Using JavaMail in Web Dynpro Java Applications for Sending e-mail to Multiple Users

Summary

This short tutorial describes about how to use Java Mail in Web Dynpro for sending e-mail to multiple users. It also describes how you can send a mail to multiple users in cc and bcc fields.

Author: Naresh Garg

Company: L&T Infotech

Created on: 11 August 2008

Author Bio

Naresh Garg has been working as Web Dynpro consultant in L&T Infotech Pvt. Ltd. for the last 9 months.

© 2008 SAP AG

Table of Contents

© 2008 SAP AG

Introduction

This short tutorial describes about how to use Java Mail in Web Dynpro for sending e-mails. It also demonstrates how to send the e-mail to the addresses given in the cc and bcc fields.

Prerequisites

Access to SAP NetWeaver AS Java and SAP NetWeaver Developer Studio installed in your system.

Before Using this Program, you need to have Javasoft's JavaMail class files which can be downloaded from here http://www.javasoft.com/products/javamail/index.html

Creating the Application

1. Create Web Dynpro Project

Open SAP NetWeaver Developer Studio. Click on New-> Web Dynpro Project. Name the project as "WD_MailTest".

2. Create Web Dynpro Application, Component, Window and View

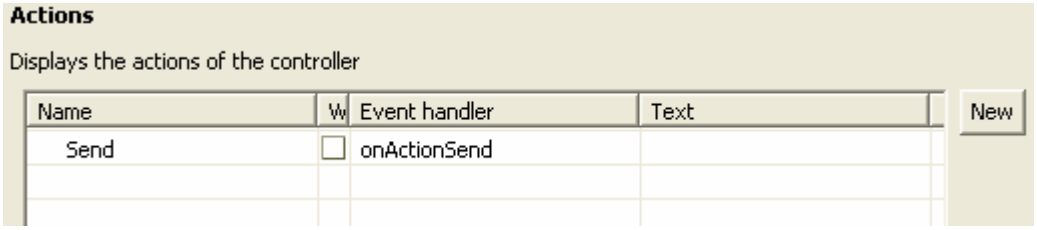
Open the created project. Right click on Application and click on "Create Application." Name the application as "MailTestApp" and specify the package details. Click on Next. Choose the option "Create a new Web Dynpro component" . Accept the default values.

3. Create Contexts and Methods in Component Controller

Open the Web Dynpro Component Controller "MailTestApp". Open the Context Tab. Create attributes of type String named "FromAddress", "ToAddress" and "TextMessage" as in the following picture.

All the context elements are of type "**string**".

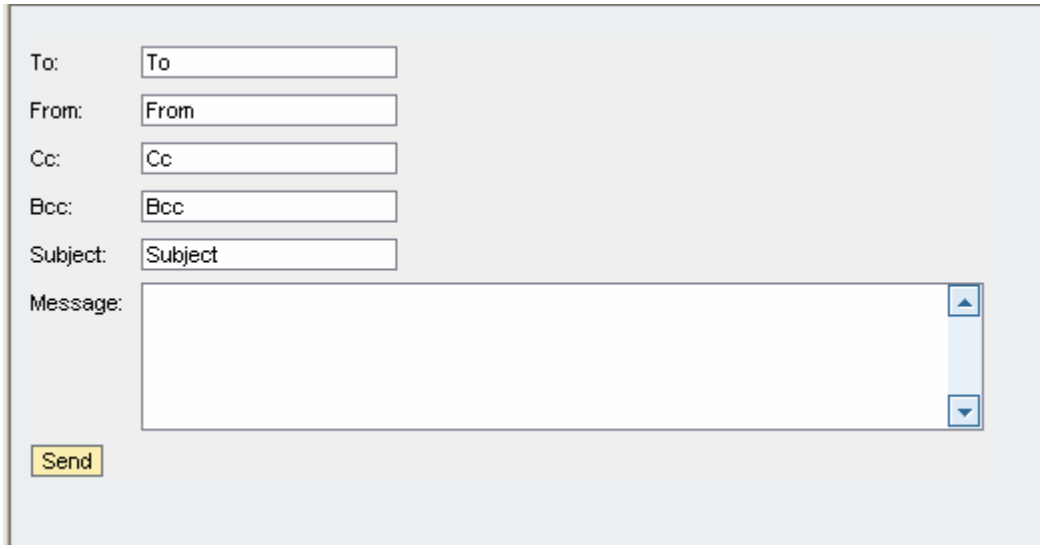
Create an action "Send". This will also generate a method "onActionSend"



© 2008 SAP AG

4. Creating Layout

Create the Layout as displayed



Here To, From, Cc, Bcc and Subject are of type input fields while the Message is of type TextEdit.

