BEA WebLogic Portal

Simplifying Portal URLs

Version 8.1 with Service Pack 5
Document Revised: October 2005

Copyright

Copyright © 2005 BEA Systems, Inc. All Rights Reserved.

Restricted Rights Legend

This software and documentation is subject to and made available only pursuant to the terms of the BEA Systems License Agreement and may be used or copied only in accordance with the terms of that agreement. It is against the law to copy the software except as specifically allowed in the agreement. This document may not, in whole or in part, be copied, photocopied, reproduced, translated, or reduced to any electronic medium or machine readable form without prior consent, in writing, from BEA Systems, Inc.

Use, duplication or disclosure by the U.S. Government is subject to restrictions set forth in the BEA Systems License Agreement and in subparagraph (c)(1) of the Commercial Computer Software-Restricted Rights Clause at FAR 52.227-19; subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013, subparagraph (d) of the Commercial Computer Software--Licensing clause at NASA FAR supplement 16-52.227-86; or their equivalent.

Information in this document is subject to change without notice and does not represent a commitment on the part of BEA Systems. THE SOFTWARE AND DOCUMENTATION ARE PROVIDED “AS IS” WITHOUT WARRANTY OF ANY KIND INCLUDING WITHOUT LIMITATION, ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. FURTHER, BEA Systems DOES NOT WARRANT, GUARANTEE, OR MAKE ANY REPRESENTATIONS REGARDING THE USE, OR THE RESULTS OF THE USE, OF THE SOFTWARE OR WRITTEN MATERIAL IN TERMS OF CORRECTNESS, ACCURACY, RELIABILITY, OR OTHERWISE.

Trademarks or Service Marks

BEA, BEA JRockit, BEA Liquid Data for WebLogic, BEA WebLogic Server, Built on BEA, Jolt, JoltBeans, SteelThread, Top End, Tuxedo, and WebLogic are registered trademarks of BEA Systems, Inc. BEA AquaLogic, BEA AquaLogic Data Services Platform, BEA AquaLogic Enterprise Security, BEA AquaLogic Service Bus, BEA AquaLogic Service Registry, BEA Builder, BEA Campaign Manager for WebLogic, BEA eLink, BEA Manager, BEA MessageQ, BEA WebLogic Commerce Server, BEA WebLogic Enterprise, BEA WebLogic Enterprise Platform, BEA WebLogic Enterprise Security, BEA WebLogic Express, BEA WebLogic Integration, BEA WebLogic Java Adapter for Mainframe, BEA WebLogic JDriver, BEA WebLogic JRockit, BEA WebLogic Log Central, BEA WebLogic Personalization Server, BEA WebLogic Platform, BEA WebLogic Portal, BEA WebLogic Server Process Edition, BEA WebLogic WorkGroup Edition, BEA WebLogic Workshop, and Liquid Computing are trademarks of BEA Systems, Inc. BEA Mission Critical Support is a service mark of BEA Systems, Inc. All other company and product names may be the subject of intellectual property rights reserved by third parties.

All other trademarks are the property of their respective companies.

Contents

URL Formats 2
Simplifying the URLs 2
Step 1. Create a Servlet to Perform the URL Mapping 2
Step 2. Register and Map the Servlet 4
Step 3. Modify Book and Page Links in the Portal Framework 5
Modify the singlelevelmenu.jsp Skeleton File 5
Modify the submenu.jsp Skeleton File 7
Modify Book and Page Definition Labels 9
Step 4. Test the URLs 9

Simplifying Portal URLs

When you create portals and portal desktops, the default URLs to the portal's books and pages contain parameters and elements you might want to eliminate to make shorter, simpler URLs. This topic shows you how to simplify your portal's URLs.

Figure 1 shows a sample default URL, and Figure 2 shows the same URL simplified.

