Creating Notifications in Web Dynpro Applications Using the JavaMail API

SDN Community Contribution

(This is not an official SAP document.)

Disclaimer & Liability Notice

Creating Notifications in Web Dynpro Applications Using the JavaMail API

Applies To:

SAP Technology, SAP Web Application Server, Web Dynpro

Summary

This tutorial describes how to design, develop, deploy, and run a Web Dynpro application that uses the JavaMail API to send notifications.

By: Suresh K R, Associate Consultant

Reviewed By: UmaSankar Subramanian, Solution Architect, SAP NetWeaver Competence Group (NWCG), Wipro Technologies Ltd., Bangalore, India

Company: Wipro Technologies Ltd, Bangalore, India

Date: 13 September 2005

Table of Contents

About this Tutorial

This tutorial describes the step-by-step procedure for how to design, develop, deploy, and run a Web Dynpro application that uses the Java mail API to send notifications. The tutorial considers a scenario in an organization in which employees must request capital goods through the intranet, a process which must meet the approval from certain authorities. In such cases, notifications are sent to the appropriate persons through email. For this purpose, the Java Mail API can be integrated with Web Dynpro applications to send notifications. This tutorial considers only the notification part; this can be further integrated with any relational database management system (RDBMS) or SAP R/3 application to develop the complete application.

Prerequisites

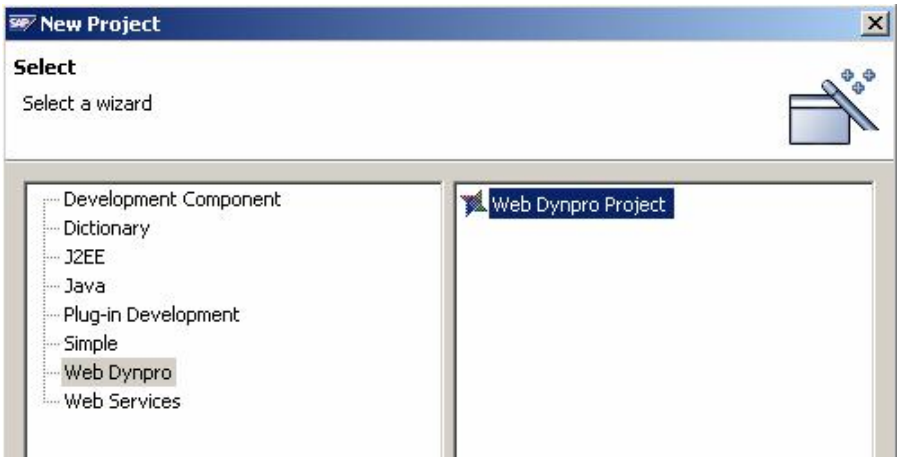
- You have launched the SAP NetWeaver Developer Studio on your computer.
- You have access to the SAP J2EE Engine (Release 6.40).
- You have access to a mail server.

