How To Apply a Custom Error Page for the "Bad User Agent" Error Message

Applicable Releases:

SAP Netweaver 7.0

SAP Netweaver 2004

IT Practice:

User Productivity Enablement

IT Scenario:

Running an Enterprise Portal

Version 1.0

March 2009

THE BEST-RUN BUSINESSES RUN SAP™

Document History

Document Version	Description
1.00	First official release of this guide

Typographic Conventions

| Type Style	Description
Example Text	Words or characters quoted from the screen. These include field names, screen titles, pushbuttons labels, menu names, menu paths, and menu options.
	Cross-references to other documentation
Example text	Emphasized words or phrases in body text, graphic titles, and table titles
`Example text`	File and directory names and their paths, messages, names of variables and parameters, source text, and names of installation, upgrade and database tools.
`Example text`	User entry texts. These are words or characters that you enter in the system exactly as they appear in the documentation.
`<Example text>`	Variable user entry. Angle brackets indicate that you replace these words and characters with appropriate entries to make entries in the system.
`EXAMPLE TEXT`	Keys on the keyboard, for example, `F2` or `ENTER`.

Icons

Icon	Description
	Caution
	Note or Important
	Example
	Recommendation or Tip

Table of Contents

1. Scenario

The SAP Enterprise Portal has the possibility to restrict access to content based on the user agent type. In the properties of an iView or page you can define the supported or unsupported user agents.
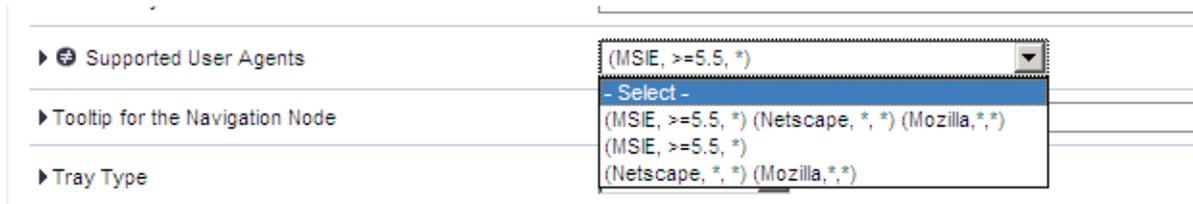
The selectable options have the following syntax:

(<user agent type>, <user agent version>, <user agent platform>)

For example the option (MSIE, >=5.5, *) means, that the page / iView can only be accessed by Internet Explorer of version 5.5 and above on all operating system platforms.

When you access an iView or page with an unsupported user agent, you will get a standard error message from the portal runtime. This "How To Guide" explains what you have to do to display your own custom error message.

In case you want to restrict access to the whole portal, you must set the "Supported User Agents" property on the frameworkpage. Since the frameworkpage is the root of all pages and iViews, the user will not be able to access and see any portal content at all.

2. Background Information

Every portal component can be rendered in different modes. The portal runtime (PRT) offers a set of predefined modes like `help`, `about`, `edit`, `error`, `badUserAgent` or `content`. If a portal component gets called by the PRT, the default mode is `content`. Therefore, to render the default UI of your portal component you must override the method `doContent`(…) in your AbstractPortalComponent class.

It is also possible to override other methods inside your portal component to react to other modes than the `content` mode. As an example let's look at the following scenario:

If the user clicks the "Help" link in the iView tray, the PRT switches the mode from `content` to `help`. In order to display a help page for the iView, you must override the `doHelp(…)` method in your portal component.

If the user accesses a portal page with a not supported user agent, the PRT makes a mode switch to the mode `badUserAgent`. To react on this, your portal component must override the `doBadUserAgent(…)` method. Inside this method you can now build your own html error message.

This approach shown so far has one limitation. You can only display a custom error message for a bad user agent, if your own components (iViews/pages) are requested. If you want to show a general error message for all pages which are accessed with a bad user agent, you must use a mechanism called **"mode delegation"**.

