

How to...

Create a Simple Web Dynpro Application

ASAP “How to...” Paper



Release: SAP Web Application Server 6.30
September 2003

CONTENTS

1	SCENARIO	1
2	RESULT.....	1
3	THE STEP BY STEP SOLUTION.....	1
3.1	Creating a Simple Web Dynpro Application	1
3.2	Creating a Web Dynpro Project.....	4
3.3	Creating a Web Dynpro Component.....	6
3.4	Creating and Designing a View	7
3.5	Defining a Web Dynpro Window	12
3.6	Defining a Web Dynpro Application	14
3.7	Displaying the Current Date Using Data Binding.....	19
3.8	Structure of a Web Dynpro Application	24

1 Scenario

Web Dynpro allows you to easily create high-quality Web-based user interfaces and so enhances your Web development. The Web Dynpro technology provides a runtime environment and a graphical development environment with special Web Dynpro tools. This document describes how to create a simple Web Dynpro application with the Web Dynpro tools that are integrated in the SAP NetWeaver Developer Studio.

2 Result

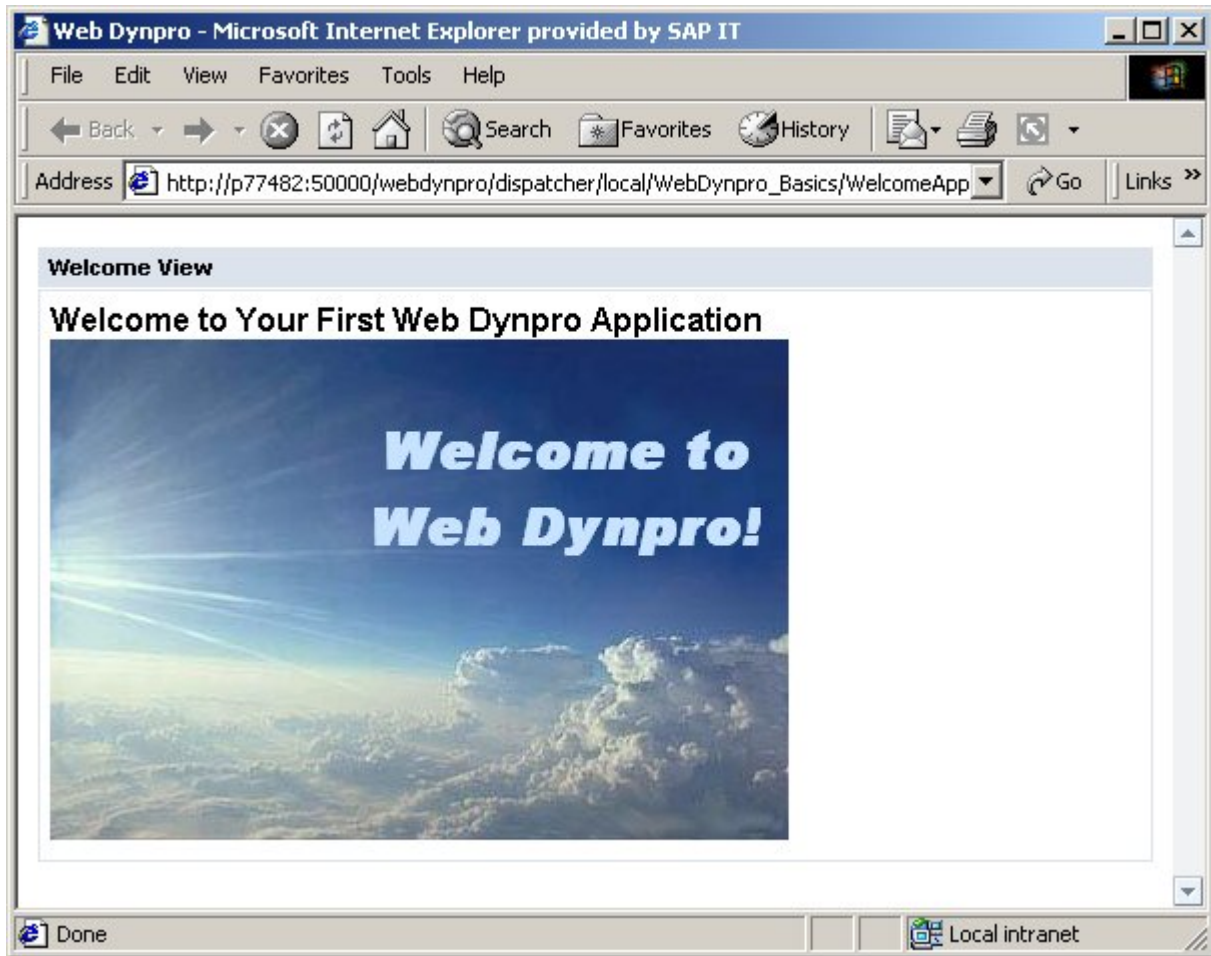
You have created a simple Web Dynpro application using the Web Dynpro tools in the SAP NetWeaver Developer Studio. You used data binding between a UI element property in the view layout and a value attribute in the view context to specify and display a text.

3 The Step By Step Solution

3.1 Creating a Simple Web Dynpro Application

3.1.1 Purpose

To start with, you create your Web Dynpro application using a simple example that illustrates the basic concepts of developing Web applications based on Web Dynpro. The example application should look like this:



The Web Dynpro application consists of only one page, which contains the static part Welcome to Your first Web Dynpro Application and a graphic. Text and graphic are displayed in the Web browser. The text is inserted using a UI element of the type *TextView*. The graphic is a MIME object, which is inserted using a UI element of the type *Image*. Both UI elements are inserted into a UI element of the type *Group*.

3.1.2 Process Flow

The exercises for the introductory example application consist of the following steps:

- [Creating a Web Dynpro Project \[Seite 4\]](#)
- [Creating a Web Dynpro Component \[Seite 6\]](#)
- [Creating and Designing a View \[Seite 7\]](#)
- [Defining a Web Dynpro Window \[Seite 12\]](#)
- [Defining a Web Dynpro Application \[Seite 14\]](#)
- [Deploying a Web Dynpro Project \[Seite 16\]](#)
- [Calling a Web Dynpro Application \[Seite 17\]](#)
- [Displaying the Current Date Using Data Binding \[Seite 19\]](#)





Create the first [Web Dynpro project \[Seite 4\]](#).

