# Best Practice of Portal Development

**By Wang Hua Hong (wanghuah@cn.ibm.com)**
**Shanghai Globalization Lab (http://ut.cn.ibm.com/sgl)**
**User Technology (http://ut.cn.ibm.com/)**
**IBM China Software Development Lab**

# Table of Contents

# 1. Preface

## 1.1 Main contents

This paper is based on information from our previous portal projects and various IBM® publications. The purpose is to give you series best practices of portal development.

In this paper, you will know what's the portal firstly. Then the author will show you the framework and process of portal development. A lot of best practices will make your portal development easily. Also I'll show you how to support globalization in portal.

## 1.2 Audience

Portal developer, designer, administrator

## 1.3 About the author

Luke (Wang Hua Hong) is a Software Engineer at the IBM Shanghai Globalization Lab (SGL: http://ut.cn.ibm.com/sgl/) in Shanghai. Luke joined IBM in 2001 and worked on IBM globalization solutions for e-business systems in these years. He has rich experience in both portal and globalization. You can reach Luke at wanghuah@cn.ibm.com.

## 1.4 Change history

| Date | Version | Description | Author |
|---|---|---|---|
| 2005-12-13 | 1.0 | Final | Luke Wang |
| | | | |
| | | | |

# 2. Portal introduction

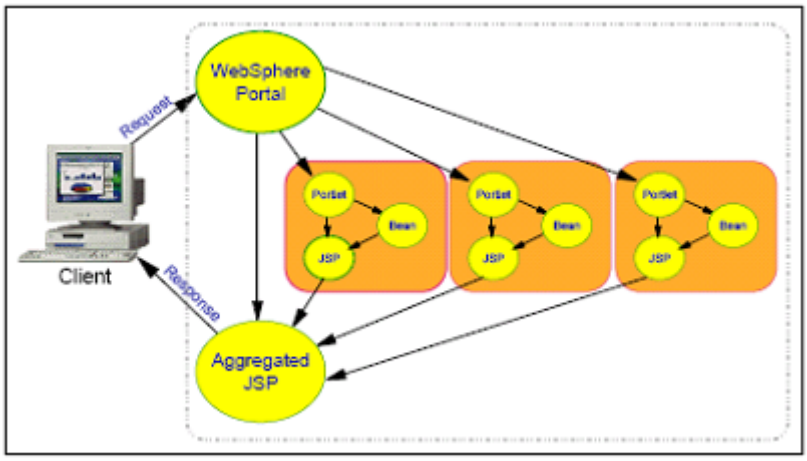
## 2.1 What is a portal?

Portals are the next-generation desktop, delivering e-business applications over the Web to all kinds of client devices. Portals provide site users with a single point of access to multiple types of information and applications.

**IBM WebSphere Portal** provides a single, secure, interactive point of access to dynamic applications, information, people and processes to help build successful Business-to-Business (B2B), Business to Employee (B2E) and Business to Consumer (B2C) portals.

## 2.2 MVC design pattern

**Model**: Encapsulating all the business logic
**View**: A JSP or even simple HTML.
**Controller:** A servlet, portlet, or a simple Java class




**Portlet MVC architecture**

The MVC architecture can be applied as a design pattern to any system needing to achieve separation of responsibilities. In fact, you will see as you continue through this paper that the Portal Server itself is architected this way. Furthermore, several of the benefits of the portlet architecture are available to you only if you employ a good MVC design.

**Model**

The Model in a portlet application is not necessarily different from the Model in any other Java server side application. The Model represents business logic and should not be concerned with the Controller or the View. The Controller could be a servlet, portlet, or a simple Java class. The View could be a JSP or even simple HTML. In theory, then, provided that existing applications employ solid MVC practices, porting the functionality WebSphere Portal should not require any changes to the logic. However, in practice, there are always applications that lack this foresight. The rich API covered later in this chapter will arm you with the tools to tackle this situation. Implementing a rigid commitment to the MVC architecture now will conserve an enormous amount of effort in later migrations or maintenance duties.

**View**

Like the servlet MVC implementation, the View is traditionally implemented using JSPs or simple HTML. However, because the HTML the View returns will be aggregated, it must not contain page-level tags and must be very mindful of the environment in which it is executing. Furthermore, the Portlet API provides tag libraries which aid in creating dynamic view resources for the portlet environment.

**Control**

The Controller is responsible for determining the requested mode, executing an appropriate Model and selecting the correct View. The portlet class itself acts as the Controller. Instead of determining the request method as in servlets, portlets need to determine the mode the user has requested. In a normal presentation, where a page is built with several portlets on it, the mode is View. The user, with appropriate permissions, may click the Edit button in order to perform some edit functionality. In this case, the mode is Edit.

### 2.3 Portal concepts


**Portal concepts**

The following are some definitions and descriptions of Portal concepts.

**Portlet**

A portlet is an application that displays page content.

**Portlet application**

Portlet applications are collections of related portlets and resources that are packaged together. All portlets packaged together share the same context that contains all resources such as images, properties files and classes.

**Page**

A portal page displays content. A page can contain one or more portlets. For example, a World Market page might contain two portlets that displays stock tickers for popular stock exchanges and a third portlet that displays the current exchange rates for world currencies. To view a page in the portal, you select its page.

**Layout**

The page layout defines the number of content areas within the page and the portlets displayed within each content area. In many cases, the portal administrator defines the page layout. The administrator can permit specified users or user groups to change the page layout to reflect individual preferences.

If you have authority to change a page, use the configure icon (wrench icon) to alter the page layout.

**Roles**

Each portal page is subdivided into one or more content areas. Each content area can contain one or more portlets. The portal administrator or a user who has authority to manage a page can control whether others who have authority to edit the page can move, edit or delete the content areas and the portlets on the page.

Portal V5.1 permission is role based. A role is a set of permissions. Roles can be assigned (or mapped) to individual principals granting those principals the corresponding permissions. If you have authority to make changes to a portal page, use the Resource Permissions page in Access under Administration to set the permissions for the page. By default, there are seven roles and they are as follows:

**Themes**

Themes represent the overall look and feel of the portal, including colors, images and fonts. There are several default themes provided with the standard installation of WebSphere Portal. Each page in the portal may have a different theme associated with it, thereby creating the appearance of virtual portals. Use the Themes and Skins under Portal User Interface to manage themes.

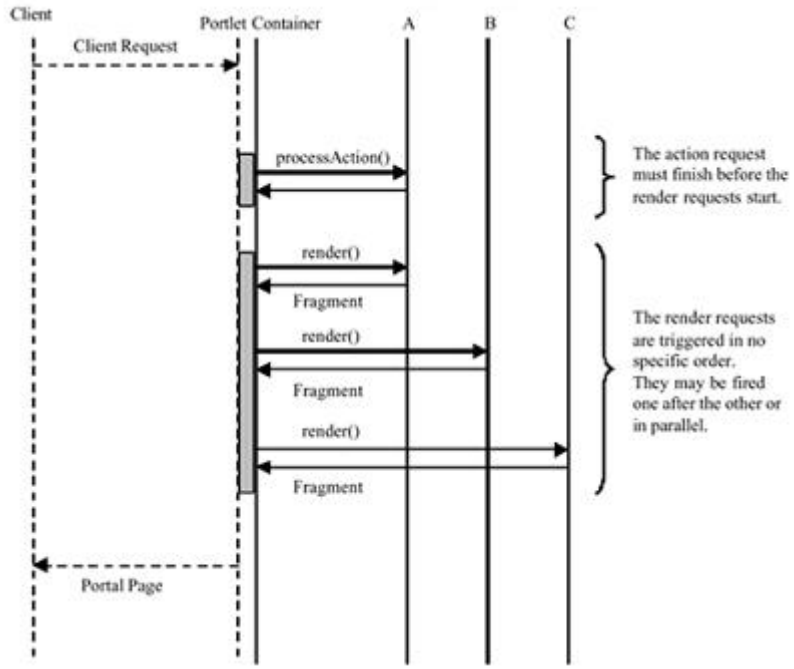
**Skins**

The term skin refers to the visual appearance of the area surrounding an individual portlet. Each portlet can have its own skin. The skins that are available for use with a portlet are defined by the portal theme that is associated with the page. The portal administrator or the designer determines the theme for pages and the available skins for the theme. The administrator can permit specified users to change the skins to reflect individual preferences. If you have authority to make changes to a portal page, use the Themes and Skins under Portal User Interface to manage themes.

### 2.4 Portlet lifecycle

The basic portlet lifecycle is to:

**Initialize**, using the init class to initialize the portlet and put it into service.
**Handle requests**, processing different kinds of actions and rendering content.
**Complete**, using the destroy class to take the portlet out of




### 2.5 Portal project types

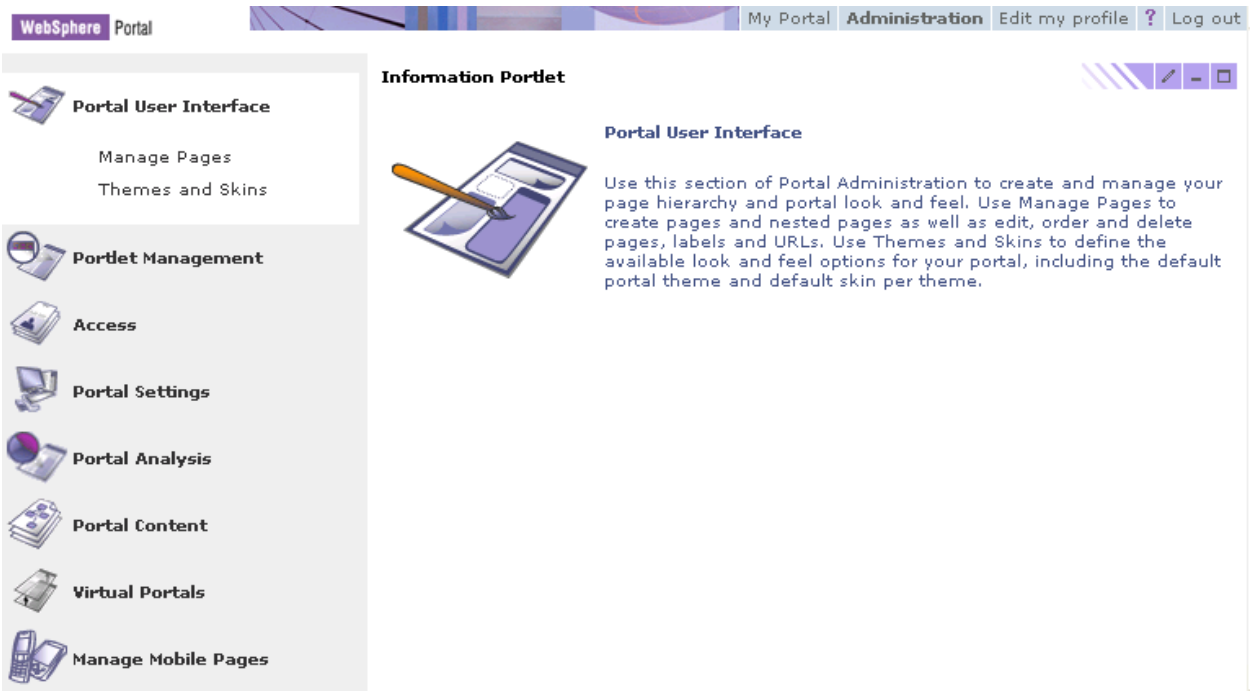
There are several types of portal project, which including IBM API portlet, JSR168 portlet, JSF portlet, XDIME portlet and etc.

For details, please refer to the section *Comparison of several types of portlet projects*

### 2.6 Portal administration

After you installed WPS environment, you can access portal by http://hostname/wps/portal.
You can login using the default user for WPS.
Username: **wpsadmin**
Password: **wpsadmin**

You will see the following page by clicking the link "**Administration**" in the top-right of portal page.




For details about how to do the administration, please refer to the section *Portal Administration*

# 3. The framework of portal project

## 3.1 Framework overview

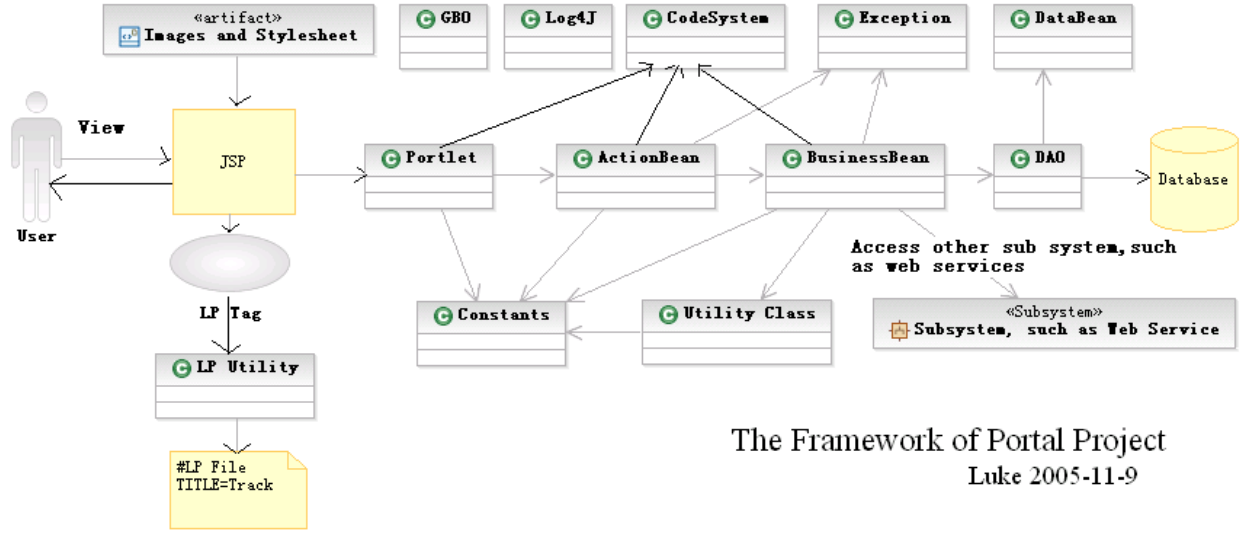
Here is the framework of portal project. It implements the MVC design pattern.

**Model**: Encapsulating all the business logic. It can be Action Bean, Business Bean, DAO, and etc.

**View**: A JSP or even simple HTML.

**Controller:** A servlet, portlet, or a simple Java class.

The main flow is User -> JSP –> Portlet –> Action Bean –> Business Bean –> DAO (DataBean) –> Database




## 3.2 Concepts in this framework

### 3.2.1 User

The user who are browsing the portal website.

### 3.2.2 JSP

The JSP page will be shown to the user. It's the **View** in MVC design pattern. Each JSP has a corresponding portlet or actionbean (in JSF portlet).

Note: In JSF portlet, the name of JSP is consistent to ActionBean, that is, for Search.jsp, the ActionBean should be Search.java.

**JSR168 Portlet – JSP**

```
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8" session="true"%>
<%@ taglib uri="http://java.sun.com/portlet" prefix="portlet" %>
<%@ page import="java.util.*,javax.portlet.*,com.ibm.sgl.oged.wemp.portlet.*" %>
```

```
<%@ page import="com.ibm.sgl.oged.constant.TrackConstansts"%>
<%@ page import="com.ibm.sgl.oged.db.databean.ShipmentBean"%>
<%@ page import="com.ibm.sgl.oged.db.databean.UserBean"%>
<%@ page import="com.ibm.sgl.oged.db.manager.UserBeanManager"%>
<%@ taglib uri="/WEB-INF/tld/lp.tld" prefix="lp" %>
<%@ taglib uri="/WEB-INF/tld/lp.tld" prefix="lp_alt" %>
<lp_alt:load file="image.properties"/>
<lp:load file="tracking.properties"/>
<%@include file="Bidi.jsp"%>


<%

String locale= com.ibm.sgl.oged.util.LPUtil.getFullLocale(request.getLocale().toString());


 %>

<link href="/wps/oged/images/<%=locale%>/Styles.css" rel="stylesheet" type="text/css">
<portlet:defineObjects />

<FORM name="hawbform" method="POST" action="<portlet:actionURL/>">
<table      width="100%" border="0" cellspacing="0" cellpadding="0">

   <tr>
      <td width="100%" height="170" align="center" valign="top"><br>
         <br>
         <table    width="90%" border="0" cellpadding="0" cellspacing="0">
            <tr>
               <td                colspan="3"                class="td5<%=dir%>"                ><lp:value
key="PORTLET_TRACKSHIPMENTSTATUS" /></td>
            </tr>
         </table>
<%
    String
errormsg=(String)renderRequest.getPortletSession().getAttribute(TrackConstansts.ERROR_MSG)
;
    if (null==errormsg || errormsg.length()==0)
{

}//end for if
```

```
else {

 %>

  <table  width="90%" border="0" cellpadding="0" cellspacing="0" >
<tr>
        <td              height="20"              class="td4<%=dir%>"              width="20"><img
src="/wps/oged/images/error-message.gif" alt="<lp_alt:value key="ERRORMSG"/>"></td>
                <td height="20"  class="td4<%=dir%>" width="90%">
                <lp:value key="<%=errormsg%>"/>
                </td>
</tr>
  </table>

<%
errormsg="";
renderRequest.getPortletSession().setAttribute(TrackConstansts.ERROR_MSG,errormsg);
}

 String
myhawb=(String)renderRequest.getPortletSession().getAttribute(TrackConstansts.TRACK_BY_H
AWB_FIELD);
    if (null==myhawb)
{
myhawb="";
}

renderRequest.getPortletSession().removeAttribute(TrackConstansts.TRACK_BY_HAWB_FIEL
D);


 %>
      <table  width="90%" border="0" cellpadding="0" cellspacing="0" >
  <tr>
            <td          width="50%"          align="center"          class="td3<%=dir%>"><font
color="#000000">HAWB:</font></td>
            <td width="50%" class="td6<%=dir%>"><font color="#000000">
              <input       name="<%=TrackConstansts.TRACK_BY_HAWB_FIELD       %>"
class=wpsFieldText  size=10 maxlength="8" value="<%=myhawb %>" >
              </font></td>
          </tr>
          <tr>
            <td colspan="3" class="td3<%=dir%>"><div align="center">
```

```
                    <input
name="<%=TrackConstansts.TRACK_BY_HAWB_SUBMIT_BUTTON%>"      type="submit"
class="button" id="track2" value="<lp:value key="BUTTON_TRACK" />" >

                </div></td>
            </tr>
          </table>
          <br>
        </td>
    </tr>

</table>
</FORM>
```

### 3.2.3 Portlet

A portlet is an application that displays page content. It's the **Controller** in MVC design pattern. It can decide to invoke which jsp according to the result from business process.
There are several types of portal project, which including IBM API portlet, JSR168 portlet, JSF portlet, XDIME portlet and etc.

**JSR168 Portlet – Portlet**

```
/*
 * Author Peter
 * Class: <TrackByHAWBBrieflyPortlet>
 *    Description: <This class is to handle the contron of the logic of UC TrackByHAWBBriefly .>
 * Version: <1.0>
 * Author: <Peter>
 * Creation date: <2005-8-9>
 * Department: Shanghai Globalization Laboratory
 * Copyright (c) 2005, International Business Machines Corporation, All rights reserved.
 */
package com.ibm.sgl.oged.wemp.portlet;

import java.io.IOException;

import javax.portlet.ActionRequest;
import javax.portlet.ActionResponse;
import javax.portlet.GenericPortlet;
import javax.portlet.PortletException;
import javax.portlet.PortletRequestDispatcher;
```

```
import javax.portlet.RenderRequest;
import javax.portlet.RenderResponse;
import com.ibm.sgl.oged.constant.TrackConstansts;
import org.apache.log4j.*;
import com.ibm.sgl.oged.wemp.actionbean.TrackByHAWBActionBean;
/**
 *
 * @author panxf
 *@version 1.0
 * TODO To change the template for this generated type comment go to
 * Window - Preferences - Java - Code Style - Code Templates
 */
public class TrackByHAWBBrieflyPortlet extends GenericPortlet {
/**
 * action bean
 */
    private TrackByHAWBActionBean trackByHAWBActBean = new
TrackByHAWBActionBean();

 /**
  * jsp name
  */
    private String JSP_NAME = null;
/**
 * logger
 */
    private static Logger logger = Logger.getLogger(TrackByHAWBPortlet.class);

    /**
     * Serve up the <code>view</code> mode.
     * @param request
     * @param response
     * @throws PortletException
     * @throws IOException
     * @see javax.portlet.GenericPortlet#doView(javax.portlet.RenderRequest,
     *      javax.portlet.RenderResponse)
     */
    public void doView(RenderRequest request, RenderResponse response)
            throws PortletException, IOException {
        // Set the MIME type for the render response
        logger.debug("#pan#    Track doView begin"); //$NON-NLS-1$

        // Set the MIME type for the render response
```

```
        response.setContentType(request.getResponseContentType());

        if (null == JSP_NAME) {
            JSP_NAME = TrackConstansts.TRACK_BY_HAWB_BRIEF;


    request.getPortletSession().removeAttribute(TrackConstansts.TRACK_BY_HAWB_FIELD);

            // Invoke the JSP to render
        }
        PortletRequestDispatcher rd = getPortletContext().getRequestDispatcher(
                getJspFilePath(request, JSP_NAME));
        rd.include(request, response);
        JSP_NAME = null;
        logger.debug("#pan# *Track doView over"); //$NON-NLS-1$

    }

    /**
     * Returns JSP file path.
     *
     * @param request
     *                  Render request
     * @param jspFile
     *                  JSP file name
     * @return JSP file path
     */
    private static String getJspFilePath(RenderRequest request, String jspFile) {
        logger.debug("#pan#    Track getJspFilePath begin"); //$NON-NLS-1$
        //$NON-NLS-1$
        String markup = request.getProperty(TrackConstansts.WPS_MARKUP);
        if ( null == markup) {
            markup = getMarkup(request.getResponseContentType());
        }
        return "/" + markup + "/" + jspFile + "." + getJspExtension(markup); //$NON-NLS-1$
//$NON-NLS-2$       //$NON-NLS-3$

    }

    /**
     * Convert MIME type to markup name.
     *
     * @param contentType
```

```
     *                   MIME type
     * @return
     */
    private static String getMarkup(String contentType) {
        logger.debug("#pan#    Track getMarkup begin"); //$NON-NLS-1$

        if (TrackConstansts.XDIME_CONTENT_TYPE.equals(contentType)) {
            return TrackConstansts.XDIME;
        }
        return TrackConstansts.HTML;
    }
    /**
     *
     * @param requet
     * @param response
     * @throws PortletException
     * @throws IOException
     * @see javax.portlet.GenericPortlet#doView(javax.portlet.RenderRequest,
     *          javax.portlet.RenderResponse)
     */
    public void processAction(ActionRequest request, ActionResponse response)
            throws PortletException, java.io.IOException {
        //PortletPreferences prenf = request.getPreferences();
        logger.debug("#pan#    Track in process action."); //$NON-NLS-1$



        if (null !=
request.getParameter(TrackConstansts.TRACK_BY_HAWB_SUBMIT_BUTTON) ) {
            logger.debug("#pan#    click TrackbyHAWB button"); //$NON-NLS-1$

            logger.debug("#pan#    do the trackbyhawb work"); //$NON-NLS-1$

            //get the hawb value
            String hawb = request
                    .getParameter(TrackConstansts.TRACK_BY_HAWB_FIELD);

            // Set the session of action bean

            trackByHAWBActBean.setSession(request.getPortletSession());

request.getPortletSession().setAttribute(TrackConstansts.TRACK_BY_HAWB_FIELD,hawb);
```

```
                if (trackByHAWBActBean.doSubmit(hawb)){    //enter the jsp
                    logger.debug("#pan# portlet doSubmit is ok"); //$NON-NLS-1$
                    JSP_NAME = TrackConstansts.TRACK_BY_HAWB_BRIEF_RESULT;

                } else {
                    logger.debug("#pan# portlet    doSubmit is not ok"); //$NON-NLS-1$
                    JSP_NAME = TrackConstansts.TRACK_BY_HAWB_BRIEF;

                }

            }

            if                                          (null                                   !=
request.getParameter(TrackConstansts.TRACK_BY_HAWB_BACK_BUTTON)) {
                JSP_NAME = TrackConstansts.TRACK_BY_HAWB_BRIEF;

            }

        }

        /**
         * Returns the file extension for the JSP file
         *
         * @param markupName
         *                  Markup name
         * @return JSP extension
         */
        private static String getJspExtension(String markupName) {
            logger.debug("#pan#    Track getJspExtension"); //$NON-NLS-1$

            return TrackConstansts.JSP;
        }
        /**
         * @return
         */
public String toString(){
        return TrackByHAWBBrieflyPortlet.class.toString();
}
}
```

### 3.2.4 ActionBean

The purpose of ActionBean is to handle the inputs in the jsp by user and invoke the methods in BusinessBean. For example, do the validation for user input. If the input is invalid or exception from BusinessBean is throwed, it will return false and set the error message to session.

**JSR168 Portlet – ActionBean**

```
package com.ibm.sgl.oged.wemp.actionbean;

import com.ibm.sgl.oged.wemp.businessbean.TrackByHAWBBrieflyBusinessBean;

import com.ibm.sgl.oged.db.databean.ShipmentBean;
import com.ibm.sgl.oged.exception.TrackException;

import javax.portlet.PortletSession;
import com.ibm.sgl.oged.constant.TrackConstansts;
import org.apache.log4j.*;

public class TrackByHAWBActionBean {
	/**
	 *  session passed from portlet
	 */
	private PortletSession mysession = null;
/**
 * logger
 */
	private static Logger logger = Logger
			.getLogger(TrackByHAWBActionBean.class);

	/**
	 * @return Returns the session.
	 */
   final  public PortletSession getSession() {
		return mysession;
	}

	/**
	 * @param session
	 *            The session to set.
	 */
    final public void setSession(PortletSession mysession) {
		this.mysession = mysession;
	}

	public boolean doSubmit(String hawb) {
```

```
        boolean mark = true;
        logger.debug("#pan# *doSubmit");//$NON-NLS-1$

        logger.debug("#pan# *doSubmit to check :::" );//$NON-NLS-1$
        logger.debug(hawb);

        if (inputValidate(hawb)) {

            logger.debug("#pan#*doSubmit is true"); //$NON-NLS-1$

            ShipmentBean sb = null;
            TrackByHAWBBrieflyBusinessBean trackBizbean = new
TrackByHAWBBrieflyBusinessBean();

            try {
                sb = trackBizbean.getShipmentDetailByHAWB(hawb);
                mysession.setAttribute(TrackConstansts.SHIPMENT_INFO, sb);

            } catch (TrackException e) {

                String errString = e.getErrorCode();
                mysession.setAttribute(TrackConstansts.ERROR_MSG, errString);
                mark = false;

            }

        } else {
            logger.debug("#pan# doSubmit is false"); //$NON-NLS-1$

            mark = false;
        }
        return mark;

    }

    public boolean inputValidate(String hawb) {

        if (null == hawb || 0 == hawb.length()|| 0 == hawb.trim().length()) {
            getSession().setAttribute(TrackConstansts.ERROR_MSG,
                    TrackConstansts.EM_HAWBEMPTY);
            logger.debug("hawb is null"); //$NON-NLS-1$
```

```
                return false;
            }

            try {
                Integer.parseInt(hawb);
            } catch (NumberFormatException e) {

                getSession().setAttribute(TrackConstansts.ERROR_MSG,
                        TrackConstansts.EM_HAWBINVALID);
                logger.debug("hawb is not numberic"); //$NON-NLS-1$

                return false;

            }
            if (8 != hawb.length()) {
                getSession().setAttribute(TrackConstansts.ERROR_MSG,
                        TrackConstansts.EM_HAWBLEN);
                logger.debug("hawb.length !=    8"); //$NON-NLS-1$

                return false;
            }

            return true;

        }

}
```

### 3.2.5 BusinessBean

The purpose of BusinessBean is to handle the business operations. For example, access database through DAO, or invoke other subsystem through web services. BusinessBean is called by ActionBean and return the result to ActionBean. If error occurs, it will throw exceptions to ActionBean. Each portlet only has one BusinessBean, and all the ActionBeans for this portlet share this BusinessBean.

**JSR168 Portlet – BusinessBean**

```
/*
 * Author: Peter Class: <TrackByHAWBBrieflyBusinessBean>
 *    Description: This class is to handle the business logic of UC TrackByHAWB and
TrackByHAWBBriefly
 * Version: <1.0>
 * Creation date: <2005-8-9>
 * Department: Shanghai Globalization
```

```
 * Laboratory Copyright (c)2005, International Business Machines Corporation,
 * All rights reserved.
 */
package com.ibm.sgl.oged.wemp.businessbean;

import com.ibm.sgl.oged.constant.TrackConstansts;
import com.ibm.sgl.oged.db.manager.ShipmentBeanManager;
import com.ibm.sgl.oged.db.databean.ShipmentBean;
import com.ibm.sgl.oged.exception.TrackException;
import com.ibm.sgl.oged.util.DBException;

import org.apache.log4j.*;

/**
 * @value 1.0
 * @author panxf
 *
 * TODO To change the template for this generated type comment go to
 * Window - Preferences - Java - Code Style - Code Templates
 */
public class TrackByHAWBBrieflyBusinessBean {
	/**
	 * Comment for <code>shipmentbeanmanager</code>
	 *
	 * @generated "UML to Java
	 *                  (com.ibm.xtools.transform.uml2.java.internal.UML2JavaTransform)"
	 */
	private ShipmentBeanManager shipmentbeanmanager = new ShipmentBeanManager();

	/**
	 * logger
	 */
	private static Logger logger = Logger
			.getLogger(TrackByHAWBBrieflyBusinessBean.class);

	/**
	 * @param hawb
	 * @return
	 * @throws TrackException
	 * @generated "UML to Java
	 *                  (com.ibm.xtools.transform.uml2.java.internal.UML2JavaTransform)"
	 */
	public ShipmentBean getShipmentDetailByHAWB(String hawb) throws TrackException {
```

```
            ShipmentBean sb = new ShipmentBean();
            logger.debug("#Peter# start    to get the shipment info"); //$NON-NLS-1$
            try {
                  sb = shipmentbeanmanager.getShipmentByHAWB(hawb);
            } catch (DBException e) {

                  String errMsg = e.getErrorCode();
                  if (TrackConstansts.EM_GENERAL.equals(errMsg)) {
                        throw new TrackException(errMsg);
                  } else {
                        logger.debug(e);
                  }
      }
            if (null == sb) {
                  logger.debug("#Peter# buz bean is    null "); //$NON-NLS-1$
            }
            logger.debug("#Peter# buz bean    pan get the shipment info"); //$NON-NLS-1$
            return sb;
      }
      /**
        * The string representation is CLASS NAME
        * Where NAME is the company name and PRODUCT is the product name.
        * @return
        */
      public String toString() {
            return TrackByHAWBBrieflyBusinessBean.class.toString();
      }

}
```

3.2.6 DAO

The purpose of DAO is to handle the database related operations. For example, insert, query, update, delete and etc.