The PRT itself uses mode delegation and delegates the handling of the `badUserAgent` mode by default to a standard portal component called `com.sap.portal.runtime.system.hooks.SystemModes`. However this delegation assignment is not fixed, it's configurable. The PRT makes a lookup in the Portal Registry to find the portal

component which should be called for the badUserAgent error message. To be able to show your own error message, you just override the standard portal component (`com.sap.portal.runtime.system.hooks.SystemModes`) which is assigned to the `badUserAgent` mode with our own portal component. This can be done inside the portal specific deployment descriptor (portalapp.xml) using the **<registry>** element.

3. Prerequisites

- You need at least SAP Netweaver Enterprise Portal 2004
- You should be familiar with the development of Portal Components
- You should be familiar with the SAP Netweaver Developer Studio

Related documentation

Source	URL
Help Portal	http://help.sap.com/saphelp_nw70ehp1/helpdata/en/53/d8b842b714b211e10000000a155106/content.htm
Help Portal	http://help.sap.com/saphelp_nw70ehp1/helpdata/en/a0/44b742cafec96ae10000000a155106/content.htm

4. Step-by-Step Procedure

You will create a portal application which includes one portal component called "BadUserAgent", which serves as the delegate. The portal component must override the `doBadUserAgent(…)` method.

In the second step, you must add the registry entry in the deployment descriptor (portalapp.xml) to define your component as the default delegate for the badUserAgent mode.

In the last step, you deploy and test your portal application.

4.1 Create a Portal Application and a Portal Component

1. Create a Portal Application called `com.demo.error`.

2. Create a new Portal Component of type `AbstractPortalComponent` called `BadUserAgent.`

3. Override the method `doBadUserAgent(…)` and return the HTML error message you want to display in the browser.

4. As a result your `BadUserAgent` component should look like this:
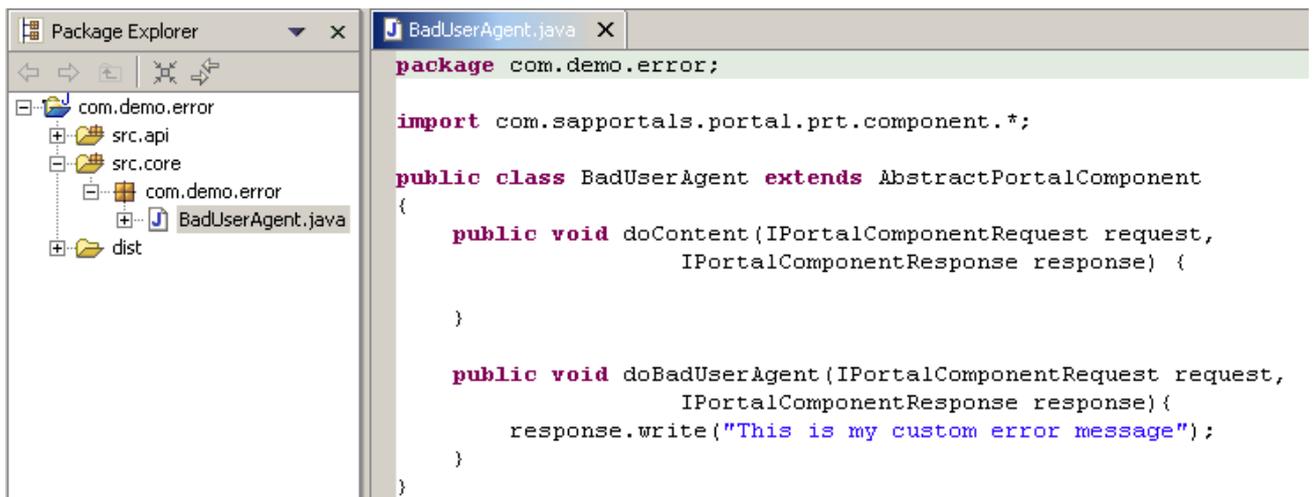
5. Instead of directly writing HTML in your `doBadUserAgent(…)` method, you could also forward the processing to a JSP page, which then renders your own error message. An example can be found in the appendix.