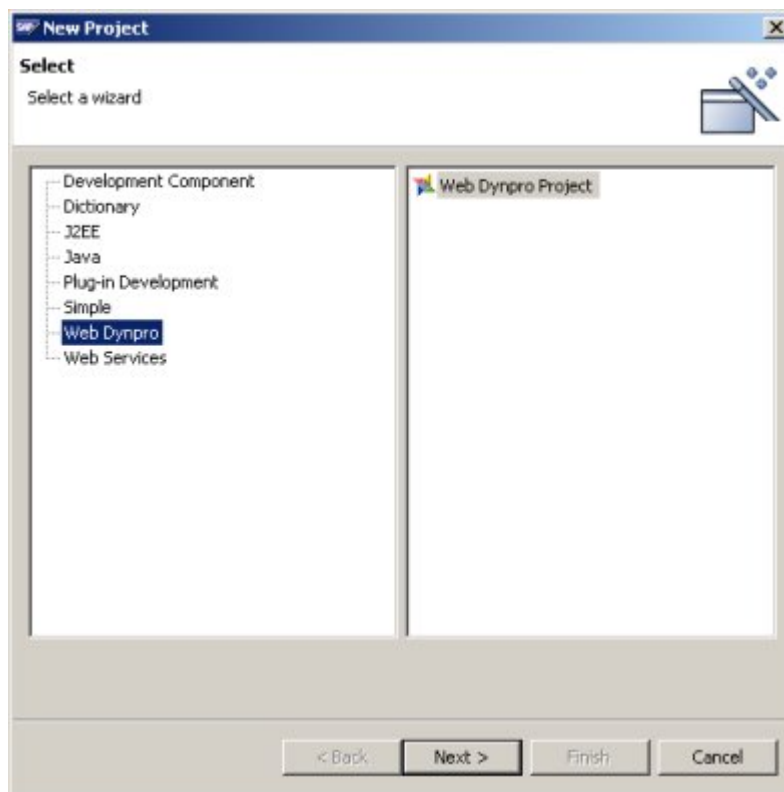
3.2 Creating a Web Dynpro Project

3.2.1 Use

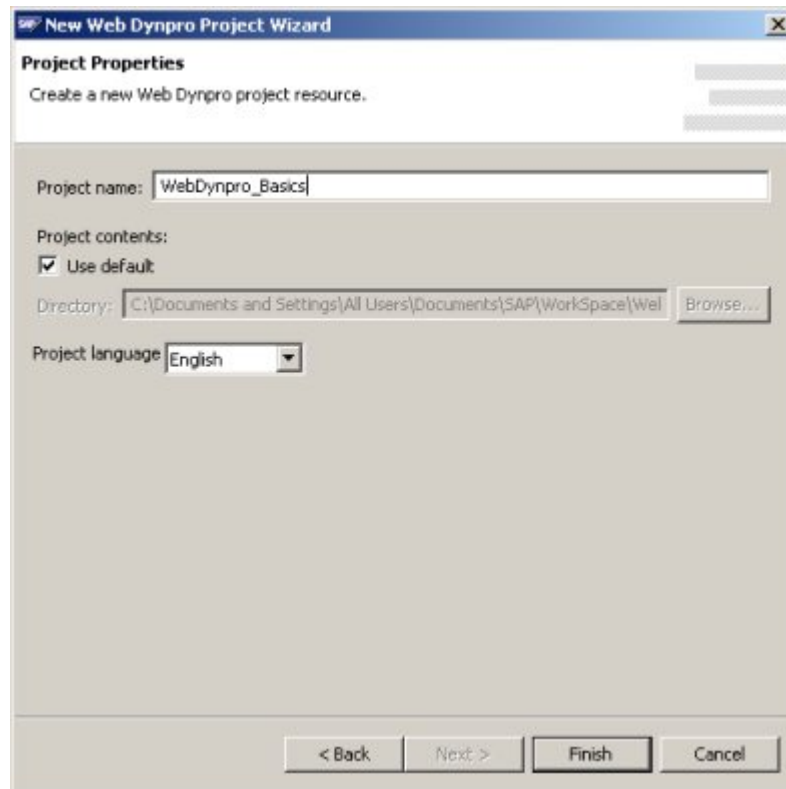
A Web Dynpro application is created within the SAP NetWeaver Developer Studio as part of a Web Dynpro project. Therefore, you must first create a new Web Dynpro project.

3.2.2 Procedure

1. Open the SAP NetWeaver Developer Studio.
2. Choose the arrow on the right of  with the tooltip *Open The New Wizard* followed by  to open the *New Project* wizard.




3. Select *Web Dynpro* and choose *Next*.
4. Enter a descriptive name for your Web Dynpro project, for example **WebDynpro_Basics**, and choose *Finish*.



By default, the corresponding directory for the project is in the workspace directory of your SAP NetWeaver Developer Studio. Therefore, *Use default* is checked under *Project contents*. If you want to store the project in a different directory, you must first uncheck the field before specifying the desired directory.

With the project language you specify which country code the language-specific files receive.

The system creates a directory structure, files, and references for the required Java archives as a framework for the still empty Web Dynpro project.

In the SAP NetWeaver Developer Studio, you will then see the *Web Dynpro Explorer*. In the function bar on the left, you can choose this perspective using the button  with the tooltip *Web Dynpro Perspective*.

3.2.3 Result

You automatically branch to the *Web Dynpro Explorer* and have created an empty shell for your first Web Dynpro application. You will continue to process your new Web Dynpro project by first creating a new Web Dynpro component and then using this component as the root component in the actual Web Dynpro application.

3.2.4 Further Procedure




The next step is to create the first [Web Dynpro Component \[Seite 6\]](#).

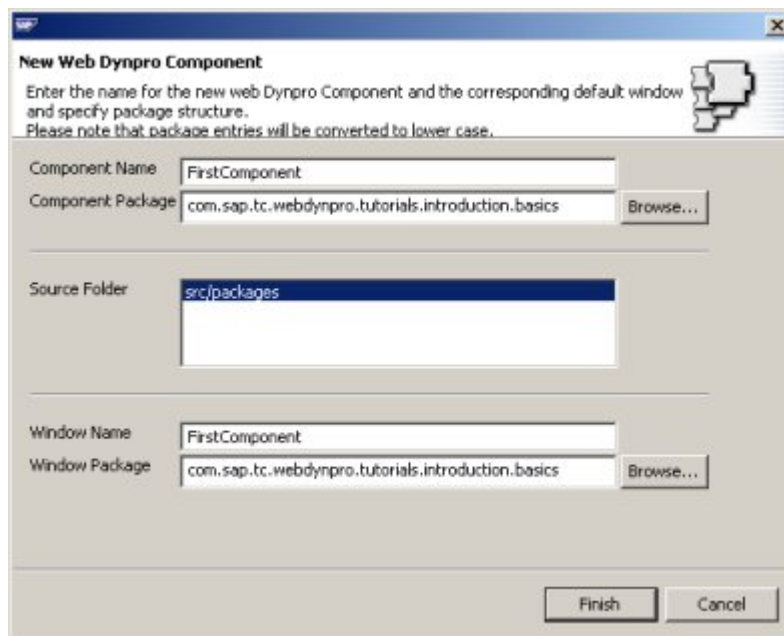
3.3 Creating a Web Dynpro Component

3.3.1 Use

A Web Dynpro application defines a starting point for a Web Dynpro component that the user can call in the Web browser. In the following procedure, you will create a new Web Dynpro component, which will then be used as the root component in your first Web Dynpro application.