Map the input fields to respective context elements.

Now, create a button and set the property onAction to value “Send”.

5. Implementation

In the implementation tab go to the method **onActionSend** and paste the following code in it.

```
//@@begin onActionSend(ServerEvent)
          String host = " ";  //specify the host
          String from = wdContext.currentContextElement().getFrom();
          String to = wdContext.currentContextElement().getTo();
          String msg = wdContext.currentContextElement().getMessage();
          String subject = wdContext.currentContextElement().getSubject();
          String cc = wdContext.currentContextElement().getCc();
          String bcc = wdContext.currentContextElement().getBcc();
          // Get system properties
          Properties props = System.getProperties();
          // Setup mail server
          props.put("mail.smtp.host", host);
          // Get session
          Session session = Session.getDefaultInstance(props, null);
          // Define message
          MimeMessage message = new MimeMessage(session);
          //Sending mail to addresses in To field
          StringTokenizer st = new StringTokenizer(to, ",");
          while (st.hasMoreTokens()) {
                 String tempTo = st.nextToken();
                 try {
                        message.setFrom(new InternetAddress(from));
                        message.setRecipient(
                               Message.RecipientType.TO,
                               new InternetAddress(tempTo));
                        message.setSubject(subject);
```


© 2008 SAP AG

```
                message.setText(msg);
                Transport.send(message);
                wdComponentAPI.getMessageManager().reportSuccess(
                        "Mail sent" + tempTo);
            } catch (AddressException e) {
                e.printStackTrace();
                wdComponentAPI.getMessageManager().reportException(
                        "exception in to",
                        false);
            } catch (MessagingException e) {
                e.printStackTrace();
                wdComponentAPI.getMessageManager().reportException(
                        "exception in to Message",
                        false);
            } catch (Exception e) {
                wdComponentAPI.getMessageManager().reportException(
                        "exception in cc",
                        false);
            }
        }
        //     Sending mail to addresses in CC field
        StringTokenizer st1 = new StringTokenizer(cc, ",");
        while (st1.hasMoreTokens()) {
            String tempCc = st1.nextToken();
            try {
                message.setFrom(new InternetAddress(from));
                message.setRecipient(
                        Message.RecipientType.CC,
                        new InternetAddress(tempCc));
                message.setSubject(subject);
                message.setText(msg);
                Transport.send(message);
                wdComponentAPI.getMessageManager().reportSuccess(
                        "Mail sent to cc" + tempCc);
            } catch (AddressException e) {
                wdComponentAPI.getMessageManager().reportException(
                        "exception in cc Address",
                        false);
                e.printStackTrace();
            } catch (MessagingException e) {
                wdComponentAPI.getMessageManager().reportException(
                        "exception in cc Message",
                        false);
                e.printStackTrace();
            } catch (Exception e) {
                wdComponentAPI.getMessageManager().reportException(
                        "exception in cc",
                        false);
            }
        }
        //     Sending mail to addresses in BCC field
        StringTokenizer st2 = new StringTokenizer(bcc, ",");
        while (st2.hasMoreTokens()) {
            String tempBcc = st2.nextToken();
            try {
                message.setFrom(new InternetAddress(from));
```