**Code sample**

```
/*
 * Class: < G11NpreferenceBeanManager >
 * Description:< This class handles all G11NPREFERENCE related operation.>
 * Version: <1.0>
 * Author: < Lori >
 * Creation date: <2005-8-9>
 * Department: Shanghai Globalization Laboratory
 * Copyright (c) 2005, International Business Machines Corporation,
```

```
 * All rights reserved.
 */
package com.ibm.sgl.oged.db.manager;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;

import org.apache.log4j.Logger;

import com.ibm.sgl.oged.db.connector.JDBCConnectionProvider;
import com.ibm.sgl.oged.db.databean.G11NPreferenceBean;
import com.ibm.sgl.oged.db.databean.ShipmentBean;
import com.ibm.sgl.oged.util.CodeSystem;
import com.ibm.sgl.oged.util.DBException;

public class G11NPreferenceBeanManager {

	private G11NPreferenceBean g11nPreferenceBean = null;
	private Logger logger = Logger.getLogger(G11NPreferenceBean.class.getName());
	private JDBCConnectionProvider _jdbcConnectionProvider = null;

	public G11NPreferenceBeanManager() {
		_jdbcConnectionProvider = new JDBCConnectionProvider();
		_jdbcConnectionProvider.setDBName(DBConstants.OGEDDBName);
		_jdbcConnectionProvider.start();
	}

	/**
	 * insert one g11npreferencebean to table "insert into G11NPREFERENCE
	 * values(?,?,?,?,?,?,?,?,?,?)"
	 *
	 * @param g11nPreferenceBean
	 *                    g11nPreferenceBean to be inserted into table
	 * @return true success, false fail
	 * @throws DBException
	 */
	public boolean insertG11NPreference(G11NPreferenceBean g11nPreferencebean)
			throws DBException {
		if (null == g11nPreferencebean) {
			logger.debug("null bean");//$NON-NLS-1$		}
```

```
boolean returnflag = false;
Connection conn = null;
PreparedStatement ps = null;
StringBuffer strSql = null;
strSql = new StringBuffer(DBConstants.InitSize);
strSql = strSql.append(DBConstants.insertG11NPreference);
conn = _jdbcConnectionProvider.getConnection();
try {
    ps = conn.prepareStatement(strSql.toString());
    ps.setString(1, g11nPreferencebean.getUserid());
    ps.setString(2, g11nPreferencebean.getLanguage());
    ps.setString(3, g11nPreferencebean.getCurrency());
    ps.setString(4, g11nPreferencebean.getDatetime());
    ps.setString(5, g11nPreferencebean.getCalendar());
    ps.setString(6, g11nPreferencebean.getWeight());
    ps.setString(7, g11nPreferencebean.getLength());
    ps.setString(8, g11nPreferencebean.getField1());
    ps.setString(9, g11nPreferencebean.getField2());
    ps.setString(10, g11nPreferencebean.getField3());
    // see if executeUpdate success
    if (0 != ps.executeUpdate()) {
        returnflag = true;
    }

}//end of try
catch (SQLException e) {
    logger.debug("Exception in db access, input may not be valid");//$NON-NLS-1$
    //    e.printStackTrace();
    DBException dbe = new DBException();
    dbe.setErrorCode(DBException.KEYEXSIT);
    throw dbe;
}//end of catch
finally {
    try {
        ps.close();
    }//end of try ps.close
    catch (SQLException e) {
        logger.debug("close preparedstatement failed");//$NON-NLS-1$
        //        e.printStackTrace();

    }//end of catch
    try {
        conn.close();
```

```
            }//end of try conn.close
            catch (SQLException e) {
                logger.debug("close connection failed");//$NON-NLS-1$
                //        e.printStackTrace();

            } //end of catch

        }//end of finally
        return returnflag;
    }

    /**
     * get G11NPreference from table by userid "select * from G11NPREFERENCE
     * where userid=?"
     *
     * @param userid
     * @return
     * @throws DBException
     */
    public G11NPreferenceBean getG11NPreference(String userid)
            throws DBException {
        if (null == userid) {
            return null;
        }
        g11nPreferenceBean = null;
        Connection conn = null;
        PreparedStatement ps = null;
        StringBuffer strSql = null;
        ResultSet rs = null;
        strSql = new StringBuffer(DBConstants.InitSize);
        strSql = strSql.append(DBConstants.getG11NPreference);
        conn = _jdbcConnectionProvider.getConnection();
        try {
            ps = conn.prepareStatement(strSql.toString());
            ps.setString(1, userid);

            rs = ps.executeQuery();
            if (rs.next()) {
                logger.debug("get one record");//$NON-NLS-1$
                g11nPreferenceBean = new G11NPreferenceBean(rs.getString(1), rs
                        .getString(2), rs.getString(3), rs.getString(4), rs
                        .getString(5), rs.getString(6), rs.getString(7), rs
                        .getString(8), rs.getString(9), rs.getString(10));
```

```
            }//end of if(rs.next)
            else {
                logger.debug("no result match hawb");//$NON-NLS-1$
                g11nPreferenceBean = null;
            }//end of else
        }//end of try
        catch (SQLException e) {
            logger.debug("Exception in db access");//$NON-NLS-1$
            //    e.printStackTrace();
            DBException dbe = new DBException();
            dbe.setErrorCode(DBException.UNKOWNPROBLEM);
            throw dbe;

        }//end of catch
        finally {
            try {
                rs.close();
                ps.close();
            }//end of try ps.close
            catch (SQLException e) {
                logger.debug("close preparedstement failed");//$NON-NLS-1$
                //        e.printStackTrace();

            }//end of catch
            try {
                conn.close();
            }//end of try conn.close
            catch (SQLException e) {
                logger.debug("Close connection failed");//$NON-NLS-1$
                //        e.printStackTrace();
            } //end of catch
        }//end of finally
        return g11nPreferenceBean;
    }

    /**
     * updateG11NPreference by G11NPreferenceBean "update G11NPREFERENCE set
     * language=?,currency=?,datetime=?,calendar=?,weight=?,length=?
     * ,field1=?,field2=?,field3=? where userid=?"
     *
     * @param g11nPreferenceBean
     *                  G11NPreferenceBean to be updated
```

```
 * @return true success,false fail
 * @throws DBException
 */
public boolean updateG11NPreference(G11NPreferenceBean g11nPreferencebean)
        throws DBException {
    if (null == g11nPreferenceBean) {
        return false;
    }

    boolean returnflag = false;
    Connection conn = null;
    PreparedStatement ps = null;
    StringBuffer strSql = null;

    strSql = new StringBuffer(DBConstants.InitSize);
    strSql = strSql.append(DBConstants.updateG11NPreference);
    conn = _jdbcConnectionProvider.getConnection();

    try {
        ps = conn.prepareStatement(strSql.toString());
        ps.setString(1, g11nPreferencebean.getLanguage());
        ps.setString(2, g11nPreferencebean.getCurrency());
        ps.setString(3, g11nPreferencebean.getDatetime());
        ps.setString(4, g11nPreferencebean.getCalendar());
        ps.setString(5, g11nPreferencebean.getWeight());
        ps.setString(6, g11nPreferencebean.getLength());
        ps.setString(7, g11nPreferencebean.getField1());
        ps.setString(8, g11nPreferencebean.getField2());
        ps.setString(9, g11nPreferencebean.getField3());
        ps.setString(10, g11nPreferencebean.getUserid());

        if (0 != ps.executeUpdate()) {
            returnflag = true;
        } else {
            DBException dbe = new DBException();
            dbe.setErrorCode(DBException.NOROWUPDATED);
            throw dbe;
        }
    }//end of try
    catch (SQLException e) {
        logger.debug("Exception in db access");//$NON-NLS-1$
        // e.printStackTrace();
        DBException dbe = new DBException();
```

```
                dbe.setErrorCode(DBException.KEYEXSIT);
                throw dbe;
            }//end of catch
            finally {
                try {
                    ps.close();
                }//end of try ps.close
                catch (SQLException e) {
                    logger.debug("close preparedstatement failed");//$NON-NLS-1$

                }//end of catch
                try {
                    conn.close();
                }//end of try conn.close
                catch (SQLException e) {
                    logger.debug("close preparedstatement failed");//$NON-NLS-1$
                } //end of catch
            }//end of finally
            return returnflag;
        }
}
```

### 3.2.7 DataBean

Commonly DataBean is related to a table in database, you can also define a DataBean as data transferring between classes. DataBean is a set of data, which will be used everywhere in the portal project. Such as portlet, ActionBean, BusinessBean, JSP and etc.

**DataBean sample**

```
/*
 * Class: <DriverBean>
 * Description:<This    bean is to handle Driver related info .>
 * Version: <1.0>
 * Author: <Lori>
 * Creation date: <2005-9-9>
 * Department: Shanghai Globalization Laboratory
 * Copyright (c) 2005, International Business Machines Corporation,
 * All rights reserved.
 */
package com.ibm.sgl.oged.db.databean;

public class DriverBean {
```

```
/**
 * userid of table driver
 */
private String userid = null;
/**
 * password field of table driver
 */
private String password = null;
/**
 * role field of table driver
 */
private String role = null;

/**
 * getPassword
 * @return Returns the password.
 */
public final String getPassword() {
	return password;
}

/**
 * setPassword
 * @param password
 *                  The password to set.
 */
public final void setPassword(String password) {
	this.password = password;
}

/**
 * getRole
 * @return Returns the role.
 */
public final String getRole() {
	return role;
}

/**
 * setRole
 * @param role
 *                  The role to set.
 */
```

```
    public final void setRole(String role) {
        this.role = role;
    }

    /**
     * getUserid
     * @return Returns the userid.
     */
    public final String getUserid() {
        return userid;
    }

    /**
     * @param userid
     *                  The userid to set.
     */
    public final void setUserid(String userid) {
        this.userid = userid;
    }

    /**
     * @return return string
     * The string representation is "name=NAME, product=PRODUCT"
     * Where NAME is the company name and PRODUCT is the product name.
     */

    public final String toString() {
        // TODO Auto-generated method stub
        return super.toString();
    }
}
```

### 3.2.8 Database

Database is the repository, which will store the data used in portal applications.

You can design the database in RSA. All fields should de defined here.Here is a sample.


Then you can generate the script in RSA.

```
-- Please run the command: db2 -tvf OGED.sql
CREATE DATABASE OGEDWEMP;
CONNECT TO OGEDWEMP;

CREATE SCHEMA DB2INST1;

CREATE TABLE DB2INST1.G11NPREFERENCE
   (USERID VARCHAR(32) NOT NULL,
     LANGUAGE VARCHAR(32) NOT NULL,
     CURRENCY VARCHAR(10) NOT NULL,
     DATETIME VARCHAR(10) NOT NULL,
     CALENDAR VARCHAR(10) NOT NULL,
     WEIGHT VARCHAR(10) NOT NULL,
     LENGTH VARCHAR(10) NOT NULL,
     FIELD1 VARCHAR(32),
     FIELD2 VARCHAR(32),
     FIELD3 VARCHAR(32));

ALTER TABLE DB2INST1.G11NPREFERENCE
   ADD CONSTRAINT C2239253 PRIMARY KEY (USERID);

COMMENT ON TABLE DB2INST1.G11NPREFERENCE IS 'For G11N Preference';

ALTER TABLE DB2INST1.G11NPREFERENCE
   ADD CONSTRAINT USERID FOREIGN KEY (USERID)
   REFERENCES DB2INST1.USER(USERID);

CONNECT RESET;
```

### 3.2.9 Exception

Each portlet or subsystem might have a corresponding exception class if required. Each exception class must have a getMessage() method to return the detailed error information.

**Exception sample**

```
package com.ibm.sgl.oged.exception;

public class TrackException extends Exception {
	/**
	 * Indicate the error message
	 */
	private String _errorCode = "";

	/**
	 * Default constructor
	 */
	public TrackException () {
	}

	/**
	 * The constructor with error code as parameter.
	 * @param errorCode
	 */
	public TrackException (String errorCode) {
		this._errorCode = errorCode;
	}

	/**
	 * Get the error code which is set in the business bean.
	 * @return errorCode indicate the Error Message.
	 */
	final public String getErrorCode () {
		return _errorCode;
	}

	/**
	 * set the error code
	 */
	final public void setErrorCode (String errorCode) {
		this._errorCode = errorCode;
	}
}
```

### 3.2.10 Constant

To void using string constants in claas, we need to collect them and put into one constant class. Each portlet or subsystem might have a corresponding constant class if required. Any direct use of constants in other classes should be avoided.

Constant sample

```
package com.ibm.sgl.oged.constant;
public class TrackConstansts {
        /**
       * Constants for JSP file name
       */
      public static final String TRACK_ADVANCED_JSP = "/html/TrackByAdvanced.jsp";
      public static final String TRACK_ADVANCED_RESULT_JSP =
"/html/TrackByAdvancedResult.jsp";
      /**
       * Constants for button name
       */
      public static final String TRACK_DETAIL_BACK = "detail_back";
      public static final String TRACK_DETAIL_SUBMIT = "detail_submit";
      /**
       * Constants for title name
       */
      public static final String TRACK_BY_HAWB_SUBMIT_BUTTON =
"TrackByHAWBSubmit";
 }
```

### 3.2.11 Utility Class

We need to define the common and reusable utilities into utility class. Then all portlets can invoke the methods in utility class. Such as, string handling, locale handling, measurement, localization pack handling and etc.

**Utility Class sample**

```
package com.ibm.sgl.oged.util;

public class MeasurementScaleConverter {
      /**
       * Unit convertion step.
       */
      public static final int SCALE_METER = 100;

      /**
```

```
 * Unit convertion step.
 */
public static final int SCALE_FOOT = 12;

private MeasurementScaleConverter() {
	super();
	// TODO Auto-generated constructor stub
}

/**
 * This method will convert the value in one unit to another unit according
 * the scale of the value.
 *
 * @param unit
 *                The input unit of this attribute. The values are from the code
 *                system.
 * @param length
 *                The input value of this attribute.
 * @return an object array which contains two objects. The first object is
 *             the converted float value. The second object is the converted
 *             unit.If the value needs not to be converted, the returned value
 *             will be the same as the parameters.
 */
public final static Object[] convert(Double length, String unit) {
	// TODO Auto-generated method stub
	double lengthValue = length.doubleValue();
	String convertedUnit = unit;
	if (CodeSystem.LENGTH_METER.equals(unit)) {
		if (1.0 > lengthValue) {
			lengthValue *= SCALE_METER;
			convertedUnit = CodeSystem.LENGTH_CENTIMETRE;
		}
	} else if (CodeSystem.LENGTH_CENTIMETRE.equals(unit)) {
		if (100.0 < lengthValue) {
			lengthValue /= SCALE_METER;
			convertedUnit = CodeSystem.LENGTH_METER;
		}
	} else if (CodeSystem.LENGTH_FEET.equals(unit)) {
		if (1.0 > lengthValue) {
			lengthValue *= SCALE_FOOT;
			convertedUnit = CodeSystem.LENGTH_INCH;
		}
	} else if (CodeSystem.LENGTH_INCH.equals(unit)) {
```

```
                if (12.0 < lengthValue) {
                    lengthValue /= SCALE_FOOT;
                    convertedUnit = CodeSystem.LENGTH_FEET;
                }
            }

            Object[] ret = new Object[2];
            ret[0] = new Double(lengthValue);
            ret[1] = convertedUnit;
            return ret;
        }
}
```

### 3.2.12 Code System

Code System is used to store the data, which will be used for data exchanging between portlets and subsystems. The constant in Code System is a key in the applications. It will be replaced with translated strings using WPS tag in JSP page.

**Code System sample**

```
package com.ibm.sgl.oged.util;

import java.util.ArrayList;

public class CodeSystem {
  //public static final String Key = "Value" //Description

      //Report Type
      public static final String REPORT_TYPE_FULL = "RT1001"; //Full
      public static final String REPORT_TYPE_MEDIUM = "RT1002"; //Medium
      public static final String REPORT_TYPE_BRIEF = "RT1003"; //Brief

      public static final ArrayList getReportTypeVector() {
          ArrayList result = new ArrayList();
            result.add(0,CodeSystem.REPORT_TYPE_FULL);
            result.add(1,CodeSystem.REPORT_TYPE_MEDIUM);
            result.add(2,CodeSystem.REPORT_TYPE_BRIEF);
            return result;
      }


}
```

### 3.2.13 Subsystem

Subsystem is other application beside portlet, which will be invoked by portal applications. For example, BusinessBean can invoke the subsystem through web services.

Here is a sample. In the class **RequestReportBusinessBean**, which invokes the email Web Service of other subsystem**.**

**Subsystem sammple**

```
EmailWebServiceProxy proxy = new EmailWebServiceProxy();
        HtmlEmail htmlEmail = new HtmlEmail();
        htmlEmail.setCustomerID(reportRequest.getUserid());
        htmlEmail.setReport_format(reportRequest.getFormat());
        String mailContent = htmlEmail.mail4Report(ReportConstansts.LOCALE_US);
        String mailSubject = LPUtil.getLPValue(locale, ReportConstansts.FILE_MYTASK,
ReportConstansts.EMAIL_SUBJECT);

        try {
            final boolean ifgroup = proxy.sendMailToGroup(group, mailSubject,
mailContent);
        } catch (RemoteException e1) {
            logger.error(e1.getMessage());
        }
```

### 3.2.14 GBO

Global Business Object (GBO) is a Java based technology, which offers a solution for constructing culturally aware graphical controls in global J2EE applications. Through encapsulating the functions into a reusable globalization tag library, GBO allows programmers to create global GUI functions in their Java Server Pages (JSP) files very conveniently.

Note: Please contact GBO team for the latest GBO version.

**GBO sample**

```
<tr>
  <td class="td3<%=dir%>"><lp:value key="TABLE_BEGINDATE" /> </td>
   <td class="td6">
      <input type="text" id="date1_Show" name="begindateshow"
readonly="readonly" value=<%=begindate%>>
      <input type="hidden" id="date1" name="begindate"
<%if(0!=begindatehidden.length())%> value=<%=begindatehidden%>>
      <input type="hidden" id="date1_Unformat" >
      <img src="/wps/oged/images/calendar.gif" alt="<lp_alt:value
```

```
key="CALENDAR"/>"
onclick="window.open('<%=renderRequest.getContextPath()%>/html/popupcalendar/PopupCalendar.jsp?locale=<%=poplocale%>','date1','height=200,width=210,scrollbars=no,resizable=yes,status=no,menu=no,toolbar=no')">
        </td>
          <tr>
```

For details about GBO, please refer to the section *Enable GBO in portal project*

### 3.2.15 Log4j

We can use Log4j to write the log in log files in portal project.

#### 3.2.15.1 Add the jar file into project

Please add the **log4j-1.2.9.jar** into your portal project first. For example: **PortalProject\WebContent\WEB-INF\lib**

#### 3.2.15.2 Add the log4j.properties into project

Add the log4j.properties file into **PortalProject\JavaSource**

#### 3.2.15.3 Modify the log4j.properties

You may need to customize the log4j.properties, such as log level, log file directory and log format.
Here is a sample.

```
# An example log4j configuration file that outputs to System.out.  The output information consists of relative time, log level, thread name, logger name, nested diagnostic context and the message in that order.

# For the general syntax of property based configuration files see the documenation of org.apache.log4j.PropertyConfigurator.

log4j.rootLogger=DEBUG, FILE

# A1 is set to be a ConsoleAppender which outputs to System.out.
#log4j.appender.A1=org.apache.log4j.ConsoleAppender

# A1 uses PatternLayout.
#log4j.appender.A1.layout=org.apache.log4j.PatternLayout

# The conversion pattern uses format specifiers. You might want to
# change the pattern an watch the output format change.
```

```
#log4j.appender.A1.layout.ConversionPattern=%-4r %-5p [%t] %37c %3x - %m%n

# In this example, we are not really interested in INNER loop or SWAP
# messages. See the effects of uncommenting and changing the levels of
# the following loggers.
# log4j.logger.org.apache.log4j.examples.SortAlgo.INNER=WARN
# log4j.logger.org.apache.log4j.examples.SortAlgo.SWAP=WARN


log4j.appender.FILE=org.apache.log4j.FileAppender
log4j.appender.FILE.File=/opt/WebSphere/PortalServer/log/oged.log
log4j.appender.FILE.Append=true
log4j.appender.FILE.layout=org.apache.log4j.PatternLayout
#log4j.appender.FILE.layout.ConversionPattern=%-4r %-5p [%t] %37c %3x - %m%n
log4j.appender.FILE.layout.ConversionPattern=[%-d{yyyy-MM-dd HH:mm:ss}] [%37c] %3x - %m%n
```

3.2.15.4 Invoke the log4j in class

There are three levels of log, info, debug and error. You can use like the following.

**logger.inf("Info here.");**
**logger.debug("Just debug here");**
**logger.error("Error occurs!")'**

Here is a sample.

```
package com.ibm.sgl.oged.wemp.businessbean;

import com.ibm.sgl.oged.constant.TrackConstansts;
import com.ibm.sgl.oged.db.manager.ShipmentBeanManager;
import com.ibm.sgl.oged.db.databean.ShipmentBean;
import com.ibm.sgl.oged.exception.TrackException;
import com.ibm.sgl.oged.util.DBException;

import org.apache.log4j.*;

public class TrackByHAWBBrieflyBusinessBean {

    private ShipmentBeanManager shipmentbeanmanager = new ShipmentBeanManager();

    /**
     * logger
     */
```

```
    private static Logger logger =
Logger.getLogger(TrackByHAWBBrieflyBusinessBean.class);

    public ShipmentBean getShipmentDetailByHAWB(String hawb) throws TrackException {
        ShipmentBean sb = new ShipmentBean();
        logger.debug("#Peter# start    to get the shipment info"); //$NON-NLS-1$
        try {
            sb = shipmentbeanmanager.getShipmentByHAWB(hawb);
        } catch (DBException e) {

            String errMsg = e.getErrorCode();
            if (TrackConstansts.EM_GENERAL.equals(errMsg)) {
                throw new TrackException(errMsg);
            } else {
                logger.debug(e);
            }
    }
        if (null == sb) {
            logger.debug("#Peter# buz bean is    null "); //$NON-NLS-1$
        }
        logger.debug("#Peter# buz bean    pan get the shipment info"); //$NON-NLS-1$
        return sb;
    }


}
```

3.2.15.5 Logs in log file

Here is the log in log file oged.log.

```
[2005-11-10 16:40:00] [com.ibm.sgl.oged.wemp.portlet.SelectLanguagePortlet] - Enter
doView method!
[2005-11-10 16:40:00] [com.ibm.sgl.oged.wemp.portlet.SelectLanguagePortlet] - The jsp is
SelectLanguage
[2005-11-10 16:40:00] [com.ibm.sgl.oged.wemp.portlet.SelectLanguagePortlet] - Exit
doView method!
```

For details, please refer to the section *Log Handling in Portal Project*

3.2.16 Images and Stylesheet

To make the portal website more beautiful, we need to add some images and apply the stylesheet

for JSP page.

#### 3.2.16.1 Add images

We can add the images into project: **PortalProject\WebContent\images**
Or we can separate them from project and place the images into wps.war directory like the following.
**opt\WebSphere\AppServer\installedApps\ogedwemp\wps.ear\wps.war\oged\images**

Note: If you want to use different image for each locale,please use the subdirectory for the locale as following.
**images\en_US**
**images\zh_CN**
**images\fr_FR**
**images\ar_EG**

#### 3.2.16.2 Apply stylesheet

The stylesheet can be the same directory with images. Please pay attention to the tag **.td1rtl** , which is used to support bidi language.
Here is the sample of stylesheet.

```
.td1{
	border-left: 5px solid #c1c1c1;
	font-weight: bold;
	color: #727272;
	padding: 5px 10px 5px 10px;
	border-bottom: 1px solid #c1c1c1;
	border-right: solid 1px #c1c1c1;
	background: #eeeeee;
}
.td1rtl{
	border-right: 5px solid #c1c1c1;
	font-weight: bold;
	color: #727272;
	padding: 5px 10px 5px 10px;
	border-bottom: 1px solid #c1c1c1;
	border-left: solid 1px #c1c1c1;
	background: #eeeeee;
}
```

For details, please refer to the section *Artwork in portal*

# 4. Portal development process

Here are the detail steps for portal development.

## 4.1 Portal development overview

Here is the portal development overview. There are several roles and scenes.

1. Design

The designer will do the design in RSA or RSM according to the use case in RequestPro. In the design phase, the designer will deliver a lot of design documents, including UI pages, portlets, interface, modeling and etc.

2. Development

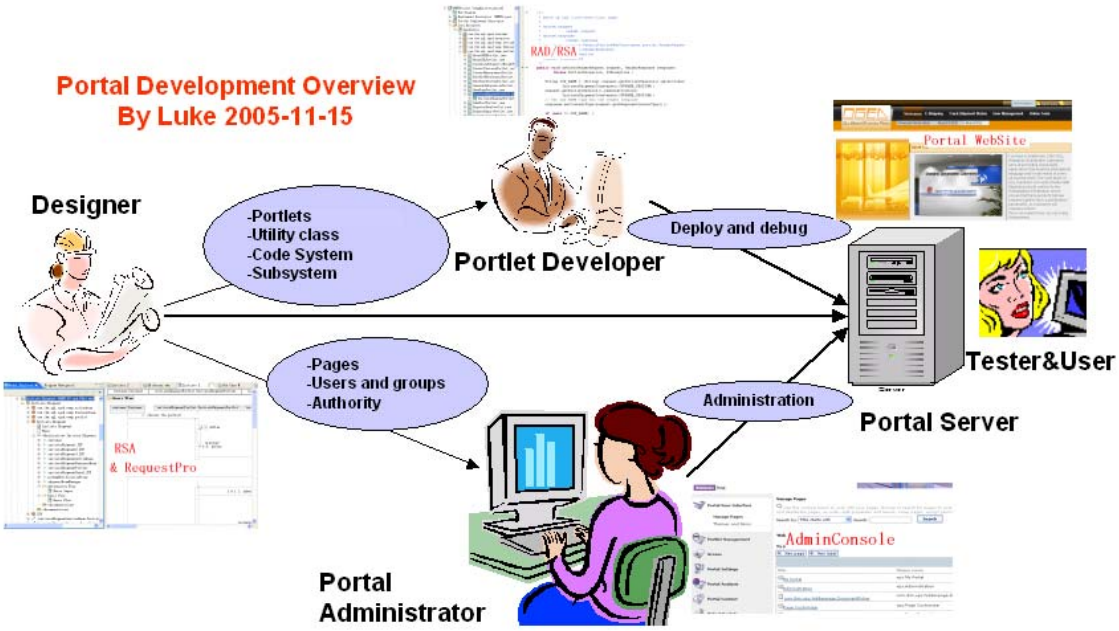
The portlet developer will do the development in RAD or RSA according to the design documents. Then deploy the war file and debug the application in portal debugging server.

3. Administration

The portal administrator will do the administration in portal admin console; for example, create users and group, create pages, set the access permissions for page or portlet.

4. Test

The testers will access the portal website in portal testing server, and then submit defects in ClearQuest or TCT.




## 4.2 Design

The designer will do the design in RSA or RSM according to the use case in RequestPro. In the design phase, the designer will deliver a lot of design documents.

- **For the developer**: UI pages, portlets, utility classes, code system, subsystems.

- **For portal administrator**: pages, users and groups, authority and etc.

## 4.3 Development

Here is the portlet project development process.

### 4.3.1 Setup Environment

Before you can develop a portal project, you need to setup the portal server environment and development environment. Here is the environment description.

**Portal Server**

There are two portal servers, one is for test and deployment, and another is for debugging. The following needs to be installed.

- VMWare (It will help you to easily backup the portal server.)
- Redhat Linux AS3.0 update3
- WPS for Multiplatform V5.1 (WAS, WPS, DB2, LDAP)
- WEMP V5.1 (**Optional**. It's used to develop XDIME portlets to support mobile device.)
- Samba (It can help you easily access the file directory in portal server.)
- SSH (Start the ssh service in portal server. It can help you control the portal server remotely.)

**Development environment**

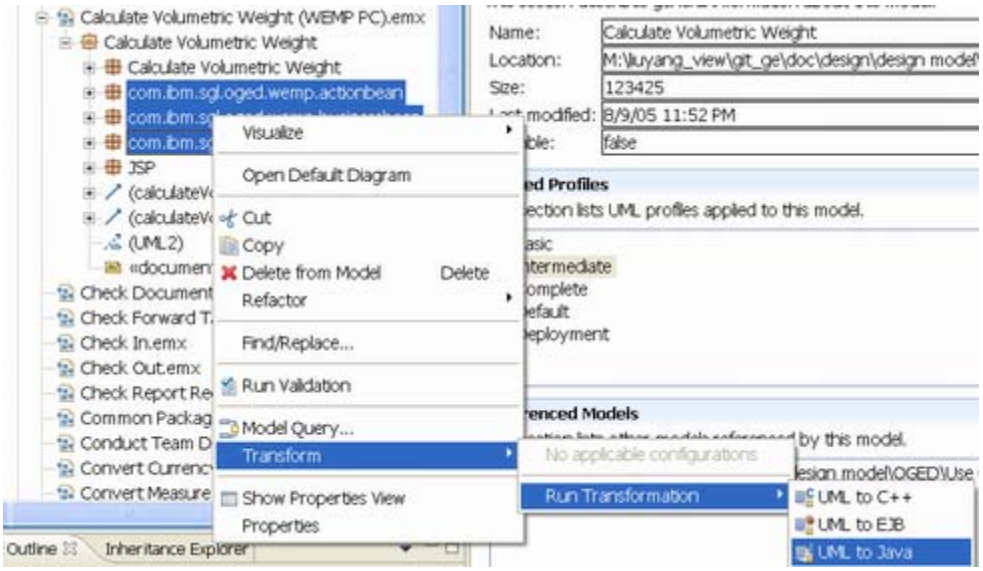
- RAD V6.0 or RSA V6.0
- WebSphere Portal V5.1 Test Environment (**Optional**. You can debug the porllet in portal server directly. )
- Portal Toolkit (It's packaged with RAD now. If you want to develop XDIME portlet, please install Mobile Portal Toolkit in RAD V6.0)
- UltraEdit
- SSH (It can help you control the portal server remotely.)

For details about how to setup environment, please refer to the InfoCenter of WPS and the Redbook **sg246681.pdf** (Rational Application Developer V6 Portlet Application Development and Portal Tools).

### 4.3.2 Transformation from Model to Java

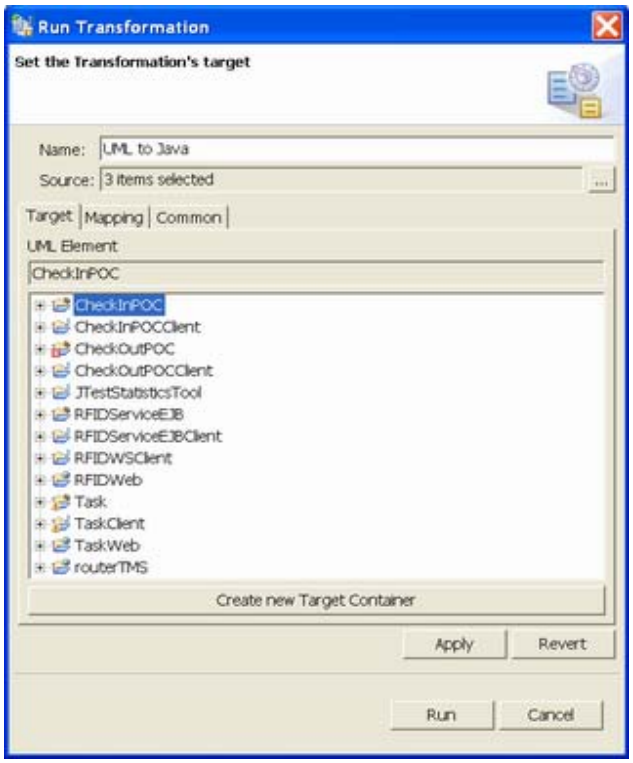
The designer has designed many classes in RSA. The developer needs to run the **Transformation from Model to Java** in RSA. The generated java file can be imported into portlet project.

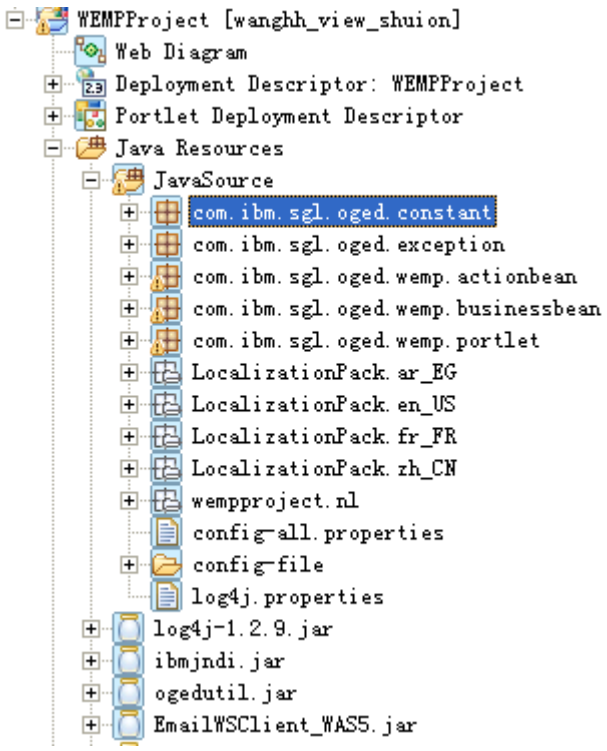
In RSA, right click the packages, select **Transform** -> **Run Transformation** -> **UML to Java**




Select a project that you want to put the generated java files and click **Run**.

Then you can view the generated java files in the target project like the following image.





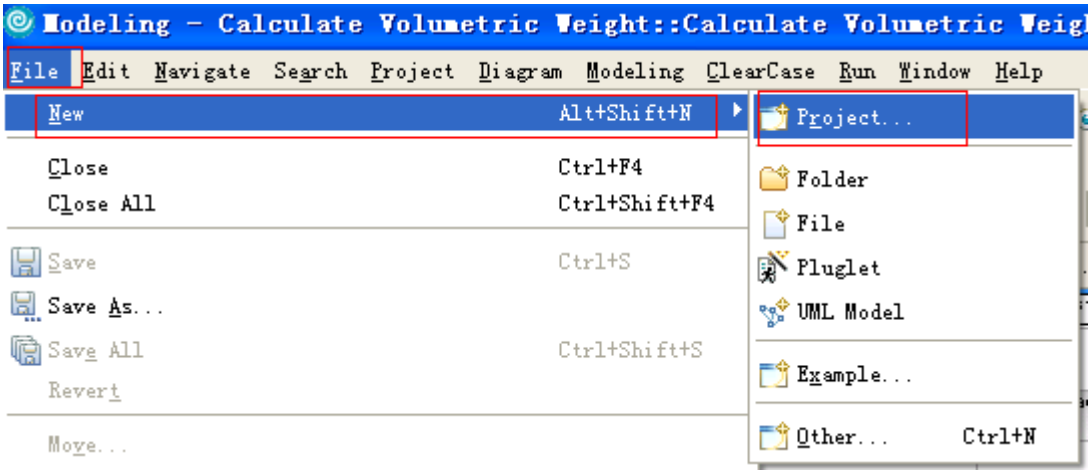

**Note:** For the referenced object in model project is platform-independent, you may need to do some modifications for the generate java file after transportation. For example, change the object to Date/Time, Calendar, ArrayList and etc.

### 4.3.3 Create a portlet project

Now you can create a portlet project in RAD. Please do as the following steps.

