How to…

Customize the Logon User Interface of SAP Enterprise Portal

ENTERPRISE PORTAL 6.0

CONFIDENTIAL

ASAP "How to…" Paper

Applicable Releases: EP 6.0

August 2003

Disclaimer

This document discusses sample coding, which does not include official interfaces and therefore is not supported. Changes made based on this information are not supported and can be overwritten during an upgrade.

1 Overall Workflow

The logon user interface (UI) of SAP Enterprise Portal is delivered in a portal archive (PAR file) named com.sap.portal.runtime.logon.par which contains all the logon UI's code and resources.

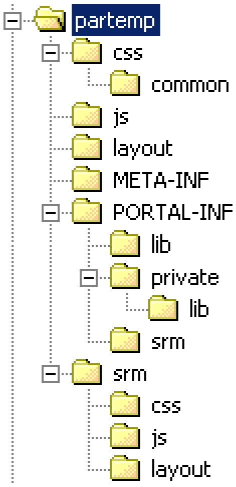
In a deployed portal, the PAR file is located at <SAPJ2EEngine-deployment-dir> \cluster\server\services\servlet_jsp\work\jspTemp\irj\root\WEB-INF\deployment\pcd. During initial portal deployment, it is renamed to com.sap.portal.runtime.logon.par.bak.

To customize the logon UI, we recommend the following procedure:

1. Make a copy of com.sap.portal.runtime.logon.par.bak and rename it. In this example, it is renamed to my.new.logon.par.
2. Move my.new.logon.par to a location outside of the <SAPJ2EEngine-deployment-dir>.
3. Extract the files from my.new.logon.par preserving the directory structure.
4. Modify files in the extracted PAR file.
5. Put the modified files back into my.new.logon.par.
6. Copy my.new.logon.par back to <SAPJ2Eengine-deployment-dir>\cluster\server\services\servlet_jsp\work\jspTemp\irj\root\WEB-INF\deployment\pcd.
7. Rename the file <SAPJ2Eengine-deployment-dir>\cluster\server\ume\authschemes.xml.bak to authschemes.xml.
8. In authschemes.xml, replace all occurrences of the string “com.sap.portal.runtime.logon” with “my.new.logon” in the tags <frontendtarget> and save the file.
9. Restart the portal.

Now, the modified logon UI is displayed at logon.

The following diagram illustrates the folder structure of the logon archive. In this example, the PAR file is extracted to a folder called *partemp*.



- partemp
  - css
    - common
  - js
  - layout
  - META-INF
  - PORTAL-INF
    - lib
    - private
      - lib
    - srm
  - srm
    - css
    - js
    - layout

.

- The PORTAL-INF folder contains the PAR file's private resources, such as jsp pages, their includes (see below) and java code
- The folders next to PORTAL-INF such as css, js, and layout contain the public resources

The resources in the folders srm and and PORTAL-INF\srm are NOT used.

2 Structure of the Logon Screens

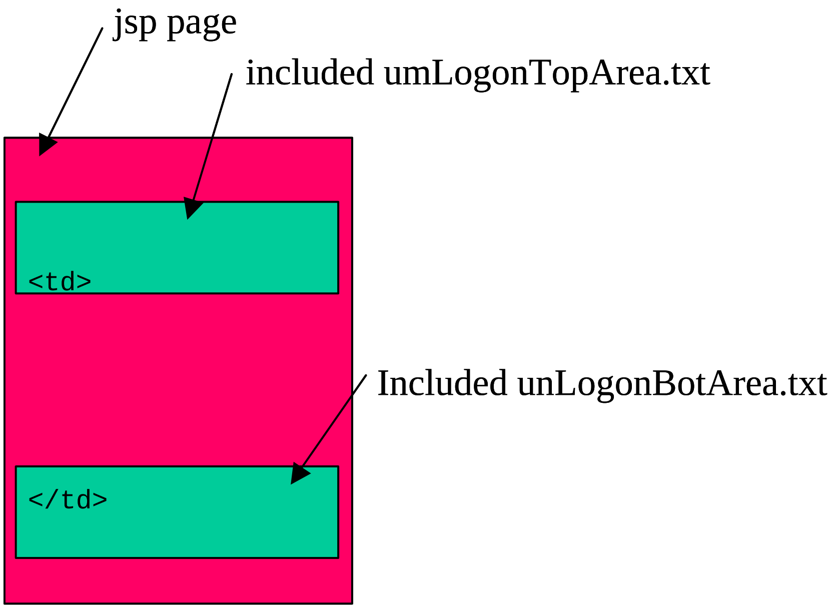
The logon UI contains the following screens:

- umLogonPage.jsp (user ID and password logon)
- umLogonCertPage.jsp (initial storage of client certificate)
- umResetPasswordPage.jsp (requesting a password reset after user forgot password)
- changePasswordPage.jsp (changing password after the old one has expired)
- umHelpPage.jsp (main logon help page where users can choose what type of help they require)
- umLogonProblemPage.jsp (sending an email to the administrator)
- umLogoffPage.jsp (displays logoff confirmation)

