

# How to...

## Extend Language Support in SAP Enterprise Portal 5.0

SAP ENTERPRISE PORTAL 5.0

PUBLIC

---

---

### ASAP “How to...” Paper



Applicable Releases: EP 5.0 SP5

September 2003

## Contents

<b>1</b>	<b>INTRODUCTION</b> .....	<b>1</b>
1.1	Implementation Considerations.....	1
1.2	Additional Documentation.....	1
1.3	SAP Notes.....	2
<b>2</b>	<b>OVERVIEW</b> .....	<b>3</b>
2.1	Prerequisites.....	4
<b>3</b>	<b>EXTENDING LANGUAGE SUPPORT TO THE .NET INFRASTRUCTURE OF THE PORTAL..</b>	<b>5</b>
3.1	Workflow: Localizing .NET Infrastructure of the Portal .....	5
3.2	Modifying the iViewServer Database Tables .....	6
3.2.1	Adding a Language to the P310_SUPPORTED_LANGUAGES Table .....	6
3.2.2	Translating Text in the Database Tables of the Portal (iViewServer).....	7
<b>4</b>	<b>EXTENDING LANGUAGE SUPPORT TO THE JAVA INFRASTRUCTURE OF THE PORTAL.</b>	<b>9</b>
4.1	Workflow: Localizing the Java Infrastructure of the Portal .....	9
4.2	Text Resources for Portal Content Directory (PCD) Objects.....	10
4.2.1	Translating Texts for PCD Objects.....	10
4.2.2	File Naming Conventions.....	11
4.2.3	Adding Text Resources to a PAR File.....	12
4.3	Adding Resource Bundles to Your Java Code.....	12
<b>5</b>	<b>VERIFYING SUPPORT FOR THE ADDED LANGUAGE</b> .....	<b>13</b>
<b>6</b>	<b>APPENDIX: SAMPLE JAVA CODE</b> .....	<b>14</b>

# 1 Introduction

SAP Enterprise Portal supports locale-specific settings that enable you to display user interface elements in six languages on the same system. The supported languages are: German, English, Spanish, French, Italian, and Portuguese.

The flexible architecture and infrastructure of SAP Enterprise Portal allow you to extend support to any other language defined by the ISO 8859-1 (Latin-1) standards.

This document explains how to extend the portal infrastructure to support additional languages for portal end users, and provides information on how to translate text elements to the supported languages.

**Warning:** Do NOT incorporate additional language and locale-specific settings unless end users need user interface elements in a locale and language other than the ones supported in the Enterprise Portal.

This document can be used by anyone who wants to add and enable support for a language in the portal, specifically:

- Developers
- Support
- Consultants
- Partners
- Customers

## 1.1 Implementation Considerations

This document is relevant if:

- You intend to extend the portal infrastructure to support languages in the ISO 8859-1 (Latin-1) standards in addition to the default languages.
- You want to enable translation of content provided to you by a content vendor in a specific locale whose language is not one of the default languages of SAP Enterprise Portal.

## 1.2 Additional Documentation

This document explains how to extend support for a language in addition to the supported default languages. Also, it explains one method of translating text elements that are displayed in the portal interface.

You can use portal tools to translate text elements once you extend support to an additional language. You can find detailed information and explanation of portal tools for editing, translating, and maintaining text strings in the supported languages at: [help.sap.com/ep](http://help.sap.com/ep)

1. Under *Administration, Developer and End User Documentation*, go to *EP 5.0 SP5*.
2. Open *SAP Enterprise Portal Documentation > Administration Guide > Translating Portal Content Metadata*.

Additional information on content translation is located at:  
*SAP Enterprise Portal Documentation > Developer Guide > Developing .NET iViews > HRNL Tags*.

### 1.3 SAP Notes

SAP Notes provide supplementary information relating to the portal. This information is comprised of limitations, workarounds, known problems, and additions to documentation made after publication.