© 2008 SAP AG

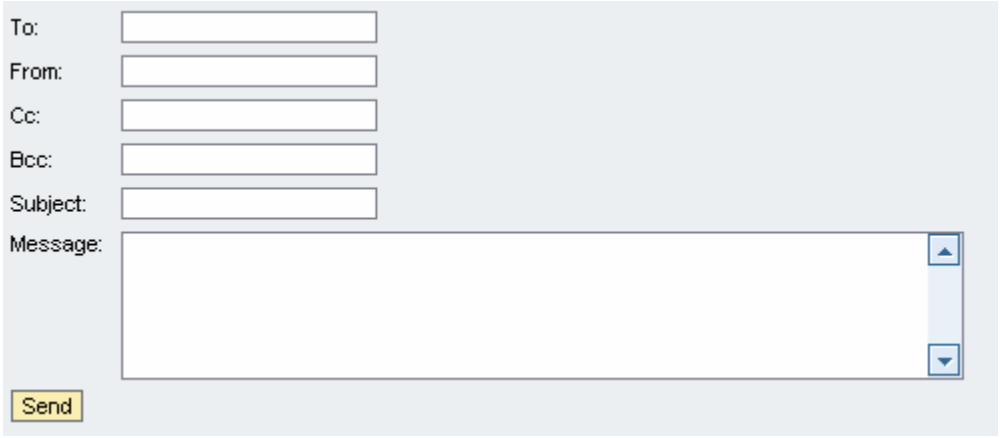
```
                        message.setRecipient(
                              Message.RecipientType.BCC,
                              new InternetAddress(tempBcc));
                        message.setSubject(subject);
                        message.setText(msg);
                        Transport.send(message);
                        wdComponentAPI.getMessageManager().reportSuccess(
                              "Mail sent to bcc" + tempBcc);
                  } catch (AddressException e) {
                        wdComponentAPI.getMessageManager().reportException(
                              "exception in bcc",
                              false);
                  } catch (MessagingException e) {
                        wdComponentAPI.getMessageManager().reportException(
                              "exception in bcc Message",
                              false);
                  } catch (Exception e) {
                        wdComponentAPI.getMessageManager().reportException(
                              "exception in cc",
                              false);
                  }
            }
      //@@end
```


© 2008 SAP AG

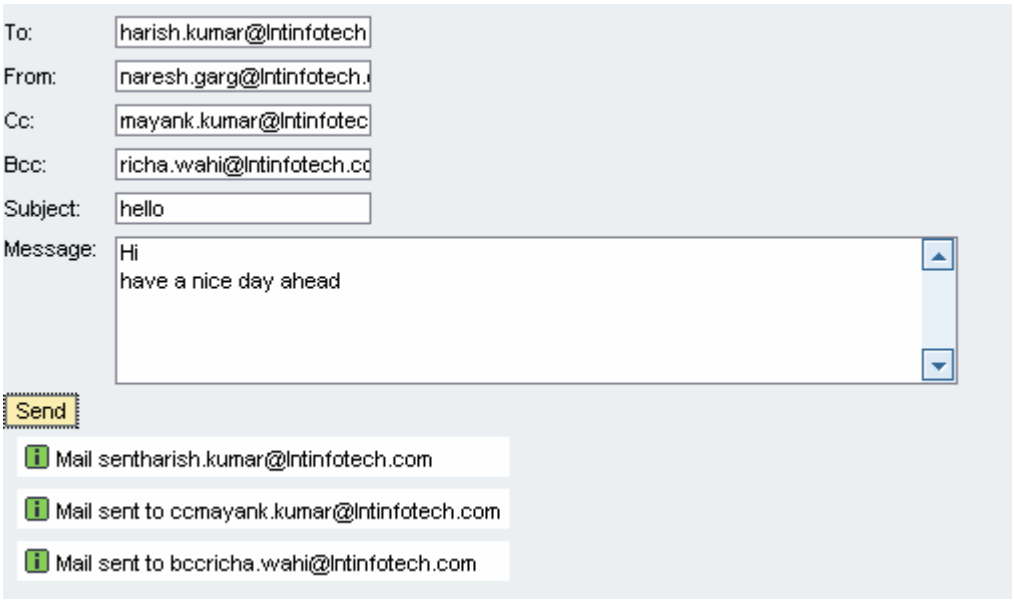
6. Result

Save the application with appropriate save metadata button on the tool bar.

Deploy and run the application, the resulting screen will be



After filling all the required fields, click on the send button, and the result will be :



© 2008 SAP AG

Related Content

Crerating First WebDynpro application

WebDynpro UI building blocks

UI development with WebDynpro Java

For more information, visit the User Interface Technology homepage.

© 2008 SAP AG

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.

© 2008 SAP AG