Getting started with Cocoon

Barry Cornelius
Computing Services, University of Oxford

Date 10th January 2006; first created: 8th January 2006
http://www.barrycornelius.com/papers/

mailto:barry.cornelius@oucs.ox.ac.uk

1 Introduction

These notes are concerned with obtaining, installing and running Cocoon on Windows.

The notes assume that Java has been installed on the computer. You can check this using the command:

```
java -version
```

These notes also assume that jar, the Java Archive utility, is available. You can check this by asking jar to output its help information:

```
jar --help
```

2 Obtaining Cocoon

Cocoon is described at http://cocoon.apache.org/. It can be downloaded from http://cocoon.apache.org/mirror.cgi. Click on the link that is next to *ZIP format (Windows platforms):*. When I did this it said *cocoon-2.1.7-src.zip*, but a later release is now available. So download that and replace any references to 2.1.7 that are given below by whatever is appropriate for you. Note the file cocoon-2.1.7-src.zip is nearly 50MB.

Save the file cocoon-2.1.7-src.zip in some directory on your computer such as c:\install.

3 Installing Cocoon

In a command shell window, type the commands:

```
c:
cd \
jar xf c:\install\cocoon-2.1.7-src.zip
```

This should create a directory called c:\cocoon-2.1.7

```
cd \cocoon-2.1.7
dir
```

The instructions for installation are in the file install.txt. So read this file.

You will see that you will need the JAVA_HOME variable to be set. Check that this is the case using:

```
echo %JAVA_HOME%
```

I got the reply:

```
c:\Program Files\Java\jdk1.5.0_01
```

The instructions in the file install.txt suggest you now issue the commands:

```
cd \cocoon-2.1.7
build
```

After about a minute it starts to report things to the command shell window. It is creating a lot of files. You will see lines like:

```
Compiling 279 source files to C:\cocoon-2.1.7\build\cocoon-2.1.7\blocks\forms\dest
```

This is making use of the Java compiler. It seems only to create files in subdirectories (of c:\cocoon-2.1.7) called tools and build. At another point it said:

```
Copying 5947 files to C:\cocoon-2.1.7\build\webapp\api\java
```

That took a long time to execute!

I ignored the error messages that it produced as they didn’t look too important.

When it finishes it says something like:

```
BUILD SUCCESSFUL
Total time: 21 minutes 42 seconds
```

4 Running Cocoon

The install.txt file says you should now type the command:

```
cocoon servlet
```

When you do this, another command shell window should appear. About 14 lines are output ending in the line:

```
08:34:52.315 EVENT  Starting Jetty/4.2.23
```

It then does a lot of work. After about 4 minutes it outputs some more lines ending in the lines:

```
08:38:39.412 EVENT  Started SocketListener on 0.0.0.0:8888
08:38:39.412 EVENT  Started org.mortbay.jetty.Server@f0eed6
```

5 Testing Cocoon

Now start up a web browser and visit the URL http://localhost:8888/.

Hopefully you will get a page displayed saying:

```
Welcome to Apache Cocoon!
Congratulations! If you are reading this page,
it means that your Apache Cocoon installation was successful.
To know more about Cocoon capabilities, look at the samples
...
```

If you click on the link behind the word *samples*, you will be taken to the page http://localhost:8888/samples/.

If you then click on *Hello World!*, you will be taken to http://localhost:8888/samples/hello-world/.

If you then click on XML, you will get some XML output. If you instead click on HTML, you will see that you are taken to http://localhost:8888/samples/hello-world/hello.html. And the XML is rendered as HTML. If you instead click on XHTML, you will see that you are taken to http://localhost:8888/samples/hello-world/hello.xhtml. And the XML is rendered as XHTML. There are other links down the left-hand side of this page that enable you to get the XML rendered in other formats.

The XML is in the file:

```
C:\cocoon-2.1.7\build\webapp\samples\hello-world\content\hello.xml
```

The XSL files that do the transformations are in the directory:

```
C:\cocoon-2.1.7\build\webapp\samples\hello-world\style\xsl
```

For example, the file that does the transformation from XML to HTML is in the file:

```
C:\cocoon-2.1.7\build\webapp\samples\hello-world\style\xsl\page2html.xsl
```

So alter both the hello.xml and the page2html.xsl files and then revisit http://localhost:8888/samples/hello-world/hello.html. In hello.xml I altered *first* to *2nd*, and in page2html.xsl I altered the h1 element to an h2 element.

When rendering the HTML, a CSS file is used. This is because page2html.xsl outputs a link element referring to a css file called main.css. The main.css file is in the directory:

```
C:\cocoon-2.1.7\build\webapp\resources\styles
```

Although this centre's an h1 element it does not centre an h2 element. So this is a change that will occur if you alter the h1 to an h2.

Note that there are no actual files called hello.html or hello.xhtml. Instead the system is configured so that if you visit a URL like http://localhost:8888/samples/hello-world/hello.html it has to take the file content\hello.xml and apply the transformation in the file xsl\page2html.xsl.

The file that controls all this is the file:

```
C:\cocoon-2.1.7\build\webapp\samples\hello-world\sitemap.xmap
```

If you look at this file, you will see the instructions that control what transformations should take place.

6 Shutting down Cocoon and restarting it

If you want to shutdown Cocoon, go to the command shell window, the one containing:

```
08:38:39.412 EVENT  Started SocketListener on 0.0.0.0:8888
08:38:39.412 EVENT  Started org.mortbay.jetty.Server@f0eed6
```

and press Control-C. The command shell window will then be removed.