1. Create a portlet project in RAD

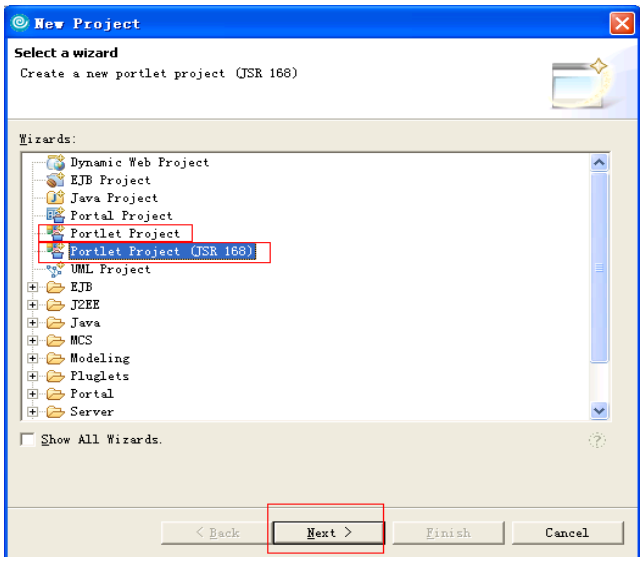
In RAD, select menu **File** -> **New** -> **Project…**

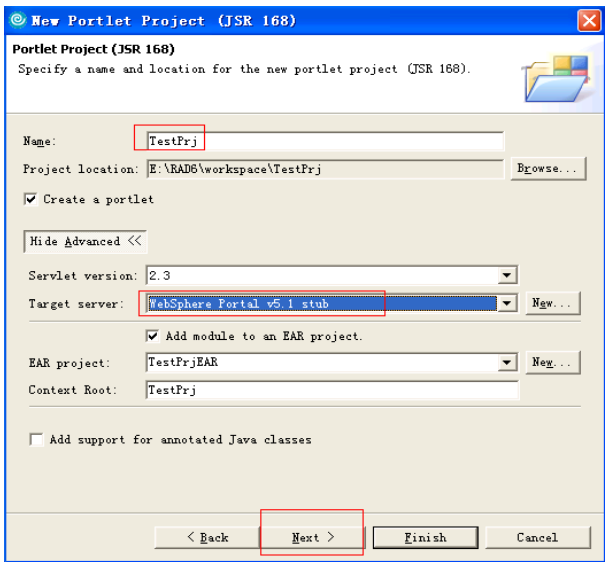



2. In the left image, you can select **Portlet Project** (It's IBM API portlet) or **Portlet Project**

**(JSR 168)**. For the difference between IBM API portlet and JSR168 portlet, please refer to the section *Comparison of several types of portlet projects*

3. Input the project name and select the target server, then click **Next**.




4. Select portlet type

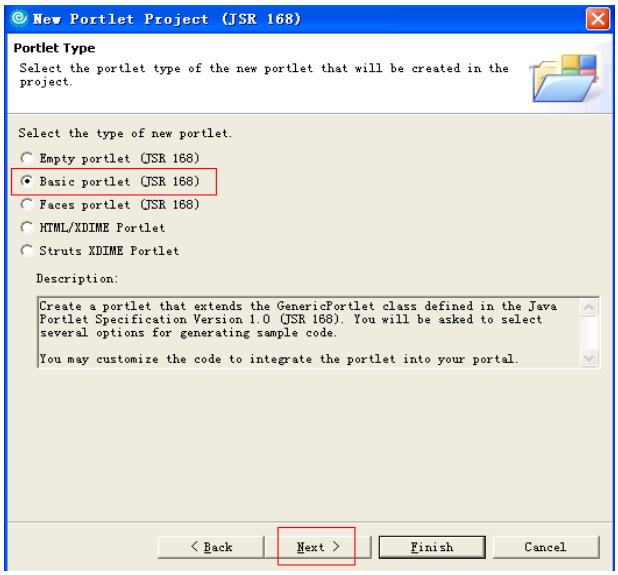
You can select the type of new portlet from list.

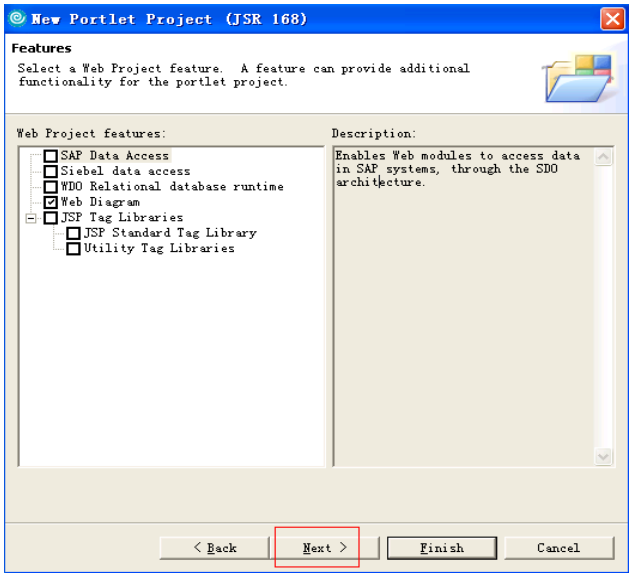
- Empty portlet (JSR168)
- Basic portlet (JSR168)
- Faces portlet (JSR168)
- HTML/XDIME portlet (JSR168)
- Structs XDIME portlet (JSR168)

For the difference between IBM API portlet, please refer to the section *Comparison of several types of portlet projects*

5. Features

Just click **Next** to the next step.



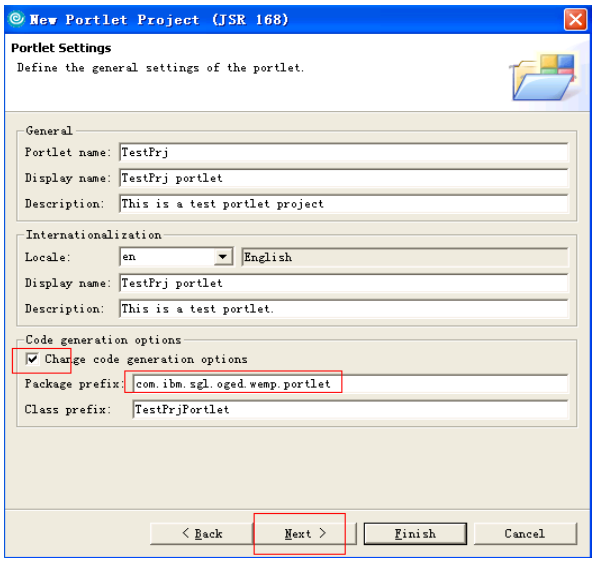



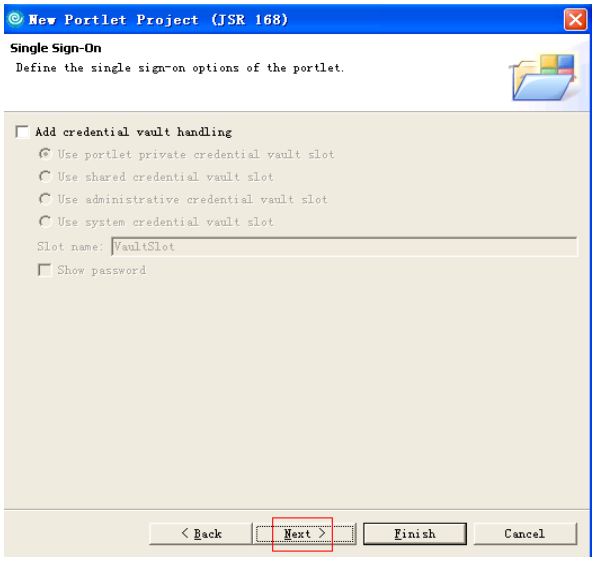

6. Portlet Settings

Please change the package prefix here, for example, com.ibm.sgl.oged.wemp.

7. Single Sign-On

Just click **Next** to the next step.



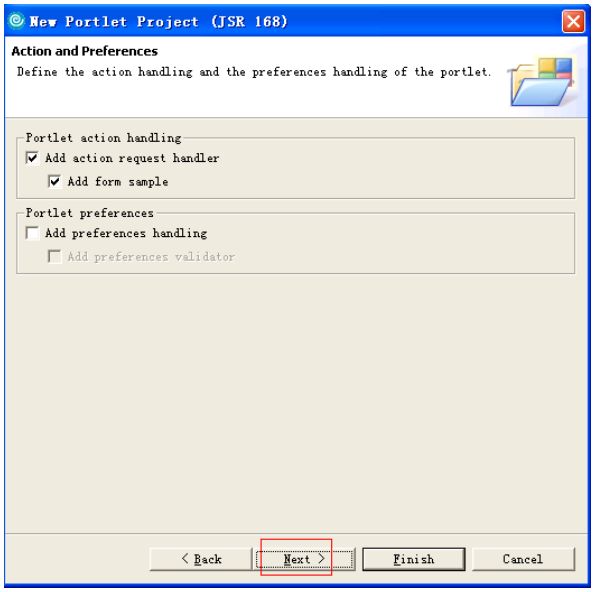



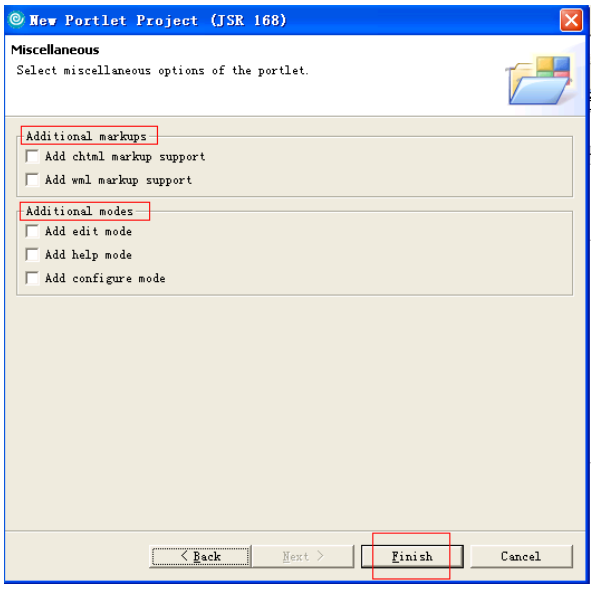

8. Action and Preferences

Just click **Next** to the next step.

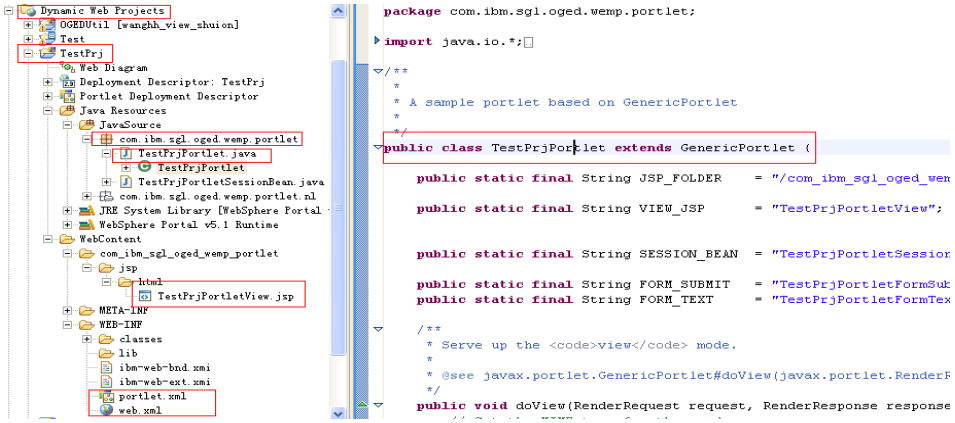
9. Miscellaneous

You can select the markup and mode, then click **Finish** to create a new portlet project.




10. After the portlet is created successfully, you can view the project in the workspace.




11. When the portlet project is created successfully, please create the java packages according to the design in model project like the following.

**For Portlet**

com.ibm.sgl.oged.portlet
com.ibm.sgl.oged.actionbean
com.ibm.sgl.oged.businessbean

**For Common packages**
com.ibm.sgl.oged.util
com.ibm.sgl.oged.exception
com.ibm.sgl.oged.constant

**For DB packages**
com.ibm.sgl.oged.db.connector
com.ibm.sgl.oged.db.manager
com.ibm.sgl.oged.db.databean

12. Import all the java files generated from modeling project.

13. Add the required jars and libs into the classpath of the portlet project.
There are some common jars, such as log4j.jar, gbo.jar, icu4j.jar and etc.

14. Create all the portlets defined in the design document before portlet development.
Just click **File** - > **New** -> **Portlet**, and follow the steps.

15. Add all the JSP files according to the UI pages.
Change the extension from html to JSP and import into **<PortletProject>\<WebContent>\<Markup>**

16. Modify the JSP file
For the JSP page of a portlet is surrounded by the theme in portal website, we can just need the main contents and remove the <html>, <head> and <body> in JSP file. Also we need to add the head for each JSP file.
Here is the sample JSP file.

```
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8" session="true"%>
<%@ taglib uri="http://java.sun.com/portlet" prefix="portlet" %>
<%@ page import="com.ibm.sgl.oged.util.LPUtil" %>
<portlet:defineObjects />

<%@ taglib uri="/WEB-INF/tld/lp.tld" prefix="lp" %>
<lp:load file="welcome.properties"/>
<%@ taglib uri="/WEB-INF/tld/lp.tld" prefix="lp_alt" %>
<lp_alt:load file="image.properties"/>
<%@include file="Bidi.jsp"%>
```

```
<link href="/wps/oged/images/<%=locale%>/Styles.css" rel="stylesheet" type="text/css">

<%
String locale= LPUtil.getFullLocale(request.getLocale().toString());
 %>
<table width="100%" border="0" cellspacing="0" cellpadding="0">
  <tr>
    <td width="250" height="170" valign="top" class="Backgroundleftbar"><img src="/wps/oged/images/<%=locale%>/pic_leftbar.gif" width="250" height="295" alt="<lp_alt:value key="PIC_LEFTBAR" />" ></td>
    <td width="75%" height="170"><br>
      <table width="100%" border="0" align="center" cellpadding="0" cellspacing="0">
        <tr>
          <td height="30" colspan="2" class="td5<%=dir%>"><lp:value key="TITLE_ABOUTSGL" /></td>

        </tr>
        <tr>
          <td width="40%" class="td3<%=dir%>"> <p><img src="/wps/oged/images/<%=locale%>/SGL.gif" width="320" height="220" alt="<lp_alt:value key="SGL" />" ></p></td>
          <td width="60%" class="td3<%=dir%>">
        <lp:value key="NOTE_ABOUTSGL" />
        <br><lp:value key="NOTE_ABOUTSLOGAN" />
      </td>
        </tr>
      </table>
      <br>
    </td>
  </tr>
</table>
```

17. Modify the configuration file

You may need to modify the configuration file of portlet project, such as web.xml, portlet.xml.

**Note:** For faces portlet, please try to define all the managed beans and navigation rules in faces-config.xml

18. Add the portlet project into ClearCase for team development.

Right click the project, and select **Team - >Share Project**.

### 4.3.4 Portlet Development

Now the developer can import the project from ClearCase and do the portlet development now. The developer needs to write the portlet application according to the design in the modeling project.

In the development, please pay attention to the following issues.

- Session
- Portlet jumping
- Portlet corporation
- Portal DB encoding
- Taglib
- Login
- Language changing
- User information
- Locale info
- Theme customizing
- Localization Pack handling
- Exception handling
- Log handling
- Code system
- DAO
- Bidi support
- GBO

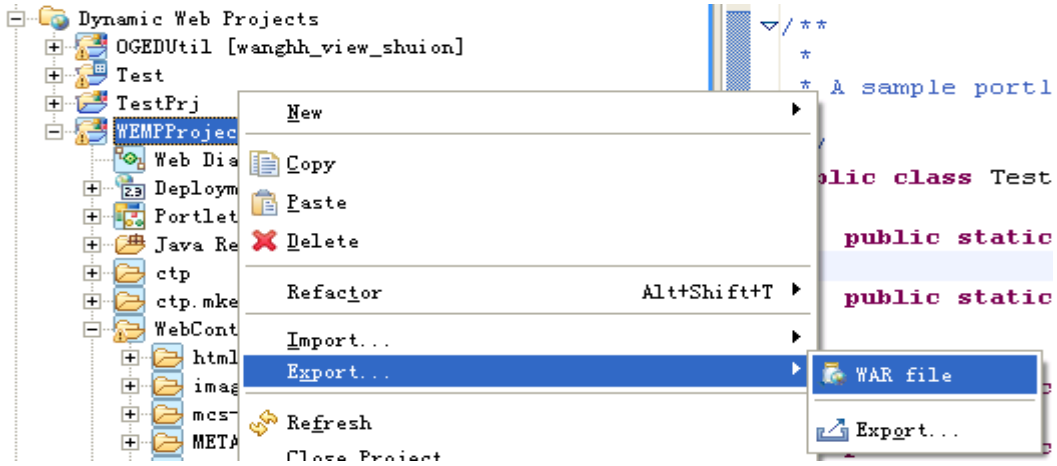
**Note:** Please check in your code in ClearCase regularly. Please do **NOT** check in the code that cannot be compiled.

For details about how to develop portlet application, please refer to the InfoCenter of WPS and the Redbook **sg246681.pdf** (Rational Application Developer V6 Portlet Application Development and Portal Tools).

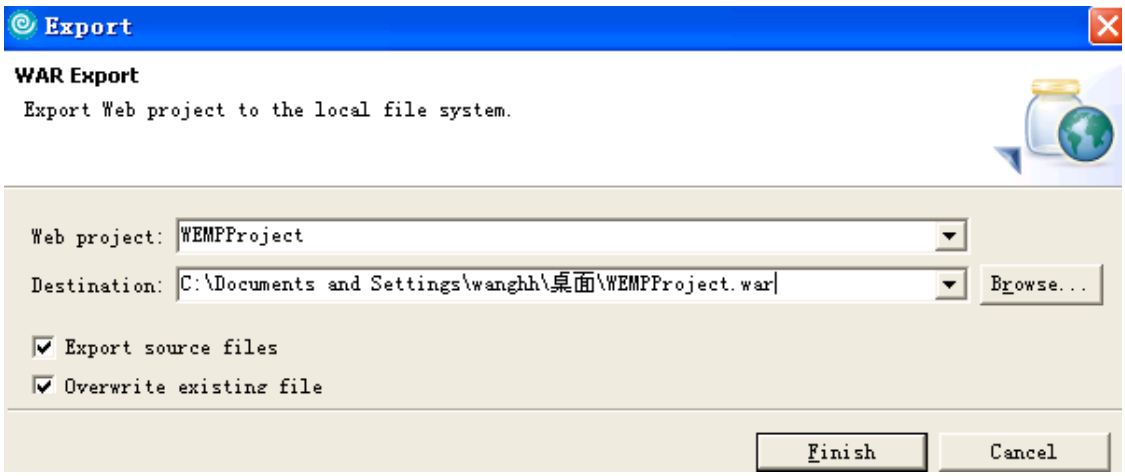
### 4.3.5 Deployment

Before you want to deploy the portlet project, you'd better reboot the portal server before deployment; otherwise it will fail due to the lack of memory.

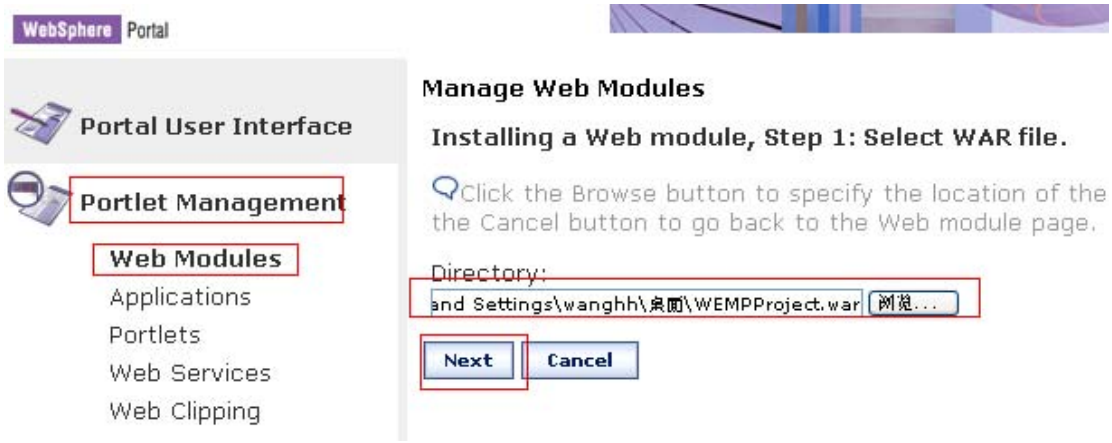
To deploy the portlet project in portal server, please right click the project name in RAD, then select **Export…** -> **WAR file**




Input the destination path and click **Finish**. Then the war file for this project will be generated.




Then you can login portal using wpsadmin/wpsadmin, select **Administration** -> **Portlet Management** -> **Web Modules** - >**Install**, input the war file directory, click **Next** to finish the installation.




After the installation is finished, the portal administrator needs to manage pages and portlets so that user can visit the portlet applications.

### 4.3.6 Administration

The portal administrator needs to do the following in adminconsole so that user can visit the

portlet applications.

- Create pages
- Set the page tile (Add multilingual title for each page)
- Create users and groups
- Assign the permissions for each page
- Assign the permissions for each portlet

For details about how to manage users, pages and portlets, please refer to the section *Portal Administration*

#### 4.3.7 Debug and Test

After the portal administrator finished the administration, the developer or tester can access the portal website and view the portlet applications.
Here are the detail steps of debugging and test.

1. Access http://hostname: 9080/wps/portal, verify the portlet application.
2. Check the error log

Log is very useful for debugging. Here are the important log files.

- /opt/WebSphere/PortalServer/log/SystemOut.log
- /opt/WebSphere/PortalServer/log/SystemErr.log
- /opt/WebSphere/PortalServer/log/wps_2005.07.21-06.52.19.log
- /opt/WebSphere/PortalServer/log/oged.log (Your own log file which uses the log4j.jar.)

3. Modify the source file in IDE, and rebuild the project.
4. You can redeploy the project again. Or just copy the JSP, class or policy file to the corresponding directory of portal server as following.

- **JSP**: <WPS>\installedApps\HelloWorld_PA_1_0_LT.ear\HelloWorld.war\helloworld\jsp\xdime
- **Class**: <WPS>\installedApps\HelloWorld_PA_1_0_LT.ear\HelloWorld.war\WEB-INF\classes\helloworld
- **Policy**: <WPS>\installedApps\HelloWorld_PA_1_0_LT.ear\HelloWorld.war\mcs-policies

5. Verify the portlet application again

### 4.4 Maintenance

We need to do the maintenance through the portlet project. Most important is backup! Please do the backup in each milestone, for example, every day or every week.

Here are the main actions for maintenance.

- Write a maintenance document, which contains the server information: IP address, hostname, username, password, VMWare image location, start service and etc.

- Backup the important configuration file
- Backup the important log files
- Backup the source code of the portal project
- Backup the VMWare images of portal server when you have deployed the war file successfully or have finished the management and authorization for page and portlets.

Each time when you have recovered the portal server from a VMWare image, please verify the portlet applications.

# 5. Best practice of portal development

## 5.1 Login

The login application is packaged with WPS. We can customize the application to meet our requirement.

### *5.1.1 Find the login war directory in the PoralServer as the following.*

```
\\10.1.1.180\opt\WebSphere\PortalServer\installedApps\Login
Port_pplication_PA_1_0_3T.ear\login.war\WEB-INF\jsp
```

### *5.1.2 Modify the following jsp file to meet your requirement*

1. LoginView.jsp:
2. BidiInclude.jsp
3. StatusMessageInclude.jsp
4. LoggedInView.jsp

### *5.1.3 Add the style sheet for LoginView.jsp to support vary style sheet for each supported language.*

```
<%

        String userLocale = getFullLocale(request.getLocale().toString()); //To get the full locale like en_US

            if (userLocale.equals("zh_CN"))
            {
            %>
      <LINK href="/wps/oged/images/zh_CN/Styles.css" rel="stylesheet"
            type="text/css">
      <% }else if(userLocale.equals("fr_FR"))
      {
      %>
      <LINK href="/wps/oged/images/fr_FR/Styles.css" rel="stylesheet"
            type="text/css">
      <% }else if(userLocale.equals("ar_EG"))
      {
      %>
      <LINK href="/wps/oged/images/ar_EG/Styles.css" rel="stylesheet"
            type="text/css">
      <% }else{ %>
      <LINK href="/wps/oged/images/en_US/Styles.css" rel="stylesheet"
```

```
        type="text/css">
<% } %>
```

*5.1.4 You can also add your new LP strings into login properties like the following.*

Add the following:

```
NOTE_INPUT=\u8bf7\u8f93\u5165\u60a8\u7684\u7528\u6237\u540d\u548c\u5bc6\u7801
```

In the file

\\10.1.1.180\opt\WebSphere\PortalServer\installedApps\Login Port_pplication_PA_1_0_3T.ear\login.war\WEB-INF\nls\loginportlet_zh.properties

*5.1.5 Invoke the lp strings using taglib*

```
<fmt:message key="NOTE_INPUT"/>
```

**5.2 Exception handling in portlet application**

Please add the exception handling in your portlet application like the following.

*5.2.1 Create the exception class for your portlet.*

If you can not find your exception in the package com.ibm.sgl.gpim.exceptions, please add your own exception class like ProductBusinessException

*5.2.2 . Set the error code and throw exception in the methods of business beans*

```
public ProductDetailContainerDataBean getProductByEPCCode (String locale,String EPCCode,
String catalogName) throws      ProductBusinessException {

        try{
            //Write your code here
            } catch (RemoteException remoteException) {
                throw new ProductBusinessException("error_code");
            }
}
```

*5.2.3 Catch the exception in the actionbeans and set the error code in the session.*

Note: The error code should be included in the lp properties file.

```
        try{
```

```
            //Call the method in the business bean here

               }catch (ProductBusinessException productBusinessException ) {

                productBusinessException.printStackTrace();

                String error_code = productBusinessException.getErrorCode();
                if (null != error_code) {
                    this.getRequestScope().put("error_code", error_code);
                 }
          }
```

*5.2.4 Get the error code from session and print the error in the jsp*

Get the error message from lp file according to the error code.

Print the error message in the top of the page.

```
<table width="100%" border="0" cellspacing="0" cellpadding="0">

    <%
        String error_code = (String)renderRequest.getAttribute("error_code");

        if (error_code != null) {
    %>

    <tr>
        <td>
          <p><font color="#FF6600"><lp:value key="<%= error_code %>"/></font></p>

        </td>
    </tr>

    <%
         }
    %>
</table>
```

### 5.3 Log handling in portal project

We can do as the following steps.

### *5.3.1 Add the log4j-1.2.9.jar into the class path of the portal project.*

Please add the **log4j-1.2.9.jar** into your portal project first. For example: **PortalProject\WebContent\WEB-INF\lib**

### *5.3.2 Add the log4j.properties in the directory <PortalProject>\JavaSource*

Here is the sample code of the log4j.properties

```
# An example log4j configuration file that outputs to System.out.  The
# output information consists of relative time, log level, thread
# name, logger name, nested diagnostic context and the message in that
# order.

# For the general syntax of property based configuration files see the
# documenation of org.apache.log4j.PropertyConfigurator.

log4j.rootLogger=DEBUG, FILE
#log4j.rootLogger=DEBUG, A1, FILE

# A1 is set to be a ConsoleAppender which outputs to System.out.
#log4j.appender.A1=org.apache.log4j.ConsoleAppender

# A1 uses PatternLayout.
#log4j.appender.A1.layout=org.apache.log4j.PatternLayout

# The conversion pattern uses format specifiers. You might want to
# change the pattern an watch the output format change.
#log4j.appender.A1.layout.ConversionPattern=%-4r %-5p [%t] %37c %3x - %m%n

# In this example, we are not really interested in INNER loop or SWAP
# messages. See the effects of uncommenting and changing the levels of
# the following loggers.
# log4j.logger.org.apache.log4j.examples.SortAlgo.INNER=WARN
# log4j.logger.org.apache.log4j.examples.SortAlgo.SWAP=WARN

log4j.appender.FILE=org.apache.log4j.FileAppender
log4j.appender.FILE.File=/opt/WebSphere/PortalServer/log/oged.log
log4j.appender.FILE.Append=true
log4j.appender.FILE.layout=org.apache.log4j.PatternLayout
#log4j.appender.FILE.layout.ConversionPattern=%-4r %-5p [%t] %37c %3x - %m%n
log4j.appender.FILE.layout.ConversionPattern=[%-d{yyyy-MM-dd HH:mm:ss}] [%37c] %3x -
%m%n
```

*5.3.3 Import the class in your java class.*

```
import org.apache.log4j.Logger;
```

*5.3.4 Define a class varible like the following.*

```
static Logger logger = Logger.getLogger(RegisterUserActionBean.class);
```

*5.3.5 Write the log in the method like the following.*

```
logger.info("It's just a information.");
logger.debug("It's a debug information.");
logger.error("There is an error here!");
```

Here is the sample code.

```
public boolean doSubmit(ActionRequest request) {
        logger.info("*****Enter the doSubmit method!*****");
        logger.debug("The email address is :" + request.getParameter("email"));
        boolean retval = false;
        if (validate(request)) {
            logger.debug("Luke--Validate user input successfully.");

            //The default role is customer
            userBean.setRole(CodeSystem.ROLE_CUSTOMER);
            userBean.setPassword("123456");

            UserBean userBean = (UserBean) request.getPortletSession()
                    .getAttribute(UserConstansts.USER_BEAN);
            try {
                retval = registerUserBusinessBean.addUser(userBean);
            } catch (UserManageException e) {
                logger.error("The error code is " + e.getErrorCode());
                request.getPortletSession().setAttribute(UserConstansts.ERRMSG_RESULT,
e.getErrorCode());
            }
            logger.debug("Luke--insert result is:" + retval);
            retval = true;

        } else {
            logger.debug("Luke--Validate Failed");
            retval = false;
        }
```

```
        logger.info("*****Exit the doSubmit method!*****retval is " + retval);
        return retval;
    }
```

*5.3.6 Find the log which you set in the file log4j.properties*

For example,view the log in **/opt/WebSphere/PortalServer/log/oged.log**

```
[2005-10-19 07:33:23] [com.ibm.sgl.oged.wemp.portlet.RegisterUserPortlet] - The jsp name is RegisterUser
[2005-10-19 07:33:31] [com.ibm.sgl.oged.wemp.portlet.RegisterUserPortlet] - *****Exit doView method!*****
[2005-10-19 07:33:56] [com.ibm.sgl.oged.wemp.portlet.RegisterUserPortlet] - *****There is an error here!****
```

*5.3.7 You can also set the log level in the properties file log4j.properties, and then restart the application or the server.*

```
log4j.rootLogger=INFO, FILE
or log4j.rootLogger=DEBUG, FILE
or log4j.rootLogger=ERROR, FILE
```

### 5.4 Code System

We often want to exchange the data between jsp, portlet, database or other system.
Commonly we need to map each other to know the exactly meaning of the value. So we can use the code system across the functions and systems. Then it will be easy to understand each other.

*5.4.1 Define the code system in the CodeSystem.java.*

```
    //City list
    public static final String CITY_ALL = "CITY1000"; //All
    public static final String CITY_NEW_YORK = "CITY1001"; //New York
    public static final String CITY_SHANGHAI = "CITY1002"; //Shanghai
    public static final String CITY_PARIS = "CITY1003"; //Paris
  public static final String CITY_CAIRO = "CITY1004"; //Cairo
```

*5.4.2 We can also define a method to get the code list defined in CodeSystem.java.*

This method will return an ArrayList,we can use it in our portlet or jsps.

```
    public static final ArrayList getCityVector() {
```

```
        ArrayList result = new ArrayList();
        result.add(0,CodeSystem.CITY_ALL);
        result.add(1,CodeSystem.CITY_NEW_YORK);
        result.add(2,CodeSystem.CITY_SHANGHAI);
        result.add(3,CodeSystem.CITY_PARIS);
        result.add(4,CodeSystem.CITY_CAIRO);

        return result;
  }
```

#### *5.4.3 Using the code system in portlet or jsps.*

LP handling:

```
<lp:value key=<%=CodeSystem.CITY_SHANGHAI%> />
```

Comparison:

```
If(theCity.equals(CodeSystem.CITY_SHANGHAI)){
//TODO here
}
```

#### *5.4.4 Or you can get the list*

```
ArrayList   cityList =   CodeSystem.getCityVector();
```

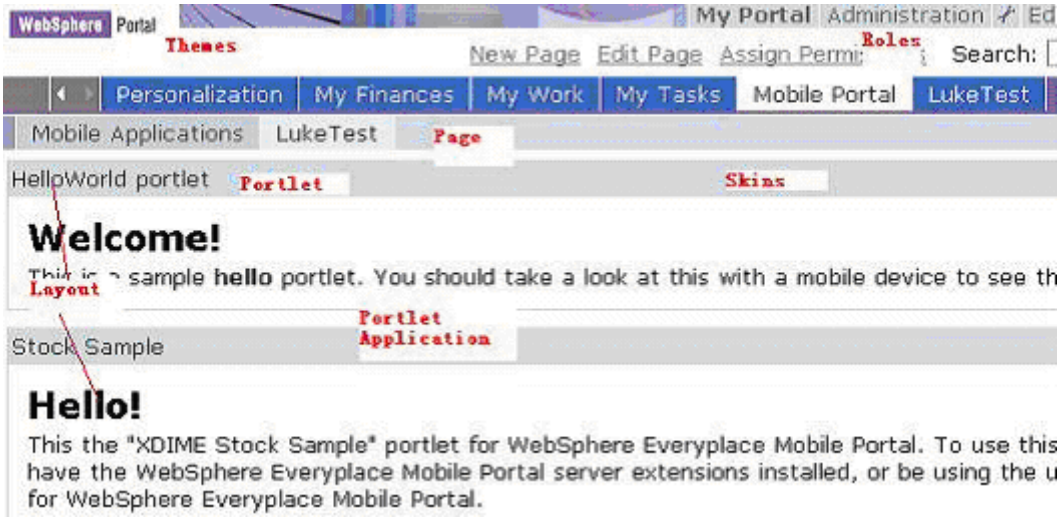
### 5.5 Customize skin of portal

We will introduce how to customize the skin of portal.

The term skin refers to the visual appearance of the area surrounding an individual portlet. Each portlet can have its own skin. The skins that are available for use with a portlet are defined by the portal theme that is associated with the page. The portal administrator or the designer determines the theme for pages and the available skins for the theme. The administrator can permit specified users to change the skins to reflect individual preferences. If you have authority to make changes to a portal page, use the Themes and Skins under Portal User Interface to manage themes.