Find SAP Notes on SAP Service Marketplace at: [service.sap.com/epinstall](http://service.sap.com/epinstall), under *Enterprise Portal 5.0 > SAP Notes*

## 2 Overview

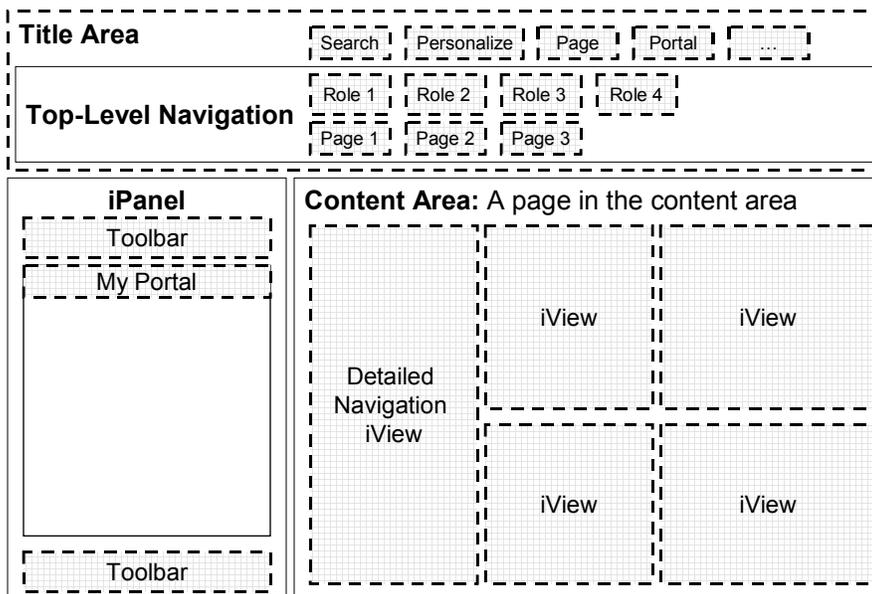
This section provides an overview of how to extend support to additional languages in the portal.

Extending support for localization requires modification to both the .NET and Java infrastructure of the portal.

To extend the .NET infrastructure of the portal to support additional languages for content objects, you configure the Portal database.

To extend the Java infrastructure of the portal to support additional languages for content objects in the Portal Content Directory (PCD), you create resource bundle files that can be referenced from the profile classes of the objects. A resource bundle is a set of related property files that share the same base name and store language and locale-specific descriptions of text for each portal object outside the Java code.

The following illustration is an example of the user interface of an out-of-the-box portal:



The shaded area with dotted lines in the illustration above indicate the areas of the portal user interface to which translated text will apply in any of the supported languages.

Specifically, you can translate and edit the strings in the following areas of the user interface:

User interface	Strings that can be translated
Title Area	Search, Personalize, Page, Portal, Logoff, Add to Favorites, including the text of the tool tips can be translated.
Top-Level Navigation	Text items in this area relate to roles in the level one tabs, and pages and services in the level two tabs. <ul style="list-style-type: none"> <li>• Role: Text representing a Role name, and folder names can be translated.</li> <li>• Page: Text denoting the Page name, description, and tool tip can be translated.</li> </ul>

iPanel	Text representing the name of a toolbar, or Web component can be translated. For example, My Portal. Note that text elements contained within a toolbar cannot be translated.
iViews	Text that represent an iView name, description, and tool tip can be translated. Note that the language in which dynamic information displays in an iView is the responsibility of the content provider.

For more information on the layout of the portal user interface, refer to the topic “The Enterprise Portal Workplace,” at: [help.sap.com/lep](http://help.sap.com/lep) >Administration, Developer and End User Documentation > EP 5.0 SP5, and navigate to: *SAP Enterprise Portal Documentation > End User Guide*

## 2.1 Prerequisites

You must have access to the following applications:

1. SAP Enterprise Portal 5.0 SP5 and above

This must be the specific portal installation for which you want to extend the support for a language. First, you work in the portal as an administrator in order to make the changes you want, and later as an end user so that you can verify the changes.

2. Portal Database (MS SQL Server or Oracle Database Server)

You need permissions in the portal database, which has the tables and indexes for the portal with which you want to work.

