

Example sitemap snippets (2.1 legacy document)

Table of contents

1 Comments.....4

Table of contents

1 Overview.....	3
2 C2 documentation.....	3

Warning:

This document was copied as is from the Cocoon 2.1 documentation, but has not yet been fully reviewed or moved to its new home.

1. Overview

The following explanations describe how some sections of the sitemaps operate. For overview, see the general [Sitemap](#) document and the specific userdocs.

2. C2 documentation

The Cocoon documentation XML files are located at documentation/xdocs/*.xml and sub-directories. The sitemap at documentation/sitemap.xmap does most of the work. It is used by build docs and it also mounted by the main sitemap to handle requests for documentation as a servlet.

Each match is presented and described in turn.

Match #1 ... `<!-- ===== C2 documentation ===== -->`
`<map:match pattern=""> <map:redirect-to uri="index.html"/> </map:match>`

Just a convenience for redirecting the null pattern to the home page.

Match #2 ... `<map:match pattern="*.html"> <map:aggregate element="site"> <map:part src="cocoon:/book-{1}.xml"/> <map:part src="cocoon:/body-{1}.xml"/> </map:aggregate>`
`<map:transform src="stylesheets/site2xhtml.xml"> <map:parameter name="use-request-parameters" value="true"/> <map:parameter name="header" value="graphics/{1}-header.jpg"/> </map:transform> <map:serialize/> </map:match>`

Generates the response for a top-level page (e.g. index.html). Note that lower-level pages (e.g. userdocs/index.html) do not match here and are processed in a similar way by Match #3.

Each document is composed of two separate parts: the side-panel menu and the actual page content. Each part is build separately and then aggregated. Consider the example of the home page (index.html). The side-panel content comes from documentation/xdocs/book.xml (and is handled by the pipeline book-index.xml). The actual page content comes from documentation/xdocs/index.xml (and is handled by the pipeline body-index.xml).

So Match #2 generates an intermediate XML file by aggregating the XML output from the two pipelines and wrapping it with a `<site>` element. (The `cocoon:/` protocol means "Use the relevant pipeline from the current sitemap".) The output stream is then transformed to XHTML and serialized.

Match #3 ... `<map:match pattern="**/*.html"> <map:aggregate element="site"> <map:part src="cocoon:/{1}/book-{1}/{2}.xml"/> <map:part src="cocoon:/body-{1}/{2}.xml"/>`
`</map:aggregate> <map:transform src="stylesheets/site2xhtml.xml"> <map:parameter name="use-request-parameters" value="true"/> <map:parameter name="header" value="graphics/{2}-header.jpg"/> </map:transform> <map:serialize/> </map:match>`

This is similar to Match #2, except that it deals with documents in sub-directories. For the side-panel, this uses the book.xml found in each relevant directory.

Match #4 ... `<map:match pattern="**book-*.xml"> <map:generate src="xdocs/{1}book.xml"/>`
`<map:transform src="stylesheets/book2menu.xml"> <map:parameter name="use-request-parameters" value="true"/> <map:parameter name="resource" value="{2}.html"/> </map:transform> <map:serialize type="xml"/> </map:match>`

This produces the additional structured content for the relevant side-panel menu. Note that this is the only match that serializes to XML (the others have the default HTML).

In essence, this Match #4 is called from Match #2 via the `cocoon:/` protocol and is aggregated with the output of Match #6

```
Match #5 ... <map:match pattern="body-todo.xml"> <map:generate src="xdocs/todo.xml"/>
<map:transform src="stylesheets/todo2document.xml"/> <map:transform
src="stylesheets/document2html.xml"/> <map:serialize/> </map:match> <map:match
pattern="body-changes.xml"> <map:generate src="xdocs/changes.xml"/> <map:transform
src="stylesheets/changes2document.xml"/> <map:transform
src="stylesheets/document2html.xml"/> <map:serialize/> </map:match> <map:match
pattern="body-faq.xml"> <map:generate src="xdocs/faq.xml"/> <map:transform
src="stylesheets/faq2document.xml"/> <map:transform src="stylesheets/document2html.xml"/>
<map:serialize/> </map:match>
```

These matches are for special cases. Note that they are defined before the general `body-*.xml` match. Remember that the sitemap reads from top to bottom and the first encountered match is used.

These special-cases need to transform the content into XML files that conform to `document-v10.dtd` first.

```
Match #6 ... <map:match pattern="body-*.xml"> <map:generate src="xdocs/{1}.xml"/>
<map:transform src="stylesheets/document2html.xml"/> <map:serialize/> </map:match>
```

This match handles the bodies of all other the other documents (which are natively conforming to `document-v10.dtd`).

In essence, this Match #6 is called from Match #2 via the `cocoon:/` protocol and is aggregated with the output of Match #4

1. Comments

add your comments