Creating a Web Dynpro Project

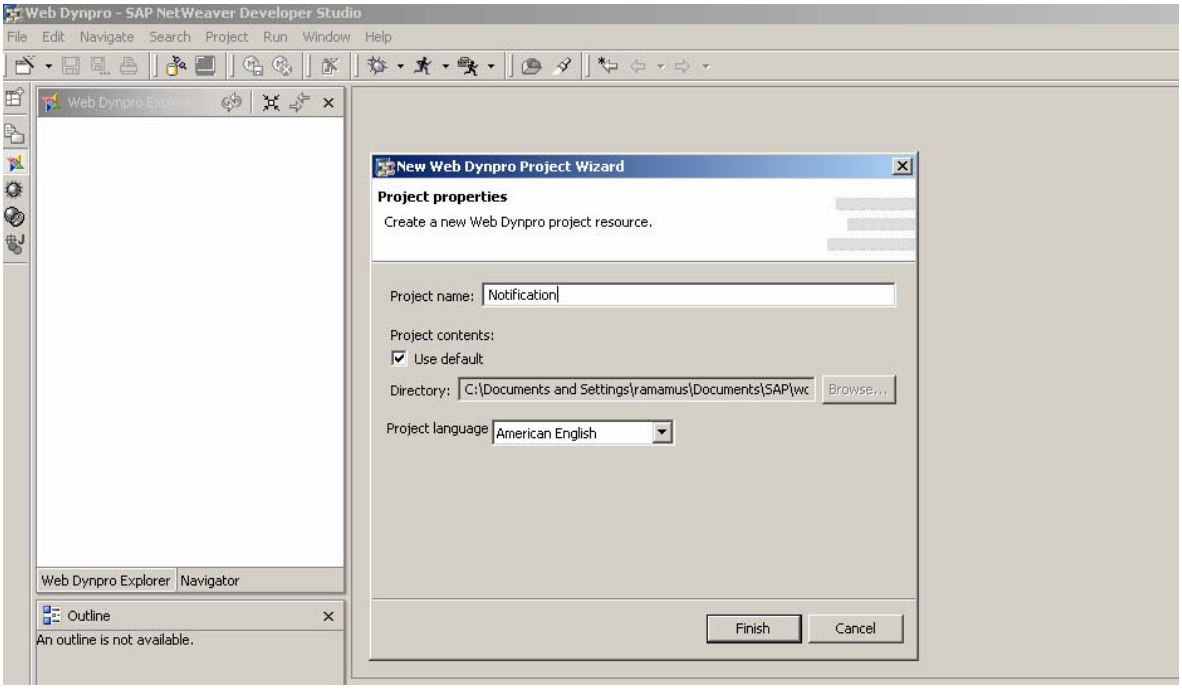
Procedure

1. Choose File → New → Project. The New Project wizard appears.

2. Select the Web Dynpro category (in the left pane), followed by Web Dynpro Project (in the right pane). Choose Next.



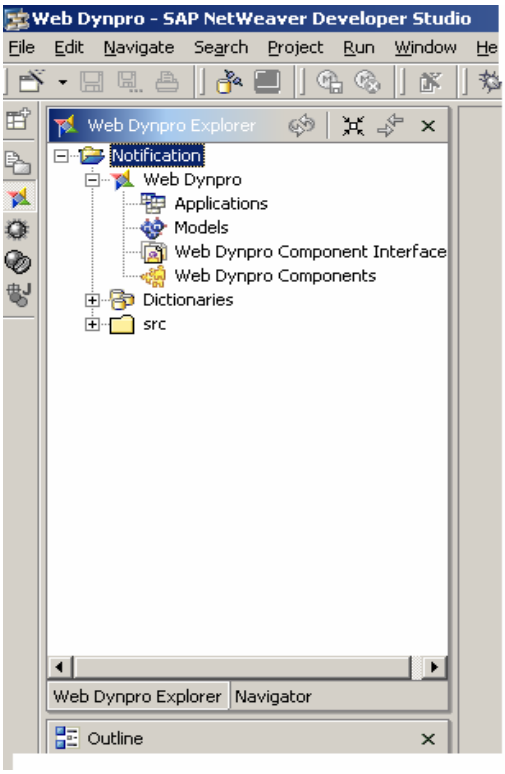
3. Enter the project name as **Notification**.



4. Choose Finish.

5. The wizard generates an initial structure for your new Web Dynpro project and automatically opens the Web Dynpro perspective.