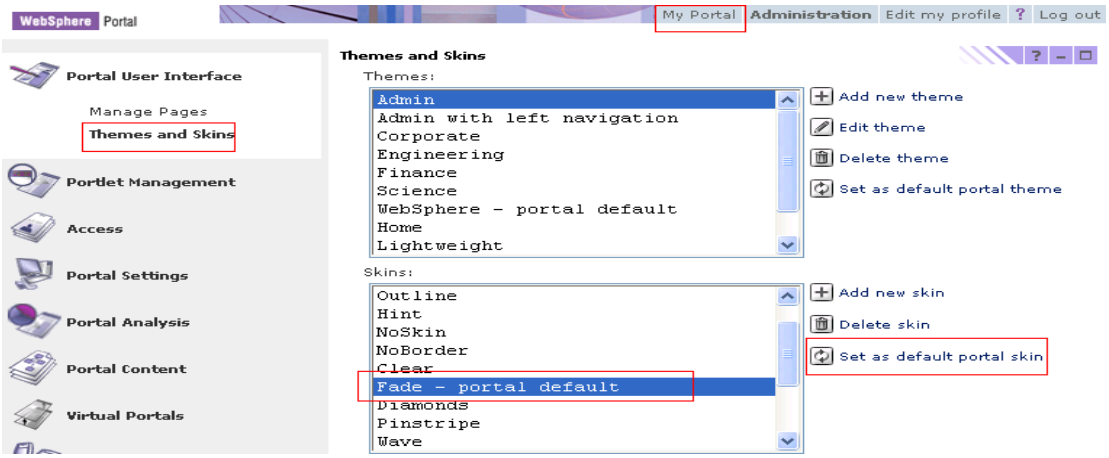
#### *5.5.1 Skin in portal*

The term skin refers to the visual appearance of the area surrounding an individual portlet.




### *5.5.2 Set the default skin in admin console*

We can create a new skin or just modify the existing skins. Here we will use an existing skin. To set the default skin, please login portal and select **Administration**, then select **Portal User Interface** -> **Themes and Skins**. Here we select **Fade** in the skin list, and set it as default portal skin.

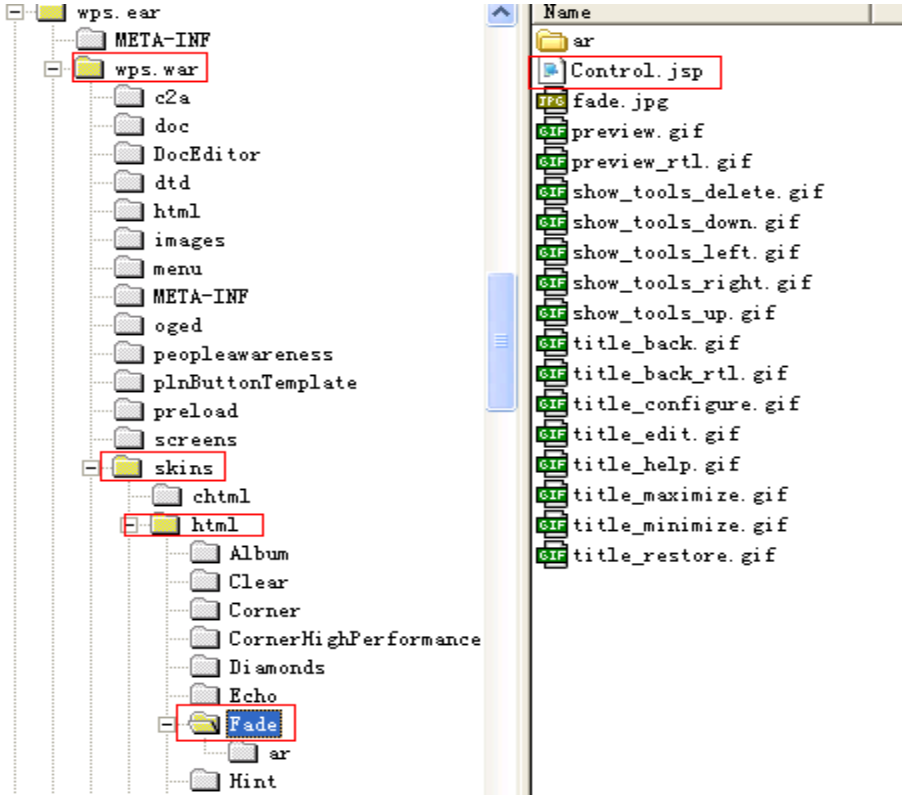



### *5.5.3 Modify the Control.jsp*

The skin contains some jsps, images and stylesheet. In most conditions, we just need to modify the Control.jsp.

/opt/WebSphere/AppServer/installedApps/ogedwemp/wps.ear/wps.war/skins/html/Fade/Control.jsp

We can change the skin to meet our requirements. For example, we can remove the maximum link, minimum link, and configuration link from skin.




*5.5.4 Verify the result*

You can see that there is no tool link in skin now.


For details about how to customize skin, please refer to WP510 Info Center: http://publib.boulder.ibm.com/pvc/wp/510/ent/en/InfoCenter/

## 5.6 Customize theme of portal

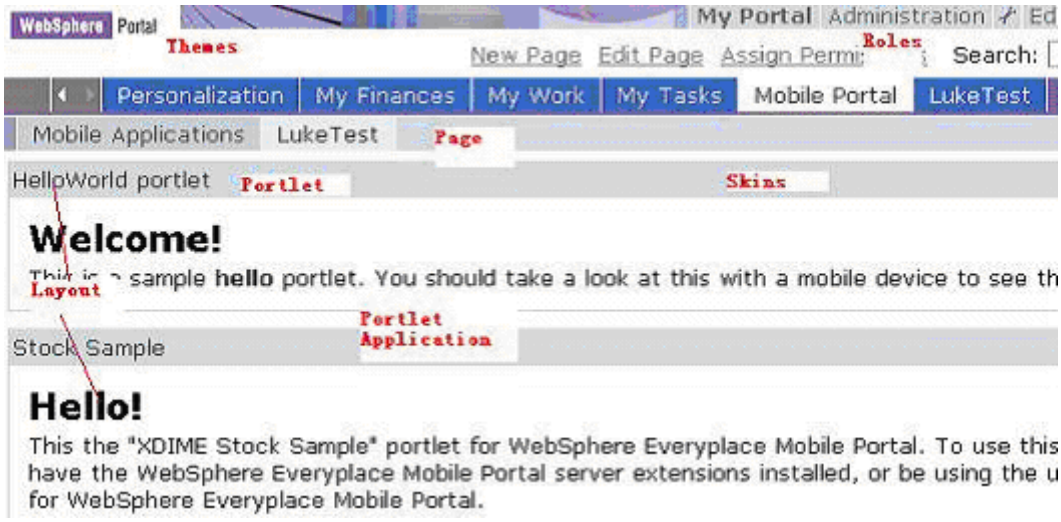
We will introduce how to customize the theme of portal.

Themes represent the overall look and feel of the portal, including colors, images and fonts. There

are several default themes provided with the standard installation of WebSphere Portal. Each page in the portal may have a different theme associated with it, thereby creating the appearance of virtual portals. Use the Themes and Skins under Portal User Interface to manage themes.

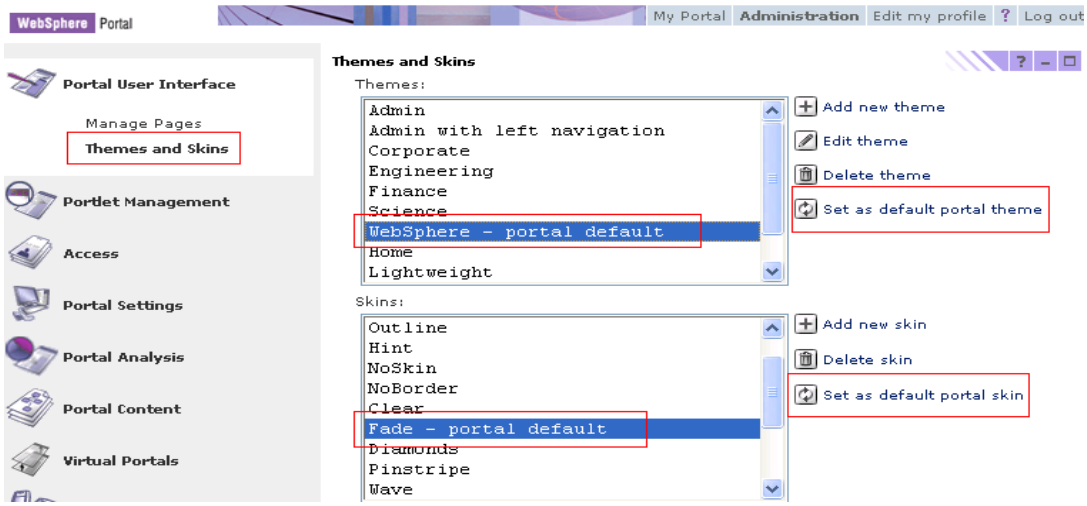
### *5.6.1 Theme in portal*

Themes represent the overall look and feel of the portal.



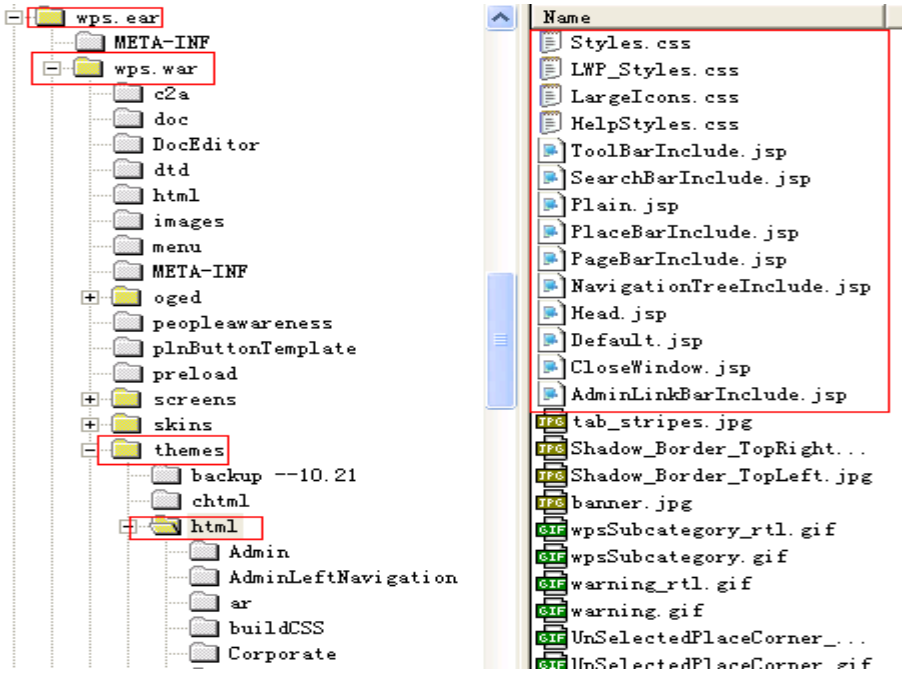

### *5.6.2 Set the default skin in admin console*

We can create a new theme or just modify the existing themes. Here we will use an existing theme. To set the default theme, please login portal and select **Administration**, then select **Portal User Interface** -> **Themes and Skins**. Here we select **WebSphere** in the theme list, and set it as default portal theme.



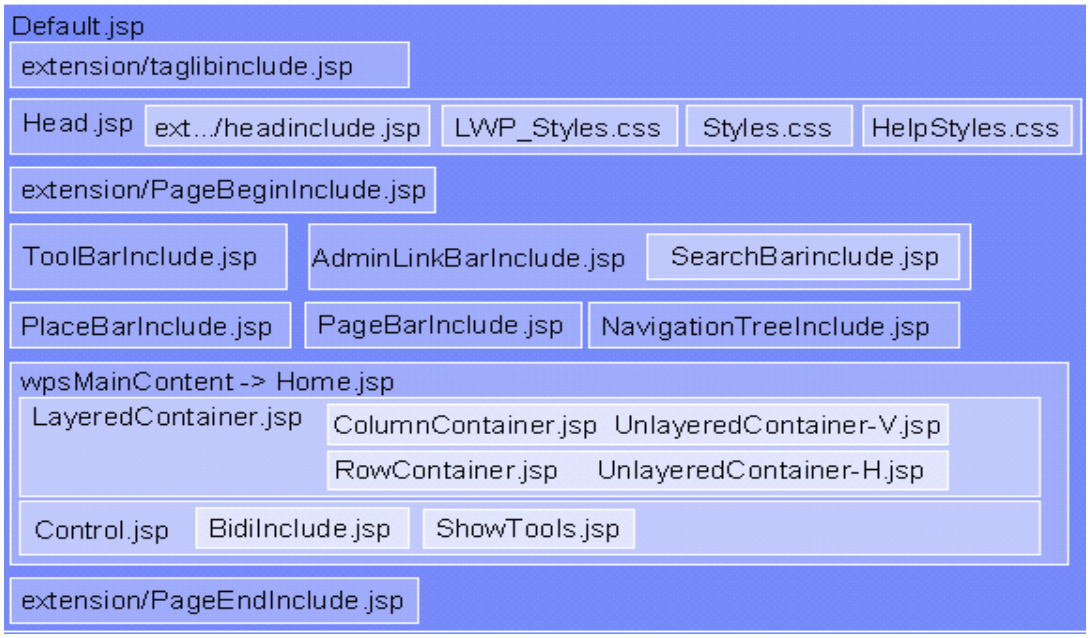

### *5.6.3 Introduction of the theme*

The theme contains some jsps, images and stylesheets. If we want to get a totally different view than the default, we need to create images, and customize jsps and stylesheet. In most conditions, the portal developer just needs to modify the following jsps: Default.jsp, PlaceBarInclude.jsp and PageBarInclude.jsp. And the artwork team can help about images and stylesheet.




*5.6.4 File dependency in theme*

Here is the file dependency in portal.



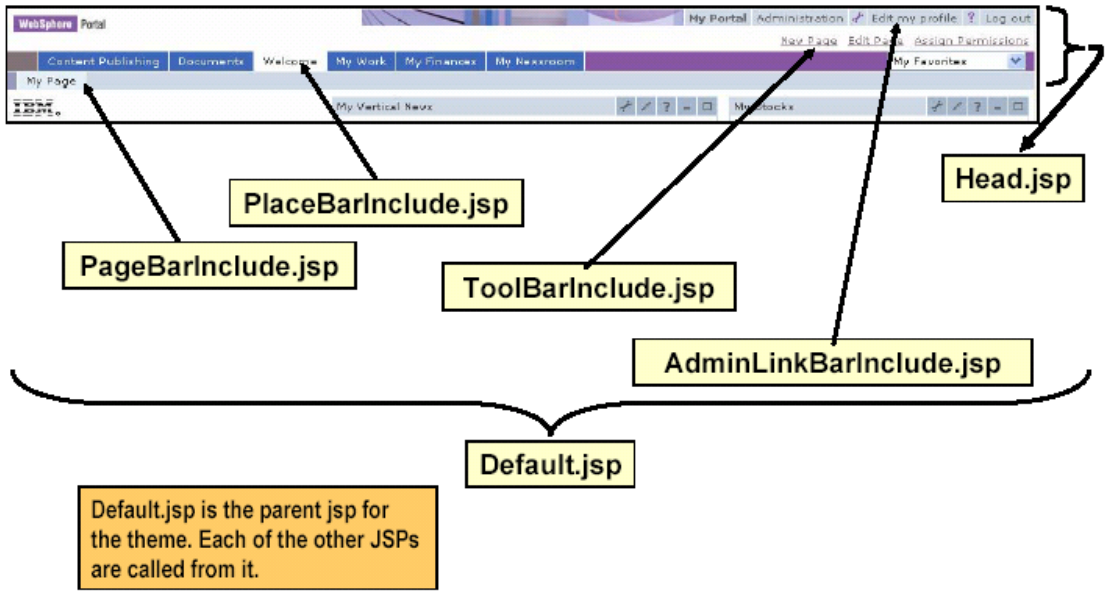

Here you see the basic file structure of a normal theme. At the top level you have the default.jsp. Included in it are the tablibinclude, the Head, the pagebegininclude, and so JSP-files. The Home.jsp is located in the screens subdirectory. The LayeredContainer and Control.jsp can be found in the skins subdirectory.

New with version 5.1 is that there is now a SearchBarInclude, which contains all of the necessary components for the integration of the portal search engine. Also the navigation except if you use

PlaceBar and PageBar is now included in the NavigationTreeInclude.jsp. With version 5.0 it was misplaced in the Skin.

Because of the usage of included jsp when you change a file for example the ToolBarInclude.jsp you need to touch the surrounding Default.jsp for the application server to recompile the whole theme. When you use the Portal Tooling normally when you change an included file and save it then also the surrounding file is automatically saved for you.

Here is the overview in portal website.




### *5.6.5 Default.jsp*

Default.jsp is an important part of your collections of jsps. Think of it as a central location that binds all the other files together. Notice what a simplified Default.jsp file contains.

Here is the short version of Default.jsp

```
<%@ page session="false" buffer="none" %>
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8" %>
<%@ taglib uri="/WEB-INF/tld/portal.tld" prefix="wps" %>
<%@ taglib uri="/WEB-INF/tld/portal-internal.tld" prefix="wps-internal" %>
<%@ include file="./extension/TagLibInclude.jsp" %>
<wps-internal:adminNavHelper/>
<wps:constants/>

<html>
        <%@ include file="./Head.jsp" %>
        <%@ include file="./extension/PageBeginInclude.jsp" %>
<table border="0" cellspacing="0" cellpadding="0">
```

```
        <tr>
          <td>
            <wps:if portletSolo="no">
                <%@ include file="ToolBarInclude.jsp" %>
                <%@ include file="PlaceBarInclude.jsp" %>
                <%@ include file="PageBarInclude.jsp" %>
            </wps:if>
      /td>
        </tr>
    <tr>
          <td>
              <a name="wpsMainContent"></a>
                  <%-- Call the WPS engine command to render the portlets for this page --%>
                  <wps:screenRender/>
          </td>
        </tr>
</table>
        <%@ include file="./extension/PageEndInclude.jsp" %>
</html>
```

### *5.6.6 Taglibs in theme*

The portalserver supplies several Custom Taglibs, which make a creation of a theme much easier. 4 are used in themes and skins. portal.tld, portal-internal.tld, menu.tld and people.tld. You can of course add others in case you need them. The point to add them would be the taglibinclude.jsp.

Here are the taglibs used in Themes and Skins

**portal.tld**

Includes basic tags for use in theme. For example urlFindInTheme to generate an url to point towards an resource/image inside the theme. Another important tag is the if tag. With it you can add conditional parts of html code to a page when for example you are in the login screen. The details, which conditions are supported, can be found in the description of the available tags ( see References )

**portal-internal.tld**

Is used for internal tags. For example adminlinkinfo initializes 2 variables, which contain the current values of the current ContentNode and the ComponentNode. The ContentNode represents the object id of the currently displayed page. The component node contains the object id of the portlet currently selected.

**portlet.tld**

Contains the different styles used in portlets / skins.

**menu.tld / people.tld** ( same as **personTag.tld** )
Integrated through ext../TagLibInclude.jsp

**JSTL definition files**
Added to enable disconnected support of Java Server Pages Standard Tag Library

**c.tld ( core ) / x.tld (xml ) / fmt.tld** ( output formatting )

**Others** available in <wp_root>/shared/app/WEB-INF/tld

### *5.6.7 Navigation in portal*

There are 3 components are used to display navigation.
**PlaceBarInclude.jsp**
Can be used to display first level of navigation
**PageBarInclude.jsp**
Can be used to display second level of navigation
**NavigationTreeInclude.jsp**
Can be used to display all remaining levels.Navigation is moved to this file. Not in skin anymore

### *5.6.8 Stylesheet in theme*

In each of the themes 3 Style sheets are used. The most important is the Styles.css. In it you will find most of the styles needed for theme and skin development. The other 2 Styles sheets are only used in 2 or 3 portlets.

Here are 3 styles used in Portal.
**Styles.css**
See appendix for further information what styles are available.
With RAD a new preview modus is available to use a packaged html-file to preview changes.
**HelpStyles.css**
Used in Common Pim Portlet / Search Center
**LWP_Styles.css**
Used in Common PIM Portlet / People Picker / Portal Doc Mgr

### *5.6.9 Before you can customize theme*

Since Default.jsp is the central file for all other JSPs, You must touch the Default.jsp in order for changes to be picked by the Server. You can touch the file by changing the saved date. Easiest way is to open Default.jsp and insert space, delete it and SAVE file. If it still does not take effect after modifying the Default.jsp, please ensure you enable JSP reloading referring to the section *Enable*

*Automatic JSP Reloadin*.

### *5.6.10 Customize theme*

I will not tell you how to do the coding in the jsps here. One of my experiences is that we should pay more attention to the user locale, stylesheet and bidi.

**Note: Please remain the administration link for the user wpsadmin.**

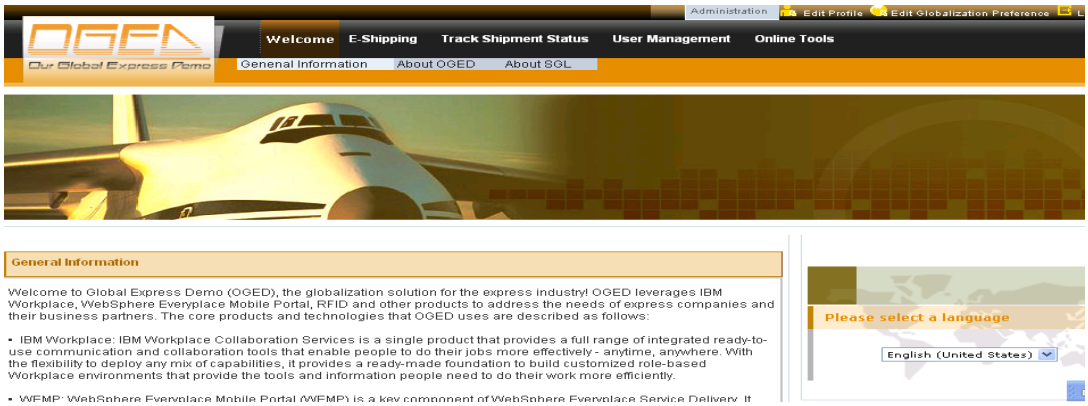
Here are the code fragments in the Default.jsp.

```
<%
        com.ibm.wps.engine.RunData rdata =com.ibm.wps.engine.RunData.from(request);
        com.ibm.wps.puma.User user = rdata.getUser();
        String userLocale = getFullLocale(rdata.getLocale().toString());
        String userid = null;
        if (null != user) {
            userid = user.getUserID();
            user.setPreferredLanguage(userLocale);
        }
%>
<table width="1000" border="0" cellspacing="0" cellpadding="0">
  <tr>
    <td width="209"><img src="/wps/oged/images/<%=userLocale%>/logo.gif" alt='Our Global Express Demo' width="209" height="101"></td>
    <td valign="top">
    <table width="791" align="<%=bidiAlignRight%>" border="0" cellspacing="0" cellpadding="0">
        <tr>
          <td height="22" align="<%=bidiAlignRight%>" class="BackgroundTitle">
          <table height="22" border="0" cellpadding="0" cellspacing="0" class="Backgroundicontitle">

  <tr>
  <wps:if loggedIn="yes" portletSolo="no">
    <td width="30">
    <%-- Administration button --%>
    <wps:urlGeneration contentNode="wps.Administration" portletWindowState="Normal">
    <td class="wpsToolBar" valign="middle" nowrap>
     <a href="<% wpsURL.write(out); %>" class="wpsToolBarLink"><wps:text key="link.administration" bundle="nls.engine"/></a>  
    </td>
        /wps:urlGeneration>
    </td>
  </wps:if>
<wps:if loggedIn="yes">
```

```
    <td >
            <wps:urlGeneration contentNode="6_0_222">
    <a class="wpsToolBarLink" href="<% wpsURL.write(out); %>"
onMouseOver="MM_swapImage('Image5','','/wps/oged/images/icon_EditProfile_1.gif',1)"
onMouseOut="MM_swapImgRestore()"><img
src="/wps/oged/images/<%=userLocale%>/icon_EditProfile.gif" alt='<wps:text
key="PORTLET_EDITPROFILE" bundle="nls.engine"/>' title='<wps:text
key="PORTLET_EDITPROFILE" bundle="nls.engine"/>' name="Image5" width="15"
height="15" border="0" id="Image5"> <font color="#FFFFFF" > <wps:text
key="PORTLET_EDITPROFILE" bundle="nls.engine"/>  </font></a>
    </td>
</wps:if>
  </tr></table>
</td></tr>
</table>
```

### *5.6.11 Verify the result*

You can see the portal website is totally different from the default portal.

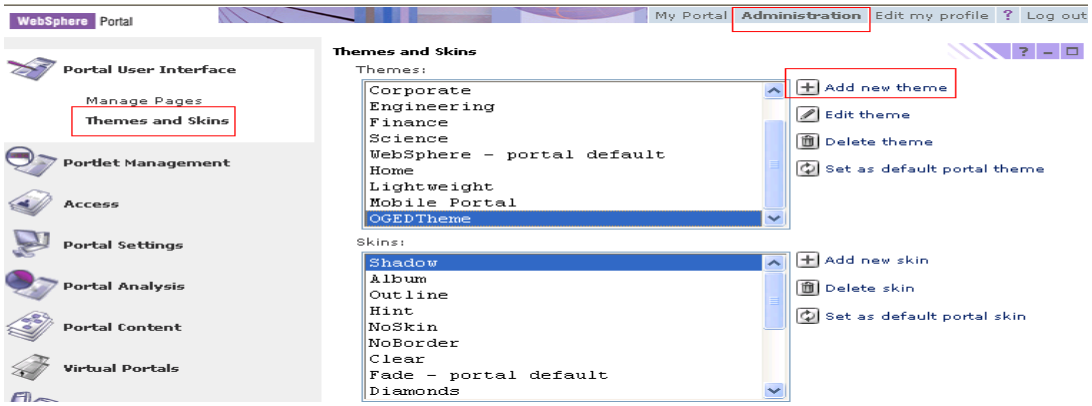



### *5.6.12 Insert a new theme*

To add a theme to the configuration of the Portal you have to copy the theme directory to the <was_root>/installedApps/<node>/wps.ear/wps.war/themes/html directory. Then add the theme in the administrative portlets as following.

1. Click on **Administration** -> **Portal User Interface** -> **Themes and Skins**
2. Click on **Add new theme**
3. Insert a name and the name of the theme-directory for example '**MyTheme'** if it is located in <was_root>/installedApps/<node>/wps.ear/wps.war/themes/html/MyTheme

Once you do that, follow the same direction as you would with existing Themes.




For details about how to customize theme, please refer to WP510 Info Center: http://publib.boulder.ibm.com/pvc/wp/510/ent/en/InfoCenter/

## 5.7 Portlet Caching

### *5.7.1 Portlet Caching*

Caching improves performance by reducing the load on the server. Caching is per user per portlet, a portlet cached for one user will not be used for another user. A JSR 168 portlet must use the portlet.xml to enable caching.The expiration cache is defined in seconds. In the following smaple, the contents of the portlet should be cached for 10 minutes. A value of **zero** indicates that the portlet caching is disabled. A **-1** value will cause the portlet cache to never expire. Any value above 0 will be interpreted as the number of seconds the content should be cached for.

Here is the piece in portlet.xml.

```
<portlet>
        <portlet-name>AboutSGL</portlet-name>
        <display-name>AboutSGL portlet</display-name>
        <display-name xml:lang="zh">AboutSGL portlet</display-name>
        <portlet-class>
            com.ibm.sgl.oged.wemp.portlet.AboutSGLPortlet
        </portlet-class>
        <init-param>
            <name>wps.markup</name>
            <value>html,xdime</value>
        </init-param>
        <expiration-cache>600</expiration-cache>
        <supports>
            <mime-type>text/html</mime-type>
            <portlet-mode>view</portlet-mode>
        </supports>
        <supports>
```

```
            <mime-type>x-application/vnd.xdime+xml</mime-type>
            <portlet-mode>view</portlet-mode>
        </supports>
        <supported-locale>zh</supported-locale>
        <resource-bundle>wempproject.nl.AboutSGL</resource-bundle>
        <portlet-info>
            <title>AboutSGL portlet</title>
        </portlet-info>
</portlet>
```

*5.7.2 Change the expiration time-out value in portlet*

If the expiration-cache is defined in the portlet.xml, the portlet may programmatically change the expiration time. The request property expiration-cache should be used to modify the caching values.

```
RenderResponse.setProperty(PortletResponse.EXPIRATION_CACHE,(new
Integer(600)).toString() );
```

While the portlet content is cached and the portlet is not a target of a request, the cached content will be used during the render phase. If there is a request for the portlet, the cached content will be disregarded and the render method of the portlet will be invoked.

### 5.8 Enable session for anonymous user

We want to maintain the session for anonymous user.

*5.8.1 Find the configuration file*

**\opt\WebSphere\PortalServer\shared\app\config\services\NavigatorService.properties**

*5.8.2 Change public.session to true like the following.*

```
public.session = true
```

### 5.9 Enable Automatic JSP Reloading

To view changes to your theme and skins JSPs without restarting the portal server, you can force the application server to automatically check for new versions of JSPs. While this is ideal for development and testing purposes, automatic JSP reloading should be disabled in a production environment because of performance issues.

If we can enable JSP automatic reloading, then we don't need to restart the application every time after modifying the jsps.

Follow these steps to enable automatic JSP reloading:

*5.9.1 Open the file*

/opt/WebSphere/AppServer/config/cells/ogedwemp/applications/wps.ear/deployments/wps/wps.war/WEB-INF/ibm-web-ext.xmi

*5.9.2 Find the following entry in this file:*

```
<webappext:WebAppExtension xmi:version="2.0" xmlns:xmi="http://www.omg.org/XMI"
        xmlns:webappext="webappext.xmi" xmlns:webapplication="webapplication.xmi"
        xmlns:commonext="commonext.xmi" xmlns:common="common.xmi"
        xmi:id="IBM_WPS_Ext" reloadInterval="3" reloadingEnabled="false"
        fileServingEnabled="true" directoryBrowsingEnabled="false"
```

*5.9.3 Change the value for reloadingEnabled to true.*

*5.9.4 Save the file and restart the portal server.*

After completing these steps, JSPs are automatically reloaded when they are changed. However, to view changes to a JSP that is included by another (parent) JSP, you must also change the parent JSP to indicate that the server must reload it.

## 5.10 How to write a WPS taglib

Our portal pages need to support several languages, such as English, Simplified Chinese, French and Egypt. We want to use taglib to access the localization pack. Here I'd like to tell you how to write a taglib for portal project.

*5.10.1 Write the class for taglib first*

We need to write one class for LoadTag, another for ValueTag.
Here is our sample code .

**LPLoadTag.java**

```
/*
 * Class: <LPLoadTag>
 * Description:<LPLoadTag>
 * Version: <1.0>
 * Author: <Luke>
 * Creation date: <2005-8-9>
 * Department: Shanghai Globalization Laboratory
 * Copyright (c) 2005, International Business Machines Corporation,
 * All rights reserved.
 */
package com.ibm.sgl.oged.util;
```

```
import java.io.IOException;
import java.io.File;
import java.util.Properties;

import javax.servlet.ServletRequest;
import javax.servlet.jsp.JspException;
import javax.servlet.jsp.JspWriter;
import javax.servlet.jsp.tagext.TagSupport;

/**
 *
 * @author tongqg
 * @version 1.0 LP
 */
public class LPLoadTag extends TagSupport {

    /**
     * load property file
     */
    private String _file = null;

    /**
     * @return
     * @exception
     */
    public int doEndTag() throws JspException {

        ServletRequest request = pageContext.getRequest();
        String locale = LPUtil.getUserLocale(request);

          //If the locale is not supported in portal project,set default to en_US.
          //By Luke 2005-10-12
          if ("en_US".equals(locale)) {
          } else if ("zh_CN".equals(locale)) {
          } else if ("fr_FR".equals(locale)) {
          } else if ("ar_EG".equals(locale)) {
          } else {
                locale = "en_US";
          }

        LPUtil lp = new LPUtil(locale, this._file);
        try {
```

```
                if (null == pageContext.getAttribute(LPConstants.PROPERTIESKEY)) {

                    pageContext.setAttribute(LPConstants.PROPERTIESKEY, lp
                            .getProperties());

                } else {
                    ((Properties) pageContext
                            .getAttribute(LPConstants.PROPERTIESKEY)).putAll(lp
                            .getProperties());
                }
            } catch (IOException e) {
                e.printStackTrace();
            }
            return EVAL_PAGE;
    }

    /**
     * @return
     */
    public final String getFile() {
            return _file;
    }

    /**
     * @param string
     */
    public final void setFile(String string) {
            _file = string;
    }

    /**
     * toString
     *
     * @return
     */
    public String toString() {
            return super.toString();
    }
}
```

**LPValueTag.java**

```
/*
 * Class: <LPValueTag>
 * Description:<LPValueTag>
 * Version: <1.0>
 * Author: <Luke>
 * Creation date: <2005-8-9>
 * Department: Shanghai Globalization Laboratory
 * Copyright (c) 2005, International Business Machines Corporation,
 * All rights reserved.
 */
package com.ibm.sgl.oged.util;

import java.io.IOException;
import java.util.Properties;

import javax.servlet.jsp.JspException;
import javax.servlet.jsp.tagext.TagSupport;

/**
 *
 * @author tongqg
 * @version 1.0
 *
 */
public class LPValueTag extends TagSupport {

	/**
	 * The key in lp properties
	 */
	private String _key = null;

	/**
	 * @return int
	 * @throws JspException
	 * @exception
	 */
	public int doEndTag() throws JspException {

		Properties p = (Properties) pageContext
				.getAttribute(LPConstants.PROPERTIESKEY);
		try {
			if (null != _key) {
```

```
                    String value = p.getProperty(_key);
                    if (null != value) {
                        //Print the value
                          pageContext.getOut().print(value);
                    }else{
                        //Print the default value if the key does not exit.
                        pageContext.getOut().print("The key " + _key + " is not found in lp file.");
//$NON-NLS-1$  $NON-NLS-2$
                    }
                }
            } catch (IOException e) {
                e.printStackTrace();
            }
            return EVAL_PAGE;
        }

        /**
         * @return key
         */
        public final String getKey() {
            return _key;
        }

        /**
         * @param string
         */
        public final void setKey(String string) {
            _key = string;
        }

        /**
         * class name
         *
         * @return string
         */
        public String toString() {
            return LPValueTag.class.getName();
        }

}
```

### *5.10.2 Write the taglib configuration file lp.tld*

We will reference to class for lp handling in the lp.tld.