3. Backup the database before making changes to it.

4. For Java iViews, first you need to prepare text resource files, which you add to the PAR files for portal objects. Then you edit the Java code that enables the portal runtime to reference the text resources.

**Note:** The assumption is that your project includes a translator to translate the text strings into the language to which you want to extend support.

## 3 Extending Language Support to the .NET Infrastructure of the Portal

When you want to extend support to an additional language for content objects in the .NET infrastructure of the portal, you work in the database server that holds the iViewServer tables.

You need administration rights to the relevant portal database server, and make sure that you backup the database before editing it.

To extend support to a language in the portal infrastructure, first, you include information about that language, and then you translate all the texts items of the portal user interface into that language.

The following provides detailed information for extending portal support to additional languages.

### 3.1 Workflow: Localizing .NET Infrastructure of the Portal

#### For Microsoft .NET Framework:

1. From the CultureInfo class in MSDN, get details of the specific language you want to add.

The CultureInfo class enables a multi-language application to display user interface (UI) elements in a specific language. The class consists of the *languagecode* and the *country/regioncode* using ISO standards.

Detailed information, including a list of language names and identifiers that are accepted by this class, exist in MSDN:

[msdn.microsoft.com/library/default.asp?url=/library/en-us/cpref/html/frlrfSystemGlobalizationCultureInfoClassTopic.asp](http://msdn.microsoft.com/library/default.asp?url=/library/en-us/cpref/html/frlrfSystemGlobalizationCultureInfoClassTopic.asp)

In addition to the list, the class explains how .NET manages culture information. From the list in the CultureInfo class, note the following information about the language you want to make available in the portal:

- Culture Name
  - Culture Identifier
  - Language Country-Region
2. Locate the database server with the tables and indexes of the iViewServer. For more information, see “Modifying the iViewServer Database Tables” on page 6.
  3. Backup the database of the portal, before editing it.
  4. Add the appropriate language data as predefined in the CultureInfo class to the tables in the portal system database. For more information, see “Adding a Language to the P310\_SUPPORTED\_LANGUAGES Table” on page 6.
  5. Translate the strings for the various user interface (UI) elements into the added language in the tables.
  6. Restart the portal Web Server (IIS).
  7. Launch the portal to verify the changes you made.

**For detailed information on language settings for Java iViews:** refer to “Extending Language Support to the Java Infrastructure of the Portal” on page 9.

## 3.2 Modifying the iViewServer Database Tables

You need access to the following tables of the portal in the **TTPsys** database:

- **P310\_SUPPORTED\_LANGUAGES**  
Holds the list of languages supported in SAP Enterprise Portal. In this table, add information about the language you intend to support as described in the section “Adding a Language to the P310\_SUPPORTED\_LANGUAGES Table” on page 6.
- **NLSTRANSLATE51**  
Contains a list of all the translated UI strings that are always available in all out-of-the-box portal installations, such as Password, Log-in, Logoff, Preview and names of the month for example, January, February.  
  
These are text elements in the title area of the portal user interface, and other iViews, such as personalization.  
  
For more information on editing this table, see the section “Translating Text in the Database Tables of the Portal (iViewServer)” on page 7.
- **NLSTRANSLATE52**  
Consists of a list of all the translated error messages in the supported portal languages. For more information on editing this table, see the section “Translating Text in the Database Tables of the Portal (iViewServer)” on page 7.
- **NLSTRANSLATE53**  
Has a list of all the translated UI strings of default iViews that are installed with an out-of-the-box portal, such as, Welcome, Collaboration, Knowledge Management, and ESS. For more information on editing this table, see the section “Translating Text in the Database Tables of the Portal (iViewServer)” on page 7.
- **NLSTRANSLATE54**  
Contains a list of all the translated strings that can be changed in an editor in the portal, such as channels, iViews, datasources, and pages.  
  
When you edit existing roles, pages, worksets and user data in the portal, the content of this table is updated to show the changes. For more information on editing this table, see the section “Translating Text in the Database Tables of the Portal (iViewServer)” on page 7.