6. Choose the Web Dynpro Explorer tab.



Creating a Web Dynpro Component

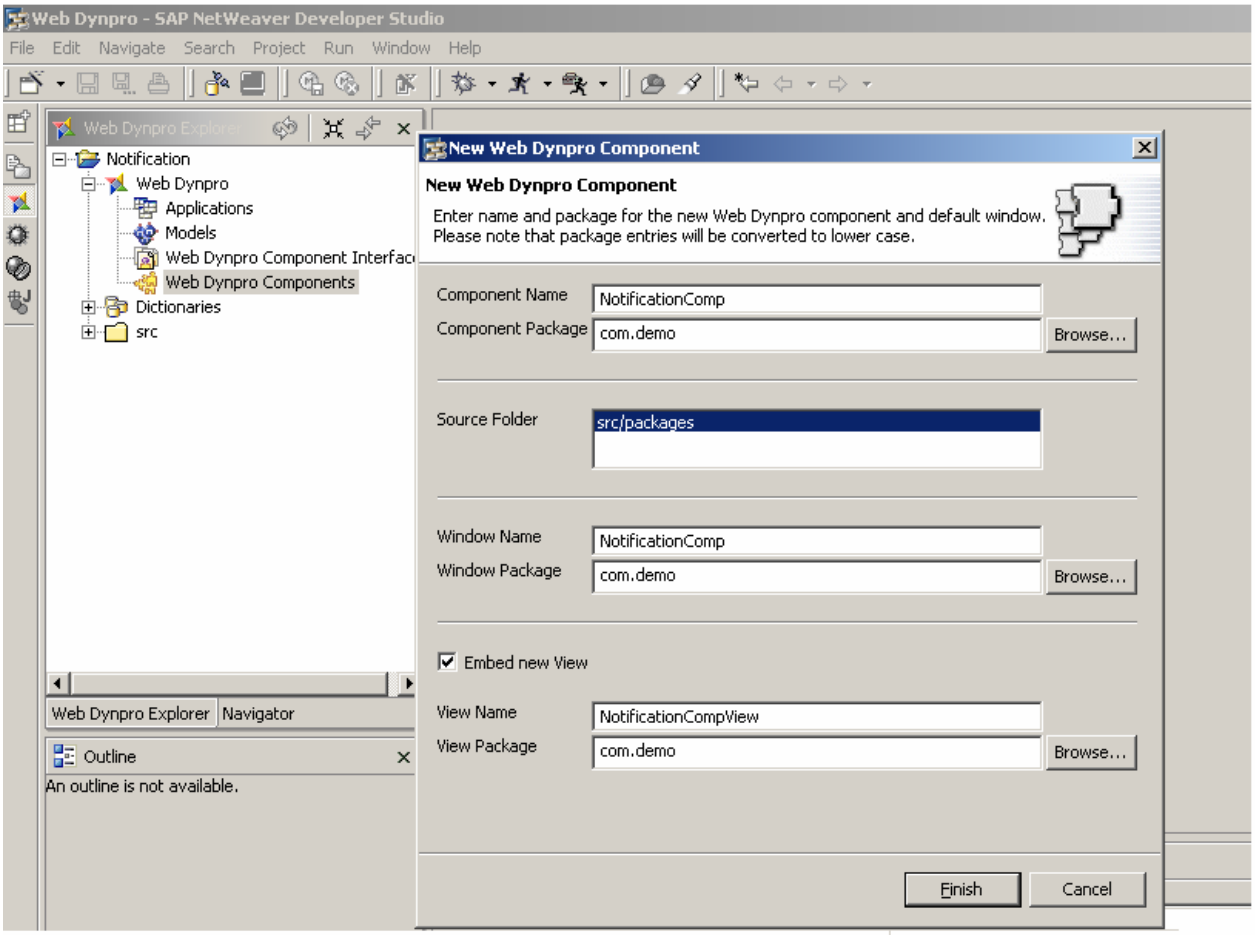
Procedure

1. Expand the *Web Dynpro* node and open the context menu for *Web Dynpro Components*.

2. To open the wizard, choose *Create Web Dynpro Component*.

3. Enter the name **NotificationComp** for your Web Dynpro component and specify the package name (such as **com.demo**) for the Java classes that will be generated.

4. Enter **NotificationView** for the view name

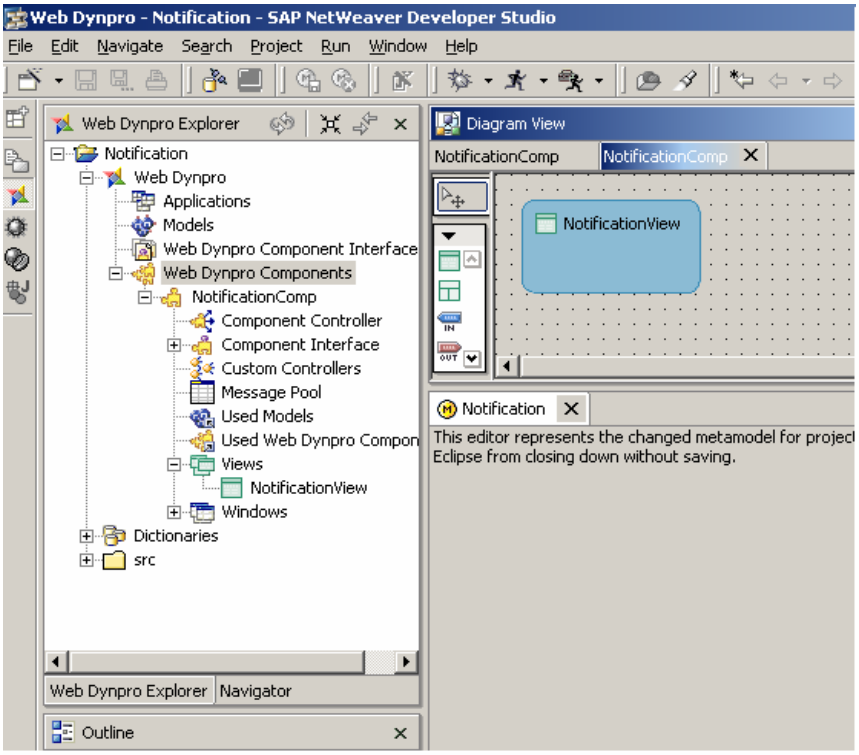


5. Accept the other suggested values and choose *Finish*.

6. Save the new project data by choosing the icon from the toolbar.

Result

The wizard performs several generation routines. Once it has finished, it will develop a Web Dynpro component with required views, models, and controllers.