Figure 1 Default URL: http://intranet/sampleportal/appmanager/myportal/mydesktop?_nfpb=true&_pageLabel=MyPage

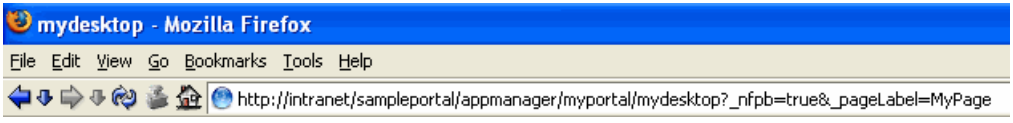
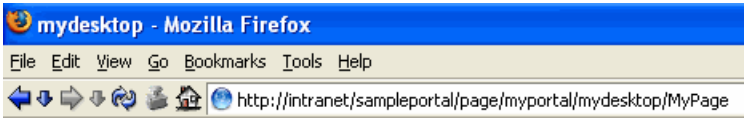


Figure 2 Simplified URL: http://intranet/sampleportal/page/myportal/mydesktop/MyPage



This topic includes the following sections:

- URL Formats
- Simplifying the URLs

Note: This topic provides instructions for simplifying URLs generated on book and page clicks. It does not provide instructions on creating simplified URLs for links within your presentation JSPs.

URL Formats

The default URLs for books and pages are different for single portal files and streaming portal desktops (see Single File vs. Streamed Rendering in the *Portal User Interface Framework Guide* Overview).

Table 1 shows examples of default URLs for single portal files and streaming portal desktops, along with examples of simplified URLs.

Table 1 Examples of default and simplified URLs

	Default URL	Simplified URL
Single Portal Files	http://intranet/ourportal/everyone.portal?_nfpb=true&_pageLabel=HomePage	http://intranet/ourportal/page/portal/everyone/HomePage
Streaming Portal Desktop	http://intranet/ourportal/appmanager/field/everyone?_nfpb=true&_pageLabel=HomePage	http://intranet/ourportal/page/field/everyone/HomePage

Simplifying the URLs

The following steps show you how to create simpler URLs:

- Step 1. Create a Servlet to Perform the URL Mapping
- Step 2. Register and Map the Servlet
- Step 3. Modify Book and Page Links in the Portal Framework
- Step 4. Test the URLs

Step 1. Create a Servlet to Perform the URL Mapping

The portal framework ultimately needs the long URLs it generates. When you set up your URL simplifying mechanism, you need to tell the framework how to map your short URL to the long URL it needs. You set up this mapping in a servlet.

In this step you create a servlet called `page` (`page.jsp`) in the web application root. This `page.jsp` handles the mapping between a simplified URL and a longer framework URL. You can name your servlet/JSP whatever you like.

1. Open your portal application in WebLogic Workshop.
2. Create `page.jsp` in the portal Web project in which you want the simplified URLs.
3. In the new JSP file that appears, switch to Source View and delete all the default code and content. Insert the code shown in Listing 1.
4. Save the file.

Listing 1 New code for page.jsp

```
<%
String [] thePath= request.getPathInfo().split("/");
   if (thePath[1].equals("portal"))
   {
      request.getRequestDispatcher("/" + thePath[2]
      + ".portal?_nfpb=true&_pageLabel=" +
      thePath[3]).forward(request, response);
   }
   else
   {
      request.getRequestDispatcher("/appmanager/"
      + thePath[1] + "/" + thePath[2] +
      "?_nfpb=true&_pageLabel=" +
      thePath[3]).forward(request, response);
   }
%>
```

The code splits the URL the servlet receives into pieces at each forward slash ("/"), detects whether the URL is for a single `.portal` file or a streaming portal, and rearranges the simplified URL to become the longer URL the framework needs.

Step 2. Register and Map the Servlet

Now that you have created the servlet, you must tell the server about it with the following steps:

1. In WebLogic Workshop, open your portal Web project's `WEB-INF/web.xml` file.
2. In the servlets section of the file, add the entry shown in Listing 2. If you used a different servlet name than page, enter that instead.