### 3.2.1 Adding a Language to the P310\_SUPPORTED\_LANGUAGES Table

Open and edit the database table P310\_SUPPORTED\_LANGUAGES. In this table, you see the details of the supported languages, as follows:

- **LANGID**  
Displays the identifier of a specific language.
- **LANGNAME**  
Displays the language name using the letters of that language.  
For example, the language for France is *Français* (written in French letters).
- **LangSYMBOL**  
Displays the language symbol as specified by the ISO standards. For example, the language symbol for French is *fr*. (Note that it is not case sensitive.)

Obtain information on a language from the list of languages in the CultureInfo class. For more information, see the section on “Workflow: Localizing .NET Infrastructure of the Portal ” on page 5.

**To add information about a language to the P310\_SUPPORTED\_LANGUAGES table:**

1. Open the table P310\_SUPPORTED\_LANGUAGES, which is in the **TTPsys** database.

The following is an illustration of the P310\_SUPPORTED\_LANGUAGES table:

langid	langname	langsymbol
-1		
0	Neutral	
7	Deutsche	de
9	English	en
10	Español	es
12	Français	fr
16	Italiano	it
22	Português	pt

The empty cells in the illustration above contain no values.

2. Enter the **decimal** value of the identifier for the language you are adding, in the **LANGID** column.  
For example, when adding the language information for *Croatian*, enter the decimal value 26. (Do not use the hex value 0x001A)
3. Enter the name of the language in the column **LANGNAME**, on the same row as the identifier.  
For example, for *Croatia* the language name is *Croatian*. (It is optional to use characters of the language, such as Croatian characters.)
4. Enter the symbol of the language in the column **LANGSYMBOL**, on the same row as the identifier and the language name.  
For example, the symbol for *Croatian* is *hr*.
5. Close the database table after making the changes.

After adding details of the language as defined in CultureInfo class to the **P310\_SUPPORTED\_LANGUAGES** table, you must specify the language information in the database tables that contain the translated strings of the various user interface (UI) elements in the portal.

The following sections provide information on how to translate text elements into the language for which you extended supported.

### 3.2.2 Translating Text in the Database Tables of the Portal (iViewServer)

You must translate text elements in the portal interface so that they can be presented in the language you have added. You can provide support for translating default text strings and other UI elements that display to end-users by configuring the tables that hold translatable text for the portal.

Configure the following tables with information on the language you added:

- NLSTRANSLATE51
- NLSTRANSLATE52
- NLSTRANSLATE53
- NLSTRANSLATE54

This section first describes the items in the database tables listed above, and then provides the procedure for configuring them.

The description and the instructions for configuring the NLSTRANS�ATE51 table is applicable to the other tables, since they contain similar items and layout. For information about each table, see the section on "Modifying the iViewServer Database Tables" on page 6.

The following illustration is an example of the NLSTranslate table:

Entryindex	Langindex	Value
2070	9	Portal
2070	16	Portale
2070	12	Portail
3094	9	You must specify a query string.
3094	16	Inserire una stringa query.
3094	12	Indiquez une cha4ne de requ4te.

When you open the NLSTRANS�ATE51 table, it displays the following details of the supported languages:

- ENTRYINDEX

The *entryindex* is a unique number that references a specific string of text.

The same entryindex number is used for a specific string of text in various languages. For example, the text "Portal", "Portale", "Portail", all have the same entryindex value 2070.

- LANGINDEX

*Langindex* is a unique number identifying a supported language.

In the example from the table above, the langindex 9 is used for all text items in the English language, and the langindex 16 is used for all the text in the Italian language.

- VALUE

*Value* is the actual string of text in a specific language. This is the text itself that is rendered in the portal and displayed to the end user.

#### To manually translate text into a language:

1. From the database server, open a table that holds text strings for the portal; for instance, NLSTranslate51, NLSTranslate52, NLSTranslate53, or NLSTranslate54.
2. Insert a new row, and do the following:
  - Translate the existing phrase or sentence under the column heading Value into the language to which you are extending support.
  - Provide the entryindex and the langindex values for each such translated text.