Designing a Layout for the NotificationView

Procedure

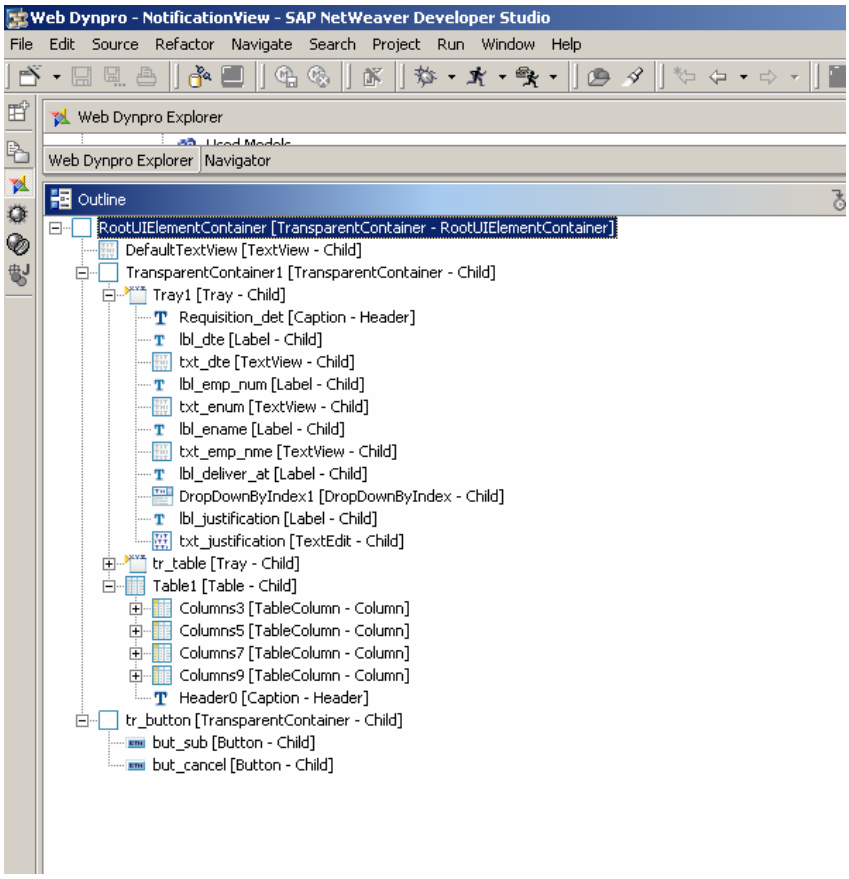
1. Launch the View Designer by double-clicking the NotificationView node in the project structure.

2. The Layout tab in the View Designer shows the NotificationView with a predefined default text. Simultaneously, the Outline view displays a list of the UI elements included. All the UI elements are arranged under a root node and are represented in order in the tree in the layout. If you select an element in the Outline view or on the Layout tab, its associated element properties are shown in the Properties view – provided you have previously selected the Properties tab that is at the bottom of your screen.

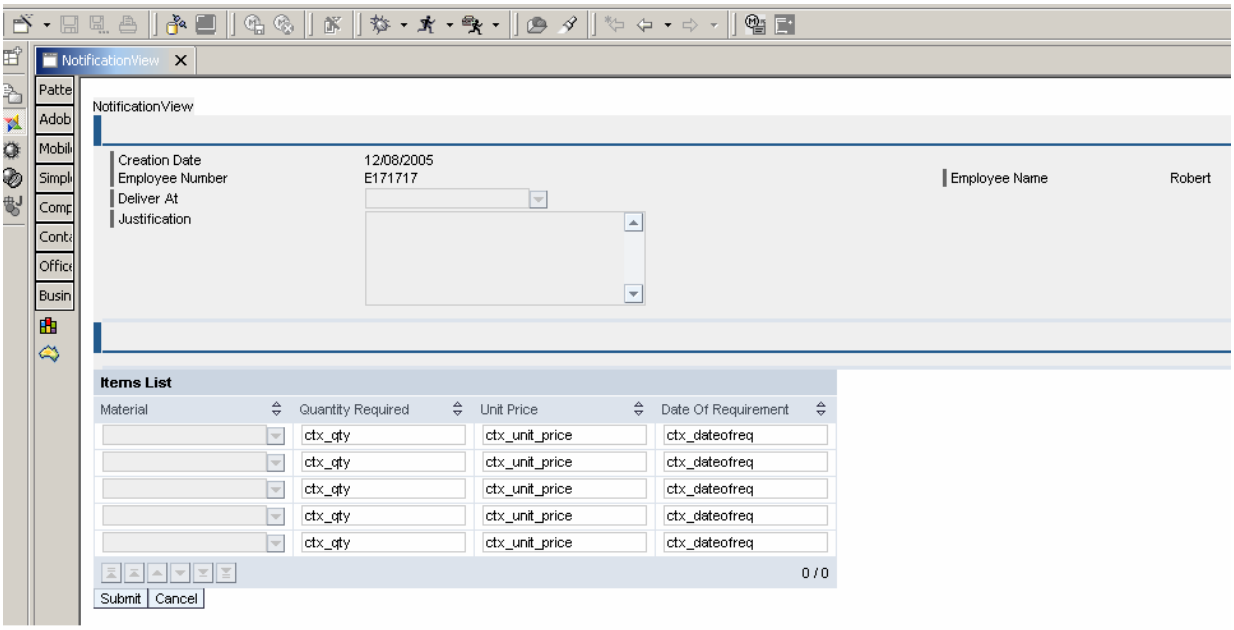
3. In the *Outline* view, select the root element **RootUIElementContainer** and choose *Insert Child* from the context menu. Create the UI elements as shown below in the Outline view.