Listing 2 Servlet declaration entry

```
<servlet>
   <servlet-name>page</servlet-name>
   <jsp-file>page.jsp</jsp-file>
</servlet>
```

This entry tells the server that a servlet called `page` exists, which is the `page.jsp` file in the portal Web project root.

3. In the servlet-mapping section of `web.xml`, add the entry shown in Listing 3.

Listing 3 Servlet-mapping entry

```
<servlet-mapping>
   <servlet-name>page</servlet-name>
   <url-pattern>/page/*</url-pattern>
</servlet-mapping>
```

This entry tells the server to use the `page` servlet whenever a user clicks a URL containing `/page/*`. You create URLs with this pattern in the following steps.

4. Save the file.

Step 3. Modify Book and Page Links in the Portal Framework

Users navigate through your portals with book and page links in a menu. The portal framework dynamically builds book and page links with menu skeleton files. In this step, you modify the menu skeletons to generate your new shorter URLs. These new URLs appear in the browser address field when users click the book and page links.

This step provides support for simplified URLs in both single-level and multi-level menus in the default skeleton.

Note: Any Look & Feel that uses the default skeleton—or at least the default `submenu.jsp` and `singlelevelmenu.jsp` skeleton files—supports the URL simplifying you implement in this step. But for any new skeletons you use in a Look & Feel that have their own `submenu.jsp` and `singlelevelmenu.jsp` files, you must repeat this step for those skeletons as well. For more information on skeletons, see Creating Skeletons and Skeleton Themes in the WebLogic Workshop help system.

Modify the singlelevelmenu.jsp Skeleton File

The `singlelevelmenu.jsp` skeleton file generates links to top-level books and pages. In this procedure, you modify the `singlelevelmenu.jsp` file in the default skeleton.

1. In WebLogic Workshop, open the following file:

   `<portalWebProject>/framework/skeletons/default/singlelevelmenu.jsp.`

2. In the file header, add the following to the list of imported classes:

   `com.bea.netuix.servlets.manager.AppContext`

3. In the file, just below the import statements, locate the following line:

   ```
   BookPresentationContext book =
   BookPresentationContext.getBookPresentationContext(request);
   ```

   Directly above that line, paste the code in Listing 1.

Listing 4 Code to Split the Request URI

```
String shortpath = "";
String [] path = request.getRequestURI().split("[/.]");
if (AppContext.getAppContext(request).isDotPortal())
   {
     shortpath = request.getContextPath() + "/page/" + path[3] + "/" + path[2] +
        "/";
```

```
   }
else
   {
     shortpath = request.getContextPath() + "/page/" + path[3] + "/" + path[4] +
        "/";
   }
```

The code splits the URI into pieces at each forward slash and dot (“[/.]”), detects whether the URL is for a single `.portal` file or a streaming portal, and rearranges the URL to use the simplified format. For example, for a `.portal` file called `sample.portal`, the split creates pieces called `sample` and `portal`, and the new simplified URL reassembles those pieces as `portal/sample`. The simplified URL is assigned to the variable `shortpath`.

4. Further down in the file, locate the code block shown in Listing 2:

Listing 5 Code block from singlefilemenu.jsp

```
else
{
   if (inactiveImage == null)
   {
     %><li class="<%= menuItemClass %>"><a href="<render:pageUrl pageLabel="<%=
       pageCtx.getDefinitionLabel() %>"/>"><%= pageCtx.getTitle() %></a></li><%
   }
   else if (rolloverImage == null)
   {
     %><li class="<%= menuItemClass %>"><a href="<render:pageUrl pageLabel="<%=
       pageCtx.getDefinitionLabel() %>"/>"><img src="<%= inactiveImage %>"></a>
       </li><%
   }
   else
   {
   %><li class="<%= menuItemClass %>"><a href="<render:pageUrl pageLabel=
      "<%= pageCtx.getDefinitionLabel() %>"/>"><img src="<%= inactiveImage
      %>" longDesc="<%= rolloverImage %>"></a></li><%
   }
}
```

This code generates links for books and pages that use images, rollover images, or text for navigation.