3.3.2 Procedure

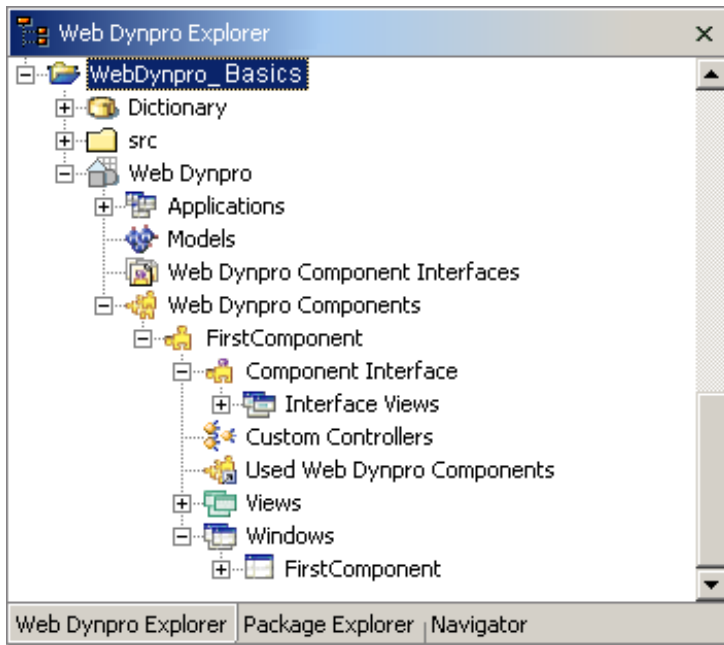
1. In the overview tree of the Web Dynpro Explorer, open your newly created project and then the *Web Dynpro* node. Open the context menu for the node *Web Dynpro Components*.
2. Choose  *Create Web Dynpro Component*.
3. Enter a name for the new Web Dynpro component, for example **FirstComponent**, and specify the Java package it is to belong to, for example **com.sap.tc.webdynpro.tutorials.introduction.basics**.



Do not change the default selection of *src/packages* as the *Source Folder*. This file is where the underlying metadata files for the Web Dynpro project are stored in the further development process.

3.3.3 Result

In the Web Dynpro Explorer, a newly added Web Dynpro component **FirstComponent** is visible under the node *Web Dynpro Components*:



You can now add the visible section to this currently empty shell.

3.3.4 Further Procedure



The next step is to create and design the first [View \[Seite 7\]](#).

3.4 Creating and Designing a View

3.4.1 Use

The layout of a rectangular unit of logically associated UI elements (for example, input fields, checkboxes, or radio buttons) is represented in a view, in which the user can interact with the application. When designing more complex user interfaces within a Web Dynpro application, a view is the smallest unit of UI elements that cannot be divided any further.

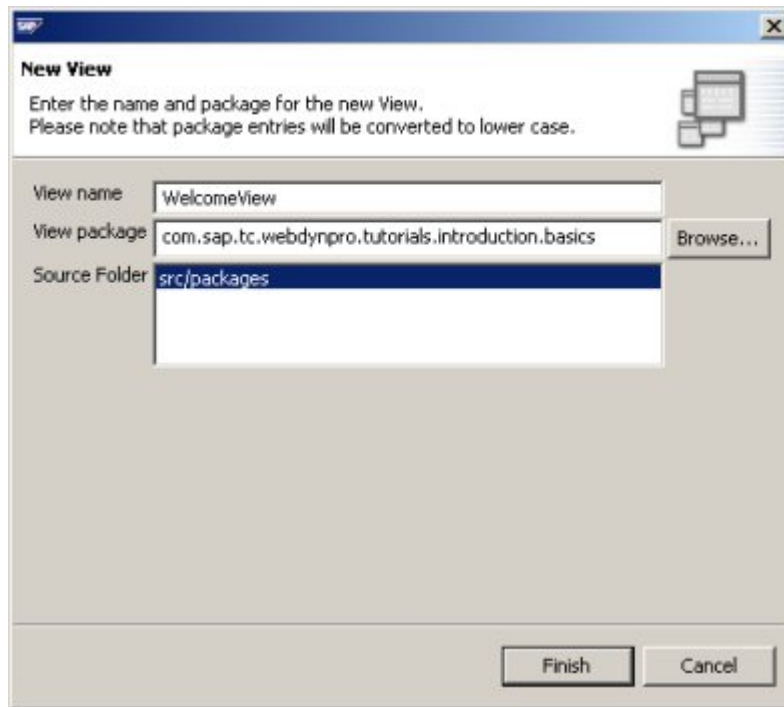
The following procedure describes how to design a simple user interface using the integrated *View Designer*.

3.4.2 Procedure

The first part of the procedure describes how to create a new view in the existing Web Dynpro component. In the second part, you design the layout and UI elements of the view using the *View Designer*.

3.4.2.1 Creating a New View

1. Open the newly created component in the overview tree of the Web Dynpro Explorer. Open the context menu for the *View* node.
2. Choose *Create View*.
3. Enter a name for the new view – for example, **WelcomeView** – and choose *Finish*.



After the completion of several generation routines, the new view is displayed under the *Views* node in the Web Dynpro Explorer.



Next to the Web Dynpro Explorer is the *View Designer*, which consists of several tabs. You can visually design a view in the *Layout* tab.

3.4.2.2 Designing the View Layout

Proceed with designing your first view. You want to embed a *TextView* UI element and an *Image* UI element in a *Group* UI element. At design time, the view is displayed in the View Designer of the SAP NetWeaver Developer Studio as follows:

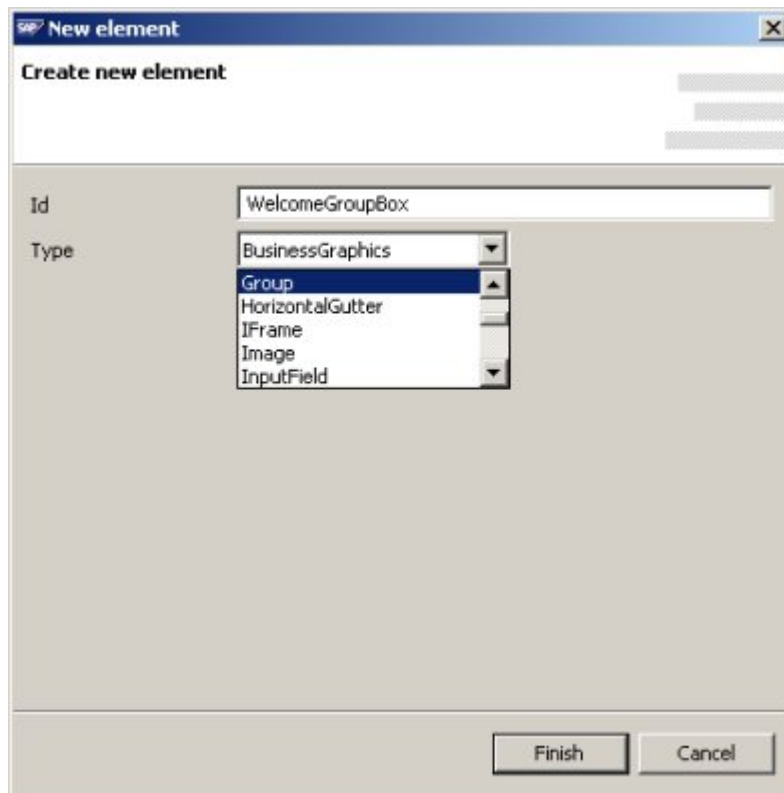


Add a *Group* UI element to the *RootUIElementContainer* UI element, which already contains a default *TextView* UI element, the *DefaultTextView* UI element.