4. Create and assign the contexts to the UI elements wherever required.

5. After creation of UI elements Outline view looks as shown below.



The *View Designer* displays the following layout for the **NotificationView**



Creating the Java Class that Sends Notifications

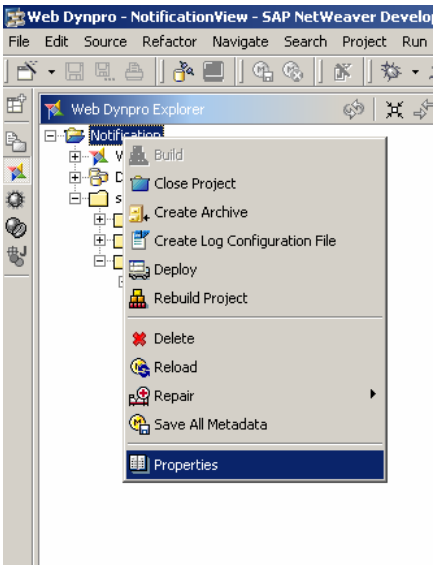
```
//Java code starts here
import java.util.Properties;
import javax.mail.Authenticator;
import javax.mail.Message;
import javax.mail.PasswordAuthentication;
import javax.mail.Session;
import javax.mail.Transport;
import javax.mail.internet.InternetAddress;
import javax.mail.internet.MimeMessage;
public class SendMail {
  public boolean sendMail(String from, String to) {
        boolean res = true;
        try {
              //enter the mail host name here
              String host = "hostname";
              // Get system properties
```

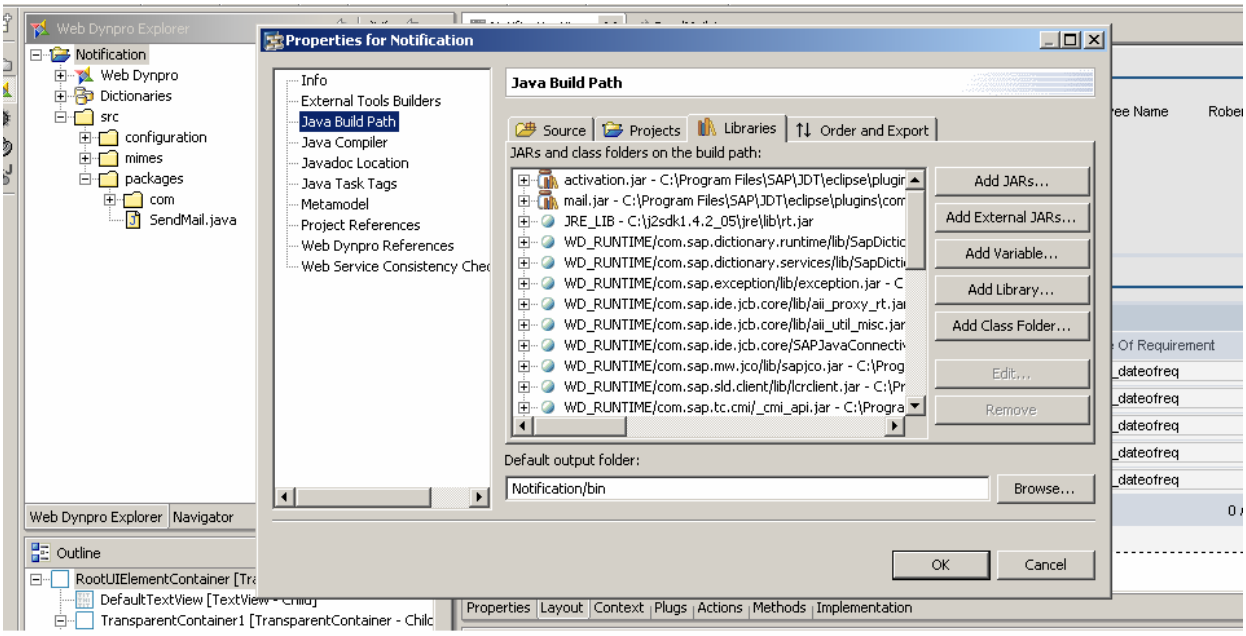
```
                Properties props = System.getProperties();
                // Setup mail server
                props.put("mail.smtp.host", host);
                props.put("mail.smtp.auth", "true");
                //  Setup authentication, get session
                Authenticator auth = new PopupAuthenticator();
                Session session = Session.getInstance(props, auth);
                // Define message
                MimeMessage message = new MimeMessage(session);
                      message.setFrom(new InternetAddress(from));
                                  message.addRecipient(
                      Message.RecipientType.TO,
                      new InternetAddress(to));
                message.setSubject("Notification");
                message.setText("A new request has been placed");
                // Send message
                Transport.send(message);
          } catch (Exception e) {
                res = false;
                System.out.println("Exception while sending mail" +
e.getMessage());
                e.printStackTrace();
          }
          return res;
    }
}

//This class is required only when authentication is needed
class PopupAuthenticator extends Authenticator {
    public PasswordAuthentication getPasswordAuthentication() {
          String username, password;
          return new PasswordAuthentication(username,password);
    }
}
//Java code ends here
```

Including the Java Class in Web Dynpro Project

Procedure

1. Copy the Java class in the Src→Packages folder.
2. Copy mail.jar and activation.jar in lib folder (these jar files are available in the plug-ins folder of NWDS)
3. Add mail.jar and activation.jar in the build path of your Web Dynpro project as shown below.





