

Configuration of Cocoon for Offline Use

(2.1 legacy document)

Table of contents

1 Comments.....4

Table of contents

1 Overview.....	3
2 The xconf Configuration Format.....	3
3 Sample xconf File.....	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

This page details the xconf configuration format for configuring Cocoon's offline page and site generation capabilities.

This page gives details of how to configure Cocoon. Details of the concepts behind offline page generation are given on the offline generation [overview](#) page.

2. The xconf Configuration Format

The `{{xconf}}` configuration format can be used both within the Cocoon [Command Line Interface](#) and within the [Cocoon Ant Task](#).

3. Sample xconf File

Below is a sample XConf file. The details of the file are detailed as comments within this sample itself.

```
<?xml version="1.0"?> <!--+ | This is an Apache Cocoon command line configuration file. |
Here you give the command line interface details of where | to find various aspects of your
Cocoon installation. | | If you wish, you can also use this file to specify the URIs | that you
wish to generate. | | The current configuration information in this file is for | building the
Cocoon documentation. Therefore, all links here | are relative to the build context dir, which,
in the build.xml | file, is set to ${build.context} | | Options: | verbose: increase amount of
information presented | to standard output (default: false) | follow-links: whether linked pages
should also be | generated (default: true) | precompile-only: precompile sitemaps and XSP
pages, but | do not generate any pages (default: false) | confirm-extensions: check the mime
type for the generated page | and adjust filename and links extensions | to match the mime
type | (e.g. text/html->.html) | context-dir: The context directory is usually the webapp |
directory containing the sitemap.xmap file. | config-file: The config file is the cocoon.xconf file,
| usually found within the WEB-INF folder | work-dir: The work directory is used by Cocoon to
| store temporary files and cache files. | dest-dir: The destination directory is where generated
| pages will be written (assuming the 'simple' | mapper is used, see below) | +--> <cocoon
verbose="true" follow-links="true" precompile-only="false" confirm-extensions="true"
context-dir="build/webapp/xconf" config-file="WEB-INF/cocoon.xconf" work-dir="build/work"
dest-dir="build/dest"> <!--+ | Broken link reporting options: | Report into a text file, one link
per line: | <broken-links type="text" report="filename"/> | Report into an XML file: |
<broken-links type="xml" report="filename"/> | Ignore broken links (default): | <broken-links
type="none"/> | | Two attributes to this node specify whether a page should | be generated
when an error has occurred. 'generate' specifies | whether a page should be generated
(default: true) and | extension specifies an extension that should be appended | to the
generated page's filename (default: none) | | Using this, a quick scan through the destination
directory | will show broken links, by their filename extension. +--> <broken-links type="xml"
file="brokenlinks.xml" generate="false" extension=".error"/> <!--+ | Load classes at startup.
This is necessary for generating | from sites that use SQL databases and JDBC. | The
<load-class> element can be repeated if multiple classes | are needed. +--> <!--
<load-class>org.firebirdsql.jdbc.Driver</load-class> --> <!--+ | Configures logging. | The
'log-kit' parameter specifies the location of the log kit | configuration file (usually called
```

logkit.xconf. | | Logger specifies the logging category (for all logging prior | to other Cocoon logging categories taking over) | | Available log levels are: | DEBUG: prints all level of log messages. | INFO: prints all level of log messages except DEBUG | ones. | WARN: prints all level of log messages except DEBUG | and INFO ones. | ERROR: prints all level of log messages except DEBUG, | INFO and WARN ones. | FATAL_ERROR: prints only log messages of this level +--> <logging log-kit="build/webapp/WEB-INF/logkit.xconf" logger="cli" level="DEBUG" /> <!--+ | Specifies the filename to be appended to URIs that | refer to a directory (i.e. end with a forward slash). +--> <default-filename>index.html</default-filename> <!--+ | Specifies a user agent string to the sitemap when | generating the site. +--> <!--

<user-agent>xxx</user-agent> --> <!--+ | Specifies an accept string to the sitemap when generating | the site. +--> <accept>*/*</accept> <!--+ | Specifies which URIs should be included or excluded, according | to wildcard patterns. | | By default, all URIs are included. If both include and exclude | patterns are specified, a URI is first checked against the | include patterns, and then against the exclude patterns. | | Multiple patterns can be given, using muliple include or exclude | nodes. | | The order of the elements is not significant, as only the first | successful match of each category is used. | | Currently, only the complete source URI can be matched (including | any URI prefix). Future plans include destination URI matching | and regexp matching. If you have requirements for these, contact | dev@cocoon.apache.org. +--> <include pattern="**"/> <exclude pattern="docs/apidocs/**"/> <!-- <include-links extension=".html"/>--> <!--+ | <uri> nodes specify the URIs that should be generated, and | where required, what should be done with the generated pages. | | Append: append the generated page's URI to the end of the | source URI: | | <uri type="append" src-prefix="documents/" src="index.html" | dest="build/dest/" /> | | Replace: Completely ignore the generated page's URI - just | use the destination URI: | | <uri type="replace" src-prefix="documents/" src="index.html" | dest="build/dest/docs.html"/> | | Insert: Insert generated page's URI into the destination | URI at the point marked with a * (example uses fictional | zip protocol) | | <uri type="insert" src-prefix="documents/" src="index.html" | dest="zip://*.zip/page.html"/> | | If in any of these scenarios, the dest attribute is omitted, | the value provided globally using the <dest-dir> node will | be used. +--> <uri type="replace" src-prefix="samples/" src="hello-world/hello.html" dest="build/dest/hello-world.html"/> <!--+ | <uri> nodes can be grouped together in a <uris> node. This | enables a group of URIs to share properties. The following | properties can be set for a group of URIs: | * follow-links: should pages be crawled for links | * confirm-extensions: should file extensions be checked | for the correct mime type | * src-prefix: all source URIs should be | pre-pended with this prefix before | generation. The prefix is not | included when calculating the | destination URI | * dest: the base destination URI to be | shared by all pages in this group | * type: the method to be used to calculate | the destination URI. See above | section on <uri> node for details. | | Each <uris> node can have a name attribute. When a name | attribute has been specified, the -n switch on the command | line can be used to tell Cocoon to only process the URIs | within this URI group. When no -n switch is given, all | <uris> nodes are processed. Thus, one xconf file can be | used to manage multiple sites. +--> <uris name="docs" follow-links="true"> <uri type="append" src-prefix="docs/" src="index.html" dest="build/dest/" /> </uris> <uris name="samples" follow-links="false" confirm-extensions="true" src-prefix="samples/" dest="build/dest/examples/" type="append" > <uri src="" /> <uri src="hello-world/" /> <uri src="hello-world/hello.html"/> <uri src="hello-world/hello.xml"/> </uris> <!--+ | File containing URIs (plain text, one per | line). +--> <!-- <uri-file></uri-file> --> </cocoon>

1. Comments

add your comments