The *RootUIElementContainer* UI element contains the *TransparentContainer* type and is integrated into each view as an invisible shell for all UI elements that are contained in it.

1. Choose *Insert Child* in the context menu of the UI element *RootUIElementContainer*[*Transparent Container*].
2. Enter a name for the UI element, for example **WelcomeGroupBox**. Select *Group* as the UI element type and choose *Finish*.



3. Fill the newly created UI element with the following UI elements. Choose *Insert Child* in the context menu of the *WelcomeGroupBox* UI element:
 - *TextView* control called **WelcomeText**. (In this example, you must delete the *DefaultTextView* element using the context menu or drag and drop it into **WelcomeGroupBox** if they have identical names.)
 - *Image* control called **WelcomelImage**

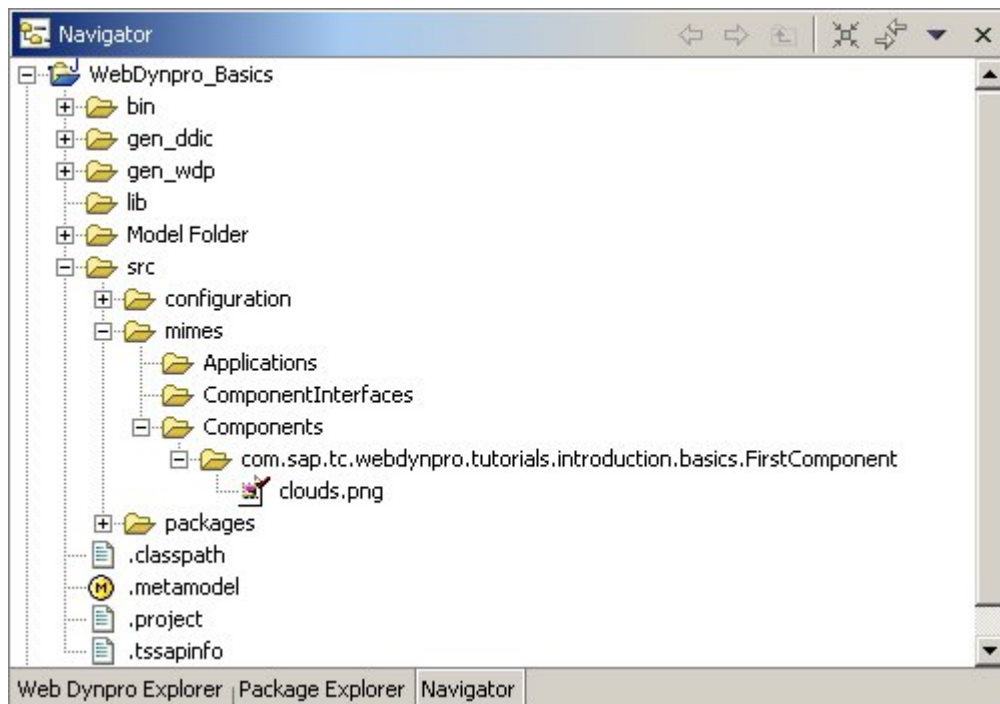
The various properties of a UI element can be edited in the *Properties* tab of the properties window in the View Designer.

4. Select the *WelcomeText* element in the outline window. Assign a static value to the individual properties in the *Properties* tab.
 - *text* property: **Welcome to Your First Web Dynpro Application**
 - *design* property: **header1** (this entry is on the right of the dropdown list box after selecting )

5. Assign the **Welcome_View** value to the text property of the WelcomeGroupBox_ Header [Caption] UI element.
6. Insert a picture by assigning its file name - in this example, **clouds.png** - as a value to the *source* property of the WelcomelmImage UI element.



7. The picture file must be stored in the directory `<WebDynproProjectFolder>/src/mimes/Components/<ComponentPackageName>`. In this example, the location is `C:/.../WebDynpro_Basics/src/mimes/Components/com.sap.tc.webdynpro.tutorials.introduction.basics`. This directory was automatically created when adding the Web Dynpro component FirstComponent.



The *layout* property of the *Group* control remains unchanged (FlowLayout). This means that all UI elements contained in it are displayed in one row and are only wrapped when the space of the corresponding container control is insufficient.

3.4.3 Result

You have developed the largest part of your first Web Dynpro application. However, the development process was purely declarative.

3.4.4 Further Procedure



Now, that you have created and designed the WelcomeView view in the Web Dynpro component FirstComponent, you can [include \[Seite 12\]](#) this view in the external visual representation of the Web Dynpro component, known as *Window*.

For further information on the UI element properties and methods that are available for the UI element interfaces, refer to UI Element Reference Guide.

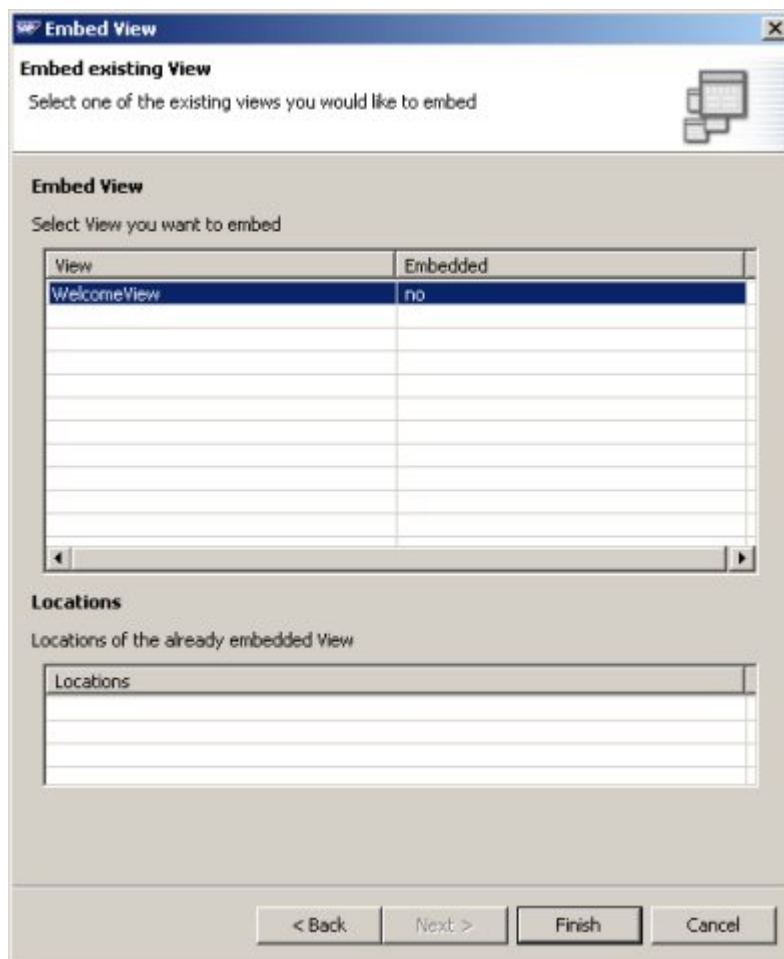
3.5 Defining a Web Dynpro Window

3.5.1 Use

A window is a representation of a Web Dynpro component that is visible to the outside, that is in the Web browser. Within a window, you can make several views visible by predefining a hierarchical rectangular group in a view composition. This example deals with the simplest case, in which a single view is embedded in the standard window of the Web Dynpro component FirstComponent.