Then place the lp.tld in the portal directory **<PortalProject>\WebContent\WEB-INF\tld**

Note that we can define the tag name in this file, such as load and value.

**lp.tld**

```
<taglib>
    <tlibversion>5.1</tlibversion>
    <jspversion>1.2</jspversion>
    <shortname>LP Taglib</shortname>

    <tag>
        <name>load</name>
        <tagclass>com.ibm.sgl.oged.util.LPLoadTag</tagclass>
        <attribute>
            <name>file</name>
            <required>true</required>
            <rtexprvalue>true</rtexprvalue>
        </attribute>
    </tag>
    <tag>
        <name>value</name>
        <tagclass>com.ibm.sgl.oged.util.LPValueTag</tagclass>
        <attribute>
            <name>key</name>
            <required>true</required>
            <rtexprvalue>true</rtexprvalue>
        </attribute>
    </tag>
</taglib>
```

### *5.10.3 Use the taglib in portal jsp file*

Write the following code in portal jsp page. We should use the tag name defined in the lp.tld file, such as lp:load and lp:value.

```
<!--Import the taglib -->
<%@taglib uri="/WEB-INF/tld/lp.tld" prefix="lp"%>

<!--Load the propertiy file -->
<lp:load file="user.properties" />
```

```
<!--Get the translated string from lp -->
<lp:value key="ERROR_PASSWORD " />
```

Here is the piece in the localization file.

```
ERROR_PASSWORD =The password is invalid!
```

### 5.11 Session in portlet

We need to use session to exchange the data between the jsp pages and portlet.

We can use get the session object from request. Please note that JSR168 portlet is different from IBM API portlet. You can see the difference in the code fragments.

#### *5.11.1 For IBM API portlet*

**JSP sample**

```
<%@ taglib uri="/WEB-INF/tld/portlet.tld" prefix="portletAPI" %>
<%@ page import="java.util.*, com.ibm.gcl.gcm.databeans.*, com.ibm.gcl.gcm.portlets.*, com.ibm.gcl.gcm.util.*, java.net.*, com.ibm.gcl.gcm.constants.FederalSearchParameters"%>

<portletAPI:init />
<%
    CaseDetails casedetails = (CaseDetails)portletRequest.getPortletSession().getAttribute(FederalSearchParameters.CASEDETAILS);
  %>
```

**Portlet sample**

```
public class CriminalSearchPortlet
      extends PortletAdapter
  implements ActionListener {

        public void doView (PortletRequest request, PortletResponse response)
            throws PortletException, IOException {
            PortletSession session = request.getPortletSession();
            String JSPFile = (String) session.getAttribute(DocumentParameters.JSPPAGENAME);
            String pageID = (String) session.getAttribute(DocumentParameters.PAGEID);

            if (null == JSPFile) {
                // First time enters into the portlet
```

```
                JSPFile = DocumentParameters.PAGE_CRIMINAL_SEARCH;
                pageID = DocumentParameters.FIRSTPAGE;
                session.setAttribute(DocumentParameters.PAGEID, pageID);
                session.setAttribute(DocumentParameters.ERRMESSAGE, null);
                session.setAttribute(DocumentParameters.OLDCRITERIA,                    new
SearchCriteriaHistory());
            }

            request.setAttribute(DocumentParameters.PAGEID, pageID);
            request.setAttribute(DocumentParameters.JSPPAGENAME, JSPFile);

            setActionURIs(request, response);

            getPortletConfig().getContext().include(JSPFile, request, response);
        }

public void actionPerformed (ActionEvent event) throws PortletException {

            String actionString = event.getActionString();
            PortletRequest request = event.getRequest();

            PortletSession session = request.getPortletSession();
            CriminalSearchActionBean actionBean = new CriminalSearchActionBean();
            actionBean.setSession(session);
            actionBean.setRequest(request);
            actionBean.setContext(getPortletConfig().getContext());

            try {
                Class actionClass = actionBean.getClass();
                Method method =
                    actionClass.getMethod(
                        DocumentParameters.ACIONT_PREFIX + actionString,
                        null);
                method.invoke(actionBean, null);
            } catch (Exception e) {
                e.printStackTrace();
            }
        }

}
```

### *5.11.2 For JSR168 portlet*

**JSP sample**

```
<%@ taglib uri="http://java.sun.com/portlet" prefix="portlet" %>
<portlet:defineObjects />
<%
String locale = com.ibm.sgl.oged.util.LPUtil.getFullLocale(request.getLocale().toString());
renderRequest.getPortletSession().setAttribute("userLocale",locale);
String name = “”;
if(renderRequest.getPortletSession().getAttribute("name")!=null)
      name = (String)renderRequest.getPortletSession().getAttribute("name");
%>
```

**Portlet sample**

```
public void processAction(ActionRequest request, ActionResponse response)
                  throws PortletException, java.io.IOException {

if (null != request.getParameter(”BUTTON_BACK”)) {
                  request.getPortletSession().setAttribute(“JSPNAME”,”InitialShipment.jsp”);
}

public void doView(RenderRequest request, RenderResponse response)
                  throws PortletException, IOException {

            String JSP_NAME = (String) request.getPortletSession().getAttribute(
                        InitiateShipmentConstansts.JSPNAME_SESSION);
          request.getPortletSession().removeAttribute(
                        InitiateShipmentConstansts.JSPNAME_SESSION);
            if (null == JSP_NAME) {
        InitiateShipmentActionBean.clearSession(request.getPortletSession());
                  JSP_NAME = InitiateShipmentConstansts.INIATE_SHIPMENT;
            }
            // Invoke the JSP to render
            PortletRequestDispatcher rd = getPortletContext().getRequestDispatcher(
                        getJspFilePath(request, JSP_NAME));
            rd.include(request, response);
}
```

## 5.12 **JavaScript in portlet**

In the portlet jsps, we need to use JavaScript to implement the following functions.

1. The event onchange and onclick, for example, refresh the page upon the user selection

without submitting.

2. Change the hidden field value and submit the form upon the user interactions. For example, select an item in a list.
3. Popup calendar
4. Pull down calendar

Here are some samples for your references.

### *5.12.1 The event onchange and onclick*

**JavaScript fragments**

```
<script language="JavaScript">
function changeLangUrl(){
	var language = document.selLangForm.Language.value;
		var submitURL;

	if(language=="en_US")
	{
		submitURL = "<wps:url command="ChangeLanguage"><wps:urlParam name="locale" value="en_US"/></wps:url>";	}
	else if(language=="zh_CN")
	{
		submitURL = "<wps:url command="ChangeLanguage"><wps:urlParam name="locale" value="zh_CN"/></wps:url>";
	}
	else if(language=="ar_EG")
	{
		submitURL = "<wps:url command="ChangeLanguage"><wps:urlParam name="locale" value="ar_EG"/></wps:url>";
	}
	else if(language=="fr_FR")
	{
		submitURL = "<wps:url command="ChangeLanguage"><wps:urlParam name="locale" value="fr_FR"/></wps:url>";
	}

 <wps:if loggedIn="yes">
	submitURL=submitURL.replace("portal","myportal");
 </wps:if>

		window.location.href = submitURL;
}
</script>
```

**Code fragments**

```
<%@ page import="java.util.*,javax.portlet.*,com.ibm.sgl.oged.wemp.portlet.*,
com.ibm.sgl.oged.db.databean.G11NPreferenceBean, com.ibm.sgl.oged.constant.UserConstansts,
com.ibm.sgl.oged.util.CodeSystem, com.ibm.sgl.gbo.core.operation.FeatureManager,
com.ibm.sgl.gbo.localelist.IGLocaleList,
com.ibm.sgl.gbo.core.preference.base.* " %>

<form name="selLangForm">
<select onchange="changeLangUrl();" name="Language" >
<%
    String localeList[]    = {"en_US", "zh_CN", "fr_FR", "ar_EG"};
    String languageList[] = {"en", "zh", "fr", "ar"};
    String countryList[] = {"US", "CN", "FR", "EG"};

    BasePreferenceFactory bpf = BasePreferenceFactory.getFactory();
    BasePreference bp = bpf.getBasePreference("languageList");

    FeatureManager fm = FeatureManager.getInstance();
    for(int i = 0; i<4; i++) {
        Locale locale = new Locale(languageList[i], countryList[i]);
        //System.out.println(locale);
        bp.setLanguage(locale);
        //System.out.println(localeList[i] + locale);

        IGLocaleList gll = (IGLocaleList) fm.getFeature(IGLocaleList.class, "languageList");

        String langList3 = gll.getLocaleName(locale);
        //System.out.println(langList3);
        String optionValue = "";
        if(CodeSystem.LOCALE_EN.equals(localeList[i])) {
            optionValue = CodeSystem.LANGUAGE_EN;
        }
        if(CodeSystem.LOCALE_CN.equals(localeList[i])) {
            optionValue = CodeSystem.LANGUAGE_CN;
        }
        if(CodeSystem.LOCALE_FR.equals(localeList[i])) {
            optionValue = CodeSystem.LANGUAGE_FR;
        }
        if(CodeSystem.LOCALE_AR.equals(localeList[i])) {
            optionValue = CodeSystem.LANGUAGE_AR;
            dir = "rtl";
            langList3                                                    =
```

```
replaceString(replaceString(langList3,")","&#8207;)" ),"(","&#8207;(");
            }
      %>
  <option dir="<%=dir%>" value=<%=optionValue %>
<%if(optionValue.equals(userLocale)) {out.print("selected");} %>>
  <div align="center" dir=<%=dir%> >
<%=replaceString(replaceString(langList3,")",bidiForceString+")" ),"(",bidiForceString+ "(")%>
</div ></option>
<%
} %>
/select>
```

*5.12.2 Change the hidden field value and submit the form upon the user interactions.*

**JavaScript fragments**

```
<script language="JavaScript">
function PrevPage(){
          this.document.prevpage.submit();

}
function NextPage(){
   this.document.nextpage.submit();

}
</script>
```

**Code fragments**

```
<tr>
<td colspan="4" class="td6<%=dir%>">
<%
if (pageID.equals(TrackConstansts.FIRST_PAGE)) {
%>
                <lp:value key="TABLE_PREVIOUS"/>
<%}else{%>
                        <a          href="#"          onclick="PrevPage()">          <lp:value
key="TABLE_PREVIOUS"/></a>
<%}%>

[<%=pageID%>/<%=TrackByAdvancedPortlet.maxPageNum%>]
<%
if (pageID.equals((new Integer(TrackByAdvancedPortlet.maxPageNum)).toString())){
```

```
%>
                    <lp:value key="TABLE_NEXT"/>
<%}else{%>
                    <a href="#" onclick="NextPage()"> <lp:value key="TABLE_NEXT"/></a>
<%} %>
    </td>
</tr>
```

```
<form name="nextpage" action=<portlet:actionURL/> method="POST">
<input name="<%=TrackConstansts.NEXT%>" type="hidden" value="<lp:value
key="BUTTON_RETURN" />"> </form>
<form name="prevpage" action=<portlet:actionURL/> method="POST">
<input name="<%=TrackConstansts.PRE %>" type="hidden" value="<lp:value
key="BUTTON_RETURN" />">
</form>
```

### *5.12.3 Popup Calendar*

**JavaScript fragments**

For details, please refer to GBO team: PopupCalendar.jsp and PopupCalendar.js

**Code fragments**

```
        <tr>
          <td class="td3<%=dir%>"><lp:value key="TABLE_BEGINDATE" /> </td>
           <td class="td6">
              <input type="text" id="date1_Show" name="begindateshow"
readonly="readonly" value=<%=begindate%>>
                <input type="hidden" id="date1" name="begindate"
<%if(0!=begindatehidden.length())%> value=<%=begindatehidden%>>
                <input type="hidden" id="date1_Unformat" >
                <img src="/wps/oged/images/calendar.gif" alt="<lp_alt:value
key="CALENDAR"/>"
onclick="window.open('<%=renderRequest.getContextPath()%>/html/popupcalendar/PopupCalen
dar.jsp?locale=<%=poplocale%>','date1','height=200,width=210,scrollbars=no,resizable=yes,statu
s=no,menu=no,toolbar=no')">
        </td>
          <tr>
```

### *5.12.4 Pull down calendar*

**JavaScript fragments**

```
<SCRIPT language="JavaScript">
//If there are more than one PulldownGBO in one page.
//You should repacle the name of year, month, date with new one.
var monthdays = new Array(12);

monthdays[0] = 31;
monthdays[1] = 28;
monthdays[2] = 31;
monthdays[3] = 30;
monthdays[4] = 31;
monthdays[5] = 30;
monthdays[6] = 31;
monthdays[7] = 31;
monthdays[8] = 30;
monthdays[9] = 31;
monthdays[10] = 30;
monthdays[11] = 31;

var back = new Array(4);


function update_days(f)

{
//      alert(f.months.parent());
        compute_days(f);
        var lengthDeleted = 31 - monthdays[f.month1.selectedIndex];

        if(back[0] == null){
                for(x=0;x<back.length ;x++) {
                        back[x] = f.date1[30 - x];
                }
        }

        if(null != back[0]){
                for(x=0;x<back.length ;x++) {
                        f.date1[30 - x] = back[x];
                }
        }
```

```
        for(x=0;x<lengthDeleted ;x++) {
                f.date1[30 - x] = null;
        }
}

function compute_days(f){
        if(!isNaN(f.year1.value) && f.year1.value.length > 0){
                var year=parseInt(f.year1.value);
                var ret_val = 0;
                if (year % 4 == 0 && year % 100 != 0 )
                        ret_val=1;

                if (year%4 == 0 && year % 400 == 0)
                        ret_val=1;

                if(ret_val == 1){
                        monthdays[1] = 29;
                }else{
                        monthdays[1] = 28;
                }
                        }

}
</SCRIPT>
```

**Code fragments**

```
<%@                                                                    page
import="java.util.*,java.text.*,java.sql.Timestamp,javax.portlet.*,com.ibm.sgl.oged.wemp.portlet.
*,com.ibm.gcl.gbo.calendar.*,com.ibm.gcl.gbo.template.*" %>

<%
CalendarGBO calendar = CalendarGBO.getInstance(locale);
String dateorder = calendar.getDateOrder();
for(int j = 0; j < dateorder.length(); j ++){
if(dateorder.charAt(j)=='m'){
%><td class="td2<%=dir%>">
<select name="month1" size="1" onchange="update_days(this.form)">
<%
        String[] monthvalue = {"1","2","3","4","5","6","7","8","9","10","11","12"};
        String[] monthcaption = calendar.getFullMonthList();
        for(int i = 0; i < monthvalue.length; i ++)     {
                if(monthvalue[i].equals(month)){
%>
```

```
                <option value="<%=monthvalue[i]%>"
selected><%=monthcaption[i]%></option>
<%          }else{ %>
                <option value="<%=monthvalue[i]%>"><%=monthcaption[i]%></option>
<%          }
      }
%>
</select>
</td>
<%
}

else if(dateorder.charAt(j)=='d'){
%><td class="td2<%=dir%>">
<select name="date1" size="1">
<%
String[] dayvalue =
{"1","2","3","4","5","6","7","8","9","10","11","12","13","14","15","16","17","18","19","20","21",
"22","23","24","25","26","27","28","29","30","31"};
String[] daycaption = calendar.getFullDayList();
for(int i = 0; i < dayvalue.length; i ++)
{
      if(dayvalue[i].equals(date)){
%>
                <option value="<%=dayvalue[i]%>" selected><%=daycaption[i]%></option>
<%          }else{ %>
                <option value="<%=dayvalue[i]%>"><%=daycaption[i]%></option>
<%          }
      }
%>
</select>
</td>
<%
}

else if(dateorder.charAt(j)=='y'){
%><td class="td2<%=dir%>">
<select name="year1" size="1" onchange="update_days(this.form)">
<%
String yearlist[] =
{"2000","2001","2002","2003","2004","2005","2006","2007","2008","2009","2010"};
for(int i = 0; i < yearlist.length; i ++){
      if(yearlist[i].equals(year)){
```

```
%>
                <option value="<%=yearlist[i]%>" selected><%=yearlist[i]%></option>
<%          }else{ %>
                <option value="<%=yearlist[i]%>"><%=yearlist[i]%></option>
<%          }
      }
%>
</select>
</td>
<%
}
}%>


<td class="td2<%=dir%>">
<select name="time" size="1">
<%
String[]                                                    times                                                    =
{"00:00","01:00","02:00","03:00","04:00","05:00","06:00","07:00","08:00","09:00","10:00","11:0
0","12:00","13:00","14:00","15:00","16:00","17:00","18:00","19:00","20:00","21:00","22:00","23
:00"};

      for(int i = 0; i < times.length; i ++){
            if(times[i].equals(time)) {
 %>
      <option value="<%=times[i]%>" selected><%=times[i]%></option>
<%  }
      else{
%>          <option value="<%=times[i]%>" ><%=times[i]%></option>

<%  }
      }
   %>
</select>
</td>
```

### 5.13 Classpath issues in portal project

Classpath is very important to the success of a project. If it's mixed up, it will be very difficult for us to find the problem.

We need to make classpath very clear in the portal project. For example, what's the use of this jar

file? What's the version of the GBO.jar?

To ensure the portal applications behavior correct, we must follow some rules.

### *5.13.1 Classpath in portal project*

Assign a person to be responsible for updating the classpath file in portal project and portal environment. Anyone else cannot check out and modify the classpath file.

### *5.13.2 Jar file in portal project*

1. Please don NOT copy the unused jar files to portal project.
2. Make sure each jar file is required and on the right version.

### *5.13.3 Jar file in portal server*

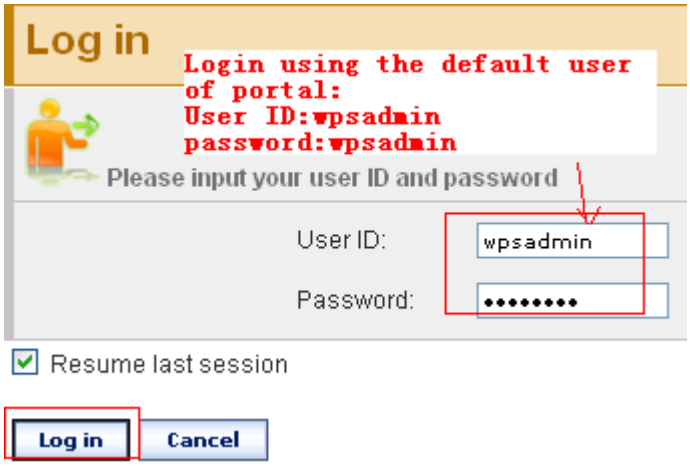
1. Please do NOT copy the unused jar files to portal project.
2. Make sure each jar file is required and on the right version.
3. **Please don NOT copy the same jar file to several locations, such as portal project, portal server, and application server. It will cause unexpected problems.**

# 6. Portal administration

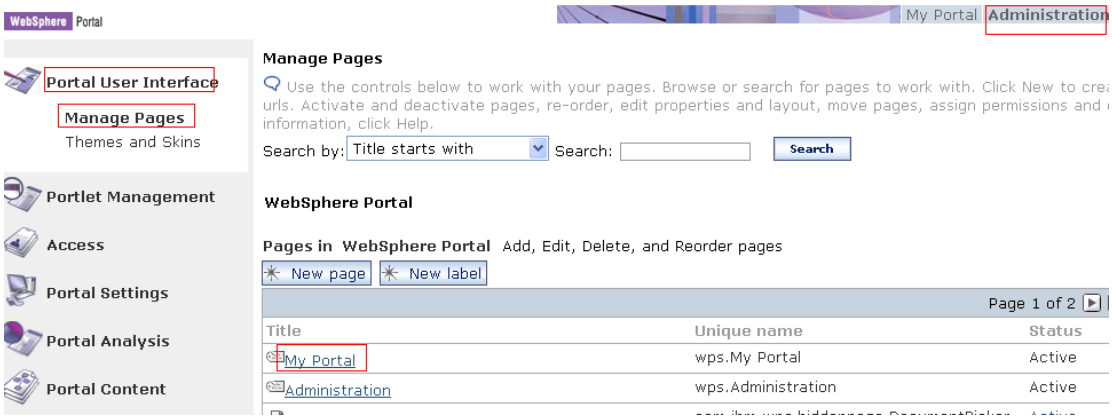
Here I will give you step-by-step instructions about how to do the administration for portal.

## 6.1 Login portal

Login using the default user of portal.
User ID: **wpsadmin**
Password: **wpsadmin**

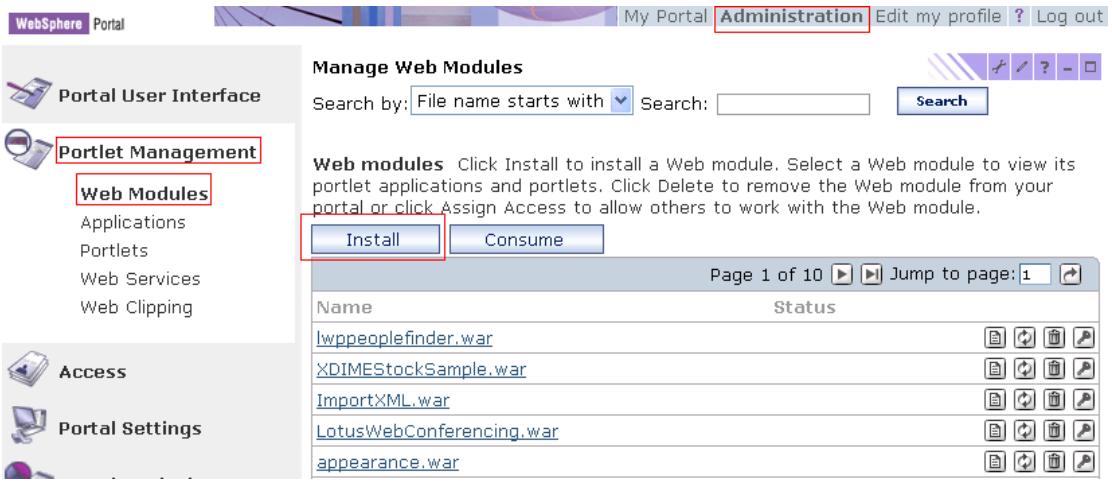



After logging in, you can click the **Administration** link to the portal admin console.

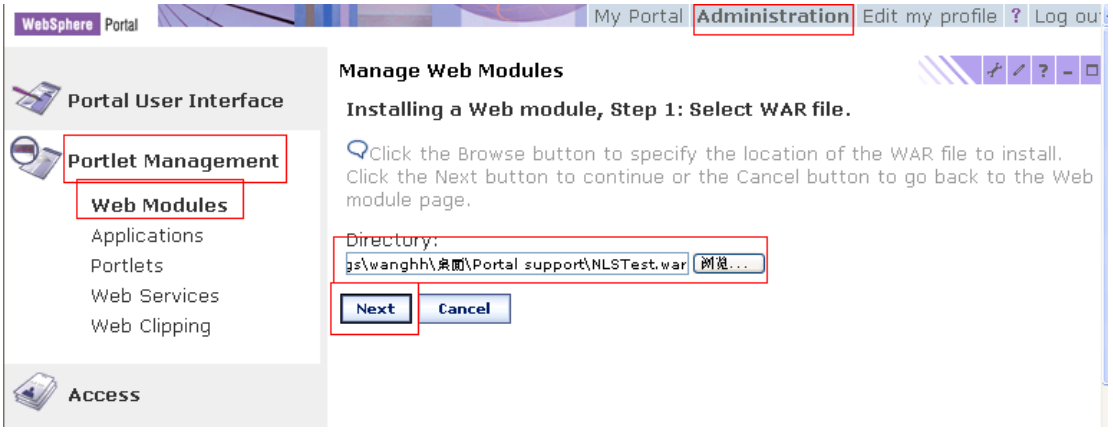



## 6.2 Deploy portal applications

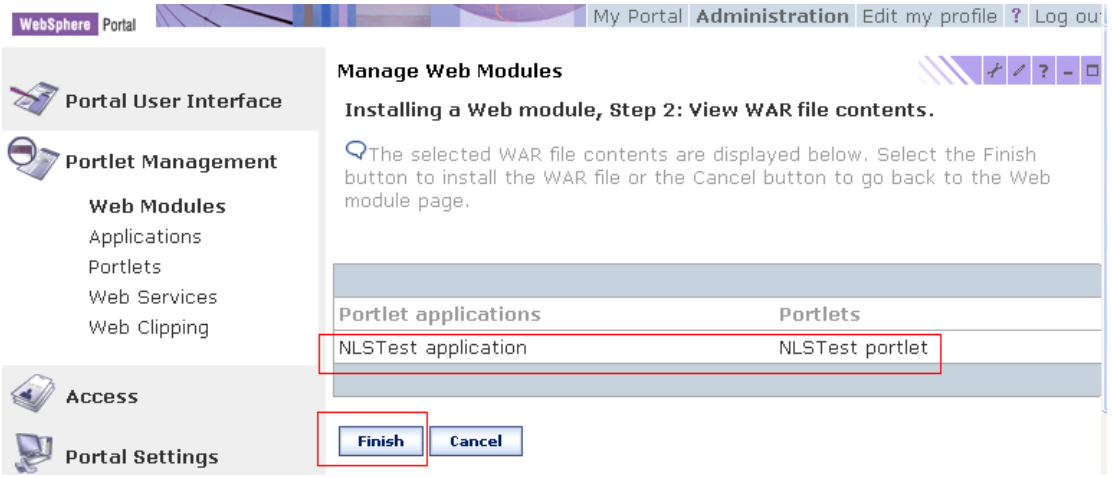
In the admin console, click **Portlet Management** -> **Web Modules** in the left navigation tree. Then you can click **Install** to install a portal application.




Browser or input the full path of the war file to be installed. Click **Next.**



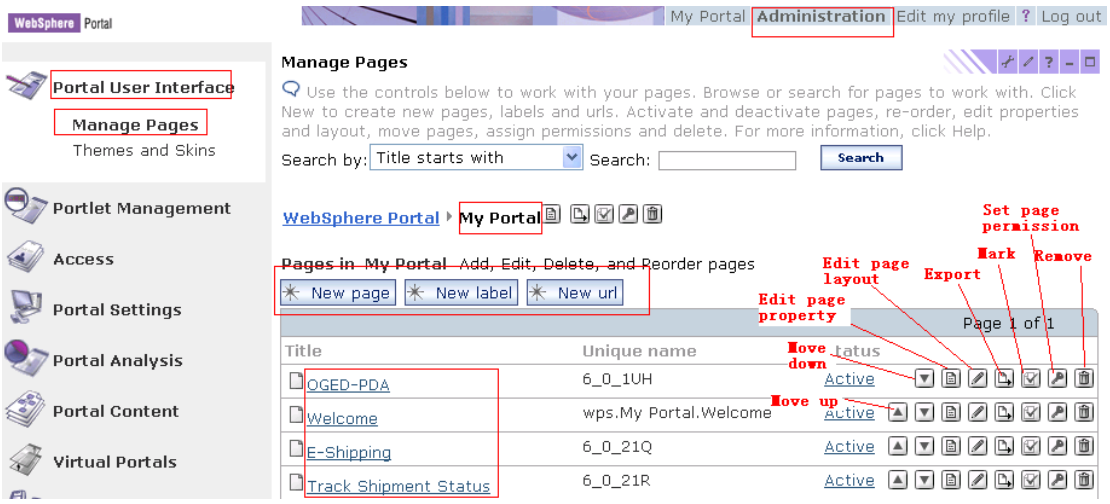

In this step, you can view the war file contents and click **Finish**.




### 6.3 Manage pages

To manage the pages, click **Portal User Interface -> Manage Pages.**

In this page, you can navigate the page tree by clicking the page title. You can click the tool icon to manage the pages. For example, set the page order; edit page layout and set page permission.




### 6.3.1 *Create a new page*

To create a new page, just click **New Page.**

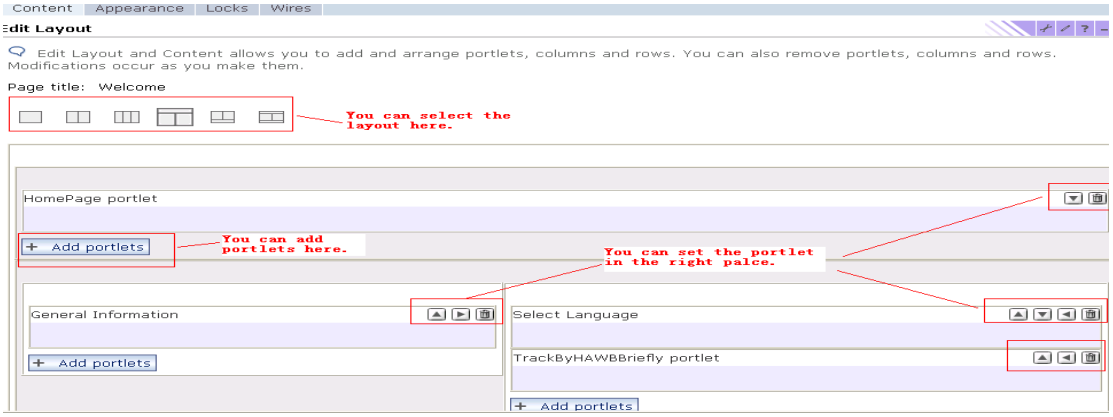
In the following page, you can set the page title, layout, supported markups and click OK to create a new page.


You can see that the new page has been created successfully.


### 6.3.2 *Edit page layout*

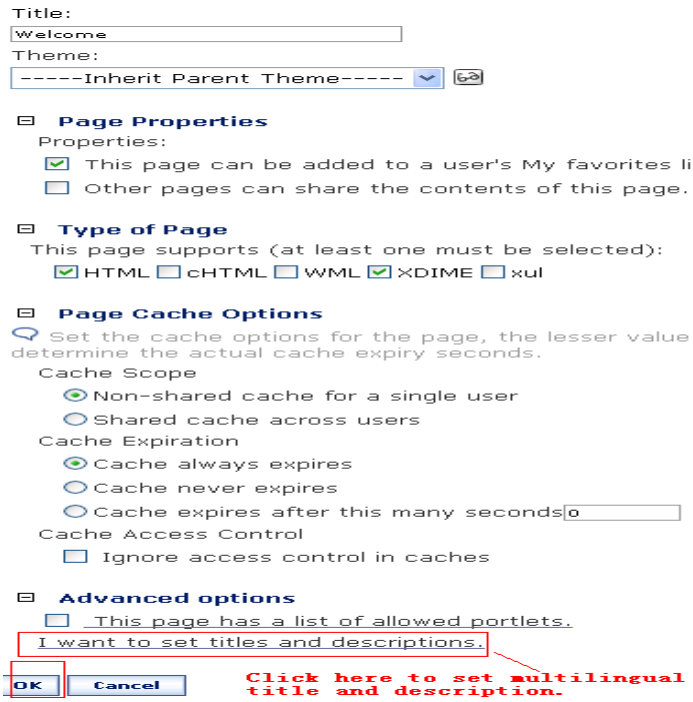
If you want to edit the page layout, please click the icon **Edit page layout**.

You can set the portlet in your wanted place as following.

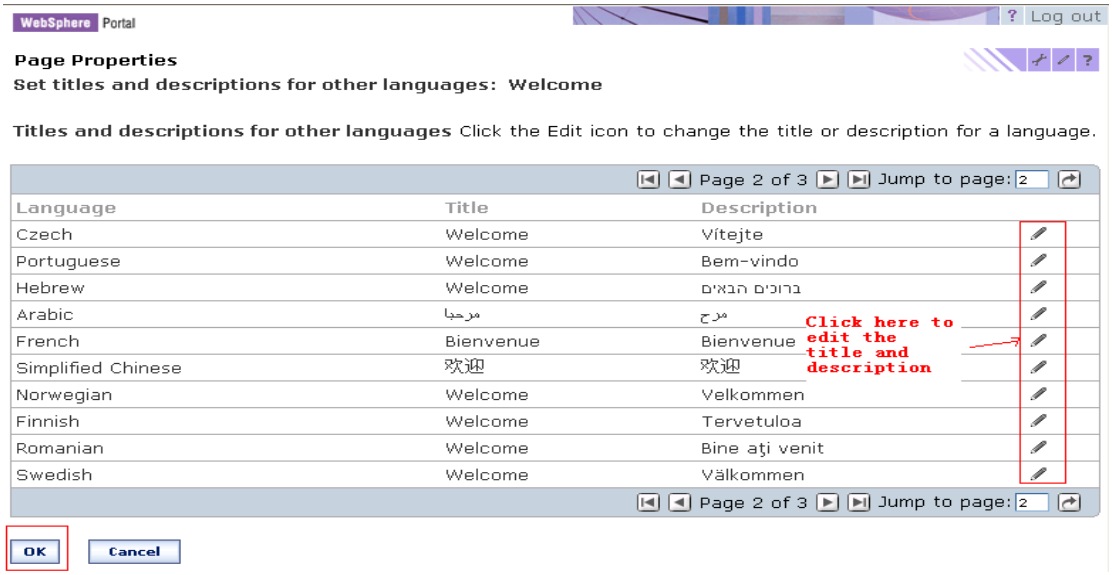



### 6.3.3 *Edit page title and description*

If you want to edit the page title and description, please click the icon **Edit page property**. Then click “I want to set titles and description”. You can set the portlet in your wanted place as following.




Click the edit icon to edit the title and description.



