

# WebDAVRTE

## RTE

[RTE](#) is a portability layer over the "designMode" attribute of IE and Mozilla's Midas API.

This tutorial shows you how to extend the "step4" sample of Cocoon's WebDAV block with RTE for WYSIWYG editing.

First copy the complete "step4" folder to "step6".

Within the "step6" folder create another folder named "rte". Download the [rte.zip](#) and extract it into this folder.

Adjust file2html.xsl to look like the following:

```
<xsl:stylesheet version="1.0"
    xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:param name="file"></xsl:param>
<xsl:param name="sitemapURI"></xsl:param>
<xsl:param name="requestURI"></xsl:param>
<xsl:param name="serverName"></xsl:param>
<xsl:param name="serverPort"></xsl:param>
<xsl:template match="/page">
<html>
  <head>
    <title>Cross-Browser Rich Text Editor</title>
    <style type="text/css">
      .btnImage {cursor: pointer; cursor: hand;}
    </style>
    <base>
      <xsl:attribute name="href">http://<xsl:value-of select="$serverName"/>:<xsl:value-of select="$serverPort"
/><xsl:value-of select="substring-before($requestURI, $sitemapURI)"/>rte/</xsl:attribute>
    </base>
  </head>
  <body>
    <form method="post">
      <xsl:attribute name="action"><xsl:value-of select="substring-before($requestURI, $sitemapURI)"/>write
/><xsl:value-of select="$file"/></xsl:attribute>
      <iframe id="testFrame" style="position: absolute; visibility: hidden; width: 0px; height: 0px;"/>
      <p>Author:<br />
        <input name="author" type="text" size="30" maxlength="30" value="{metapage/author}" />
      </p>
      <p>Category:<br />
        <input name="category" type="text" size="30" maxlength="30" value="{metapage/category}" />
      </p>
      <p>State:<br />
        <input name="state" type="text" size="30" maxlength="30" value="{metapage/state}" />
      </p>
      <p>Title:<br />
        <input name="title" type="text" size="30" maxlength="30" value="{page/title}" />
      </p>
      <hr/>
      <script language="JavaScript" type="text/javascript">
        <xsl:attribute name="src"><xsl:value-of select="substring-before($requestURI, $sitemapURI)"/>rte
/browserdetect.js</xsl:attribute>
      </script>
      <script language="JavaScript" type="text/javascript">
        <xsl:attribute name="src"><xsl:value-of select="substring-before($requestURI, $sitemapURI)"/>rte
/richtext.js</xsl:attribute>
      </script>
      <script language="JavaScript" type="text/javascript">
        <xsl:comment>
          function submitForm() {
            try {
              document.getElementById('para').value = document.getElementById('edit').contentWindow.document.
body.innerHTML;
            }
            catch (e) {
              document.getElementById('para').value = document.getElementById('edit').value;
            }
          }
        </xsl:comment>
      </script>
    </form>
  </body>
</template>
</xsl:stylesheet>
```

```

        document.getElementById('editor').submit();
    }
    Start();
    //</xsl:comment>
</script>
<noscript><p><b>Javascript must be enabled to use this form.</b></p></noscript>
<iframe id="edit" width="510px" height="200px">
    <xsl:attribute name="src">http://<xsl:value-of select="$serverName"/>:<xsl:value-of select="$serverPort"
/><xsl:value-of select="substring-before($requestURI, $sitemapURI)"/>para/<xsl:value-of select="$file"/></xsl:
attribute>
</iframe>
<textarea id="para" name="para" style="position: absolute; visibility: hidden; width: 0px; height: 0px;"
>abc</textarea>
<p><input type="submit" value="Submit" onclick="submitForm()"/></p>
</form>
</body>
</html>
</xsl:template>
</xsl:stylesheet>

```

Change all contents to contain only a single "para" element.

Within the richtext.js file uncomment this line

```
//    document.writeln('<iframe id="edit" width="510px" height="200px"></iframe>');
```

and change the timeout to be called after 1000 ms

```
setTimeout("enableDesignMode()", 1000);
```

add the following match node to the sitemap

```

<map:match pattern="para/**">
  <map:generate src="{global:staging}repo/{1}"/>
  <map:transform src="{global:staging}styles/file2para.xsl"/>
  <map:serialize type="html"/>
</map:match>

```

and adjust the "repo/\*\*" match node to look like this:

```

<map:match pattern="repo/**">
  <map:aggregate element="page" label="content">
    <map:part src="cocoon:/page/{1}"/>
    <map:part src="cocoon:/metapage/{1}"/>
  </map:aggregate>
  <map:transform src="{global:staging}styles/file2html.xsl">
    <map:parameter name="file" value="{1}"/>
    <map:parameter name="requestURI" value="{request:requestURI}"/>
    <map:parameter name="sitemapURI" value="{request:sitemapURI}"/>
    <map:parameter name="serverName" value="{request:serverName}"/>
    <map:parameter name="serverPort" value="{request:serverPort}"/>
  </map:transform>
  <map:serialize type="html"/>
</map:match>

```

Create a file2para.xsl stylesheet

```
<?xml version="1.0"?>
<xsl:stylesheet version="1.0"
    xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:template match="/page">
  <html>
    <xsl:copy-of select="content/para/node()" />
  </html>
</xsl:template>
</xsl:stylesheet>
```

Declare the HTMLTransformer at the beginning of the sitemap

```
<map:transformers default="xalan">
  <map:transformer name="html" src="org.apache.cocoon.transformation.HTMLTransformer">
    <jtidy-config>jtidy-config.txt</jtidy-config>
  </map:transformer>
</map:transformers>
```

"jtidy-config.txt" has to be created with the following content:

```
output-xhtml: no
tidy-mark: no
```

Modify the writing pipeline to contain the HTMLTransformer:

```
<map:match pattern="write/**">
  <map:generate type="request" label="content" />
  <map:transform src="{global:staging}styles/request2doc.xml" />
  <map:transform type="html">
    <map:parameter name="tags" value="para" />
  </map:transform>
  <map:transform src="{global:staging}styles/doc2write.xml">
    <map:parameter name="file" value="{global:staging}repo/{1}" />
  </map:transform>
  <map:transform type="write-source" />
  <map:serialize type="xml" />
</map:match>
```

As a last step you have to modify the "doc2write.xml" stylesheet

```

<?xml version="1.0"?>
<xsl:stylesheet version="1.0"
    xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
    xmlns:req="http://apache.org/cocoon/request/2.0"
    xmlns:source="http://apache.org/cocoon/source/1.0">
<xsl:param name="file"></xsl:param>
<xsl:template match="request/parameters">
<page>
  <source:write create="true">
    <source:source><xsl:value-of select="$file"/></source:source>
    <source:path>page</source:path>
    <source:fragment>
      <title><xsl:value-of select="title"/></title>
      <content>
        <xsl:for-each select="content/para">
          <para>
            <!--the following line is the one to change-->
            <xsl:copy-of select="html/body/node()" />
          </para>
        </xsl:for-each>
      </content>
    </source:fragment>
  </source:write>
  <source:write create="true">
    <source:source><xsl:value-of select="$file"/>.meta</source:source>
    <source:path>metapage</source:path>
    <source:fragment>
      <author><xsl:value-of select="author"/></author>
      <category><xsl:value-of select="category"/></category>
      <state><xsl:value-of select="state"/></state>
    </source:fragment>
  </source:write>
</page>
</xsl:template>
</xsl:stylesheet>

```

Finally add the following match node to the sitemap.

```

<map:match pattern="rte/**">
  <map:read src="{global:staging}rte/{1}"/>
</map:match>

```