3.5.2 Procedure

1. In the overview tree of the Web Dynpro Explorer, open your project and then the node *WebDynpro* → *Web Dynpro Components* → *First Component* → *Windows*.
2. Choose *Embed View* in the context menu of the node *FirstComponent*.
3. Choose *Embed Existing View*. In the next wizard window, select the only existing view *WelcomeView* and exit the wizard by choosing *Finish*.



This step allows you to make the Web Dynpro component FirstComponent visible to the outside by displaying the view WelcomeView.



Double-clicking the window FirstComponent displays the [Application Modeler \[Extern\]](#), which is integrated in the SAP NetWeaver Developer Studio.

3.5.3 Result

A Web Dynpro component is made visible to the outside using a corresponding window. After you have embedded the previously created view WelcomeView in the default view FirstComponent of the corresponding Web Dynpro component, you can use it as the root component for a Web Dynpro application that is addressed in the Web browser using a URL.

3.5.4 Further Procedure



In the next step, you can easily [define a new Web Dynpro application \[Seite 14\]](#).

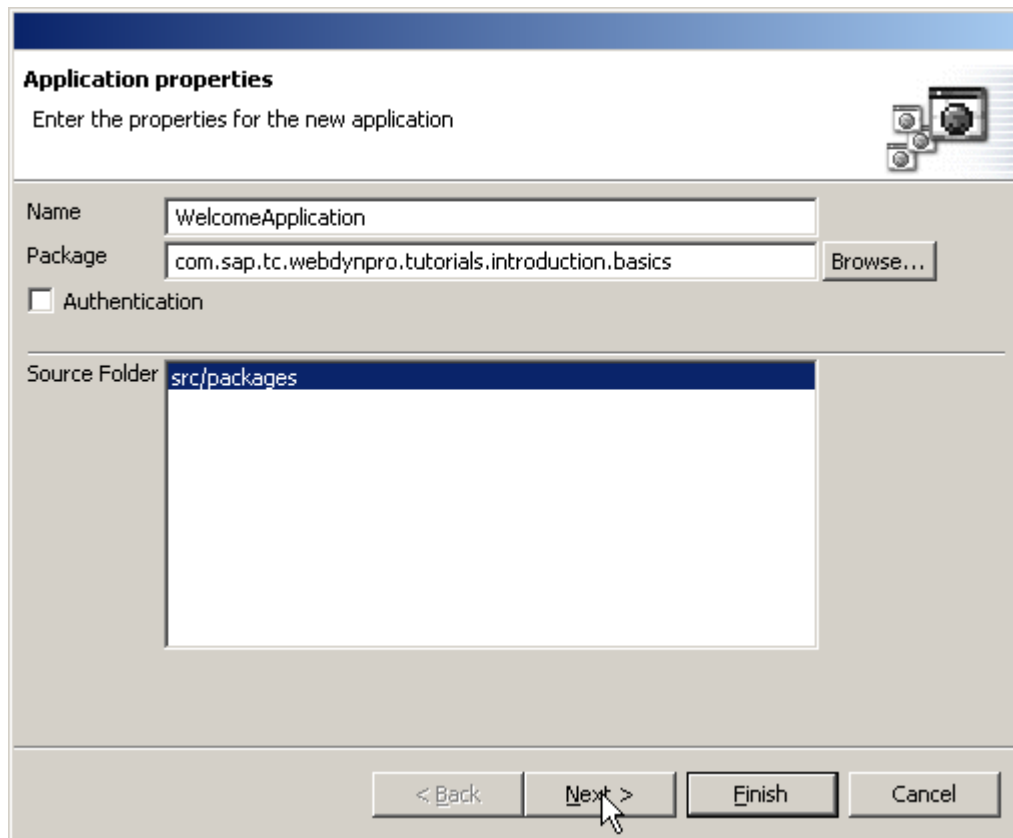
3.6 Defining a Web Dynpro Application

3.6.1 Use

The execution of a Web Dynpro application starts in the Web browser; the visual part of one Web Dynpro component (the root component of this Web Dynpro application) is displayed. The visual parts of a Web Dynpro component, which can be used from the outside, are the windows, of which exactly one is linked to the Web Dynpro application using the interface view of the window.

3.6.2 Procedure

1. In the Web Dynpro Explorer, choose *Create Application* in the context menu of the *Applications* node in your project.
2. Specify a descriptive name for the Web Dynpro application, for example **WelcomeApplication**, and choose *Next*.



3. Choose *Use existing Component* and *Next*.
4. Select the only available value for the Web Dynpro component, the interface view, and the startup plug.

3.6.3 Result

Your first Web Dynpro project now has callable Web Dynpro application.

3.6.4 Further Procedure



Based on the achieved state of the underlying metadata for the Web Dynpro project, you can now generate corresponding Java source code files. After these have been compiled, you can start creating a development unit that you can deploy in the SAP J2EE Engine. Next step: [Deploying a Web Dynpro Project \[Seite 16\]](#).

3.6.5 Deploying a Web Dynpro Project

3.6.6 Use



The deployment process must be executed to display the Web Dynpro application, which belongs to the Web Dynpro project, using a URL in the Web browser. The Web Dynpro project is stored as an independent deployment unit in an appropriate directory in the SAP J2EE Engine.

3.6.7 Requirements



Before you start the tutorial, make sure that you have started the SAP J2EE Engine. The deployment unit of the Web Dynpro project will be stored in its directory structure.

3.6.8 Procedure

1. Choose  with the tooltip *Save All Metadata* to save the metadata. This function is inactive if it is not necessary to store the metadata again.
2. In the Web Dynpro Explorer, choose  (*Rebuild Project*) in the context menu of the project node. The Java source text files and additional project-specific files, such as Java resource bundles or Java property files, are generated.
3. Choose *Create Archive* in the same context menu that you used before. The deployable development unit (archive) of your Web Dynpro project is generated. As a result, the file <Web Dynpro project name>.ear is stored in the corresponding Web Dynpro project directory. In this example, the file WebDynpro_Basics.ear is created.
4. Choose *Deploy* in the same context menu that you used before. The archive that you stored in the project directory is now available for the SAP J2EE Engine.

3.6.9 Result

You have successfully deployed your Web Dynpro project and can start your first Web Dynpro application in the Web browser. For further information, refer to [Structure of a Web Dynpro Application \[Seite 24\]](#).

3.6.10 Further Procedure



Take a look at the WelcomeView view developed by yourself by [calling the Web Dynpro application \[Seite 17\]](#) in your Web browser.

