

Configuring ColdFusion to process requests with custom file extensions

Posted At : February 17, 2006 9:01 AM | Posted By : Steven Erat

Related Categories: ColdFusion, Macromedia

Technote 18289 describes how to configure ColdFusion MX to handle custom template extensions such as .dbm. The technote is outdated and this blog entry will show you how to configure a custom extension for ColdFusion MX 7. On some occasions I've been asked how to alter the template extension handling in order to mask that ColdFusion runs the web application, so this procedure could be used to even map .htm extensions to ColdFusion.

- Configure the webserver by using the **wsconfig.jar's** -map switch with a comma separated list of extensions in quotes, such as -map ".htm,.html". ColdFusion ships connector installation scripts in the cf_root/bin/connectors/ directory that can help facilitate using the wsconfig utility. You could edit the script IIS_connector.bat provided for you so that it uses the -map switch like this:

```
%CFUSION_HOME%\untime\in\wsconfig -server coldfusion -ws IIS -site 0 -coldfusion -cfwebroot -map ".htm,.html" -v
```

- After using **wsconfig** with the -map switch or using the modified IIS_connector.bat script, the web server is configured to send those extensions to the ColdFusion server, but CF has no internal mapping for them to associate them with the underlying engine, the CFMServlet.
- If a request for index.htm was made at this point, IIS would hand off the request to the connector, but the connector would have a list of mappings that it got from ColdFusion's web.xml file, and since web.xml does not yet contain mappings for .htm, the result will be that the connector returns an HTTP 503 status code to the web server and the web server sends that to the client browser.
- Add the internal mappings by backing up and then editing cf_root/wwwroot/WEB-INF/web.xml to create new custom mappings that associate the extensions with the CFMServlet.
- Use an existing mapping from web.xml as a starting point, like this one:

```
<servlet-mapping id="macromedia_mapping_8">
  <servlet-name>CFMServlet</servlet-name>
  <url-pattern>*.cfm</url-pattern>
</servlet-mapping>
```

- Just after the last servlet-mapping node, copy the starter code into the file, and then and change the *.cfm extension in the url-pattern element to a custom mapping like *.htm or even *.foo, and finally remove the id attribute:

```
<servlet-mapping>
  <servlet-name>CFMServlet</servlet-name>
  <url-pattern>*.htm</url-pattern>
</servlet-mapping>
```

- Then copy and paste that custom mapping to create a variation on the URL Pattern, ending in /* like this:

```
<servlet-mapping>
  <servlet-name>CFMServlet</servlet-name>
  <url-pattern>*.htm/*</url-pattern>
</servlet-mapping>
```

- Repeat for each custom mapping such as .dbm, .htm, etc... Each custom mapping will require two servlet-mapping nodes, one for *.ext and another for *.ext/* (where the *.ext/* mapping helps map Search Engine Safe URLs such as /blog/index.cfm/2006/2/17/CF-Custom-File-Extensions for example)
- Save web.xml

- Restart the ColdFusion server AND restart the web server.