Decorating with Parameters

As discussed in Decorating Beyond URL Patterns, SiteMesh out of the box uses sitemesh-default.xml which contains mappers to how decorators are applied.

In some cases you want to go beyond the defaults and use more advanced mappers or even define you own custom mappers.

This example will take you through configuring your own sitemesh.xml mapper file to decorate using query string parameters. After completing this tutorial you will be able to apply similar steps to make use of other advanced mappers.

Creating Your Own sitemesh.xml

To start extract the sitemesh-default.xml file from your SiteMesh jar file into your web applications WEB-INF directory and rename it sitemesh.xml.

Alternatively, download the sitemesh-default.xml matching the version you are using from GitHub.

Here are the contents of sitemesh-default.xml from sitemesh-2.4.1.jar.

```
<sitemesh>
    <property name="decorators-file" value="/WEB-INF/decorators.xml"/>
    <excludes file="${decorators-file}"/>

 <page-parsers>
  <parser content-type="text/html"
class="com.opensymphony.module.sitemesh.parser.HTMLPageParser" />
 </page-parsers>
 <decorator-mappers>
  <mapper class="com.opensymphony.module.sitemesh.mapper.PageDecoratorMapper">
   <param name="property.1" value="meta.decorator" />
   <param name="property.2" value="decorator" />
  </mapper>
  <mapper class="com.opensymphony.module.sitemesh.mapper.FrameSetDecoratorMapper"/>
  <mapper class="com.opensymphony.module.sitemesh.mapper.PrintableDecoratorMapper">
   <param name="decorator" value="printable" />
   <param name="parameter.name" value="printable" />
   <param name="parameter.value" value="true" />
  </mapper>
  <mapper class="com.opensymphony.module.sitemesh.mapper.FileDecoratorMapper"/>
  <mapper class="com.opensymphony.module.sitemesh.mapper.ConfigDecoratorMapper">
   <param name="config" value="${decorators-file}" />
  </mapper>
 </decorator-mappers>
</sitemesh>
```

Decorate using Query Parameters

With this modification, individual pages will be able to specify the decorator to use with a query string parameter. For example, the following url would override the default coffeebreak pattern based decorator and load a Paris themed decorator.

```
http://sitemesh.homeip.net/coffeebreak/menu.jsp?decorator=parisTheme
```

First, add to the sitemesh.xml the PageDecoratorMapper,

```
<sitemesh>
    <property name="decorators-file" value="/WEB-INF/decorators.xml" />
    <excludes file="${decorators-file}" />

    <page-parsers>
        <parser content-type="text/html"
            class="com.opensymphony.module.sitemesh.parser.HTMLPageParser" />
        <parser content-type="text/html;charset=ISO-8859-1"
            class="com.opensymphony.module.sitemesh.parser.HTMLPageParser" />
    </page-parsers>

    <decorator-mappers>
        <!-- Mapper allows pages to specify decorator using meta tag, -->
        <!-- <meta name="decorator" content="yourDecoratorName"/> -->
        <mapper class="com.opensymphony.module.sitemesh.mapper.PageDecoratorMapper">
            <param name="property.1" value="decorator" />
            <param name="property.2" value="meta.decorator" />
        </mapper>

        <!-- The ConfigDecoratorMapper MUST be located after other mappers. -->
        <mapper class="com.opensymphony.module.sitemesh.mapper.ConfigDecoratorMapper">
            <param name="config" value="${decorators-file}"/>
        </mapper>
    </decorator-mappers>
</sitemesh>
```

Ordering Mapper Elements

The order of the mapper elements determine precedence. A well behaved application should use the following order,

1. Parameter query = ParameterDecoratorMapper
2. Page specific using meta tag = PageDecoratorMapper
3. Pattern = ConfigDecoratorMapper

Keep in mind that the nature of the mapper elements can override the order set by the developer.

For example, a common mistake is to put ConfigDecoratorMapper before PageDecoratorMapper. Because ConfigDecoratorMapper is so general, technically all pages request would cause a decorator to be loaded and as such the PageDecoratorMapper never gets loaded. Even in the case of an exclude in decorators.xml, the match is to not apply a decorator and again PageDecoratorMapper is never loaded.