3.6.11 Calling a Web Dynpro Application

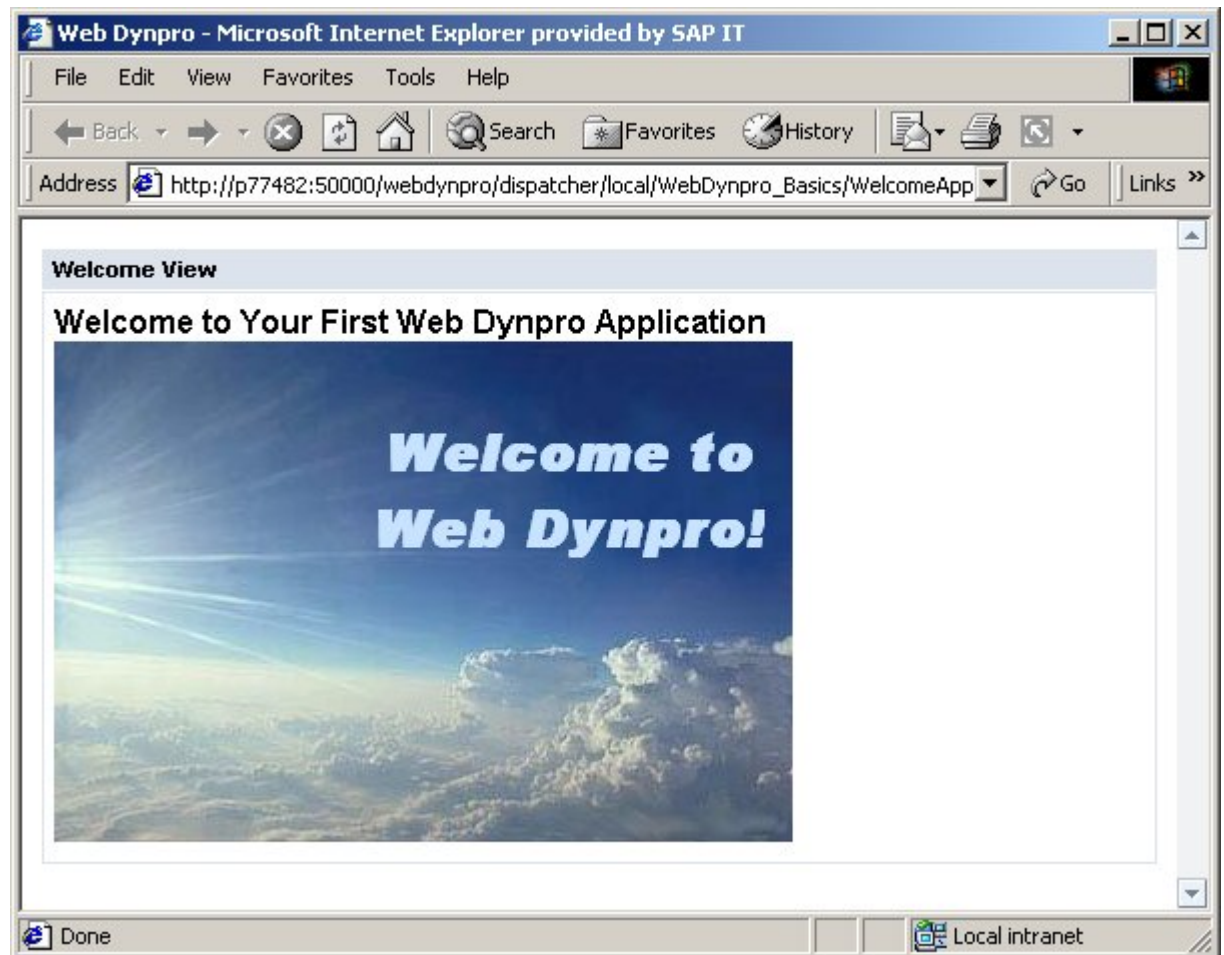
3.6.12 Use

You can call a Web Dynpro application using a unique URL in the Web browser. To do this, you must have successfully deployed the relevant archive of the Web Dynpro project in the SAP J2EE Engine.

3.6.13 Procedure

1. In the Web Dynpro Explorer, choose *Run* in the context menu for the node *WebDynpro* → *Applications* → *WelcomeApplication* of your Web Dynpro project.

A Web browser opens automatically displaying the view you designed.



3.6.14 Result

You have successfully created your first Web Dynpro project including an integrated Web Dynpro application.

The entire development process of the application was purely declarative and carried out without programming a single line of Java source code. The required Java source code was generated automatically by the SAP NetWeaver Developer Studio.

3.6.15 Further Procedure

This Web Dynpro project will now be extended to include a programmatic section, which you will implement in your Web Dynpro application yourself. The view is to display the current data and time. In so doing, you will be familiarized with the concept of [Data Binding \[Extern\]](#) of user interface elements to a [Context \[Extern\]](#) as a structured storage place.



Carry out the enhancement [Displaying the Current Date Using Data Binding \[Seite 19\]](#).

3.7 Displaying the Current Date Using Data Binding

3.7.1 Use

To display programmatically specified data (for example, the current time) in a view, you can use data binding between the properties of UI elements and a structured storage place, which is known as context.



This simple example only uses the view context, which is contained in every view.

In the following exercises you use the static text *Welcome to Your first Web-Dynpro-Application*, which is displayed in the *WelcomeText* UI element of your view. You will add the programmatically specified information of the time and date to the static text.

3.7.2 Procedure

You declaratively define a value attribute for the Web Dynpro context **WelcomeText** and programmatically assign a value to the value attribute in a predefined handling routine of the view controller. Finally, you bind the **WelcomeText** property of the *TextView* control to this context attribute (declaratively).

3.7.2.1 Declaring a Value Attribute for the Web Dynpro Context

1. Open your *WelcomeView* view in the View Designer and select the *Context* tab.
2. Choose *New* → *Value Attribute* in the context menu of the *Context* node. Enter the name **WelcomeText** for this value attribute and choose *Finish*.

The properties that belong to a value attribute are displayed in the *Properties* tab in the lower section of the View Designer. By default, the underlying data type of a value attribute for the Web Dynpro context is declared as string.

3. Select the *Implementation* tab. You can modify the implementation source code of the view controller in the corresponding user-specific program sections. This source code is generated by the system.

3.7.2.2 Changing the State of the View Context Using the Program

4. Add the following code to the *wdDoInit()* method, which is automatically executed when the view controller is initialized by the Web Dynpro runtime environment.

```
/** Hook method called to initialize controller. */
public void wdDoInit()
{
    //@ @begin wdDoInit()
    String welcomeText = "Welcome to Web Dynpro, it's ";
    welcomeText += Calendar.getInstance().getTime().toString();
    welcomeText += "!";
    wdContext.currentContextElement().setWelcomeText(welcomeText);
    //@ @end
}
```



After defining the new value attribute for the view context and regenerating several classes and interfaces, you can access the root node of the view context using `wdContext.currentContextElement()` and assign a new value to the value attribute `WelcomeText` using the `setWelcomeText(String text)` method.



In this example, the text to be displayed is part of the source text. You should not use this procedure if you want to internationalize your application.

