpro Explorer View, trigger *Deploy New Archive and Run* for the *DynamicFormApp*. The result should look like the following screenshot:



## 9 Integrate Images into PDF Forms

In this section, we will integrate images into the confirmation. The following graphic shows how the images will be integrated:



The left graphic represents an application-specific image that will be integrated dynamically, and the right one represents the company logo, which is a static image.

- 9-1** In order to integrate images in the form layout, switch back to the Adobe LiveCycle Designer view of the *InteractiveForm1* element.  
For the right image, drag and drop the *Static Image* element from the standard *Library* tab to the *Body Pages* pane.
- 9-2** Select the *Static Image* element on the *Body Pages* pane, click on the *Object* tab and select the *sap\_corporate.gif* for the *URL* entry under the component folder in the corresponding mimes folder of the project *TutWD\_DynamicForm\_Init* (→ `<Eclipse_Workspace>\TutWD_DynamicForm_Init\src\mimes\Components\com.sap.tut`

.wd.dynamicform.DynamicFormComp). In addition, activate *Embed Image Data* and select *Use Image Size* for the *Sizing* field.

**9-3** For the left image, drag and drop an *Image Field* element from the standard *Library* tab to the *Body Pages* pane. Select this image field and edit the following properties:

- Click on the *Layout* tab and choose *None* for the *Caption* position.
- Click on the *Object*, then the *Binding* tab and choose *None* for *Default Binding*.
- Click on the *Field* tab, enter *\$record.SapOnlineShopUrl* for the *URL* entry, and select *Use Image Size* for the *Sizing* field.
- Click on the script editor and enter the following *FormCalc* script statement, which enables the dynamic integration of the image.  
Show: *initialize*  
Script: *this.value.image.href = xfa.resolveNode(this.value.image.href).value;*  
Language: *FormCalc*  
Run At: *Client*

**9-4** Save the new metadata by choosing the  (*Save All Metadata*) icon from the toolbar.

**9-5** Switch to the *DynamicFormView* and select the *Implementation* page. Add the following source code at the end of the *wdDoInit* method:

```
try
{
    String url = WDURLGenerator.getAbsoluteWebResourceURL(
        wdComponentAPI.getDeployableObjectPart(), "sap_online_shop.jpg");
    wdContext.currentDataSourceElement().setSapOnlineShopUrl(url);
}
catch(Exception e)
{
    throw new WDRuntimeException(e);
}
```

These coding lines initialize the *SapOnlineShopUrl* context element, which is bound to the corresponding image field and visible in the form.

**9-6** On the *Implementation* page of the *DynamicFormView*, choose *Source* → *Organize Imports*.

**9-7** In the Web Dynpro Explorer View, trigger *Deploy New Archive and Run* for the *DynamicFormApp*. The result should look like the screenshot in the Overview section at the beginning of this document.