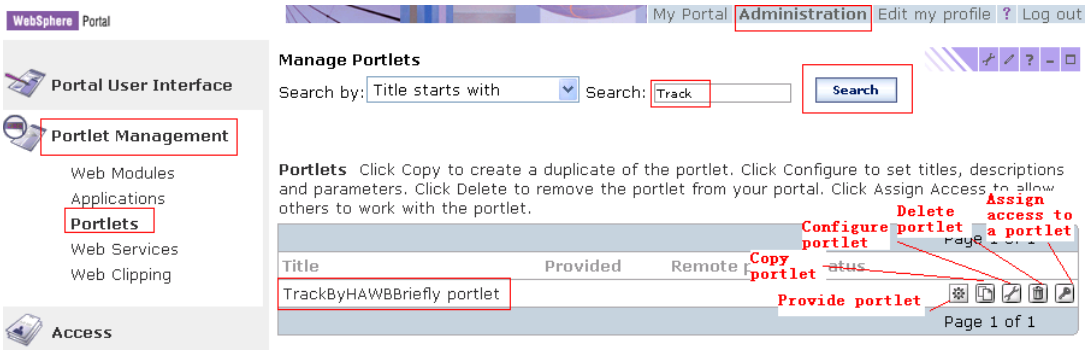

Input the title and description here and click **OK**.


### 6.4 Manage Portlets

To manage portlets, you can click **Portlet Management** -> **Portlets.**

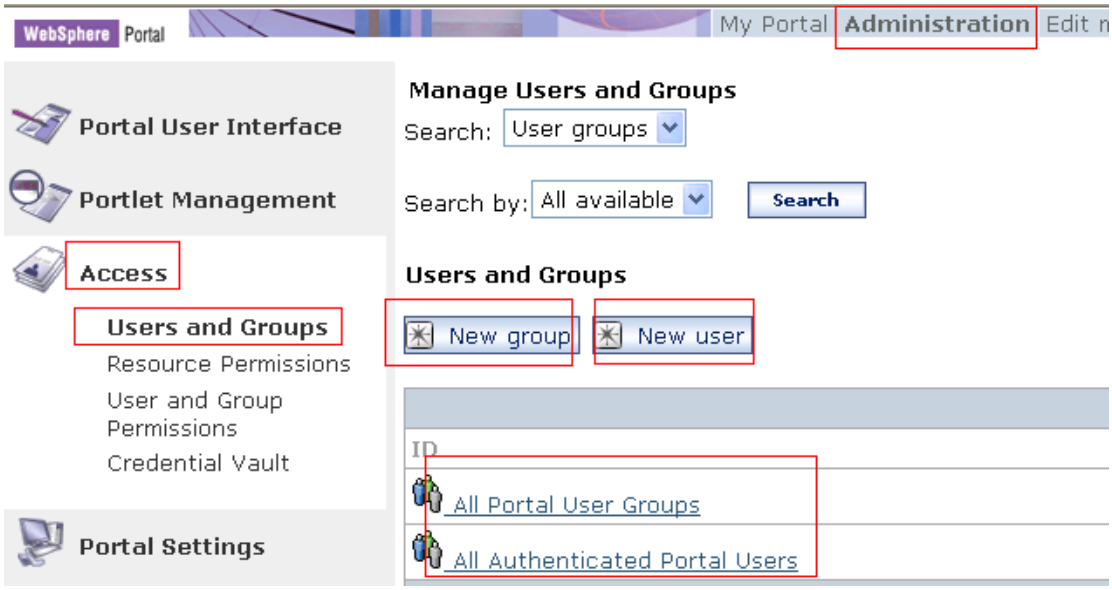
In this page, you can manage portlets, for example, configure portlet, assign access to a portlet and etc.

Also, you can click **Search** to search a portlet.

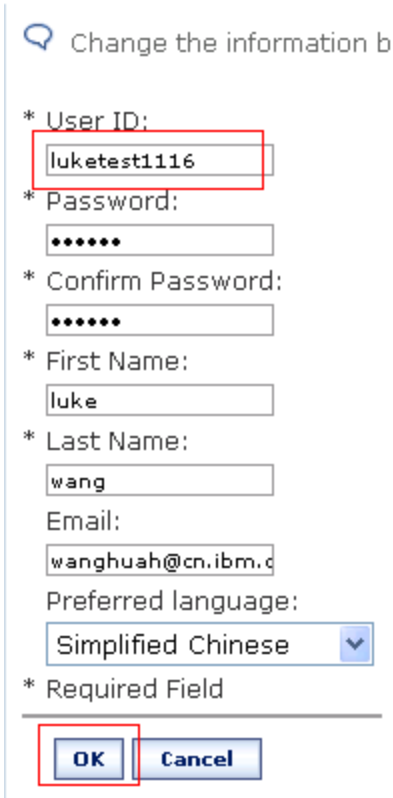



### 6.5 Manage Users and Groups

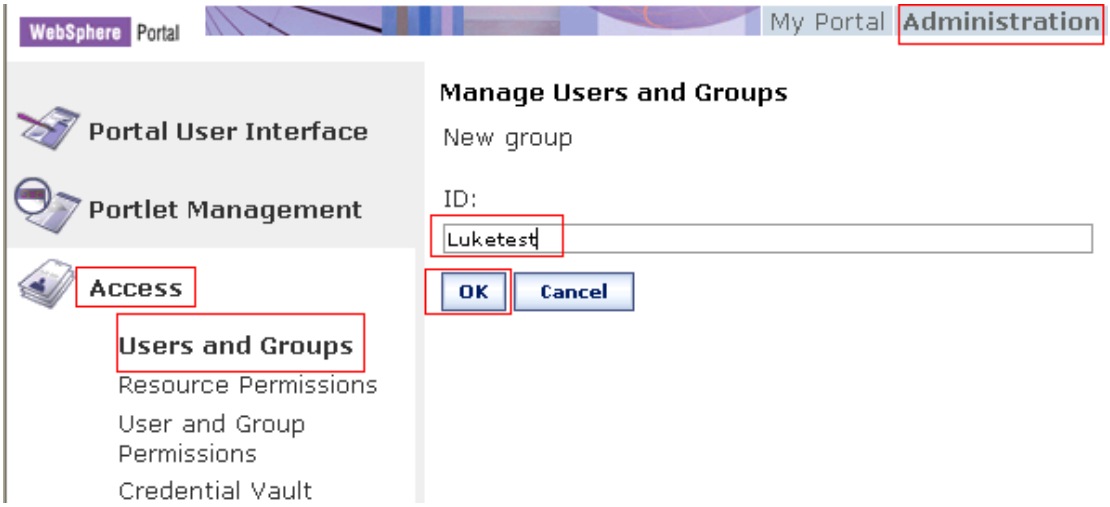
To manage the user and groups, please click **Access** -> **Users and Groups**




To add a new user, please click **New user**. Input the user information and click **OK**.




To add new group, please click **New group**. Input the group ID and click **OK**.

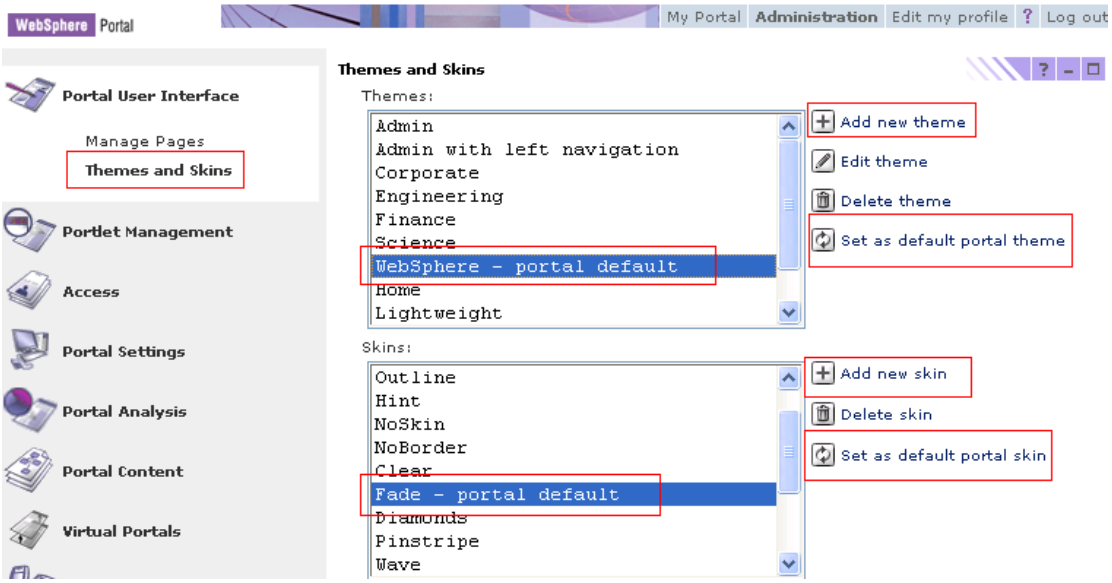



### 6.6 Manage themes and skins

To manage themes and skins, please click **Portal User Interface -> Themes and Skins**

You can select the theme or skin from the list, and set it as default in portal. You can also create a new theme or skin, and then add new theme or skin in this page.

For details, please refer to the section *Customize theme of portal* and *Customize skin of portal*




### 6.7 Set resource permissions

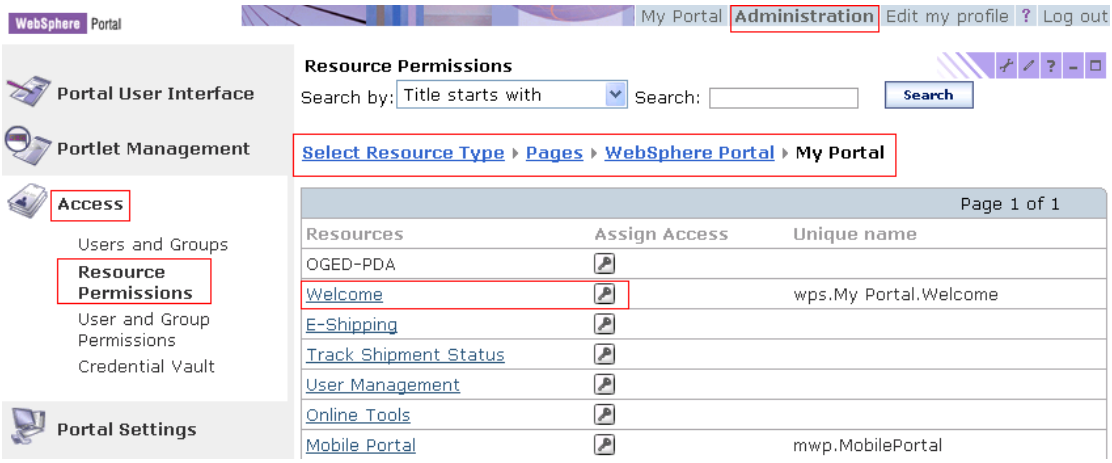
To set access permissions for page , we have two ways.

1. Click **Portal User Interface** -> **Manage Pages**, then navigate to the page, click icon **Set page permission**
2. Click **Access** ->**Resource Permission,** select **Pages** as resource type, then navigate to the page, click icon **Assign Access**

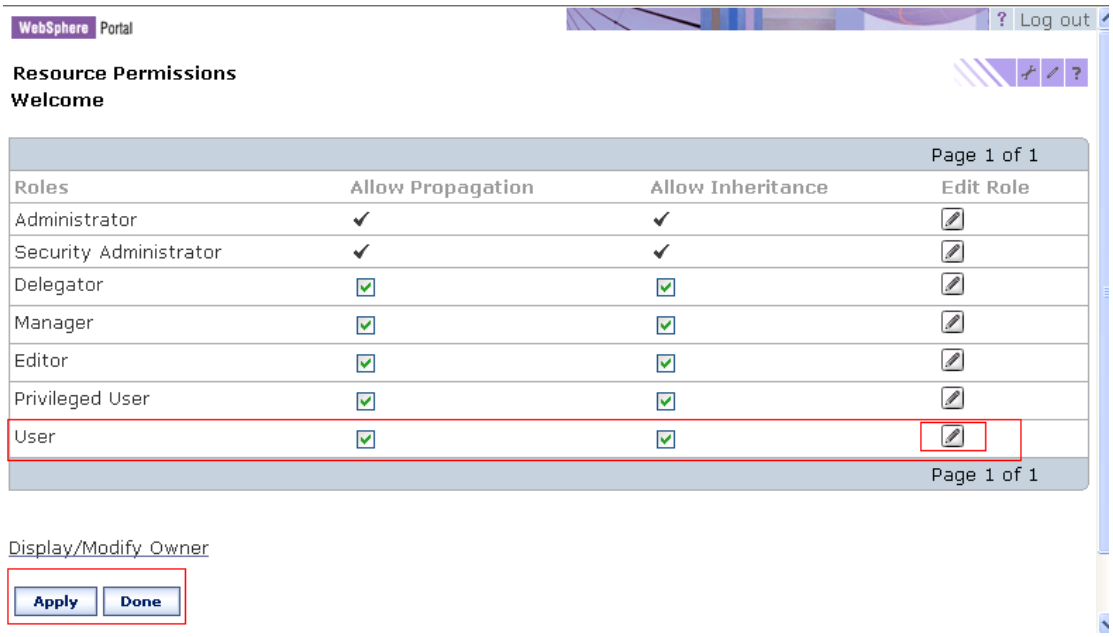
To set access permission for portlet, we also have two ways.

1. Click **Portlet Management** -> **Portlets**, then navigate to the portlet, click icon **Assign access to portlet**
2. Click **Access** ->**Resource Permission,** select **Portlets** as resource type, then navigate to the page, click icon **Assign Access**

Here I take the page and the second way as sample. It's similar for portlets.

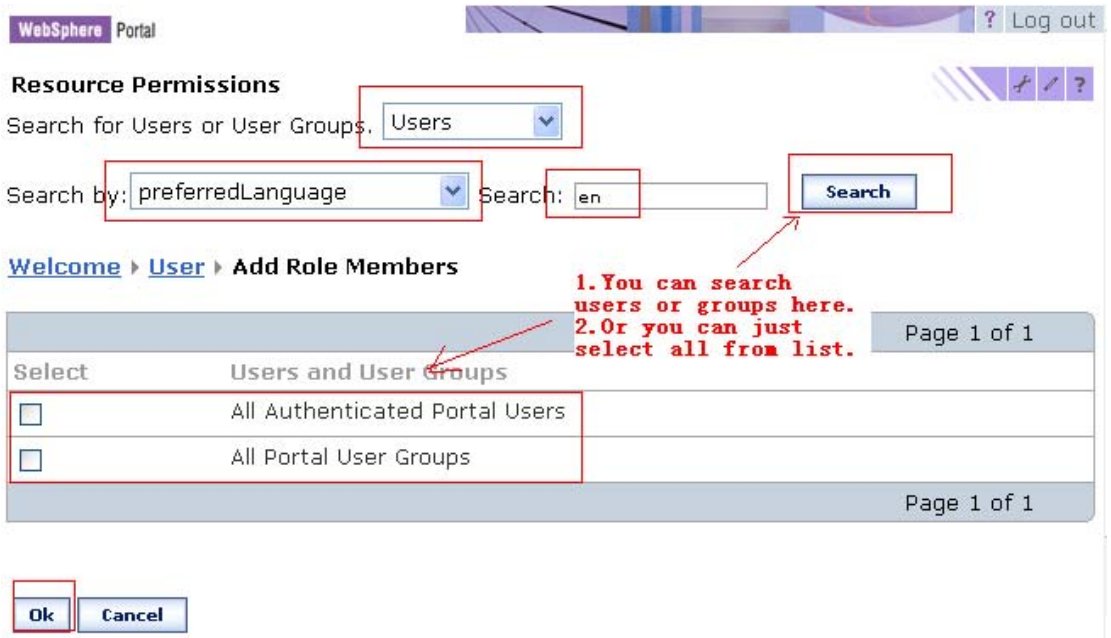



After you click the icon **Assign Access**, you will see the **Resource Permissions** page.
You can set the permission according to the role, such as Administrator, Manager, Editor, User and etc. Commonly, we set the permission for the role User.
Click the icon **Edit Role** in the User line.

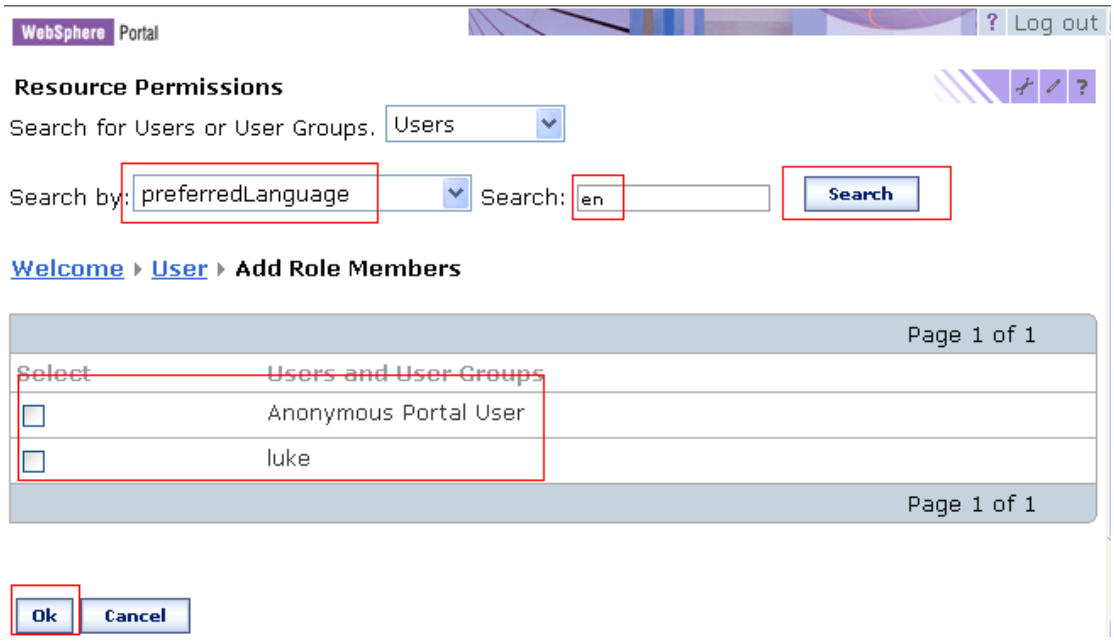



Click the button **+Add** to add the user or group.

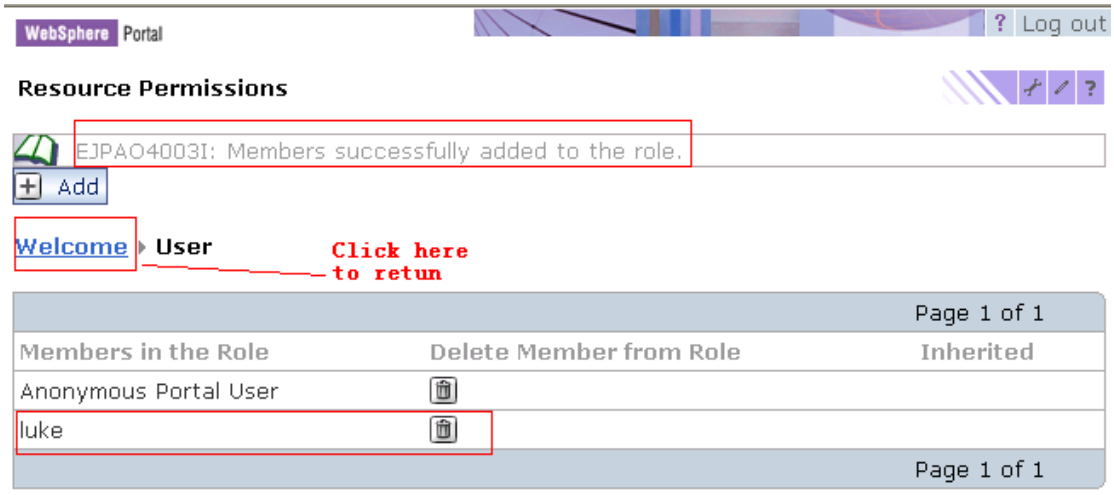

You can search user or group in this page. Commonly, we can select **Users**, then search by **prefferredLanguage**.




Input **en** as preferred language, we can search the user named luke in LDAP of portal.
Select the user luke and click OK to add the user. Then the user luke will act as the User role for the current resource.




Click the navigation bar to exit the setting page.



# 7. Bring globalization into portal

In this section, you will know how to support globalization in portal.

## 7.1 Globalization overview in portal project

To support Globalization in portal project, please pay attention to the following aspects.

### *7.1.1 Localization Pack*

To support multi languages, we need to collect all the strings which will be shown in UI interface, such as portal web page, email and etc. We can package the lp files and send to TSC for translation. After translation, we need to convert the original UTF-8 strings into \U format ASCII strings. Then we will use WPS taglib or resource bundle to get the strings in LP file.
Note: We will use \U format ASCII file in portal project. But in other projects, such as PvC projects or Web Services projects, we will use UTF-8 format LP file.

For details, please refer to the section *Localization Pack Handling*

### *7.1.2 Select Language*

To support multi languages, we need to support language changing in portal website. Then user can select the preferred language to view the portal website.

For details, please refer to the section *Select Language*

### *7.1.3 Globalization Features*

Globalization is not just a matter of translation. Cultural information should also be considered, as people from different cultural background would expect different behaviors of a same application.

For details, please refer to the section *Enable GBO in portal project* and *Globalization Preference in portal project*

### *7.1.4 Globalization Preference*

User can set his (or her) globalization preference after logging in portal website. For example, the user can browser the portal website in English, but the measurement format or calendar format can be set to Simplified-Chinese format.
We can provide the following g11N preferences for users.

- Popup Calendar
- Pull down Calendar
- Currency Format
- Length Format
- Weight Format
- Date/Time Format

- Honorific
- Name format
- Address format

For details, please refer to the section *Globalization Preference in portal project*

#### *7.1.5 Bidi support*

Commonly we choose several languages in our portal project; there should be one bid language, such as Hebrew or Egyptian. Bidi support is very unique and special for us for it's display from right to left in website. Portal developers should pay more attentions to bidi.

For details, please refer to the section *Bidi in portal*

#### *7.1.6 Artwork in portal*

In a globalization portal project, we may need to modify the stylesheet to meet the requirement in each locale.

For details, please refer to the section *Artwork in portal*

#### *7.1.7 Locale handling*

In portal project, locale is everywhere. Developer needs to know how to get the locale and use it in jsp, portlet and business beans.

For details, please refer to the section *How to get userid and locale in portlet* and *Locale fallback in portal*

#### *7.1.8 Encoding*

If the encoding is not right, the words may be error code, and the portal page will not be right to users. So we need to set UTF-8 encoding in jsp page, localization pack files, java string handling.

For details, please refer to the section *Encoding of portlet*

### 7.2 How to get userid and locale in portlet

Please get the locale and userid as the following.

#### *7.2.1 User Locale*

**1. Theme**

```
com.ibm.wps.engine.RunData rdata = com.ibm.wps.engine.RunData.from(request);
String userLocale = rdata.getLocale().toString();
```

```
out.println("userLocale is: " + userLocale);
```

**2. Portlet (In the following methods: processAction, doView,actionPerformed)**

```
//Get the locale from request
String userLocale = com.ibm.sgl.oged.util.LPUtil.getUserLocale((javax.servlet.ServletRequest)request);
//Set the user locale to session
request.getPortletSession().setAttribute("userLocale",userLocale);
//or set it in application scope
request.getPortletSession().setAttribute("userLocale",userLocale,PortletSession.APPLICATION_SCOPE);
```

**3. Portlet jsp files**

```
//Get the locale from request
String userLocale = com.ibm.sgl.oged.util.LPUtil.getFullLocale(request.getLocale().toString();
//or get it from application scope
renderRequest.getPortletSession().getAttribute("userLocale",PortletSession.APPLICATION_SCOPE);
```

**4. Java Classes**

3.1 If you have get the request firstly.

```
//Get the locale from request object
String userLocale = com.ibm.sgl.oged.util.LPUtil.getUserLocale((javax.servlet.ServletRequest)request);
```

3.2 Or you can get from session if you have set locale in session.

```
//Get the locale from session
String userLocale = request.getPortletSession().getAttribute("userLocale");
```

*7.2.2 User ID*

**1. Theme jsp file**

```
//Get user id from request
com.ibm.wps.engine.RunData rdata = com.ibm.wps.engine.RunData.from(request);
com.ibm.wps.puma.User user = rdata.getUser();
String userid = null;
if (null != user) {
      userid = user.getUserID();
      user.setPreferredLanguage(userLocale);
}
 out.println("userid is: " + userid);
```

or you can write as the following

```
String userid = request.getRemoteUser()
```

**2. Portlet class (In the following methods: processAction, doView,actionPerformed)**

```
//Get the locale from request
String userid = request.getRemoteUser();
//Set the user locale to session
request.getPortletSession().setAttribute("userid",userid);
```

**3. Portlet jsp files**

```
//Get the user id from request
String userid = request.getRemoteUser());
```

**3. Java Classes**

3.1 You should get the request firstly.

```
//Get the locale from request
String userid = request.getRemoteUser();
```

3.2 Or you can get from session if you have set it.

```
//Get the locale from session
String userid= request.getPortletSession().getAttribute("userid")
```

### 7.3 Localization Pack Handling

To support multi languages, we need to collect all the strings which will be shown in UI interface, such as portal web page, email and etc. We can package the lp files and send to TSC for translation. After translation, we need to convert the original UTF-8 strings into \U format ASCII strings. Then we will use WPS taglib or resource bundle to get the strings in LP file.

Here I will show you the life cycle of the Localization Pack.

#### *7.3.1 Collect the LP strings and organize the Localization Packs*

Please collect all the English strings into the LP properties file like the following. Please pay attention to those strings which may not be displayed in UI pages, such as portlet tiles, page titles, status , code system.

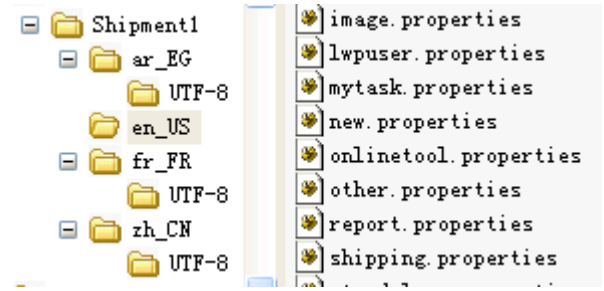
**Note: If the string is too long, please separate it into several strings in LP file**.

**Note: The line "# NLS_MESSAGEFORMAT_NONE" is required in LP file**

```
#welcome.properties
#This file is for OGED WEMP welcome function
# By Luke 2005-8-3
```

```
# NLS_MESSAGEFORMAT_NONE

#Portlet titles
PORTLET_WELCOME=Welcome

#Title informations
TITLE_GENERALINFORMATION=General Information

#Table column name
TABLE_TRACKSHIPMENT=Track shipment status

#Button label
BUTTON_TRACK=Track

#Notification information
NOTE_ABOUTSLOGAN=If you can make it here, you can make it everywhere.

#Error messages
EM_GENERAL=The system is temporarily unavailable, please contact the system administrator.

#Other string
```

Organize the LP files by locale like the following.




**Note: Please use notepad instead of UltraEdit to view and update LP files. For the Unicode support in UltraEdit is not very good.**

### *7.3.2 Translation of LP*

After the LP files are packaged, we need to contact translation service provider for transltation

Sometime we can just send UTF-8 format file, sometimes we need to send XLIFF files. So we need to use the tool XLIFF to convert the property file to XLIFF file. When the LP files are returned, we need to convert from XLIFF to property file.

**Note: Please ensure the returned files are using UTF-8 encoding.**

For details of XLIFF, please refer to the readme.txt in xliff

```
java -jar Prop2XLF.jar [-version] [-without-locale-suffix] [-reverse] [-mode aggregate|mirror|given]
```

[-target-languages ab-CD,ef-GH,ij-KL,xx-XX] [Source] [Result]

**Note: Please use notepad instead of UltraEdit to view and update LP files. For the Unicode support in UltraEdit is not very good.**
**Note: It's very important to maintain the LP file version! Please ensure all the keys in each locale the same.**

### *7.3.3 Write WPS tag or use resource bundle*

We have two methods to get the LP strings in LP file. Write a WPS tag or use resource bundle directly.

**1. For WPS tag, please refer to the section *How to write aWPS Tag***

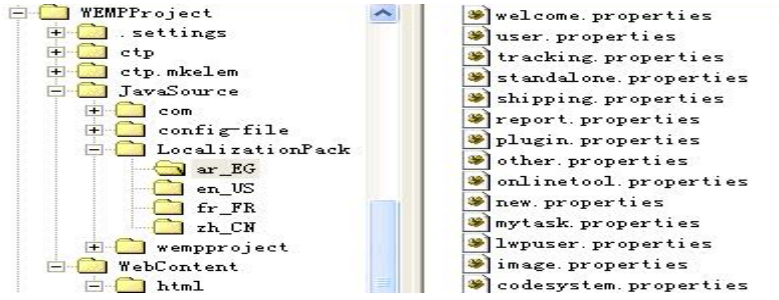
**2. About writing resource bundle code, please refer to java JDK.**

Here is a sample.

```
package com.ibm.sgl.oged.lwp.plugin.utils;

import java.util.Locale;
import java.util.*;
import java.io.*;
import org.apache.log4j.*;

public class LocaleUtil {
	public static String getResource(String name) {
		Locale locale = Locale.getDefault();
		StringBuffer bundleBuffer = new StringBuffer()
		.append("conf").append(File.separator)//$NON-NLS-1$
		.append("resource");//$NON-NLS-1$
		ResourceBundle bundle =
			ResourceBundle.getBundle(bundleBuffer.toString(), locale);
		return bundle.getString(name);
	}
}
```

### *7.3.4 Using LP in portal project*

**1. Add the LP files into portal project like the following**



**2. In JSP file**

Write the following code in jsp page:

```
<!--Import the taglib -->
<%@taglib uri="/WEB-INF/tld/lp.tld" prefix="lp"%>
<!—Or you can init another instance -->
<%@taglib uri="/WEB-INF/tld/lp.tld" prefix="lpnew"%>

<!--Load the property file -->
<lp:load file="user.properties" />
<!--You  can load another properties file using the new instance -->
<lp:load file="other.properties" />

<!--Get the translated string from lp -->
<lp:value key="USER_NAME" />
<lpnew:value key="ERR_GENERAL" />
```

**3. In Portlet or Business Bean**

We can use the utility class LPUtil.class in portlet and business bean. Please refer to LPUtil.java for details.

I

```
String mailSubject = (LPUtil.getLPValue(locale,”other.properties”,”EMAIL_TITLE”));
```

*7.3.5 Verify multi languages in portal website*

After the portal project is deployed into portal server. User can also browser the website in each locale by changing the language in homepage as the following.

Here are two images for two languages.

**English**


**Simplified-Chinese:**


### *7.3.6 TVT*

In TVT phase, tester may find that some words are not right.

We need to contact the translate to correct it and update the following places:

1. The properties file in Localization Pack directory of CC
2. The properties file in portal project
3. The properties file in portal deployed directory (You may need to re-deploy the portal project or restart the portal server to take effect.)

### *7.3.7 How to Support Complex Sentence in Localization Pack File*

We collect all the strings into localization pack files, which will be translated by TSC. Sometimes we need to use a variable in the lp strings. For example, RESULT=There are {%number} results found.

We can write a method **replaceString()** to replace the variable with true value.

Here is the sample:

#### 7.3.7.1 String in Localization Pack file

Here is the LP string in LP file.

```
# {%number} represents the tickets number ; {%acocount} represents the total cost of this order.
BOOKING_RESULT =You have booked {%number} tickets. Total cost is {%account}.
```

#### 7.3.7.2 Code in jsp file

We call the method reaplceString() to replace the variable with true value in LP string.

```
<%= replaceString(replaceString(LPUtil.getLPValue(lpFile, "BOOKING_RESULT") ,
"{%number}" , ticketNumber), "{%account}",totalAccount) %>
```

#### 7.3.7.3 The method replaceString()

The method replaceString().

```
<%!
    public String replaceString(String original, String source, String target) {
        String result = original;
        int pos = 0;

        pos = original.indexOf(source);

        if (pos != -1) {
            result = original.substring(0, pos);
            result += target;
            result += original.substring(pos + source.length(), original.length());
        }

        return result;
    }
%>
```

## 7.4 Select language

To support multi languages, we need to support language changing in portal website. Then user can select the preferred language to view the portal website.

Here I attach two images of two languages.