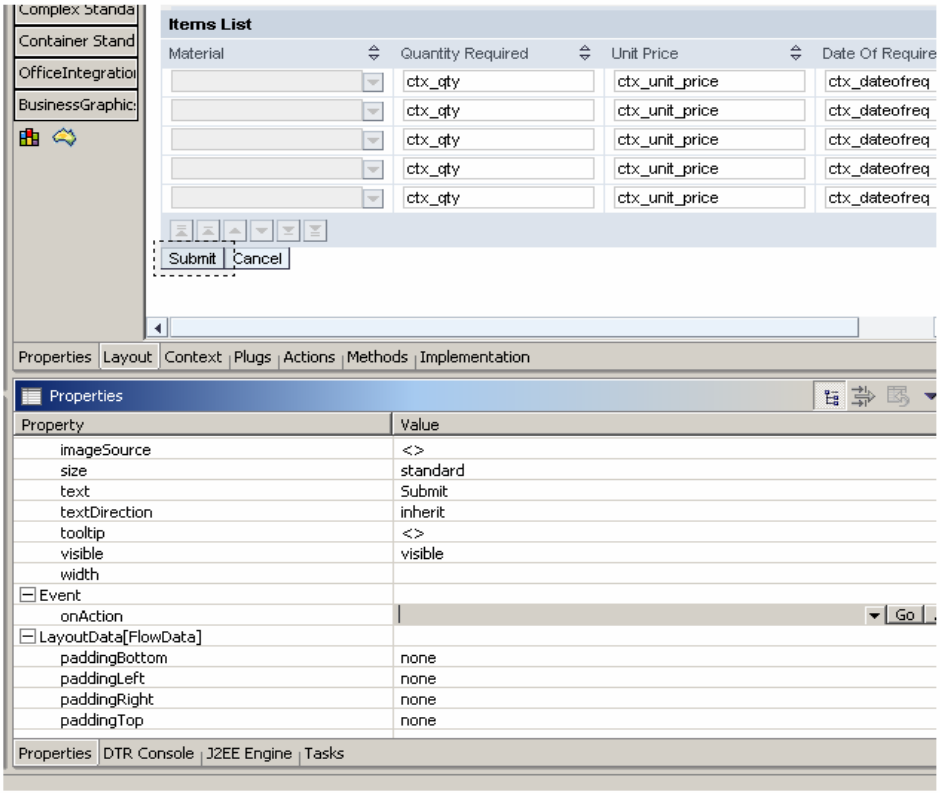
Creating Action for the Submit Button

Procedure

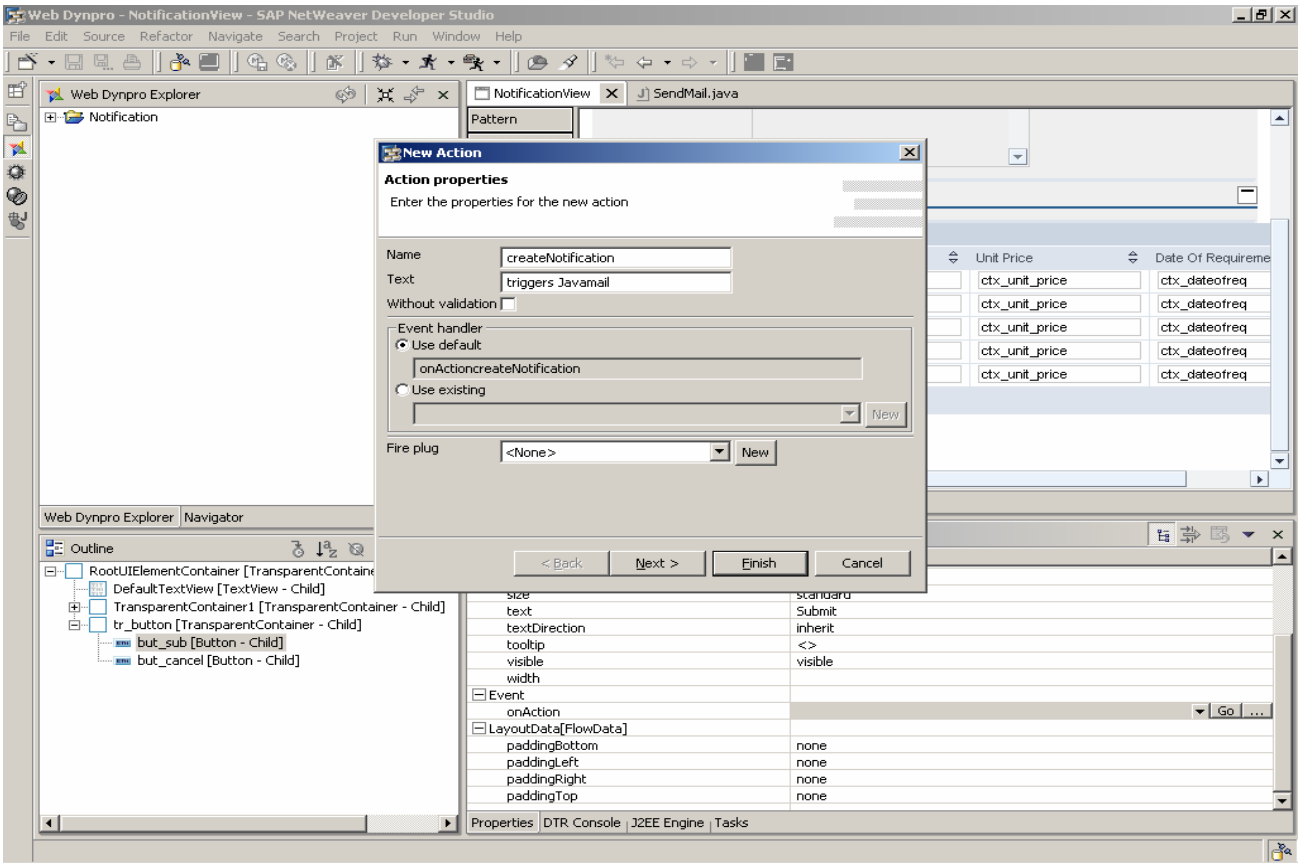
Creating the Submit Action

1. Open the *View Designer* for the NotificationView.

2. Open the properties of Submit button UI



3. Create a new action for this submit button as shown below.



Result

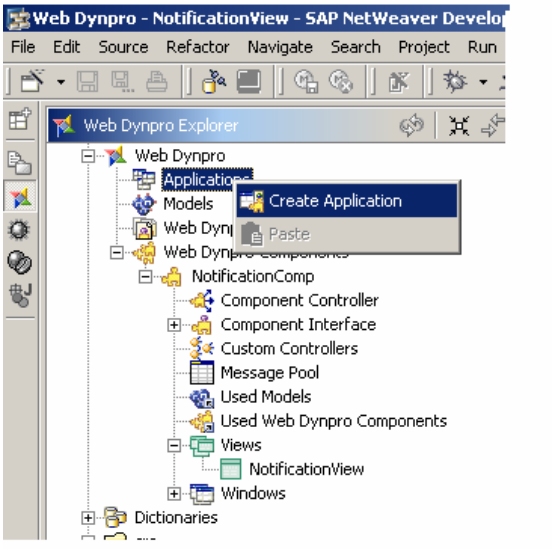
This will create onActioncreateNotification method in NotificationView.

Implementing the Action Event Handler

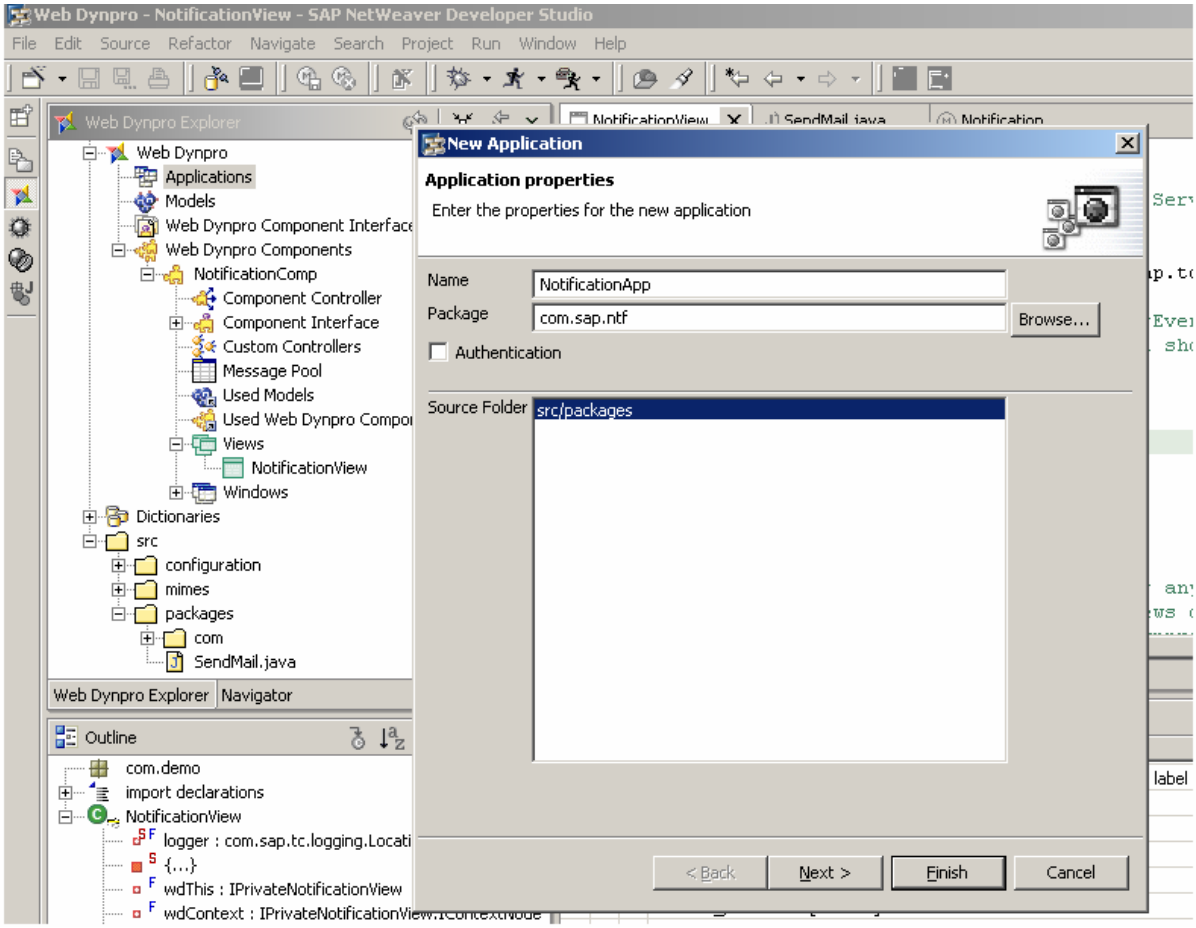
1. In the *View Designer,* choose the *Implementation* tab for the **NotificationView**. The Developer Studio runs several generation routines, and then displays the updated source code for the implementation of the view controller.
2. Insert the following line of code in the onActioncreateNotification method:

```
boolean sent=true;
SendMail snd=new SendMail();
sent=snd.sendMail("from","to");
   if(sent==true){
   msgMgr.reportSuccess("Your request has been sent for approval");
   }
   else
   {
   msgMgr.reportException ("Failed to create request",false);

   }
```

