

# UI Elements Solution



**Chapter:** UI Elements

**Theme:** Container UI Elements



At the end of this Exercise, you are able to:

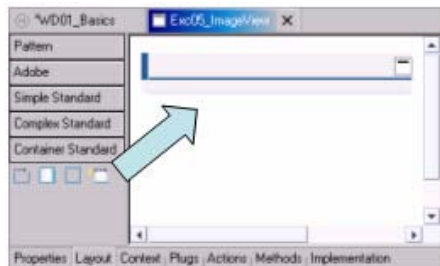
- Define an Image UI Element.
- Place the image within a Tray UI Element.

## 4 Developing, Step-by-Step

4-1 Add a Tray UI element to the view *ImageView*.

In the Web Dynpro Explorer: Expand the nodes *WD01\_Basics\_UI\_01 / Web Dynpro / Web Dynpro Components / Exc\_UI\_01 / Views*.

Select the node *ImageView* and choose *Edit* from the context menu.



The View Designer for the view *ImageView* appears on the right pane

In the toolbar, choose *Container Standard*. Drag and drop *Tray* to the editor pane on the right.

Assign the following property values to the Tray UI element:

TrayHeader: text	Image view, Tray title
------------------	------------------------

Save the new project data by choosing *Save All Metadata* from the toolbar.

#### 4-2 Add a TextView to the Tray UI element.



In the Outline view, select the Tray element and choose *Insert Child* from the context menu.

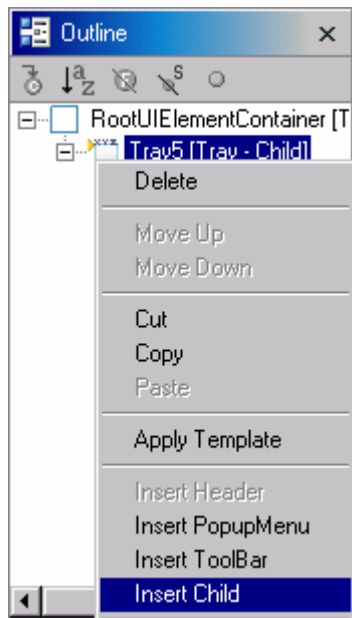
Add a TextView UI element to the Tray element.

Assign the following property values to the TextView UI element:

text	Tray text
------	-----------

Save the new project data by choosing *Save All Metadata* from the toolbar.

#### 4-3 Add a Group to the Tray UI element.



In the Outline view, select the Tray element and choose *Insert Child* from the context menu.

Or:

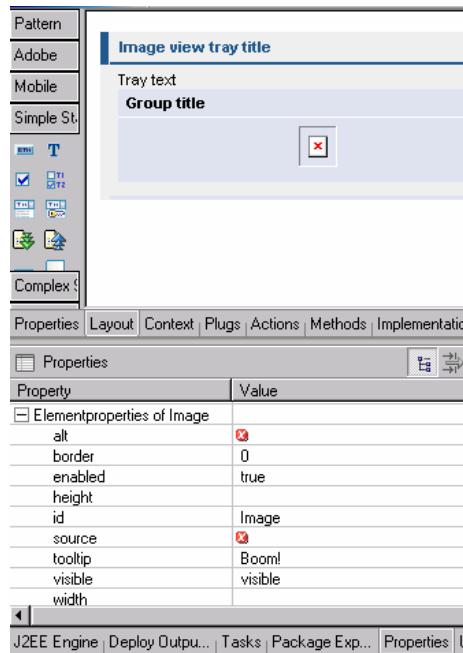
From the toolbar, choose *Container Standard* and drag and drop *Group* to the editor pane.

Assign the following property value to the Group Caption UI element:

text	Group title
------	-------------

Save the new project data by choosing *Save All Metadata* from the toolbar.

#### 4-4 Add an Image to the Group UI element.



In the Outline view, select the Group element and choose *Insert Child* from the context menu.

Or:

From the toolbar, choose *Simple Standard* and drag and drop *Image* to the editor pane.

Assign the following property values to the Image UI element:

alt	explosion.gif
Source	explosion.gif
Tooltip	Boom!

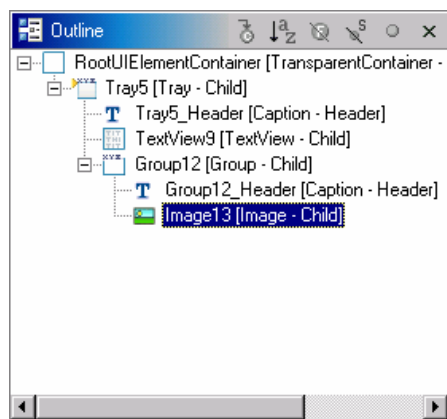
The simplest way to insert a picture is by assigning its file name - in this example, *explosion.gif* - as a value to the *source* property of the Image UI element. The picture file must be stored in the directory `<ProjectFolder>/src/mimes/Components/<ComponentPackageName>`.

This directory was automatically created when adding the Web Dynpro component Exc\_UI\_01.

(Optional: Have a look at Step 6. There you will find, how you can set the *source* property programmatically)

Save the new project data by choosing *Save All Metadata* from the toolbar.

#### Successful Result



The Outline View shows the nested UI containers and UI elements.

#### 4-5 Create a Web Dynpro application

In the Web Dynpro Explorer, expand the node *WD01\_Basics\_UI\_01*.  
Expand the node *Web Dynpro* and open the context menu for *Applications*.

To open the wizard, choose *Create Application*.

In the wizard, enter:

Name	WD01_Basics_UI_01
Package	com.sap.training.wd.basics.ui01

Accept the other suggested values and choose *Next*.

Select *Use existing component* and choose *Next*.