## 4 Extending Language Support to the Java Infrastructure of the Portal

In the Java environment, SAP Enterprise Portal employs resource bundles. A resource bundle consists of related classes that inherit from the class `ResourceBundle`. Each file in a resource bundle has a language code, country code, and locale variant.

When the portal determines that a request contains a language code, it accesses a locale-specific file from the bundle that contains information of the requested language. From this file, it provides the necessary text resources in the requested language for rendering. This approach enables developers to separate the implementation code from locale-specific information contained in bundles.

More information on how the portal determines user locale settings is available at:

**help.sap.com/ep** > EP 5.0 SP5, *SAP Enterprise Portal Documentation* > *Administration Guide* > *Translating Portal Content Metadata*

For additional information on internationalization in Java programs using resource bundles, refer to the section on “Internationalization” in the documentation set for the Portal Development Kit for Java on **www.iviewstudio.com**

### 4.1 Workflow: Localizing the Java Infrastructure of the Portal

- Make sure that you provide text resources in the language you want to support. Text resources are files in a resource bundle. These files contain text entries in the supported languages for portal content such as roles, worksets, pages, and external services in the Portal Content Directory (PCD). The PCD is in the local file system, where the portal is deployed.
- Objects in the PCD that implement portal content are placed in separate folders. For each portal object, there are two text resource files: an original language file *obj.properties*, and the English language file *obj\_en.properties*.

The original language file determines the language resources for maintaining and rendering text in the portal user interface. The language from the original language file can also be used as the neutral language for maintaining and displaying text in the portal.

For more information on language resources for maintaining and displaying text elements in the portal, go to: **help.sap.com/ep**  
*SAP Enterprise Portal Documentation* > *Administration Guide* > *Translating Portal Content Metadata*.

The English language file determines the language resources for displaying text in the English language. In an out-of-the-box portal, both the original language and the English language files contain exactly the same data in the same language.

The following is the workflow to enable localization in Java for portal objects:

1. In the portal, create a role, page, workset, or external service, in English. Text resources in the English language are automatically created in the PCD for the role, page, workset, or external service.

Note that the portal automatically creates text resources only in the English language. You must manually create text resources in the language to which you want to extend support.

The original language and the English language files are the text resources that are created and placed in a folder designated for text resources. For more information, see the section on “Text Resources for Portal Content Directory (PCD) Objects” on page 10.

2. Go to the folder designated for each portal object and create a duplicate of the original language file with a different name. For more information, see the section on “Text Resources for Portal Content Directory (PCD) Objects” on page 10.
3. Give the duplicate file a name that conforms to the file naming conventions for supported locales. For more information on file naming conventions, see the section on “File Naming Conventions.”  
For example; duplicate the file *obj\_en.properties* and give it a new name, such as *obj\_de.properties*, if you intend to support the German language.
4. Open the newly created language file in a text editor, such as *Notepad*.
5. Translate each text entry into the language you intend to support in the portal.  
As the language file is stored using the Windows standard encoding (ISO-8859-1), add Unicode escape sequences for characters that cannot be represented in this encoding. For example, the Unicode escape sequence for the German letter Ü is added as Unicode escape sequence `\u00DC`.
6. Include the additional text resource files in the portal archive PAR file for the portal objects.
7. In the Java class for the portal components of each object such as role, page, workset, define a parameter called “**ResourceBundleName**.”  
The value of this parameter is the resource bundle name. For more information, see “Adding Resource Bundles to Your Java Code” on page 12.
8. Restart the SAP J2EE Engine service.
9. Launch the portal to verify the changes you made. For more information, see “Verifying Support For the Added Language” on page 13.

## 4.2 Text Resources for Portal Content Directory (PCD) Objects

Properties for the portal content in the PCD are defined in the file *pcd.properties*. This file is stored in the folder:

```
.../<irj>/web-inf/plugins/portal/system/properties/pcd.properties (pcd.home)
```

Portal objects for content such as roles, pages, worksets and external services are separated from the text resources for the content. There are two subfolders under the folder: *.../Enterprise Portal/DataFile/Data*.

These are:

- PCD  
The PCD folder contains the objects in the PCD.
- *TR* (TextResources)  
The TR folder contains the translatable texts for the objects.