All the screens have the same structure:

- The jsp page contains the "functional part" of the page, for example, the HTML form and its submit and cancel button). It is basically a HTML table, of which the elements are mostly formatted using CSS and some extra style attributes. This is the part that is different for all the jsps listed above. In a rendered, non-customized page, this is the part to the left of the branding image.
- Just before the functional part, there is the directive <%@ include file="/umLogonTopArea.txt"%> and just after it the is the directive <%@ include file="/umLogonBotArea.txt"%>. These two text files contain the beginning and the end of the HTML container, in which the functional part resides. They are common to all the jsps listed above.

umLogonTopArea.txt ends with an opening <td>-tag and umLogonBotArea.txt starts with a closing </td>-tag. The part of the jsps in between the two include directives is a complete HTML element, in which all opening tags are closed. Each jsp, umLogonTopArea.txt, and umLogonBotArea.txt together form a syntactically correct document. Some of the opening tags in umLogonTopArea.txt are closed in umLogonBotArea.txt.



Note: Modifications in umLogonTopArea.txt and umLogonBotArea.txt affect all logon screens. This is useful if, for example, you want a text to be displayed on all screens.

3 Making New Resources Available

Create a new folder called myres under partemp. Put new images, style sheets, and javascript files into myres.

The availability of the new resources depends on where you include them:

- If you include the new resources in UmLogonTopArea.txt or umLogonTopArea.txt, because these files are included on all jsp pages, the resources are available on all of them
- If you include the new resources in one specific jsp page, the resources are available on this page only

3.1 Including a New Style Sheet (CSS)

For example, you can provide a style sheet called test.css in the myres folder. Use the following code to make it accessible:

```
<%
   IResource newstyle = componentRequest
            .getResource(IResource.CSS, "myres/test.css");
   aResponse.include(componentRequest, newstyle);
%>
```

Once the CSS is accessible, the jsp can reference classes from test.css.

If you want the style sheet to be available on all logon screens, you must insert the above code at the beginning of umLogonTopArea.txt.

If you want the style sheet to be available on one specific screen only, you must insert the above code in the desired jsp page directly before the include directive for umLogonTopArea.txt.

3.2 Including a New Image

For example, you can provide an image called testimg.gif in the myres folder. Then you can use this image as follows:

```
<img src="<%=webpath%>myres/testimg.gif">
```

3.3 Including a New Javascript File

For example, you can provide a script called test.js in the myres folder. Then you can use this script as follows:

```
<script language="JavaScript" src="<%=webpath%>myres/test.js">
</script>
```

4 Special Parts of the Logon UI

4.1 Branding Image

The branding image branding-image.jpg is referenced in umLogonBotArea.txt.

Note: As of EP6 SP2, it is possible to change the branding image without having to modify code. Instead, you can set the user management property `ume.logon.branding_image` to a valid URL. For example:

```
ume.logon.branding_image=/irj/newImage.gif.
ume.logon.branding_image=http://www.company.com/Image.gif
```

4.2 SAP Logo

The SAP logo is referenced in umLogonBotArea.txt as sapLogo.gif.

5 Adding Language Dependent Texts

Text resources for language-dependent texts are contained in the JAR file PORTAL-INF/lib/umlogonbase.jar. The language specific files are called logonLabels_xx.properties, where xx is replaced by the language code. For example logonLabels_en.properties is standard English. The logonLabels.properties file contains the default language that is used if the current browser language is not found among the logonLabels_xx.properties files.

1. Extract the file logonLabels.properties and the files logonLabels_xx.properties for the required languages from the JAR file.
2. Each text has a key, which is the same in all logonLabels_xx.properties files. Find a new key, that is not yet contained in logonLabels. In this example, we use the key NEW_KEY.
3. Add a new line for each new key that is to be added in each of the extracted files as a key=value pair. For example:
   NEW_KEY=my new text
   logonLabels_en.properties should contain the English text, logonLabels_de.properties should contain the German text, and so on.
4. Replace the old files in the jar with the updated ones.

6 Using Language Dependent Texts

In the jsps, umLogonTopArea.txt, and umLogonTopArea.txt, you can reference the texts from the language files as follows:

```
<%=logonLocale.get("NEW_KEY")%>
```

In the HTML code that is generated from the jsps, the above line is replaced by the value of the key NEW_KEY from the logonLabels_xx.properties, where xx is the current browser language.

If the text is in a table-data-tag, use the following code:

```
<td><%=logonLocale.get("NEW_KEY")%></td>
```

7 Changing the General Layout

You can change the general layout as long as you respect the restrictions that are imposed by the structure of the txt files and the jsps (opening/closing tags).