5. Replace the `href` values, highlighted in Listing 2, with the following code:

   ```
   <%= shortpath + childPageCtx.getDefinitionLabel() %>
   ```

   The links become the `shortpath` you defined, along with the book/page definition labels needed to complete the URL.

6. Save the file.

Modify the submenu.jsp Skeleton File

The `submenu.jsp` skeleton file generates the links to books and pages below the top-level book and page links. In this procedure, you modify the `submenu.jsp` file in the default skeleton. Modifying the `submenu.jsp` skeleton file is almost identical to modifying `singlelevelmenu.jsp` in the previous procedure.

1. In WebLogic Workshop, open the following file:

   `<portalWebProject>/framework/skeletons/default/submenu.jsp.`

2. In the file header, add the following to the list of imported classes:

   `com.bea.netuix.servlets.manager.AppContext`

3. In the file, locate the following line:

   `if (!bookCtx.isHidden()`

   Directly above that line, paste the code in Listing 3.

Listing 6 Code to Split the Request URI

```
String shortpath = "";
String [] path = request.getRequestURI().split("[/.]");
if (AppContext.getAppContext(request).isDotPortal())
{
   shortpath = request.getContextPath() + "/page/" + path[3] + "/" + path[2] +
      "/";
}
else
{
   shortpath = request.getContextPath() + "/page/" + path[3] + "/" + path[4] +
      "/";
}
```

4. Further down in the file, locate the code block shown in Listing 4.

Listing 7 Code block from submenu.jsp

```
<a class="<%= menuItemLinkClass %>" href="<%=PageURL.createPageURL(request,
response, childPageCtx.getDefinitionLabel()).toString()
%>"><%=childPageCtx.getTitle() %></a>
```

5. Replace the `href` value, highlighted in Listing 4, with the following code:

   `<%= shortpath + childPageCtx.getDefinitionLabel() %>`

The link becomes the `shortpath` you defined, along with the book/page definition label needed to complete the URL.

6. Save the file.

Modify Book and Page Definition Labels

The portal framework uses book and page Definition Labels in the book and page URLs rather than the book and page Titles your visitors see. That means the book or page Definition Label appears in the Web browser address bar when a visitor clicks that book or page.

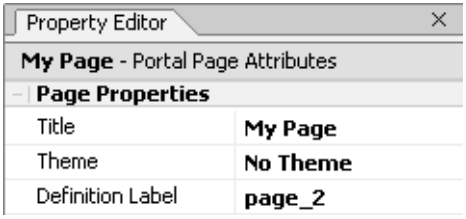
For example, a user sees a page called My Page (the Title), but when the user clicks My Page, the URL in the address bar ends with /page_2 (the Definition Label). Change your Definition Labels to the text you want to appear for the book/page name in the URL.

Note: You cannot use spaces or special characters in Definition Labels.

To modify a definition label:

1. With a portal file open in WebLogic Workshop, select a book or page.
2. In the Property Editor (Figure 3), modify the Definition Label, and press **Enter** or **Tab**.
3. Save the portal file.

Figure 3 Page Title and Definition Label



Step 4. Test the URLs

1. With a portal file open in WebLogic Workshop, make sure the Look & Feel value in the Property Editor is set to a Look & Feel that uses the default skeleton.
2. Select **Portal > Open Current Portal**. Your portal appears in a browser window. Navigate among the books and pages to see the simplified URLs.

You can also create a streaming portal desktop using your portal file as a template in the WebLogic Administration Portal. After you create the desktop, view it and navigate among the books and pages to see the simplified URLs.

For instructions, see Create a Desktop in the WebLogic Administration Portal help system.