Creating the Web Dynpro Application



1. Enter the application name and package name.



2. Select default values for the remaining options and create the application

Result

This will create the Notification application.

Deploying and Running Application

Deploying the Project

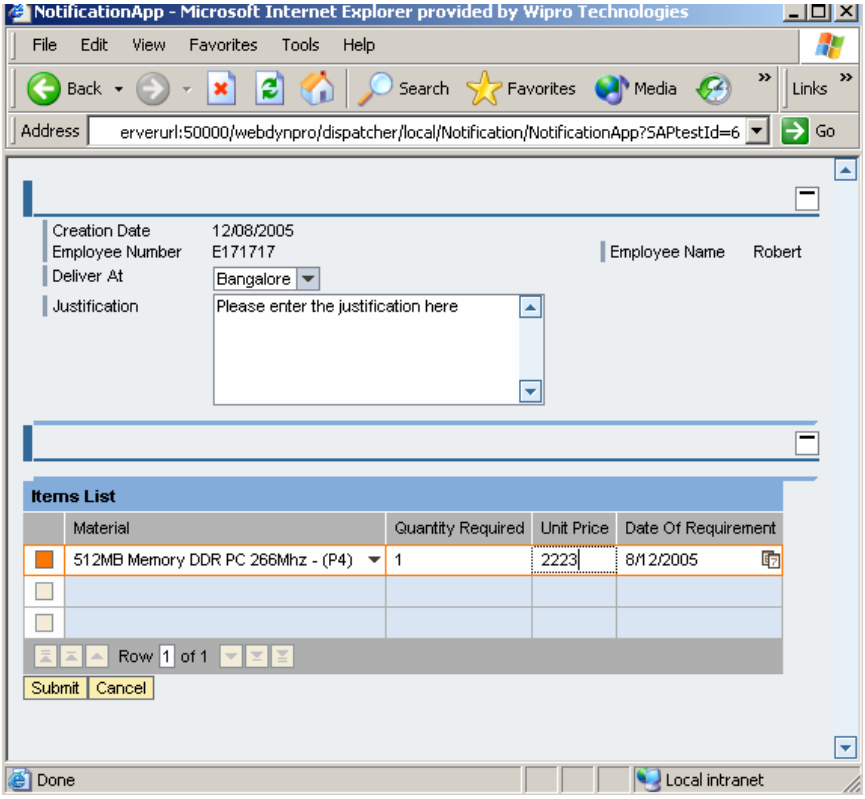
In the *Web Dynpro Explorer*, expand the project node **`Notification`** and select the application **NotificationApp**

Choose *Deploy New Archive and Run*.

Result

The Developer Studio performs the deployment process and then automatically launches the Notification application in the web browser.

Test the Web Dynpro application by clicking on the submit button. It will send the mail to the ID which is sent to the application.



Conclusion:

This tutorial explained step by step procedure to create notifications in Web Dynpro application by using Java Mail API. This considered a scenario in an organization where employee needs to raise a request through company intranet for getting an item which needs approval from his higher authorities. In such condition mail will be sent to the concerned person as notification, mentioning that a request has been created which needs approval.

Author Bio

Suresh K R has been working as an associate consultant for the SAP NetWeaver Competence Group (NWCG) at Wipro Technologies, Bangalore, India since October 2004. He has five years of experience in the IT industry and is an SAP-certified Web AS consultant. His areas of interest include cutting-edge technologies such as Web Application Servers, Web Dynpro integration, JDI, Enterprise Portal and J2EE with SAP NetWeaver Developer Studio.