To startup Cocoon again, first ensure that JAVA_HOME is set and then execute the commands:

```
cd \cocoon-2.1.7
cocoon servlet
```

Once again, it will start up another command shell window. Wait for lines like:

```
09:25:04.787 EVENT  Started SocketListener on 0.0.0.0:8888
09:25:04.787 EVENT  Started org.mortbay.jetty.Server@f0eed6
```

to appear before using a web browser to go to http://localhost:8888/.

7 Building your own Cocoon application by copying

If you want to build your own Cocoon application, then create a subdirectory in the build\webapp directory and put the files in there. The easiest way is to copy an existing subdirectory. For example, here are some instructions for creating a subdirectory called hw and putting the *Hello World!* files in that subdirectory:

```
cd c:\cocoon-2.1.7\build\webapp
mkdir hw
xcopy /s/e samples\hello-world hw
```

Now you can visit URLs like http://localhost:8888/hw/ and http://localhost:8888/hw/hello.html.

8 Building a Cocoon application from scratch

The *Hello World!* example demonstrates that it is easy to transform an XML file into a large number of different formats. It uses a directory structure to keep this organised.

The following instructions show you how to build a subdirectory that has no structure and has the smallest number of files that are needed to do a transformation.

First create a new subdirectory:

```
cd c:\cocoon-2.1.7\build\webapp
mkdir hwsmall
```

And then copy some files across:

```
copy hw\content\hello.xml hwsmall
copy hw\style\xsl\page2html.xsl hwsmall
```

Now move to the hwsmall subdirectory:

```
cd hwsmall
```

In the hwsmall subdirectory, create a file called sitemap.xmap that just has the lines:

```
<?xml version="1.0"?>
<map:sitemap xmlns:map="http://apache.org/cocoon/sitemap/1.0">
 <map:pipelines>
  <map:pipeline>
   <map:match pattern="">
    <map:generate src="hello.xml"/>
    <map:serialize type="xml"/>
   </map:match>
   <map:match pattern="hello.html">
    <map:generate src="hello.xml"/>
    <map:transform src="page2html.xsl"/>
    <map:serialize type="html"/>
   </map:match>
   <map:match pattern="hello.txt">
    <map:generate src="hello.xml"/>
    <map:serialize type="text"/>
   </map:match>
  </map:pipeline>
 </map:pipelines>
</map:sitemap>
```

Note: do not include the line numbers (and the colons).

Now use your web browser to go to the URL http://localhost:8888/hwsmall/. That should display the XML. And http://localhost:8888/hwsmall/hello.html should render the XML as HTML. And http://localhost:8888/hwsmall/hello.txt should produce some text.

9 Using the TEI Stylesheets

To download the TEI Stylesheets, go to http://tei.sf.net/ and click on the *Released Files* link that is towards the bottom of the page. On the new page, there is a table of packages. One of these is the *Stylesheets* package. This has a *Release* number such as 5.2.9. Click on the *Download* button that is on the same line. On the new page, there is a link that is labelled something like *tei-xsl-5.2.9.zip*. Click on this link. On the new page, click on the *Download* link that is on a line for a suitable mirror site. On the new page, it says that the download will start automatically. Save the download into a directory on your computer such as c:\install.

Now issue the commands:

```
cd c:\cocoon-2.1.7\build\webapp
mkdir hwtei
cd hwtei
mkdir content
mkdir style
cd style
jar xf c:\install\tei-xsl-5.2.9.zip
move tei-xsl-5.2.9 tei
```

That has created a big directory called:

```
c:\cocoon-2.1.7\build\webapp\hwtei\style\tei
```

Now use a text editor to create the file

```
c:\cocoon-2.1.7\build\webapp\hwtei\content\hello.xml
```

It should contain the following lines:

```
<?xml version="1.0"?>
<TEI.2>
  <teiHeader>
    <fileDesc>
      <titleStmt>
        <title>Hello World! page</title>
      </titleStmt>
      <publicationStmt>
        <date>2006-01-08</date>
      </publicationStmt>
    </fileDesc>
  </teiHeader>
  <text>
    <body>
      <div>
        <head>Introduction</head>
        <p>This is an introduction to my latest Cocoon page!</p>
      </div>
      <div>
        <head>Hello</head>
        <p>This is my latest Cocoon page!</p>
      </div>
    </body>
  </text>
</TEI.2>
```

Note: once again do not include the line numbers (and the colons).

Use a text editor to create the file:

```
c:\cocoon-2.1.7\build\webapp\hwtei\sitemap.xmap
```

It should contain the lines:

```
<?xml version="1.0"?>
<map:sitemap xmlns:map="http://apache.org/cocoon/sitemap/1.0">
 <map:pipelines>
  <map:pipeline>
   <map:match pattern="">
    <map:generate src="content/hello.xml"/>
    <map:serialize type="xml"/>
   </map:match>
   <map:match pattern="hello.html">
    <map:generate src="content/hello.xml"/>
    <map:transform src="style/tei/p4/html/tei.xsl"/>
    <map:serialize type="html"/>
   </map:match>
   <map:match pattern="hello.pdf">
    <map:generate src="content/hello.xml"/>
    <map:transform src="style/tei/p4/fo/tei.xsl"/>
    <map:serialize type="fo2pdf"/>
   </map:match>
  </map:pipeline>
 </map:pipelines>
</map:sitemap>
```

Note: once again do not include the line numbers (and the colons).

Now go to the URL http://localhost:8888/hwtei/. This will output the XML. And if you go to http://localhost:8888/hwtei/hello.html the XML will be transformed into HTML using the TEI-to-HTML stylesheets. And if you go to http://localhost:8888/hwtei/hello.pdf the XML will be transformed into PDF using the TEI-to-FO stylesheets.