4.2 Register your component as a delegate

1. Open the portalapp.xml file. Here you define, that your portal component should be called by the PRT, if the mode is switched to `badUserAgent`. This is done by an registry entry using the tag <registry>. The following screenshot shows how the portalapp.xml file looks like.

```
J BadUserAgent.java    portalapp.xml  X
 1 <?xml version="1.0" encoding="UTF-8"?>
 2 <application>
 3   <registry>
 4     <entry path="/runtime/prt.modes/badUserAgent"
 5            name="BadUserAgent"
 6            type="component" rebind="true"/>
 7   </registry>
 8   <application-config>
 9     <property name="startup" value="true"/>
10   </application-config>
11   <components>
12     <component name="BadUserAgent">
13       <component-config>
14         <property name="ClassName" value="com.demo.error.BadUserAgent"/>
15       </component-config>
16       <component-profile/>
17     </component>
18   </components>
19   <services/>
20 </application>
21
```

Explanation of the portalapp.xml file:

Lines 3-7: This defines the portal component `BadUserAgent` as the delegate for the mode
   `badUserAgent`. Technically you create a new binding in the Portal Registry for the path
   "`/runtime/prt.modes/badUserAgent`". Setting the rebind property to true means, that if
   the component is reloaded, that the binding is done again.

Lines 8-10: You must also set the property `startup` to `true`. In the case of a server restart, the
   portal component is loaded automatically and the binding is done. If you omit this parameter,
   your binding would be lost after a server restart.

4.3 Deploy your portal component to the server and test it.

After you have deployed your portal application to the server, you can check if the binding is correct
using the Portal Registry Browser. You must be logged in as an administrator user with superadmin
permissions. Navigate to

`System Administration` `Support` `Portal Runtime` `Portal Registry`
`Browser` `runtime` `prt.modes`

If you look at the binding for "`badUserAgent`", you should see your own portal component class (see
following picture).

You are here: ROOT/runtime/prt.modes

Child list of the current context:

Name	Class name
config	java.lang.String - com.sap.portal.runtime.system.hooks.ConfigComponent
help	java.lang.String - com.sap.portal.runtime.system.hooks.SystemModes
iPartSettings	java.lang.String - com.sap.portal.coll.ipartpropertyeditor.IPartPropertyEditorComponent
_reinitcache	java.lang.String - com.sap.portal.runtime.system.hooks.ReleaseComponent
badUserAgent	java.lang.String - com.demo.error.BadUserAgent
_releasecache	java.lang.String - com.sap.portal.runtime.system.hooks.ReleaseComponent
_release	java.lang.String - com.sap.portal.runtime.system.hooks.ReleaseComponent
about	java.lang.String - com.sap.portal.pagebuilder.DefaultAboutModeDelegation
edit	java.lang.String - com.sap.portal.appdesigner.framework.profile
error	java.lang.String - com.sap.portal.runtime.system.hooks.ErrorComponent

To really test your custom error message, you should create a page with an iView on it. Set the property "Supported User Agents" of your page to (MSIE, >=5.5, *).

This means, that your page can only be accessed by Internet Explorer version 5.5 or higher. If you now try to access your page with Firefox, you should see your custom error message instead of the standard error message.

5. Appendix

Implementation of method `doBadUserAgent(…)` and including the JSP `badUserAgentErrorPage.jsp` into the response.

```
public void doBadUserAgent(IPortalComponentRequest request,
                IPortalComponentResponse response){
    IResource res = request.getResource(IResource.JSP, "jsp/badUserAgentErrorPage.jsp");
    response.include(request,res);

}
```

The JSP must be put into folder */dist/PORTAL-INF/jsp*.
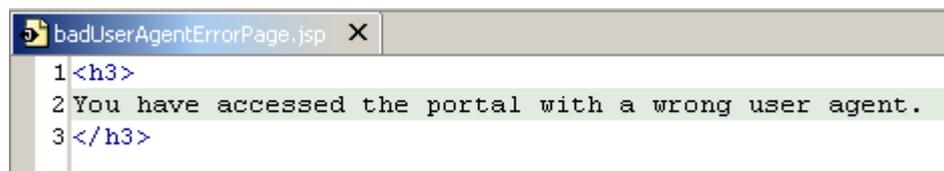
```
badUserAgentErrorPage.jsp  X
1 <h3>
2 You have accessed the portal with a wrong user agent.
3 </h3>
```

www.sdn.sap.com/irj/sdn/howtoguides