### *7.4.1 English*


### *7.4.2 Simplified-Chinese:*

### *7.4.3 Code Fragment*

We implement this function using **JavaScript** and WPS command **ChangeLanguage**.
Here are the code fragments.

```
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8" session="true"%>
<%@ taglib uri="http://java.sun.com/portlet" prefix="portlet" %>
<%@ page import="java.util.*,javax.portlet.*,com.ibm.sgl.oged.wemp.portlet.*" %>
<%@ page import="java.util.*,javax.portlet.*,com.ibm.sgl.oged.wemp.portlet.*,
com.ibm.sgl.oged.db.databean.G11NPreferenceBean, com.ibm.sgl.oged.constant.UserConstansts,
com.ibm.sgl.oged.util.CodeSystem, com.ibm.sgl.gbo.core.operation.FeatureManager,
com.ibm.sgl.gbo.localelist.IGLocaleList,
com.ibm.sgl.gbo.core.preference.base.* " %>

<%@ page import="com.ibm.sgl.oged.util.*"%>
<%@ taglib uri="/WEB-INF/tld/engine.tld" prefix="wps"%>

<%@ taglib uri="/WEB-INF/tld/lp.tld" prefix="lp" %>
<lp:load file="welcome.properties"/>
<portlet:defineObjects />
<%@ include file="Bidi.jsp" %>

<script language="JavaScript">
function changeLangUrl(){
    var language = document.selLangForm.Language.value;
    //Set the cookie locale for XUL function in Mozila
        var Then = new Date();
        Then.setTime(Then.getTime() + 60*1000 ); //60second
        document.cookie = "locale=ogedwemp.cn.ibm.com_"+language+";expires="+
Then.toGMTString();

    var submitURL;
    if(language=="en_US")
    {
        submitURL = "<wps:url command="ChangeLanguage"><wps:urlParam name="locale"
value="en_US"/></wps:url>";    }
    else if(language=="zh_CN")
    {
        submitURL = "<wps:url command="ChangeLanguage"><wps:urlParam name="locale"
value="zh_CN"/></wps:url>";
    }
    else if(language=="ar_EG")
    {
        submitURL = "<wps:url command="ChangeLanguage"><wps:urlParam name="locale"
```

```
value="ar_EG"/></wps:url>";
      }
      else if(language=="fr_FR")
      {
          submitURL = "<wps:url command="ChangeLanguage"><wps:urlParam name="locale"
value="fr_FR"/></wps:url>";
      }

  <wps:if loggedIn="yes">
      submitURL=submitURL.replace("portal","myportal");
  </wps:if>

            window.location.href = submitURL;

}
</script>

<%
//Cookie cookie = new Cookie("locale", "en_US");
//cookie.setMaxAge(100000000);
//String name = cookie.getName();
//String value =    cookie.getValue();

//response.addCookie(cookie);
//response.flushBuffer();
//out.println("Finish adding cookie - "+name+" | "+ value);

String userLocale = com.ibm.sgl.oged.util.LPUtil.getFullLocale(request.getLocale().toString());

//String curLocale = request.getLocale().toString();
//out.println("*****getRemoteUser is " + request.getRemoteUser());
//out.println("*****getLocale is " + userLocale);
  %>

<%!
      public String replaceString(String original, String source, String target) {
            String result = original;
            int pos = 0;

            pos = original.indexOf(source);

            if (pos != -1) {
                  result = original.substring(0, pos);
```

```
                result += target;
                result += original.substring(pos + source.length(), original.length());
            }

            return result;
        }
%>
<form name="selLangForm">

<link href="/wps/oged/images/<%=userLocale %>/Styles.css" rel="stylesheet" type="text/css">

<table width="270" border="0" cellspacing="0" cellpadding="0">
            <tr>
              <td><img  src="/wps/oged/images/<%=userLocale%>/homemap1.gif"  width="271"
height="50" alt='<lp:value key="TABLE_SELECTLANGUAGE"/>'></td>
            </tr>
</table>


<table width="100%" border="0" cellpadding="0" cellspacing="0"
class="protalboxsmall<%=dir%>">
                    <tr>
                      <td height="29" class="protaltd2map<%=dir%>" ><div ><lp:value
key="TABLE_SELECTLANGUAGE" /></div></td>
                    </tr>
                    <tr>
                      <td height="65"  class="protaltd3map<%=dir%>" ><div align="center" >
                      <select onchange="changeLangUrl();" name="Language" >
<%
      String localeList[]   = {"en_US", "zh_CN", "fr_FR", "ar_EG"};
      String languageList[] = {"en", "zh", "fr", "ar"};
      String countryList[] = {"US", "CN", "FR", "EG"};

      BasePreferenceFactory bpf = BasePreferenceFactory.getFactory();
      BasePreference bp = bpf.getBasePreference("languageList");

      FeatureManager fm = FeatureManager.getInstance();
      for(int i = 0; i<4; i++) {
            Locale locale = new Locale(languageList[i], countryList[i]);
            //System.out.println(locale);
            bp.setLanguage(locale);
            //System.out.println(localeList[i] + locale);
```

```
            IGLocaleList gll = (IGLocaleList) fm.getFeature(IGLocaleList.class, "languageList");

            String langList3 = gll.getLocaleName(locale);
            //System.out.println(langList3);
            String optionValue = "";
            if(CodeSystem.LOCALE_EN.equals(localeList[i])) {
                  optionValue = CodeSystem.LANGUAGE_EN;
            }
            if(CodeSystem.LOCALE_CN.equals(localeList[i])) {
                  optionValue = CodeSystem.LANGUAGE_CN;
            }
            if(CodeSystem.LOCALE_FR.equals(localeList[i])) {
                  optionValue = CodeSystem.LANGUAGE_FR;
            }
            if(CodeSystem.LOCALE_AR.equals(localeList[i])) {
                  optionValue = CodeSystem.LANGUAGE_AR;
                  dir = "rtl";
                  langList3                                                                    =
replaceString(replaceString(langList3,")","&#8207;)" ),"(","&#8207;(");
            }
      %>
      <option dir="<%=dir%>" value=<%=optionValue %>
            <%if(optionValue.equals(userLocale)) {out.print("selected");} %>>
        <div align="center" dir=<%=dir%> >
            <%=replaceString(replaceString(langList3,")",bidiForceString+")"   ),"(",bidiForceString
+ "(")%>
               </div ></option>
<%
} %>
/select>

                        </div></td>
                  </tr>
               </table>
</form>
```

### 7.5 Globalization Preference in portal projects

User can set his (or her) globalization preference after logging in portal website. For example, the user can browser the portal website in English, but the measurement format or calendar format can be set to Simplified-Chinese format.

### *7.5.1 Globalization Preference*

We can provide the following g11N preferences for users.

- Popup Calendar
- Pull down Calendar
- Currency Format
- Length Format
- Weight Format
- Date/Time Format
- Honorific
- Name format
- Address format

### *7.5.2 Design a table base for G11n preference*

We need to store the G11n preference setting in database. Any G11n feature should be related to a field.For example, Currency, Length, Weight and etc.

Here is a sample.


Then you can generate the script in RSA.

```
-- Please run the command: db2 -tvf OGED.sql
CREATE DATABASE OGEDWEMP;
CONNECT TO OGEDWEMP;
CREATE SCHEMA DB2INST1;
CREATE TABLE DB2INST1.G11NPREFERENCE
  (USERID VARCHAR(32) NOT NULL,
    LANGUAGE VARCHAR(32) NOT NULL,
    CURRENCY VARCHAR(10) NOT NULL,
    DATETIME VARCHAR(10) NOT NULL,
    CALENDAR VARCHAR(10) NOT NULL,
    WEIGHT VARCHAR(10) NOT NULL,
    LENGTH VARCHAR(10) NOT NULL,
  );
ALTER TABLE DB2INST1.G11NPREFERENCE  ADD CONSTRAINT C2239253 PRIMARY
KEY (USERID);
```

```
ALTER TABLE DB2INST1.G11NPREFERENCE ADD CONSTRAINT USERID FOREIGN KEY (USERID)
  REFERENCES DB2INST1.USER(USERID);
CONNECT RESET;
```

Here is the G11n preference in database.

Note WM1001and LM1001 are constants in Code System and represent Metric or English.

```
USERID    LANGUAGE    CURRENCY DATETIME     CALENDAR WEIGHT    LENGTH
------ ---------- -------------------------------- -------------------------------- --------------------------------
lukewang          en_US                en_US                en_US                en_US
WM1001       LM1001
```

### *7.5.3 Write DAO and DataBean*

Commonly DataBean is related to a table in database, you can also define a DataBean as data transferring between classes. DataBean is a set of data, which will be used everywhere in the portal project. Such as portlet, ActionBean, BusinessBean, JSP and etc. We need to write DAO to handle the database related operations. For example, insert, query, update, delete and etc.

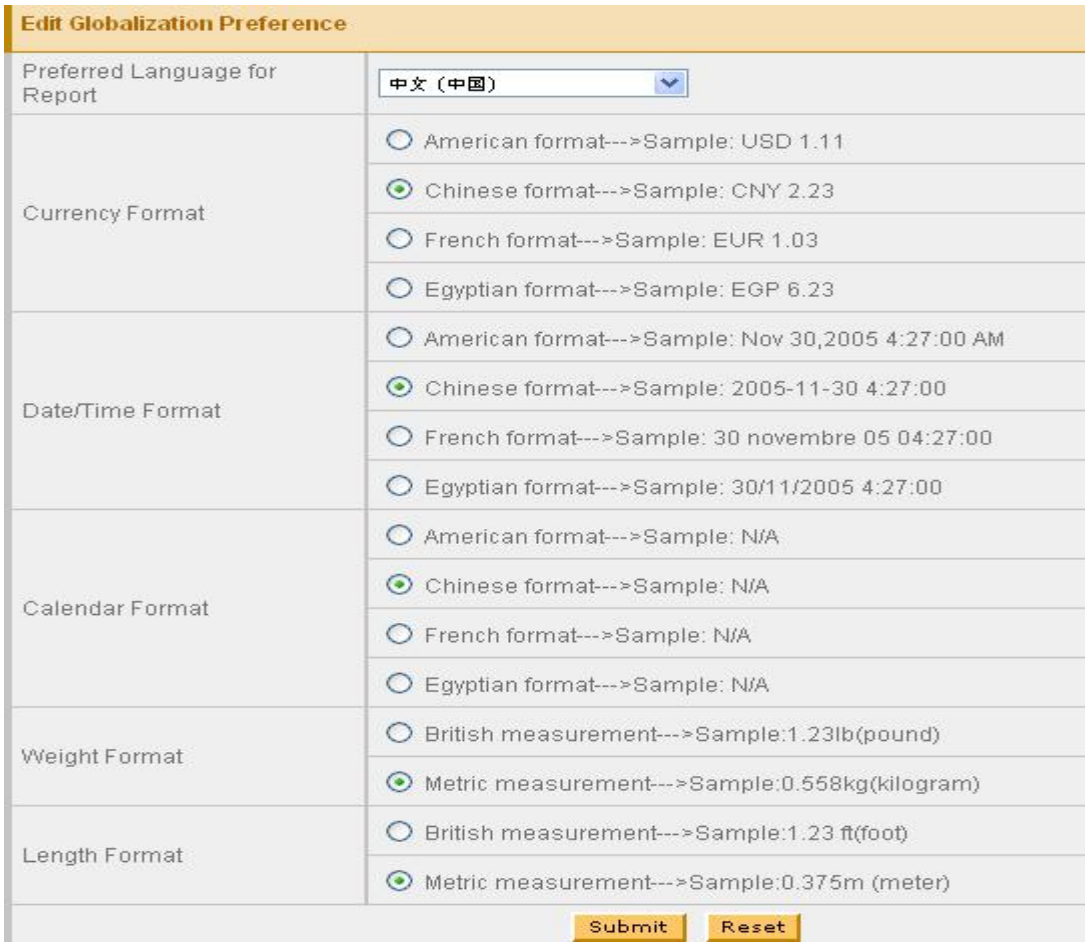
### *7.5.4 Write a portlet for G11n preference*

A portlet can be provided for user to set the G11n preference.

### *7.5.5 Verify the results in portal website*

Set the value in the portlet.

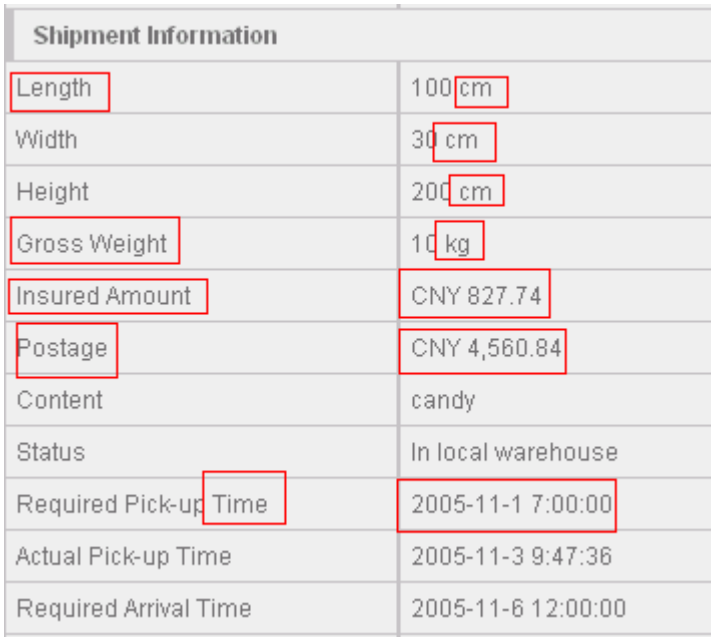
Assume that the current user browsers the website using English, but he (or she) wants to set the G11n preference to Chinese format.



| Edit Globalization Preference | |
|---|---|
| Preferred Language for Report | 中文（中国） |
| Currency Format | ○ American format--->Sample: USD 1.11 |
| | ◉ Chinese format--->Sample: CNY 2.23 |
| | ○ French format--->Sample: EUR 1.03 |
| | ○ Egyptian format--->Sample: EGP 6.23 |
| Date/Time Format | ○ American format--->Sample: Nov 30,2005 4:27:00 AM |
| | ◉ Chinese format--->Sample: 2005-11-30 4:27:00 |
| | ○ French format--->Sample: 30 novembre 05 04:27:00 |
| | ○ Egyptian format--->Sample: 30/11/2005 4:27:00 |
| Calendar Format | ○ American format--->Sample: N/A |
| | ◉ Chinese format--->Sample: N/A |
| | ○ French format--->Sample: N/A |
| | ○ Egyptian format--->Sample: N/A |
| Weight Format | ○ British measurement--->Sample:1.23lb(pound) |
| | ◉ Metric measurement--->Sample:0.558kg(kilogram) |
| Length Format | ○ British measurement--->Sample:1.23 ft(foot) |
| | ◉ Metric measurement--->Sample:0.375m (meter) |
| | Submit Reset |

Verify the result in portal website.

You can see that the Currency, Date/time is set to Chinese format, and Length, Weight are set to Metric format.



| Shipment Information | |
|---|---|
| Length | 100 cm |
| Width | 30 cm |
| Height | 200 cm |
| Gross Weight | 10 kg |
| Insured Amount | CNY 827.74 |
| Postage | CNY 4,560.84 |
| Content | candy |
| Status | In local warehouse |
| Required Pick-up Time | 2005-11-1 7:00:00 |
| Actual Pick-up Time | 2005-11-3 9:47:36 |
| Required Arrival Time | 2005-11-6 12:00:00 |

### 7.6 Enable GBO in portal project

It's a well-known fact that an application should provide a friendly, "globalized" interface for its customers. However, there is a category of culturally sensitive GUI (graphical user interface) functionalities that is indispensable for the globalized appliations but not currently supported by Java or other existing technologies and that often requires developers to write complex code from scratch.

Global Business Object (GBO) is a technical preview that provides a set of libraries for constructing culturally aware graphical controls. GBO encapsulates the functions into a reusable globalization tag library in a J2EE environment. Programmers can easily create global GUI functions in Java Server Pages (JSP) files by simply using GBO's tags.

Here I will list some sample code used in portal project. you can visit GBO intranet website for details. http://9.181.106.149:9080/gbo/.
Fore more details about GBO, please visit the following link.
Build culturally aware apps with GBO:
http://www-128.ibm.com/developerworks/java/library/j-gbo/?ca=dnt-649
Global Business Object:
http://www.alphaworks.ibm.com/tech/gbo

#### *7.6.1 Pop up Calendar*

**JavaScript fragments**

For details, please refer to GBO team: PopupCalendar.jsp and PopupCalendar.js

**Code fragments**

```
            <tr>
               <td class="td3<%=dir%>"><lp:value key="TABLE_BEGINDATE" /> </td>
                 <td class="td6">
                     <input type="text" id="date1_Show" name="begindateshow"
readonly="readonly" value=<%=begindate%>>
                     <input type="hidden" id="date1" name="begindate"
<%if(0!=begindatehidden.length())%> value=<%=begindatehidden%>>
                     <input type="hidden" id="date1_Unformat" >
                     <img src="/wps/oged/images/calendar.gif" alt="<lp_alt:value
key="CALENDAR"/>"
onclick="window.open('<%=renderRequest.getContextPath()%>/html/popupcalendar/PopupCalen
dar.jsp?locale=<%=poplocale%>','date1','height=200,width=210,scrollbars=no,resizable=yes,statu
s=no,menu=no,toolbar=no')">
          </td>
            <tr>
```

### *7.6.2 Pull down Calendar*

**JavaScript fragments**

```
<SCRIPT language="JavaScript">
//If there are more than one PulldownGBO in one page.
//You should repacle the name of year, month, date with new one.
var monthdays = new Array(12);

monthdays[0] = 31;
monthdays[1] = 28;
monthdays[2] = 31;
monthdays[3] = 30;
monthdays[4] = 31;
monthdays[5] = 30;
monthdays[6] = 31;
monthdays[7] = 31;
monthdays[8] = 30;
monthdays[9] = 31;
monthdays[10] = 30;
monthdays[11] = 31;

var back = new Array(4);


function update_days(f)

{
//      alert(f.months.parent());
        compute_days(f);
        var lengthDeleted = 31 - monthdays[f.month1.selectedIndex];

        if(back[0] == null){
                for(x=0;x<back.length ;x++) {
                        back[x] = f.date1[30 - x];
                }
        }

        if(null != back[0]){
                for(x=0;x<back.length ;x++) {
                        f.date1[30 - x] = back[x];
                }
```

```
        }

        for(x=0;x<lengthDeleted ;x++) {
                f.date1[30 - x] = null;
        }
}

function compute_days(f){
        if(!isNaN(f.year1.value) && f.year1.value.length > 0){
                var year=parseInt(f.year1.value);
                var ret_val = 0;
                if (year % 4 == 0 && year % 100 != 0 )
                        ret_val=1;

                if (year%4 == 0 && year % 400 == 0)
                        ret_val=1;

                if(ret_val == 1){
                        monthdays[1] = 29;
                }else{
                        monthdays[1] = 28;
                }
                        }

}
</SCRIPT>
```

**Code fragments**

```
<%@                                                                    page
import="java.util.*,java.text.*,java.sql.Timestamp,javax.portlet.*,com.ibm.sgl.oged.wemp.portlet.
*,com.ibm.gcl.gbo.calendar.*,com.ibm.gcl.gbo.template.*" %>

<%
CalendarGBO calendar = CalendarGBO.getInstance(locale);
String dateorder = calendar.getDateOrder();
for(int j = 0; j < dateorder.length(); j ++){
if(dateorder.charAt(j)=='m'){
%><td class="td2<%=dir%>">
<select name="month1" size="1" onchange="update_days(this.form)">
<%
        String[] monthvalue = {"1","2","3","4","5","6","7","8","9","10","11","12"};
        String[] monthcaption = calendar.getFullMonthList();
        for(int i = 0; i < monthvalue.length; i ++)     {
```

```
            if(monthvalue[i].equals(month)){
%>
                <option value="<%=monthvalue[i]%>"
selected><%=monthcaption[i]%></option>
<%          }else{ %>
                <option value="<%=monthvalue[i]%>"><%=monthcaption[i]%></option>
<%          }
      }
%>
</select>
</td>
<%
}

else if(dateorder.charAt(j)=='d'){
%><td class="td2<%=dir%>">
<select name="date1" size="1">
<%
String[] dayvalue =
{"1","2","3","4","5","6","7","8","9","10","11","12","13","14","15","16","17","18","19","20","21",
"22","23","24","25","26","27","28","29","30","31"};
String[] daycaption = calendar.getFullDayList();
for(int i = 0; i < dayvalue.length; i ++)
{
      if(dayvalue[i].equals(date)){
%>
                <option value="<%=dayvalue[i]%>" selected><%=daycaption[i]%></option>
<%          }else{ %>
                <option value="<%=dayvalue[i]%>"><%=daycaption[i]%></option>
<%          }
      }
%>
</select>
</td>
<%
}

else if(dateorder.charAt(j)=='y'){
%><td class="td2<%=dir%>">
<select name="year1" size="1" onchange="update_days(this.form)">
<%
String yearlist[] =
{"2000","2001","2002","2003","2004","2005","2006","2007","2008","2009","2010"};
```

```
for(int i = 0; i < yearlist.length; i ++){
      if(yearlist[i].equals(year)){
%>
                  <option value="<%=yearlist[i]%>" selected><%=yearlist[i]%></option>
<%        }else{ %>
                  <option value="<%=yearlist[i]%>"><%=yearlist[i]%></option>
<%        }
      }
%>
</select>
</td>
<%
}
}%>


<td class="td2<%=dir%>">
<select name="time" size="1">
<%
String[]                                                times                                                =
{"00:00","01:00","02:00","03:00","04:00","05:00","06:00","07:00","08:00","09:00","10:00","11:0
0","12:00","13:00","14:00","15:00","16:00","17:00","18:00","19:00","20:00","21:00","22:00","23
:00"};

      for(int i = 0; i < times.length; i ++){
            if(times[i].equals(time)) {
 %>
      <option value="<%=times[i]%>" selected><%=times[i]%></option>
<%  }
      else{
%>          <option value="<%=times[i]%>" ><%=times[i]%></option>

<%  }
      }
   %>
</select>
</td>
```

*7.6.3 Measurement*

**Measurement sample**

```
/*
 * Class: MeasurementUtil.java
 * Description:
 * Version: 1.0
 * Author: swj
 * Creation date: Sep 12, 2005
 * Department: Shanghai Globalization Laboratory
 * Copyright (c) 2004, International Business Machines Corporation,
 * All rights reserved.
 */

package com.ibm.sgl.oged.util;

import com.ibm.gcl.gbo.measure.MeasureGBO;

/**
 *
 *
 * @author swj
 * @version 1.0
 */
public class MeasurementUtil {
	/**
	 * Unit name for cm.
	 */
	public static final String UNIT_CM = "cm";

	/**
	 * Unit name for inch.
	 */
	public static final String UNIT_INCH = "inch";

	/**
	 * Unit name for kg.
	 */
	public static final String UNIT_KG = "kg";

	/**
	 * Unit name for pound.
	 */
	public static final String UNIT_POUND = "pound";

	/**
```

```
 * Format the length value to display. The formatted number will contain the
 * unit information.
 *
 * @param initValue
 *                  The initial value. This value is in the unit "cm".
 * @param convertTo
 *                  The target unit system. This is one of the two values:
 *                  CodeSystem.LENGTH_ENGLISH CodeSystem.LENGTH_METRIC
 * @param displayLocale
 *                  The result locale.
 * @return The formatted string.
 */
public static String formatLength (double initValue, String convertTo,
        String displayLocale) {
    String initUnit = UNIT_CM;
    String toUnit = UNIT_CM;
    if (CodeSystem.LENGTH_ENGLISH.equals(convertTo)) {
        toUnit = UNIT_INCH;
    }

    return MeasureGBO.convertByUnit(initUnit, toUnit, initValue, displayLocale);
}

/**
 * Format the length value to display. The formatted number will contain the
 * unit information.
 *
 * @param initValue
 *                  The initial value. This value is in the unit "cm".
 * @param convertTo
 *                  The target unit system. This is one of the two values:
 *                  CodeSystem.LENGTH_ENGLISH CodeSystem.LENGTH_METRIC
 * @param displayLocale
 *                  The result locale.
 * @param precision
 *                  The number of digits that will be kept.
 * @return The formatted string.
 */
public static String formatLength (double initValue, String convertTo,
        String displayLocale, int precision) {
    String initUnit = UNIT_CM;
    String toUnit = UNIT_CM;
    if (CodeSystem.LENGTH_ENGLISH.equals(convertTo)) {
```

```
                toUnit = UNIT_INCH;
            }

            return MeasureGBO.convertByUnit(initUnit, toUnit, initValue, displayLocale,
precision);
        }

        /**
         * Format the length value. Only the number part of the value will be
         * returned.
         *
         * @param initValue
         *                  The value to be converted.
         * @param fromLocale
         *                  The original locale.
         * @param toLocale
         *                  The target locale.
         * @param displayLocale
         *                  The display locale.
         * @return The formatted value.
         */
        public static String formatLengthValue (double initValue,
                String fromLocale, String toLocale, String displayLocale) {
            String initUnit = UNIT_CM;
            String toUnit = UNIT_CM;
            if (CodeSystem.LENGTH_ENGLISH.equals(initUnit)) {
                initUnit = UNIT_INCH;
            }
            if (CodeSystem.LENGTH_ENGLISH.equals(toUnit)) {
                toUnit = UNIT_INCH;
            }
            return MeasureGBO.convertValueByUnit(initUnit, toUnit, initValue,displayLocale);
        }

        /**
         * Convert the length value to the standard unit. This method shall be
         * called before values are added to database.
         *
         * @param initValue
         *                  The initial value.
         * @param initMetric
         *                  The initial unit.
         * @return The converted value without unit.
```

```
 */
public static double convertLength (double initValue, String initMetric) {
        String initUnit = UNIT_CM;
        String toUnit = UNIT_CM;
        if (CodeSystem.LENGTH_ENGLISH.equals(initMetric)) {
                initUnit = UNIT_INCH;
        }
        return MeasureGBO.convertValueByUnit(initUnit, toUnit, initValue);
}

/**
 * Format the weight value to display. The formatted number will contain the
 * unit information.
 *
 * @param initValue
 *                  The initial value. This value is in the unit "kg".
 * @param convertTo
 *                  The target unit system. This is one of the two values:
 *                  CodeSystem.WEITHT_ENGLISH CodeSystem.WEIGHT_METRIC
 * @param displayLocale
 *                  The result locale.
 * @return The formatted string.
 */
public static String formatWeight (double initValue, String convertTo,
                String displayLocale) {
        String initUnit = UNIT_KG;
        String toUnit = UNIT_KG;
        if (CodeSystem.WEIGHT_ENGLISH.equals(convertTo)) {
                toUnit = UNIT_POUND;
        }

        return MeasureGBO.convertByUnit(initUnit, toUnit, initValue, displayLocale);
}

/**
 * Format the weight value to display. The formatted number will contain the
 * unit information.
 *
 * @param initValue
 *                  The initial value. This value is in the unit "kg".
 * @param convertTo
 *                  The target unit system. This is one of the two values:
 *                  CodeSystem.WEITHT_ENGLISH CodeSystem.WEIGHT_METRIC
```

```
     * @param displayLocale
     *                The result locale.
     * @param precision
     *                The number of digits that will be kept.
     * @return The formatted string.
     */
    public static String formatWeight (double initValue, String convertTo,
                String displayLocale, int precision) {
        String initUnit = UNIT_KG;
        String toUnit = UNIT_KG;
        if (CodeSystem.WEIGHT_ENGLISH.equals(convertTo)) {
            toUnit = UNIT_POUND;
        }

        return MeasureGBO.convertByUnit(initUnit, toUnit, initValue, displayLocale,
precision);
    }

    /**
     * Format the weight value. Only the number part of the weight will be
     * returned.
     *
     * @param initValue
     *                The value to be formatted.
     * @param fromLocale
     *                The original locale.
     * @param toLocale
     *                The target locale.
     * @param displayLocale
     *                The display locale.
     * @return The formatted value.
     */
    public static String formatWeightValue (double initValue,
                String fromLocale, String toLocale, String displayLocale) {
        String initUnit = UNIT_KG;
        String toUnit = UNIT_KG;
        if (CodeSystem.WEIGHT_ENGLISH.equals(initUnit)) {
            initUnit = UNIT_POUND;
        }
        if (CodeSystem.WEIGHT_ENGLISH.equals(toUnit)) {
            toUnit = UNIT_POUND;
        }
```

```
            return MeasureGBO.convertValueByUnit(initUnit, toUnit, initValue, displayLocale);
        }

        /**
         * Convert the weight value to the standard unit. This method shall be
         * called before values are added to database.
         *
         * @param initValue
         *                  The initial value.
         * @param initMetric
         *                  The initial unit.
         * @return The converted value without unit.
         */
        public static double convertWeight (double initValue, String initMetric) {
            String initUnit = UNIT_KG;
            String toUnit = UNIT_KG;
            if (CodeSystem.WEIGHT_ENGLISH.equals(initMetric)) {
                initUnit = UNIT_POUND;
            }
            return MeasureGBO.convertValueByUnit(initUnit, toUnit, initValue);
        }

        public static void main (String args[]) {
            double result = MeasureGBO.convertValueByUnit("inch", "cm", 12345.0);
            String result1 = MeasureGBO.convertByUnit("cm", "inch", result, "en_US");
            System.out.println(result);
            System.out.println(result1);

            System.out.println(31356.3 / 2.54);
        }
}
```

### *7.6.4 Title*

**Title sample**

```
<%@ page import="com.ibm.gcl.gbo.name.NameGBO"%>

<select class="select-title" name="Title">
<%
            NameGBO name = NameGBO.getInstance(curLocale);
            String strArr[] = null;
            String strHon = null;
            String strIdx = "";
```

```
            strArr = name.getHonorificArr();
            for (int i=0,j=0; i<strArr.length; i++) {
                  strIdx = i + "";
                  strHon = strArr[i];
                  if (strHon.equals("")) {
                        continue;
                  } else {
                        j++;
                  }
                  if (j == 0) {
%>
                        <option value="<%=i%>" <%=(strIdx.equals(su.getTitle())) ? "selected" :
""%>><%=strHon%></option>
<%
                  } else {
%>
                        <option value="<%=i%>" <%=(strIdx.equals(su.getTitle())) ? "selected" :
""%>><%=strHon%></option>
<%                }
            }
%>
</select>
```

*7.6.5 Address*

**Address sample**

```
<%@ page import="com.ibm.gcl.gbo.address.AddressGBO"%>

<select                                                    class="select-nationality"
onchange="window.document.theform.elements['counOnChange'].value='changed';checkFormatB
ean();" name="Country">
                  <%
                    AddressGBO addr = AddressGBO.getInstance(curLocale);
                    String strArr2[][] = null;

                    strArr2 = addr.getCountryList();
                    for (int i=0; i<strArr2.length; i++) {
                          String locale = strArr2[i][0];
                          String country = strArr2[i][1];
                  %>
                    <option                                          value="<%=locale%>"
<%=su.getCountry().equals(locale)?"selected":""%>><%=country%></option>
```

```
                    <%
                      }
                    %>
</select>
```

### *7.6.6 Currency*

**Currency sample**

```
/*
 * Class: CurrencyUtil.java
 * Description:
 * Version: 1.0
 * Author: swj
 * Creation date: Sep 13, 2005
 * Department: Shanghai Globalization Laboratory
 * Copyright (c) 2004, International Business Machines Corporation,
 * All rights reserved.
 */

package com.ibm.sgl.oged.util;

import com.ibm.gcl.gbo.currency.CurrencyGBO;

/**
 *
 *
 * @author swj
 * @version 1.0
 */
public class CurrencyUtil {
	/**
	 * Convert the currency value to USD. This method shall be called before
	 * values are added to database.
	 *
	 * @param initValue
	 *                  The initial value.
	 * @param initLocale
	 *                  The initial locale.
	 * @return The converted value without unit.
	 */
	public static double convertCurrency (double initValue, String initLocale) {
		String toLocale = CodeSystem.LANGUAGE_EN;
		return CurrencyGBO
```

```
                .convertValueByLocale(initLocale, toLocale, initValue);
    }

    /**
     * Format the currency value to display.
     *
     * @param initValue
     *                  The currency amount in USD.
     * @param toLocale
     *                  The target locale.
     * @param displayLocale
     *                  The locale for display.
     * @return The formatted currency string.
     */
    public static String formatCurrency (double initValue, String toLocale,
            String displayLocale) {
        String initLocale = CodeSystem.LANGUAGE_EN;

        return CurrencyGBO.convertByLocale(initLocale, toLocale, initValue,
                displayLocale);
    }

    /**
     * Format the currency value. Only the number part of the result will be
     * returned.
     *
     * @param initValue
     *                  The currency ammount to be formatted.
     * @param fromLocale
     *                  The original locale.
     * @param toLocale
     *                  The target locale.
     * @param displayLocale
     *                  The display locale.
     * @return The formatted currency string.
     */
    public static String formatCurrencyValue (double initValue,
            String fromLocale, String toLocale, String displayLocale) {
        return CurrencyGBO.convertValueByLocale(fromLocale, toLocale,
                initValue, displayLocale);
    }

    public static void main (String[] args) {
```

```
            System.out.println("Test currency util...");
            System.out.print("100 FR = ");
            System.out.print(convertCurrency(100, "fr_FR"));
            System.out.println(" USD");

            System.out.println();
            System.out.print("100 USD = ");
            System.out.print(formatCurrency(12.0811, "zh_CN", "ar_EG"));
            System.out.println();

            System.out.println();
            System.out.print("100 USD = ");
            String result = CurrencyGBO.convertValueByCurrency("USD", "CNY", 100.0,
                        "fr_FR");
            System.out.println(result);
            System.out.println();
      }
}
```

### *7.6.7 Locale List*

**Code fragments**

```
<%@ page import="java.util.*,javax.portlet.*,com.ibm.sgl.oged.wemp.portlet.*,
com.ibm.sgl.oged.db.databean.G11NPreferenceBean, com.ibm.sgl.oged.constant.UserConstansts,
com.ibm.sgl.oged.util.CodeSystem, com.ibm.sgl.gbo.core.operation.FeatureManager,
com.ibm.sgl.gbo.localelist.IGLocaleList,
com.ibm.sgl.gbo.core.preference.base.* " %>

<form name="selLangForm">
<select onchange="changeLangUrl();" name="Language" >
<%
      String localeList[]    = {"en_US", "zh_CN", "fr_FR", "ar_EG"};
      String languageList[] = {"en", "zh", "fr", "ar"};
      String countryList[] = {"US", "CN", "FR", "EG"};

      BasePreferenceFactory bpf = BasePreferenceFactory.getFactory();
      BasePreference bp = bpf.getBasePreference("languageList");

      FeatureManager fm = FeatureManager.getInstance();
      for(int i = 0; i<4; i++) {
            Locale locale = new Locale(languageList[i], countryList[i]);
            //System.out.println(locale);
```

```
            bp.setLanguage(locale);
            //System.out.println(localeList[i] + locale);

            IGLocaleList gll = (IGLocaleList) fm.getFeature(IGLocaleList.class, "languageList");

            String langList3 = gll.getLocaleName(locale);
            //System.out.println(langList3);
            String optionValue = "";
            if(CodeSystem.LOCALE_EN.equals(localeList[i])) {
                  optionValue = CodeSystem.LANGUAGE_EN;
            }
            if(CodeSystem.LOCALE_CN.equals(localeList[i])) {
                  optionValue = CodeSystem.LANGUAGE_CN;
            }
            if(CodeSystem.LOCALE_FR.equals(localeList[i])) {
                  optionValue = CodeSystem.LANGUAGE_FR;
            }
            if(CodeSystem.LOCALE_AR.equals(localeList[i])) {
                  optionValue = CodeSystem.LANGUAGE_AR;
                  dir = "rtl";
                  langList3                                                                    =
replaceString(replaceString(langList3,")","&#8207;)" ),"(","&#8207;(");
            }
      %>
  <option dir="<%=dir%>" value=<%=optionValue %>
<%if(optionValue.equals(userLocale)) {out.print("selected");} %>>
  <div align="center" dir=<%=dir%> >
<%=replaceString(replaceString(langList3,")",bidiForceString+")" ),"(",bidiForceString+ "(")%>
</div ></option>
<%
} %>
/select>
```

### *7.6.8 Time Zone*

Use JDK directly.

```
import java.util.TimeZone;
```

### *7.6.9 Date format*

Use JDK instead

```
import java.sql.Timestamp;
import java.text.DateFormat;
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.Locale;
import java.util.StringTokenizer;
import java.util.TimeZone;
```

### *7.6.10 Greeting*

```
<%@ page import="com.ibm.gcl.gbo.greeting.GreetingGBO" %>
GreetingGBO greeting = GreetingGBO.getInstance(locale);
if (title!=null&&!title.equals("")){
        int honorificId = Integer.parseInt(title);
    greetStr = "<tr bgcolor='#A5DBFC' class='Weathertable'><td height='19' align='center' nowrap class='header2'><table>" + greeting.getGreetingMsg(name,honorificId) + "</table><br></td></tr>";
} else {
    greetStr = "<tr bgcolor='#A5DBFC' class='Weathertable'><td height='19' align='center' nowrap class='header2'><table>" + greeting.getGreetingMsg(name,locale) + "</table><br></td></tr>";
}
```

### *7.6.11 Name*

```
<%@ page import="com.ibm.gcl.gbo.name.*" %>
<%
int GBO_NAME_TEMPLATE = 9;

                try {

                        NameGBO nameGBO = null;
                        nameGBO = NameGBO.getInstance(locale);
                        NameTemplate nameTemplate = new NameTemplate(NameTemplate.HTML, GBO_NAME_TEMPLATE);
                        nameTemplate.setHasTable(false);
                        out.println(nameGBO.genModifyLayout(nameArr, vHonorificId, GBO_NAME_SN, nameTemplate));
```

```
                    } catch (Exception e) {
                        out.println("<TR>Error when loading NameGBO</TR>");
                    }
%>
```

### 7.7 Bidi in portal

We will introduce how to support bidi in portal project.
We will use wps tag to support bidi language (ar_EG and iw_IL) in portal project.