### 4.2.1 Translating Texts for PCD Objects

The *TR* folder contains subfolders for storing texts for the following objects:

- Roles:  
`.../TR/roles/<role_technical_name>/`

For example, `.../TR/roles/Documentation/`, where *Documentation* is the technical name of the role that was created in the portal.

- Worksets:  
.../TR/global/folders/<workset\_technical\_name>/  
For example, .../TR/global/folders/**Support**/, where Support is the technical name of the workset that was created in the portal.
- Pages:  
.../TR/global/pages/<page\_technical\_name>/  
For example, .../TR/global/pages/**Project Assistants**/, where Project Assistants is the technical name of the page that was created in the portal.
- External Services:  
.../TR/global/services/service-iviews/<external-service\_technical\_name>/  
For example, .../TR/global/services/service-iviews/**Tracking Support**/, where Tracking Support is the technical name of the external service that was created in the portal.

Each <object\_technical\_name> folder contains a metafile called *obj.info*, which consists of general information about the stored texts to be translated.

In addition, the folder contains the two text resource files, *obj\_en.properties* and *obj.properties*. The former file supports English language texts, and the latter file is the original language file.

Duplicate the file *obj\_en.properties* for translation of text into the language you intend to support, and give it a name that conforms to the naming conventions in the following section.

In each file, there are unique text keys. Each key is the same for each text string in its original language and in the various languages to which it was translated.

1. Translate each existing phrase or sentence into the language for which you are extending support.
2. For such translated text, use the same text key.

For instance, *MySentence1* is the text key in the example below:

File Name	Translated Text
<i>ResourceTexts.properties</i>	MySentence1= Sentence in English
<i>ResourceTexts_en.properties</i>	MySentence1= Sentence in English
<i>ResourceTexts_de.properties</i>	MySentence1= Satz in deutscher Sprache

The text resources in the original language file is used as a fallback if the requested language is not available, or if no language-specific text is maintained for the object. For information on the original language file, see "Workflow: Localizing the Java Infrastructure of the Portal".

#### 4.2.2 File Naming Conventions

Texts to be translated are stored in properties files.

For text resource files, follow the procedure in this section for property file naming conventions:

- Each filename must conform to the following format:  
*obj\_<2\_letter\_language\_code>.properties*  
For example, the filename for roles in English is *roles\_en.properties*.

The portal obtains language resources of an object based on the filename of the resource bundle for that object as follows:

Part of the filename	Description
<i>obj</i>	This is the object name such as, <i>roles</i> . This part of the filename enables the portal to get the specific object's resource bundle.
<2_letter_language_code>	This part of the filename indicates the language code such as <i>en</i> , <i>fr</i> , <i>de</i> , for which language resources are available in the bundle.
<i>properties</i>	The extension to the filename informs the portal of the class to search for.

### 4.2.3 Adding Text Resources to a PAR File

After creating the text resource files, add them to the PAR file for the portal objects.

For instance, the file *MyTranslatableJavaView.par* contains *.../private/lib/myClasses.jar* and *.../private/profiles/default.properties* files.

1. Add the following text resource files to *myClasses.jar*:
  - *.../private/classes/ResourceTexts.properties* (this is the original language file)
  - *.../private/classes/ResourceTexts\_en.properties* (this is the file with English text)
  - *.../private/classes/ResourceTexts\_de.properties* (this is the file with German text)
2. Edit the Java code (class file) for the portal component of each object by adding a parameter that assigns a name to the resource bundle in the PAR file.

By so doing, you enable the portal runtime to reference the text resources it needs for displaying user interface elements.

### 4.3 Adding Resource Bundles to Your Java Code

The portal must know the name of the resource bundles that have been assigned to the profile in each portal component.

You provide the resource bundle name by adding a parameter in the profile called "ResourceBundleName." Use the segment of the resource bundle name that is common to the name in the set of text resources and the original language files of each portal object.

The following is an example of a profile in the file; */private/profiles/default.properties*:

- *ClassName=com.myCompany.myDefaultClass*
- *AuthRequirement=admin*
- *ResourceBundleName=ResourceTexts*

In the source code, call your portal component and request for the resource bundle of your profile. It returns the string from the requested language resource bundle.