5. Insert the following text at the beginning of the source text to access the static methods of the `java.util.Calendar` class:

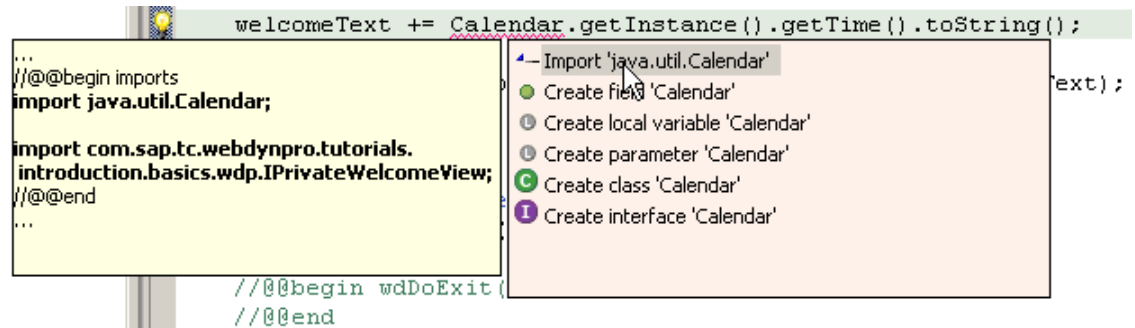
```

@@@begin imports
import com.sap.tc.webdynpro.tutorials.introduction.basics.wdp.IPrivateWelcomeView;
import java.util.*;
6.  @@@end

```



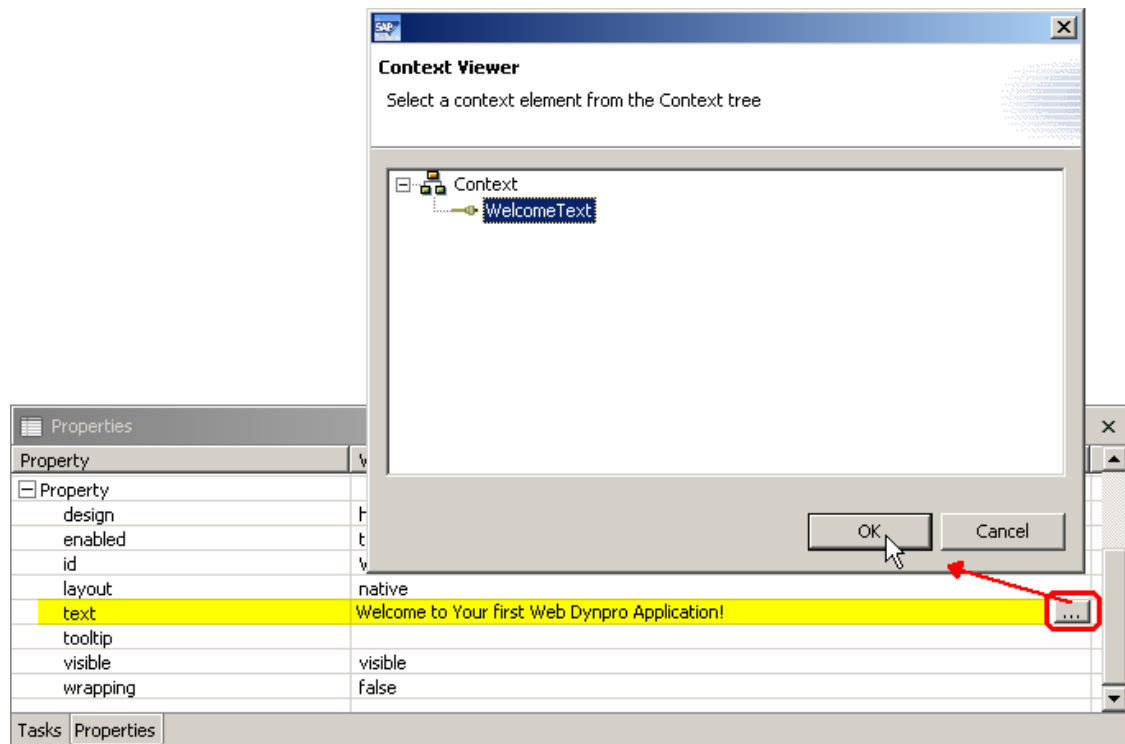
You can also insert the missing source code using the Quick Fix function of the SAP NetWeaver Developer Studio. Place the cursor on the wavy underlined character string `Calendar`. Select **CTRL + 1** and double-click **Import 'java.util.Calendar'**. The import line is inserted automatically.



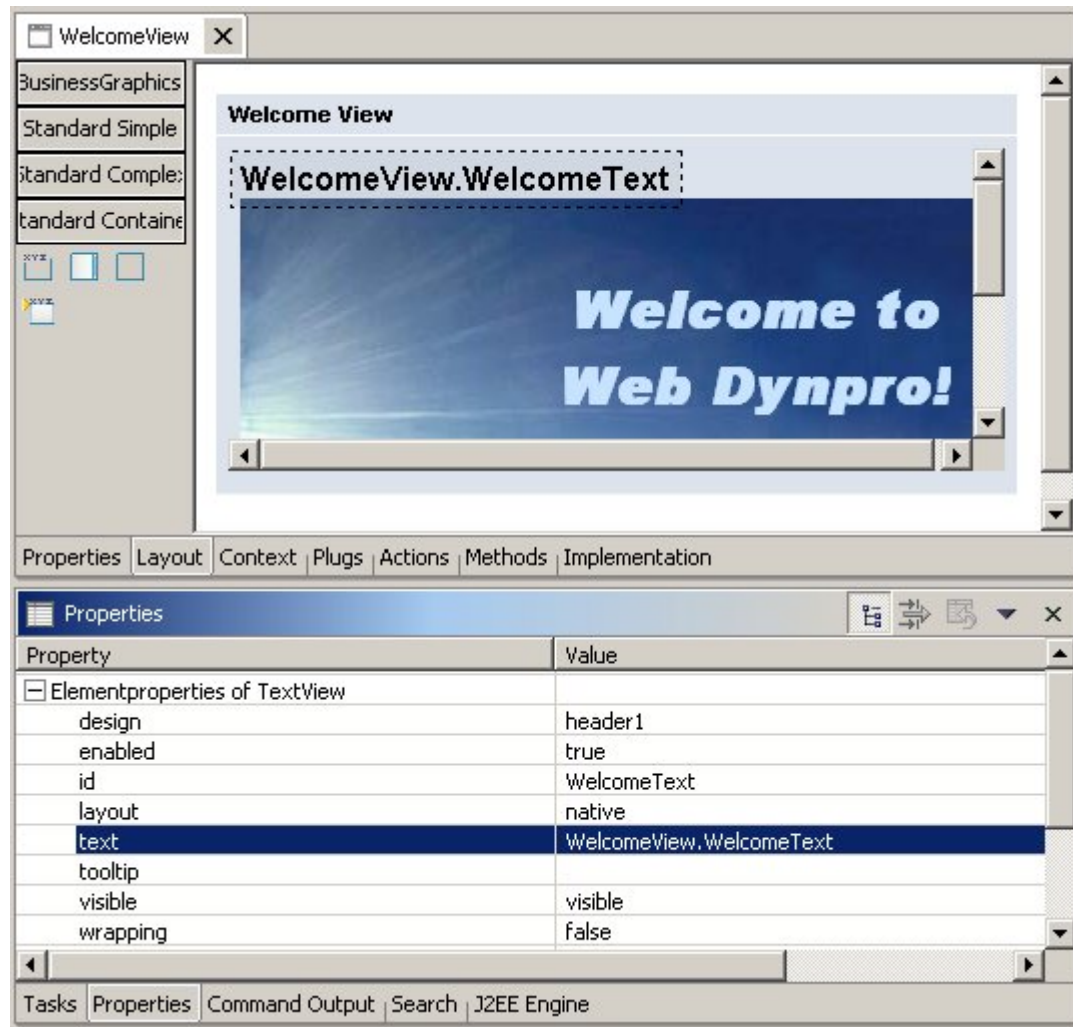
3.7.2.3 Data Binding

For the data binding between your `TextView` UI element and the value attribute of the view context, proceed as follows:

7. Select the *Layout* tab in the View Designer and highlight your *TextView* UI element `WelcomeText` in the outline area.
8. In the lower section of the View Designer, on the *Properties* tab of the text property, choose the button at the right of the value field.
9. Select the `WelcomeText` value attribute in the Context Viewer and choose *OK*.