#### *7.7.1 Bidi tag in portal taglib*

Please import the taglib first

```
<%@ taglib uri="/WEB-INF/tld/engine.tld" prefix="wps"%>
```

The use the tag wps:bidi in the jsp file.

```
<wps:bidi is="rtl"> </wps:bidi>
```

#### *7.7.2 Define the bidi attributes in the Bidi.jsp*

We can define some bidi variables in the Bidi.jsp. Such as dir and bidiForceString.
Here are the contents of Bidi.jsp

```
<%--
    The following bit of code is to make BIDI support a little easier.
    These variables are meant to suppliment the <wps:bidi ...> tag.
    There are situations where it is easier to use these variables instead of the tag itself.
--%>
<%@ taglib uri="/WEB-INF/tld/engine.tld" prefix="wps"%>
<%
    String bidiAlignRight = "right";    // BIDI sensitive value for align="right" attribute
    String bidiAlignLeft  = "left";     // BIDI sensitive value for align="left" attribute
    String bidiDirAttr    = "";         // BIDI sensitive value for dir="rtl" attribute
    String bidiImageRTL   = "";         // BIDI sensitive value for adding _rtl to a graphic
                String dir="ltr";
    String bidiForceString = "";
    boolean isRTL = false;
%>
<wps:bidi is="rtl">
    <%
        bidiAlignRight = "left";      // BIDI sensitive value for align="right" attribute
        bidiAlignLeft  = "right";         // BIDI sensitive value for align="left" attribute
        bidiDirAttr    = "dir=\"rtl\"";   // BIDI sensitive value for dir="rtl" attribute
```

```
        bidiImageRTL    = "_rtl";          // BIDI sensitive value for adding _rtl to a graphic
        dir="rtl";
                bidiForceString = "&#8207;";    //Force bidi
                        isRTL = true;

    %>
</wps:bidi>
```

*7.7.3 Include the Bidi.jsp in your portal jsp file.*

```
<%@ include file="Bidi.jsp"%>
```

*7.7.4 Now we can use the bidi attributes deifined in the Bidi.jsp.*

```
<table width="1000" <%=bidiDirAttr%>    border="0" cellspacing="0" cellpadding="0">
<td class="td5<%=dir %>"    colspan="3" ><lp:value key="TITLE_REGISTERUSER"/></td>
<%=replaceString(replaceString(langList3,")",bidiForceString+")" ),"(",bidiForceString+ "(")%>
```

*7.7.5 Support bidi in XUL function*

We will use the attribute **dir** with value **normal** and **reverse** for XUL tag.

```
<window id="CalculateVolumetricWeight" title="CalculateVolumetricWeight"
      xmlns:html="http://www.w3.org/1999/xhtml"
      xmlns="http://www.mozilla.org/keymaster/gatekeeper/there.is.only.xul"
      style="background-color: #F9E9CC">

<hbox dir="reverse">
   <html:input id="width" maxlength="6" dir="RTL"/>
    <label hidden="true"/>
</hbox>
```

*7.7.6 Bidi in portlet class*

Sometime we need to use bidi in portlet class, such as composing an email in businessBean.
Firstly we need to know if the locale is bidi language, which is ar_EG or iw_IL. We have two methods.
One is comparing the locale. For example:

```
boolean bidi    = false;
String bidiForceString = "";
If(“ar_EG”.equals(userLocale) || “iw_IL”.equals(userLocale)){
```

```
    bidi = true;
        bidiForceString = "&#8207;"
}
```

Another is getting the bidi attributes from session.

```
String bidi = request.getPortletSession().getAttribute("bidi");
String bidiForceString = ""
If("yes".equals(bidi)){
    bidiForceString = "&#8207;"
}
```

### *7.7.7 Bidi in stylesheet*

Sometime we may need to modify the stylesheet to support bidi.

For example.

```
.td5{
    border-left: 5px solid #B5B5B5;
    color: #B97400;
        padding-left: 10px;
    background: White url(bg_map2.gif);
}
.td5rtl{
    border-right: 5px solid #B5B5B5;
    color: #B97400;
        padding-right: 10px;
    background: White url(bg_map2.gif);
}
```

We will use a bidi variable defined in Bidi.jsp like the following.

```
<td class="td5<%=dir %>"    colspan="3" ><lp:value key="TITLE_REGISTERUSER"/></td>
```

### *7.7.8 Difference of bidi support between in IE6.0 and Mozilla Firefox V1.0.7*

**1.1. Source Code in jsp**

```
…
String bidi = "";
String langList3 = ……      // Get the language here
if(CodeSystem.LOCALE_AR.equals(localeList[i])) {
    optionValue = CodeSystem.LANGUAGE_AR;
        dir = "rtl";
```

```
        langList3 = replaceString(replaceString(langList3,")","&#8207;)" ),"(","&#8207;(");
}
…
<option dir="<%=dir%>" value=<%=optionValue %> <%if(optionValue.equals(userLocale))
{out.print("selected");} %>> <div align="center" dir=<%=dir%> >
<%=replaceString(replaceString(langList3,")",bidiForceString+")" ),"(",bidiForceString+
"(")%></div ></option>
…
```

**8.2. The result in the browser IE6,0**

In locale zh_CN(Simplified Chinese)


In Locale ar_EG(Egypt)


**8.3. The result in the Mozilla Firefox V1.0.7**

In locale zh_CN(Simplified Chinese)

**You can see the it's different with IE6.0.**


In Locale ar_EG(Egypt)

## 7.8 Encoding of portlet

If the encoding is not right, the portal page will not be right to users. The words may be error code.

### *7.8.1 JSP page encoding*

To ensure the page rendering to user in right,please add the following line into your jsps.

```
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8" session="true"%>
```

### *7.8.2 Localization Pack file*

When doing the LP translation, please ensure you save the original string file as UTF-8 encoding. Then you need to convert the **UTF-8 format** file to **\u format** file.

2.1. Original UTF-8 file user.properties:

```
PORTLET_ONLINETOOLS=在线工具
```

2.2. Run the native2ascii command in command console: (Ensure you have JDK in your path)

```
C:\native2ascii –encoding UTF-8 user.properties \ASCII\user.properties
```

2.3. ASCII file is like the following.

```
PORTLET_ONLINETOOLS=\u5728\u7ebf\u5de5\u5177
```

### *7.8.3 Multilingual string handling in Java code, such as sending email.*

Please ensure you UTF-8 to handle all the strings.

1. You can check if the portal page is rendered using UTF-8

Right Click in the portal page -> **Select Encoding** – should be **UTF-8**

## 7.9 Encoding of portal database

This piece describes how to change the default encoding of WPS DB to avoid the wrong character displaying.

By default, WebSphere Portal Server has the globalization features and supports multilingual applications. However, there could be some wrong characters on the homepage of an application, e.g., when access http://gwpcwps.cn.ibm.com/wps/portal, you are supposed to see some strange characters like '?' in some cases.

The cause of the problem is the encoding setting of WPS DB. Explore the WPSDB in the DB2

Control center, and open the MARKUP_DESC_LOD table. Check the CHARSET of your supported locale, if it is not UTF-8, update the filed with UTF-8.

# 8. Artwork in Portal

Artwork is very important in portal project. In a globalization portal project, we may need to modify the stylesheet to meet the requirement in each locale.

## 8.1 UI design

Before we can write the codes, we need to do the UI design. UI design is very important to a portal project. A good design can save you much time.

Here are some principles we can follow in UI design.

### *8.1.1 Naming conversion*

Please use the word associated for html file name, such as GenerateReport.html.

### *8.1.2 Deal with UI pages of different roles*

If there several roles in your portal website, and they have the similar UI pages, please design only one page for them and put necessary notes to the page. In addition, notes must be outstanding so that the artwork team members are able to know what you mean. Some common items should be put in an independent file, for example bidi info and stylesheet.

### *8.1.3 Keep the same style for all of UI pages.*

Keep the same style for all of UI pages is important to the overall feeling of the portal website.

### *8.1.4 List all the error messages on the UI pages.*

The error messages will be collected into localization pack files and send to TSC for translation..

### *8.1.5 As for some dynamic contents, enough explanation should be shown on the page.*

The explanation would be help for developers in the development.

### *8.1.6 As for the JSP components, please put notes indicating the type of the used component.*

The explanation would be help for developers in the development.

### *8.1.7 Except the homepage, it is better to put a "back" button on every page.*

A "Back" button would be helpful when user wants to input the data again or want to view the previous page.

### *8.1.8 For some special input, it is better to write some limitation notes besides the input.*

For example, the length the password must be more than 6. This would be helpful for developer to validate the user input in portlet.

*8.1.9 If there are many inputs in a page, a "reset" button is a must.*

A "reset" button can allow use to reset all the input and input the data again.sa

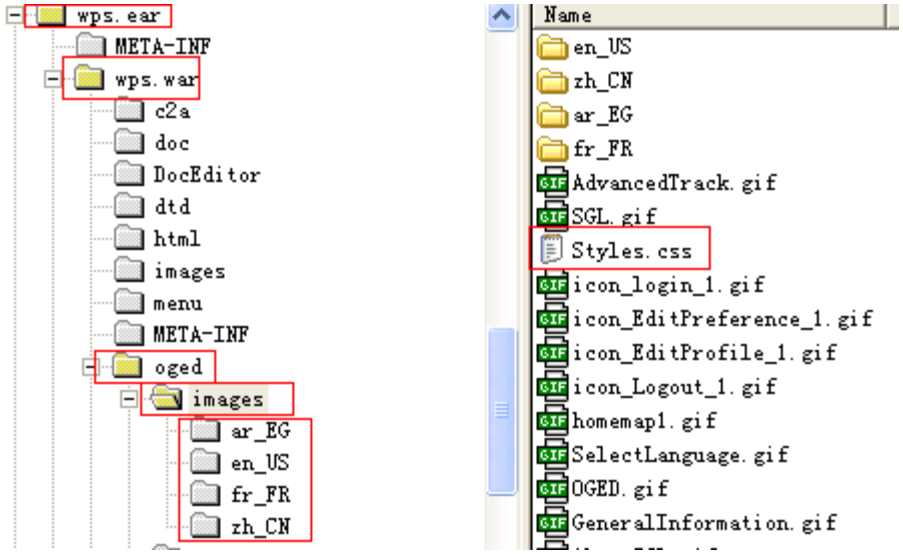
*8.1.10 Convert html files to JSP files.*

Firstly, you need to rename the html files name to JSP file name and add them into portal projects.
Then you collect all the strings into localization pack files from JSP.
Now you can start coding.

### 8.2 Images

We can add the images into project: **PortalProject\WebContent\images**
Or we can separate them from project and place the images into wps.war directory like the following.
**/opt/WebSphere/AppServer/installedApps/ogedwemp/wps.ear/wps.war/oged/images**

If you want to use different image for each locale, please use the subdirectory for the locale as following.




**Note: Please pay attention to bidi language; some images may need to be converted from right to left. Here is a sample.**

### 8.3 Apply stylesheet

To make the portal website more beautiful, we need to apply stylesheet. The stylesheet can be the same directory with images.

Please pay attention to the tag **.td1rtl** in the following sample , which is used to support bidi language. Here is the sample of stylesheet.

```
.td1{
	border-left: 5px solid #c1c1c1;
	font-weight: bold;
	color: #727272;
	padding: 5px 10px 5px 10px;
	border-bottom: 1px solid #c1c1c1;
	border-right: solid 1px #c1c1c1;
	background: #eeeeee;
}
.td1rtl{
	border-right: 5px solid #c1c1c1;
	font-weight: bold;
	color: #727272;
	padding: 5px 10px 5px 10px;
	border-bottom: 1px solid #c1c1c1;
	border-left: solid 1px #c1c1c1;
	background: #eeeeee;
}
```

### 8.4 Coordination between artwork team and portal team

Good communication between artwork team and portal team can contribute more to the success of a portal project.

Of course, to avoid frequent UI changes, a good UI design is very important.

If we have to change the UI pages, please work closely with artwork team and make our requirement clear. If the changes don't affect our java code, it would be the best.

Notes: Please pay more attention to the theme of portal for it's the overall impression of the portal website.

## 9. Comparison of several types of portlets

You will know the differences of several portal projects, which including IBM API portlet, JSR168 portlet, JSF portlet, XDIME portlet and etc.

Here I list the sample code in our previous projects to show the difference of each type portal project.

### 9.1 IBM API Portlet

*9.1.1 IBM API Portlet- Portlet*

```
import org.apache.jetspeed.portlet.*;
public class BookAir extends PortletAdapter implements PortletTitleListener, ActionListener {

public void doView(PortletRequest request, PortletResponse response) throws PortletException, IOException {
      PortletContext context = getPortletConfig().getContext();
      PortletSession session = request.getPortletSession();
      String jspFileName = (String) session.getAttribute(SESSION_PAGE_TYPE);
       context.include(jspFileName, request, response);
}
public void actionPerformed(ActionEvent event) throws PortletException {
    PortletRequest request = event.getRequest();
    PortletSession session = request.getPortletSession();
    String action = event.getActionString();
    if (ACTION_CANCEL_BOOKING.equals(action)){
    session.setAttribute(SESSION_PAGE_TYPE,PAGE_HOTEL_HISTORY_CACLEL);
    }
}
}
```

*9.1.2 IBM API Portlet- JSP*

```
<%@ page language="java“ contentType="text/html; charset=UTF-8"pageEncoding="UTF-8"%>
<%@ taglib uri="/WEB-INF/tld/portlet.tld" prefix="portletAPI" %>
<%@ taglib uri="/WEB-INF/tld/engine.tld" prefix="wps" %>
<%@ taglib uri="/WEB-INF/tld/lp.tld" prefix="lp" %>
<lp:load file="search.properties"/>
<portletAPI:init />

<form name="<portletAPI:encodeNamespace value='SearchReturn'/>" action=<%=(String)request.getAttribute(FederalSearchParameters.SEARCH_RETURN_ACTION_URI)%>
method ="POST">
```

```
<lp:value key="case_search_page_title"/>

<input type="button" name="search" value="<lp:value key="search"/>"
onclick="document.<portletAPI:encodeNamespace value='search'/>.submit();">

</form>
```

*9.1.3 IBM API Portlet – portlet.xml*

```
<portlet id="BookHistory" href="WEB-INF/web.xml#Servlet_1063354122906"
major-version="1" minor-version="0">
        <portlet-name>BookHistory</portlet-name>
        <supports>
            <markup name="html">
            </markup>
            <markup name="kiosk">
                <view />
            </markup>
            <markup name="palm">
                <view />
            </markup>
            <markup name="pda">
                <view />
            </markup>
            <markup name="wap">
                <view />
            </markup>
        </supports>
    </portlet>
    <concrete-portlet href="#BookHistory">
        <portlet-name>BookHistory</portlet-name>
        <default-locale>en</default-locale>
        <language locale="en">
            <title>BookHistory</title>
            <title-short></title-short>
            <description></description>
            <keywords></keywords>
        </language>
    </concrete-portlet>
```

*9.1.4 IBM API Portlet– web.xml*

```
<servlet id="Servlet_1063354122906">
```

```
            <servlet-name>BookHistory</servlet-name>
            <display-name>BookHistory</display-name>
            <servlet-class>com.ibm.gcl.gtod.portlet.BookHistory</servlet-class>
        </servlet>

 <servlet-mapping id="ServletMapping_6">
            <servlet-name>BookHistory</servlet-name>
            <url-pattern>/BookHistory/*</url-pattern>
    </servlet-mapping>
```

## 9.2 JSR168 Portlet

### *9.2.1 Introducing JSR 168*

The Java Standardization Request 168 (JSR 168) defines a portlet specification, including a contract between the portlet container and the portlet. JSR 168 is defined by the Java Community Process (JCP). The JSR 168 was co-led by IBM and Sun and had a large Expert Group that helped to create the final version which is now available. This Expert Group consisted of Apache Software Foundation, Art Technology Group Inc.(ATG), BEA, Boeing, Borland, Citrix Systems, Fujitsu, Hitachi, IBM, Novell, Oracle, SAP, SAS Institute, Sun, Sybase, Tibco, Vignette. More details about this JSR can be found at http://jcp.org/en/jsr/detail?id=168.

### *9.2.2 JSR168 Portlet - Portlet*

```
public class JSR168Portlet extends GenericPortlet {
public void doView(RenderRequest request, RenderResponse response) throws PortletException,
IOException {
response.setContentType(request.getResponseContentType());
      if(null == JSP_NAME) JSP_NAME = “jsppage1";
      // Invoke the JSP to render
      PortletRequestDispatcher rd =
getPortletContext().getRequestDispatcher(getJspFilePath(request, JSP_NAME));
            rd.include(request,response);
    }
 }
public void processAction(ActionRequest request, ActionResponse response) throws
PortletException, java.io.IOException {
PortletSession session = request.getPortletSession();
if(request.getParameter("initiateNext1")!=null){
    //Do something
   JSP_NAME = “jsppage2";
   PortletPreferences prefs = request.getPreferences();
   prefs.setValue(request.getParameter("name"),request.getParameter("value"));
   prefs.store();
```

```
 }
}
```

### *9.2.3 JSR168 Portlet - JSP*

```
<%@ taglib uri="http://java.sun.com/portlet" prefix="portlet"%>
<%@ page language="java" contentType="text/html; charset=UTF-8"
      pageEncoding="UTF-8" session="false"%>
<portlet:defineObjects/>
<%
PortletPreferences prefs = renderRequest.getPreferences();
   Enumeration e = prefs.getNames();

   while (e.hasMoreElements()) {
      String name = (String)e.nextElement();
      String value = prefs.getValue(name,"test");
}
%>
<FORM ACTION="<portlet:actionURL/>" METHOD="POST">
<input name="initiateNext1" type="submit" value="Next">
</FORM>
```

### *9.2.4 JSR168 Portlet– portlet.xml*

```
<portlet>
        <portlet-name>Xdime</portlet-name>
        <display-name>Xdime portlet</display-name>
        <display-name xml:lang="zh">Xdime portlet</display-name>
        <portlet-class>xdime.XdimePortlet</portlet-class>
        <init-param>
            <name>wps.markup</name>
            <value>html,xdime</value>
        </init-param>
        <expiration-cache>0</expiration-cache>
        <supports>
            <mime-type>text/html</mime-type>
            <portlet-mode>view</portlet-mode>
        </supports>
        <supports>
            <mime-type>x-application/vnd.xdime+xml</mime-type>
            <portlet-mode>view</portlet-mode>
        </supports>
```

```
        <supported-locale>zh</supported-locale>
        <resource-bundle>xdime.nl.XdimePortletResource</resource-bundle>
        <portlet-info>
            <title>Xdime portlet</title>
        </portlet-info>
    </portlet>
```

## 9.3 Faces Portlet

### *9.3.1 JavaServer Faces (JSF)*

JavaServer Faces is a GUI framework for developing J2EE Web applications (JSR 127). It includes reusable user interface components, input validation, state management, server-side event handling, page lifecycle management,
accessibility, and internationalization. Faces-based application development can be applied to portlets, similar to the way that Faces development is implemented in Web applications. Similar to Struts, there are many wizards to help you with Faces development. Both WebSphere Portal V5.0.2.2 and V5.1support JavaServer Faces.
There are certain limitations to Faces portlet development in the current release. Service Data Objects (SDO), formerly referred to as WebSphere Data Objects (WDO), are limited to prototyping purposes only. Applications that rely on SDOs should be limited in a production environment. File upload and binary download are not supported for Faces components. Finally, document root-relative URLs are not supported for images.

Refer to the Rational Application Developer Faces documentation in the InfoCenter for usage details. Alternatively you can refer to the following web site: http://www.jcp.org/en/jsr/detail?id=127

### *9.3.2 Faces Portlet - Portlet*

```
public class SearchProduct extends PageCodeBase {
  protected HtmlCommandExButton getButton1 () {
        if (button1 == null) {
            button1 = (HtmlCommandExButton)    findComponentInRoot("button1");
        }
        return button1;
    }

public String doSearchAction () {
javax.portlet.PortletRequest portletRequest = (javax.portlet.PortletRequest) facesContext
                .getExternalContext().getRequest();
        searchProductDataBean = this.getSearchProductDataBean();

getSessionScope().put("productListContainerDataBean",productListContainerDataBean);
```

```
            return "success";
}

public SearchProductDataBean getSearchProductDataBean () {
}

}
```

### *9.3.3 Faces Portlet - JSP*

```
<%@taglib uri="http://java.sun.com/jsf/core" prefix="f"%>
<%@taglib uri="http://java.sun.com/portlet" prefix="portlet"%>
<%@taglib uri="http://java.sun.com/jsf/html" prefix="h"%>
<%@taglib uri="http://www.ibm.com/jsf/html_extended" prefix="hx"%>
<%@taglib uri="/WEB-INF/tld/lp.tld" prefix="lp"%>
<%@page language="java" contentType="text/html; charset=UTF-8"
      pageEncoding="UTF-8" session="true"%>
<portlet:defineObjects />
<lp:load file="productsearch.properties" />
<f:view>
<%if (locale.equals("zh_CN")) {%>
      <f:loadBundle basename="LocalizationPack/zh_CN/productsearch"
            var="lpnew" />
      <%} else if (locale.equals("fr_FR")) {%>
      <f:loadBundle basename="LocalizationPack/fr_FR/productsearch"
            var="lpnew" />
      <%} %>
<hx:scriptCollector id="scriptCollector1">
<%--                              jsf:pagecode                              language="java"
location="/JavaSource/com/ibm/sgl/gpim/actionbeans/SearchProduct.java"          --%><%--
/jsf:pagecode --%>
            <h:form styleClass="form" id="SearchProduct_form1">
<hx:commandExButton              styleClass="commandExButton“              id="button1"
value="#{lpnew.COMMANDLINK_SEARCHBYPRODUCT}“  action="#{pc_SearchProduct.do
SearchAction}"></hx:commandExButton>
            </h:form>
      </hx:scriptCollector>
</f:view>
```

### *9.3.4 Faces Portlet – portlet.xml*

```
<portlet>
            <description>This is SearchProduct portlet base</description>
            <description xml:lang="en-US">This is SearchProduct portlet en_US</description>
```

```
        <portlet-name>SearchProduct</portlet-name>
        <display-name>SearchProduct portlet</display-name>
        <display-name xml:lang="en-US">
            SearchProduct portlet en_US
        </display-name>
        <portlet-class>
            com.ibm.faces.webapp.FacesGenericPortlet
        </portlet-class>
        <init-param>
            <name>com.ibm.faces.portlet.page.view</name>
            <value>/html/SearchProduct.jsp</value>
        </init-param>
        <init-param>
            <name>wps.markup</name>
            <value>html</value>
        </init-param>
        <expiration-cache>0</expiration-cache>
        <supports>
            <mime-type>text/html</mime-type>
            <portlet-mode>view</portlet-mode>
        </supports>
        <supported-locale>en_US</supported-locale>
        <resource-bundle>lp.nl.SearchProductResource</resource-bundle>
        <portlet-info>
            <title>SearchProduct portlet</title>
        </portlet-info>
    </portlet>
```

### *9.3.5 Faces Portlet – faces-config.xml*

```
    <managed-bean>
      <managed-bean-name>pc_SearchProduct</managed-bean-name>
      <managed-bean-class>
          com.ibm.sgl.gpim.actionbeans.SearchProduct
      </managed-bean-class>
      <managed-bean-scope>request</managed-bean-scope>
    </managed-bean>
    <managed-bean>
        <managed-bean-name>searchProductDataBean</managed-bean-name>
        <managed-bean-class>
            com.ibm.sgl.gpim.databeans.SearchProductDataBean
```

```
            </managed-bean-class>
            <managed-bean-scope>session</managed-bean-scope>
          </managed-bean>
<navigation-rule>
            <from-view-id>/html/SearchProduct.jsp</from-view-id>
            <navigation-case>
                  <from-action>
                        #{pc_SearchProduct.doSearchAction}
                  </from-action>
                  <from-outcome>success</from-outcome>
                  <to-view-id>/html/ProductSearchProductList.jsp</to-view-id>
            </navigation-case>
            <navigation-case>
                  <from-action>
                        #{pc_SearchProduct.doSearchAction}
                  </from-action>
                  <from-outcome>failure</from-outcome>
                  <to-view-id>/html/ProductSearchResultNULL.jsp</to-view-id>
            </navigation-case>
</navigation-rule>
```

## 9.4 XDIME Portlet

### *9.4.1 Introduction of WEMP and XDIME*

WebSphere Everyplace Mobile Portal (hereafter called WEMP) is a key component of WebSphere Everyplace Service Delivery. It extends WebSphere Portal software with technology to meet the requirements of mobile and wireless service providers by delivering the promise of 'write once, render many' to mobile devices through device-independent authoring and allows new devices to be introduced without changes to the style guide or impact to the existing applications or products.

XML Device-Independent Markup Extensions (hereafter called XDIME) is an abstract, device-independent markup language. XDIME can be processed and converted by the components of WEMP, such as MCS (Multi Channel Server), into a wide variety of markup languages, such as WML 1.1, WML 1.3, XTHML Basic, XHTML-MP, HTML 3.2, HTML 4.0, and others. So that developers can just write one version XDIME-enabled source files for a wide variety of mobile devices.

### *9.4.2 WEMP Portlet–Portlet*

```
public class XdimePortlet extends GenericPortlet {
public void processAction(ActionRequest request, ActionResponse response) throws
PortletException, java.io.IOException {
PortletSession session = request.getPortletSession();
if(request.getParameter("initiateNext1")!=null){
```

```
        JSP_NAME = "InitiateShipment2";
}
public void doView(RenderRequest request, RenderResponse response) throws PortletException,
IOException {
response.setContentType(request.getResponseContentType());
        if(null == JSP_NAME) JSP_NAME = "InitiateShipment";
        // Invoke the JSP to render
        PortletRequestDispatcher                                rd                                =
getPortletContext().getRequestDispatcher(getJspFilePath(request, JSP_NAME));
        rd.include(request,response);
    }
  }
}
```

### *9.4.3 WEMP Portle – html JSP*

```
<%@ taglib uri="http://java.sun.com/portlet" prefix="portlet"%>
<%@ page language="java" contentType="text/html; charset=UTF-8"
        pageEncoding="UTF-8" session="false"%>
<portlet:defineObjects/>

<FORM ACTION="<portlet:actionURL/>" METHOD="POST">
<input name="initiateNext1" type="submit" value="Next">
</FORM>
```

### *9.4.4 WEMP Portlet– XDIME JSP*

```
<%@ page session="false" contentType="x-application/vnd.xdime+xml" %>
<%@taglib uri="http://java.sun.com/portlet" prefix="portlet" %>
<portlet:defineObjects/>
<canvas layoutName="/layout.mlyt" type="portlet" >
<pane name = "sample_pane">
    <p>Initiate shipment request</p>
</pane>
<xfform name="testForm" method="POST" action="<portlet:actionURL/>" >
        <xftextinput      name="shipmentContent"      type="text"      caption="Shipment      content:"
captionPane="contentpane" entryPane="contentpane"/>
        <xfaction type="submit"    name="initiateNext1"
            caption="Next"    captionPane="buttonpane" entryPane="buttonpane" />

</xfform>
</canvas>
```

## 9.5 Comparison between IBM Portlet and JSR168 Portlet

### *9.5.1 Deciding which API to use*

As the IBM portlet API will continue to be supported in future versions of WebSphere Portal, there is no immediate need to migrate existing portlets unless the portlet is required to interoperate with other portals that support JSR 168. For new portlets, consider using JSR 168 when the functionality it provides is sufficient for the portlet's needs or when the portlet is expected to be published as Web Service for Remote Portlets (WSRP) service. As the JSR 168 and WSRP are closely aligned, it is possible to publish a JSR 168 portlet as a WSRP service. If the portlet needs more functionality than that provided by JSR 168, the IBM portlet API should be used.

### *9.5.2 Future directions for WebSphere Portal JSR 168 support*

IBM plans to provide further enhancements to the JSR in follow-on versions to make the JSR 168 portlet API as useful as the current IBM counterpart.

### *9.5.3 The comparison of IBM API portlet and JSR168 portlet*

| | **IBM Portlet API** | **JSR168** |
|---|---|---|
| **Request,response** | PortletRequest, PortletResponse, ActionEvent | RenderRequest, RenderResponse ActionRequest, ActionResponse |
| **Life cycle** | ActionPerformed() | ProcessAction() |
| **Methods to create URI to the portlet** | PortletResponse.createURI() | RenderResponse.createActionURL() and RenderResponse.createRenderURL() |
| **Including JSPs** | PortletContext.include() | PortletRequestDispatcher.include() |
| **JSP tag library** | <portletAPI:init/> - | portlet:defineObjects/> |
| | <portletAPI:createURI/> | PortletResponse.createURI() |
| | <portletAPI:createURI/> | <portlet:renderURL/> <portlet:actionURL/> |
| | <portletAPI:encodeNamespace/> | <portlet:namespace/> |
| **Namespace encoding** | PortletResponse.encodeNamespace() | RenderResponse.getNamespace() |
| | | |

For more, please refer to:
http://www-128.ibm.com/developerworks/websphere/library/techarticles/0312_hepper/hepper.html

## 10. Portal Resource Center

Here are some useful resourcs of portal.

### 10.1 WebSphere Portal for Multiplatforms

http://www.ibm.com/software/genservers/portal

### 10.2 WebSphere Portal Express

http://www.ibm.com/software/genservers/portalexpress/

### 10.3 WP510 Info Center

http://publib.boulder.ibm.com/pvc/wp/510/ent/en/InfoCenter/

### 10.4 RAD V6 Portlet Application Development and Portal Tools

http://w3.itso.ibm.com/redpieces/abstracts/sg246681.html?Open

### 10.5 IBM WebSphere Portal for Multiplatforms V5 Handbook

http://www.redbooks.ibm.com/abstracts/sg246098.html

### 10.6 IBM WebSphere Portal V5 A Guide for Portlet Application Development

http://www.redbooks.ibm.com/abstracts/sg246076.html

### 10.7 WebSphere Portal library:

http://www.ibm.com/software/genservers/portal/library/

### 10.8 WebSphere Portal zone in DeveloperWorks

http://www.ibm.com/developerworks/websphere/zones/portal/

### 10.9 WebSphere Portal Forum

http://www.ibm.com/developerworks/forums/dw_forum.jsp?forum=168&cat=9

### 10.10 IBM user groups

http://www.ibm.com/developerworks/websphere/usergroups/index.html

### 10.11 JSR 168 Specification:

http://www.jcp.org/en/jsr/detail?id=168

### 10.12 Comparing JSR 168 to the IBM portlet API

http://pvcid.raleigh.ibm.com/wpf/ic/510/ent/en/InfoCenter/wps/jsrcpr.html
http://www-128.ibm.com/developerworks/websphere/library/techarticles/0312_hepper/hepper.html

### 10.13 Tags used by the portal JSPs

http://publib.boulder.ibm.com/pvc/wp/510/ent/en/InfoCenter/wps/dgn_ptltld.html

## 11. Appendix 1: Abbreviations and acronyms

RAD – Rational Application Developer
WPS – WebSphere Portal Server
RSA – Rational Software Architecture
G11n - Globalization
WEMP – WebSphere Everyplace Mobile Portal
XDIME –XML Device Independent Markup Extension
LP – Localization Pack
JSR –Java Specification Request
JSF – JavaServer Faces
JCP - Java Community Process

## 12. Appendix 2: Acknowledge

I want to say thanks to my colleagues, Elton, Terry, Martin, Benny, and Catherine. With your kindly help, I can present this paper, a series of best practices to the portal developers and administrators.

Also I'll say thanks to you, my dear reader. Thank you for your time in this document, hope it can help you in your portal development. If you have any question, please don't hesitate to contact me at wanghuah@cn.ibm.com.


**<End>**