

# XPath Directory Generator (2.1 legacy document)

## Table of contents

1 Comments.....4

## Table of contents

1 XPath Directory Generator.....	3
2 Additional Configuration.....	3
3 Extended DTD.....	3
4 Example.....	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. XPath Directory Generator

Generates an XML directory listing performing XPath queries on XML files.

- Name: xpathdirectory
- Class: org.apache.cocoon.generation.XPathDirectoryGenerator
- Cacheable: yes

The XPath Directory Generator provides all the functionality of the [Directory Generator](#). Additionally it is possible to get XML snippets out of the XML files the Directory Generator finds.

## 2. Additional Configuration

```
<map:generate type="xpathdirectory" src="the/requested/directory"> <map:parameter
name="xpath" value="/article/title|/article/abstract"/> <map:parameter name="xmlFiles"
value="\.x.*$"/> </map:generate>
```

The XPath Directory Generator has two additional parameters, both are optional.

- xpath: Sets the XPath the XPath Directory Generator should use for queries on XML files. If you don't set this parameter it will behave like the Directory Generator.
- xmlFiles: The xml files pattern. Specifies the files that should be handled as XML files. XPath queries will only be tried on files matching this pattern. The XPath Directory Generator does not fail on non-XML files or files that are not well-formed or not valid. All Exceptions on parsing the files will be caught and ignored. But of course, useless parsing, throwing and catching exceptions is very time consuming, so the xmlFiles pattern should not be too generic. If you specify an empty pattern all files will be handled as XML files. The default pattern when not specifying this parameter is `\.xml$`, so that all files ending `.xml` are handled as XML files. The pattern is a regular expression as described in the API docs of the [Apache RegExp project](#).

## 3. Extended DTD

```
<!ELEMENT directory (directory|file)*> <!ATTLIST directory name CDATA #REQUIRED
lastModified CDATA #REQUIRED date CDATA #REQUIRED size CDATA #REQUIRED
requested CDATA #IMPLIED sort CDATA #IMPLIED reverse CDATA #IMPLIED>
<!ELEMENT file (xpath?)> <!ATTLIST file name CDATA #REQUIRED lastModified CDATA
#REQUIRED date CDATA #REQUIRED size CDATA #REQUIRED> <!ELEMENT xpath
#ALL> <!ATTLIST xpath query CDATA #REQUIRED>
```

## 4. Example

The current XPath Directory Generator may generate following xml:

```
<dir:directory xmlns:dir="http://apache.org/cocoon/directory/2.0" name="articles"
lastModified="1057183738609" date="03.07.03 00:08" size="0" requested="true"
sort="name" reverse="false"> <dir:directory name="images" lastModified="1057183738609"
date="03.07.03 00:08" size="0"/> <dir:file name="article1.xml"
lastModified="1057183738609" date="03.07.03 00:08" size="123"> <dir:xpath
query="/article/title"> <title>My first article!</title> </dir:xpath> </dir:file> <dir:file
name="article2.html" lastModified="1057183738609" date="03.07.03 00:08" size="345"/>
```

```
<dir:file name="article2.xml" lastModified="1057183738609" date="03.07.03 00:08"
size="234"> <dir:xpath query="/article/title"> <title>My second article!</title> </dir:xpath>
</dir:file> </dir:directory>
```

## 1. Comments

add your comments