For a sample code for this implementation, see "Appendix: Sample Java Code" on page 14.

## 5 Verifying Support For the Added Language

After extending language support to both the .NET and the Java parts of the portal infrastructure, you are ready to check if portal user interface elements display in the supported language.

**To verify support for the new language, do the following:**

1. Launch the portal client.

Information displayed by your portal client is appropriate to the language defined by the client locale. For example, if the client locale indicates English, the portal is displayed in English. If the client locale indicates German, the portal is displayed in German.

**Important:**

If the portal is running while you configure the language settings, then you have to **RESTART** the portal Web server (IIS).

2. Change the language in which the portal displays the user interface. Language settings for the portal can be changed by selecting: *Personalize* → *Portal* → *Language setting*

Expected Results:

- The portal client should display the user interface in the language of choice. When no translation is available, the portal applies its neutral language settings.

## 6 Appendix: Sample Java Code

The following is a sample of the Java code for referencing text resource bundle files:

```
package com.myCompany

import com.sapportals.portal.prt.component.*;
import com.sapportals.portal.prt.resource.*;
import com.sapportals.portal.prt.event.*;
import com.sapportals.portal.prt.runtime.*;
import com.sapportals.portal.prt.util.StringUtils;

import java.util.ResourceBundle;

public class LicenseComponent extends AbstractPortalComponent
{
    /*****
     * This method is used to encode Unicode characters from resource bundle strings
     * to the corresponding HTML representation for correct rendering of e.g. german
     * umlauts or characters like ">"
     *
     * @param unescapedText    The text as it comes from the resource bundle
     * @return                 The
     *****/
    private String htmlEncode(String unescapedText)
    {
        return com.sapportals.portal.prt.util.StringUtils.escapeToHTML(unescapedText);
    }

    /*****
     * This method handles the portal component request.
     *
     * @param request          The portal component's request
     * @param response         The portal component's response
     *****/
    protected void doContent(IPortalComponentRequest request,
                             IPortalComponentResponse response)
    {
        ResourceBundle myBundle = request.getResourceBundle();

        String sentenceInRequestsLanguage = myBundle.getString("MySentence1");

        String encodedSentence = htmlEncode( sentenceInRequestsLanguage );

        response.write( encodedSentence );
    }
}
```

©Copyright 2003

■No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP AG. The information contained herein may be changed without prior notice.

■Some software products marketed by SAP AG and its distributors contain proprietary software components of other software vendors.

■Microsoft®, WINDOWS®, NT®, EXCEL®, Word®, PowerPoint® and SQL Server® are registered trademarks of Microsoft Corporation.

■IBM®, DB2®, DB2 Universal Database, OS/2®, Parallel Sysplex®, MVS/ESA, AIX®, S/390®, AS/400®, OS/390®, OS/400®, iSeries, pSeries, xSeries, zSeries, z/OS, AFP, Intelligent Miner, WebSphere®, Netfinity®, Tivoli®, Informix and Informix® Dynamic Server™ are trademarks of IBM Corporation in USA and/or other countries.

■ORACLE® is a registered trademark of ORACLE Corporation.

■UNIX®, X/Open®, OSF/1®, and Motif® are registered trademarks of the Open Group.

■Citrix®, the Citrix logo, ICA®, Program Neighborhood®, MetaFrame®, WinFrame®, VideoFrame®, MultiWin® and other Citrix product names referenced herein are trademarks of Citrix Systems, Inc.

■HTML, DHTML, XML, XHTML are trademarks or registered trademarks of W3C®, World Wide Web Consortium, Massachusetts Institute of Technology.

■JAVA® is a registered trademark of Sun Microsystems, Inc.

■JAVASCRIPT® is a registered trademark of Sun Microsystems, Inc., used under license for technology invented and implemented by Netscape.

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

■SAP, SAP Logo, R/2, R/3, mySAP, mySAP.com and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP AG in Germany and in several other countries all over the world. All other product and service names mentioned are trademarks of their respective companies.