You can recognize the data binding of a UI element property to a value attribute of the view context by the address `WelcomeView.WelcomeText`, which is the property value. The general naming convention for this address is as follows: `<view name>.<value attribute name>`.

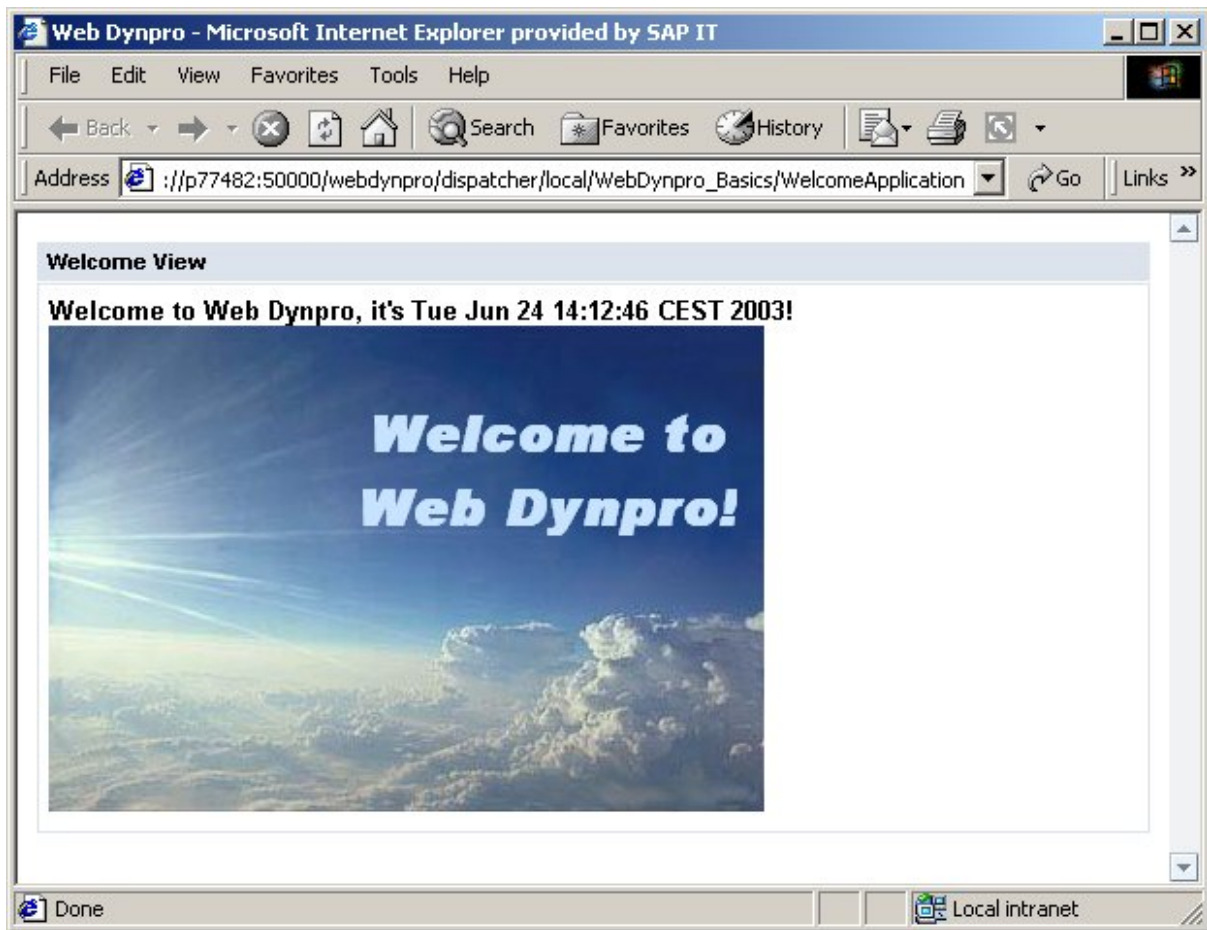


3.7.2.4 Redeploying and Starting the Application

For detailed information, refer to [Deploying a Web Dynpro Projects \[Seite 16\]](#).

10. Save the metadata.
11. Start the generation.
12. Start the deployment and the execution of the application in the context menu of the application.

3.7.3 Result

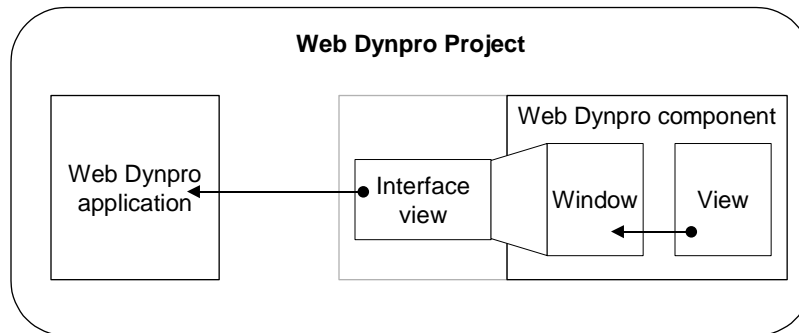


You have successfully created your first Web Dynpro application. You used data binding between a UI element property in the view layout and a value attribute in the view context to programmatically define a text and display it.

3.8 Structure of a Web Dynpro Application

3.8.1 Definition

A Web Dynpro application has the following structure:



The Web Dynpro project represents an external unit that contains different parts and can be deployed. The Web Dynpro applications are the parts of a Web Dynpro project that can be called from a Web browser.

Each Web Dynpro application contains exactly one Web Dynpro component (root component). A Web Dynpro component contains one or more windows, which are the visual representation of the Web Dynpro component. Windows consist of one or more nested views that contain the UI elements.

You establish a connection between a window of the Web Dynpro component and the application to bind the component to a Web Dynpro application. The interface view is used as the interface and there is exactly one interface view for each window.

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®, NT®, EXCEL®, Word®, PowerPoint® and SQL Server® 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 and Informix® Dynamic Server™ are trademarks of IBM Corporation in USA 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®, the Citrix logo, ICA®, Program Neighborhood®, MetaFrame®, WinFrame®, VideoFrame®, MultiWin® and other Citrix product names referenced herein are trademarks of Citrix Systems, Inc.

HTML, DHTML, XML, XHTML 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.

MarketSet and Enterprise Buyer are jointly owned trademarks of SAP AG and Commerce One.

SAP, SAP Logo, R/2, 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 trademarks of their respective companies.