Select

Web Dynpro Component	Exc_UI_01
Interface View	Exc_UI_01_WindowInterfaceView
Startup Plug	Default

and choose *Finish*.

Successful result:

The generated Web Dynpro application object completes your project structure.  
You can now deploy your project and start your Web Dynpro application.  
*WD01\_Basics\_UI\_01* enables you to address the Web application as a whole,  
when you launch this complete application in the next step.

## 5 Building, Deploying, and Running, Step-by-Step

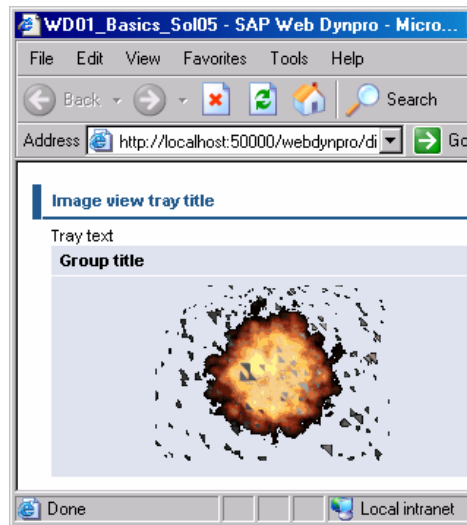
Deploy and run the Web Dynpro application.

In the Web Dynpro Explorer:

Expand the node *WD01\_Basics\_UI\_01 / Web Dynpro / Applications*.  
Open the context menu for *WD01\_Basics\_UI\_01*

To deploy and run the application, choose *Deploy new Archive and Run*

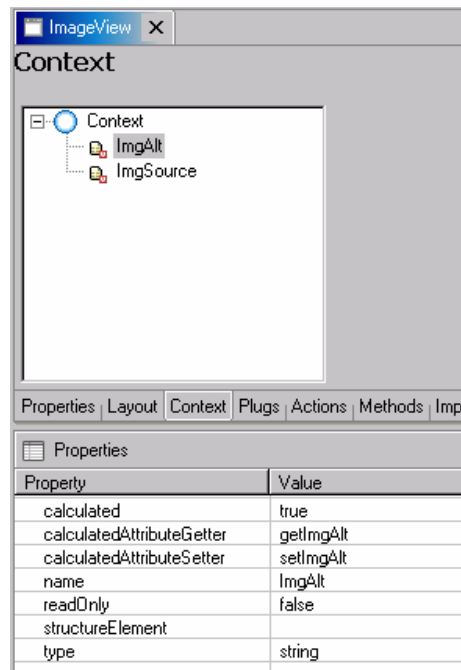
Successful result:



The Developer Studio launches the Web browser and chooses the active view *ImageView*.

## 6 Optional: Setting image properties programmatically, Step-by-Step

### 6-1 Bind the image properties *alt* and *source* to the context of the view.



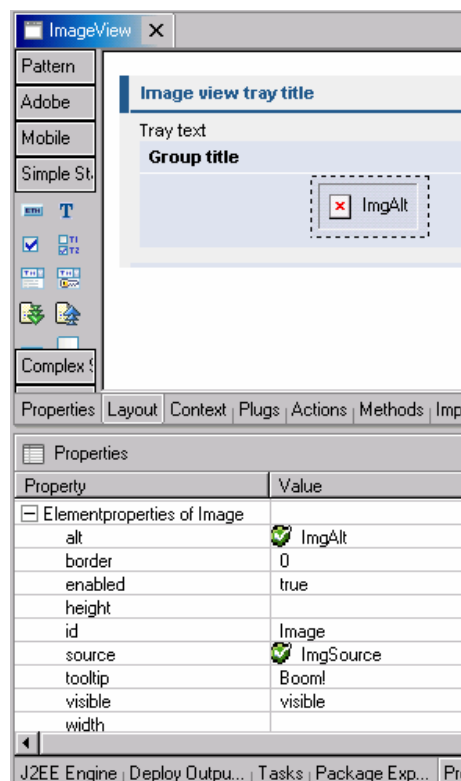
In the View Designer, choose the *Context* tab.

Add two value attributes (*ImgAlt* and *ImgSource*) to the root node *Context*.

Set the following property values of the context elements *ImgSource* and *ImgAlt*:

calculated	true
------------	------

This will automatically generate a getter- and a setter-method for the context value attributes *ImgSource* and *ImgAlt*, respectively.



In the View Designer, choose the tab *Layout*.

Assign the following property values to Image UI element

alt	ImgAlt
source	ImgSource

This will bind the image properties to the view context.

Save the new project data by choosing *Save All Metadata* from the toolbar.

## 6-2 Set the image properties *source* and *alt* programmatically.

In the View Designer, choose the tab *Implementation*, select the generated methods *getImgSource* and *getImgAlt*. Add the following return statement:

```
//@@begin javadoc:getImgSource( . . .)
/**
 * Declared getter method for attribute ImgSource of node
 * Context @param element the element requested for the value
 * @return the calculated value for attribute ImgSource
 */
//@@end
public java.lang.String getImgSource(
    IPrivateImageView.IContextElement element) {
    //@@begin getImgSource( . . .
    return "explosion.gif";
    //@@end
}

. . .

//@@begin javadoc:getImgAlt( . . .)
/**
 * Declared getter method for attribute ImgAlt of node
 * Context @param element the element requested for the value
 * @return the calculated value for attribute ImgAlt
 */
//@@end
public java.lang.String getImgAlt(
    IPrivateImageView.IContextElement element) {
    //@@begin getImgAlt( . . .
    return "This image is not available";
    //@@end
}
```

Save the new project data by choosing *Save All Metadata* from the toolbar.

Rebuild, deploy and run the application.