

# WHAT YOU \*NEED\* TO KNOW ABOUT COCOON

*A path through the jungle*

**author:** Bertrand Delacrétaz, [www.codeconsult.ch/bertrand](http://www.codeconsult.ch/bertrand), [bdelacretaz@codeconsult.ch](mailto:bdelacretaz@codeconsult.ch)

**event:** Cocoon GetTogether, Ghent, October 12th, 2004

**copyright:** (C) 2004 Bertrand Delacrétaz, codeconsult

## 1. WHAT YOU \*NEED\* TO KNOW ABOUT COCOON

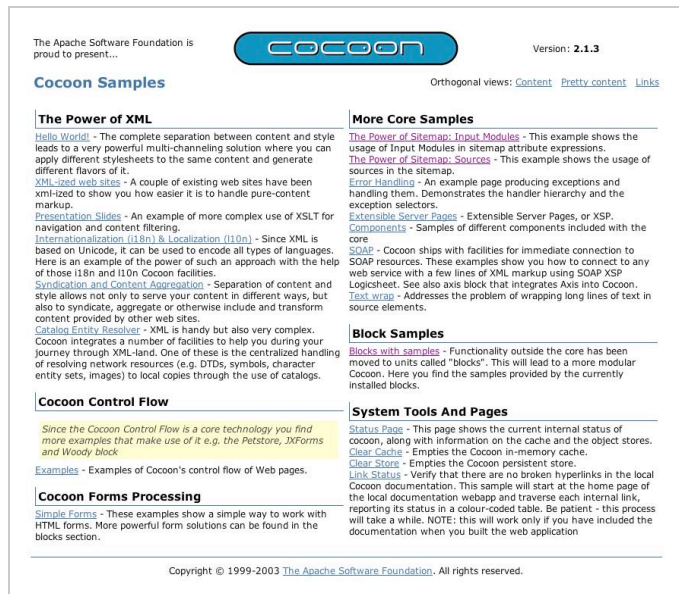
Apache Cocoon: a scary jungle of functionality?

Where to start?

What to learn?

Will I make it?

Will it be worth it?



The Apache Software Foundation is proud to present...

**Cocoon** Version: 2.1.3

Orthogonal views: [Content](#) [Pretty content](#) [Links](#)

**Cocoon Samples**

**The Power of XML**

[Hello World!](#) - The complete separation between content and style leads to a very powerful multi-channelling solution where you can apply different stylesheets to the same content and generate different flavors of it.

[XML-sized web sites](#) - A couple of existing web sites have been xml-ized to show you how easier it is to handle pure-content markup.

[Presentation Slides](#) - An example of more complex use of XSLT for navigation and content filtering.

[Internationalization \(i18n\) & Localization \(l10n\)](#) - Since XML is based on Unicode, it can be used to encode all types of languages. Here is an example of the power of such an approach with the help of those i18n and l10n Cocoon facilities.

[Syndication and Content Aggregation](#) - Separation of content and style allows not only to serve your content in different ways, but also to syndicate, aggregate or otherwise include and transform content provided by other web sites.

[Catalog Entity Resolver](#) - XML is handy but also very complex. Cocoon integrates a number of facilities to help you during your journey through XML-land. One of these is the centralized handling of resolving network resources (e.g. DTDs, symbols, character entity sets, images) to local copies through the use of catalogs.

**Cocoon Control Flow**

[Since the Cocoon Control Flow is a core technology you find more examples that make use of it e.g. the Petstore, JXForms and Woody block](#)

[Examples](#) - Examples of Cocoon's control flow of Web pages.

**Cocoon Forms Processing**

[Simple Forms](#) - These examples show a simple way to work with HTML forms. More powerful form solutions can be found in the blocks section.

**More Core Samples**

[The Power of Sitemap: Input Modules](#) - This example shows the usage of Input Modules in sitemap attribute expressions.

[The Power of Sitemap: Sources](#) - This example shows the usage of sources in the sitemap.

[Error Handling](#) - An example page producing exceptions and handling them. Demonstrates the handler hierarchy and the exception selectors.

[Extensible Server Pages](#) - Extensible Server Pages, or XSP.

[Components](#) - Samples of different components included with the core.

[SOAP](#) - Cocoon ships with facilities for immediate connection to SOAP resources. These examples show you how to connect to any web service with a few lines of XML markup using SOAP XSP Logsheet. See also axis block that integrates Axis into Cocoon.

[Text wrap](#) - Addresses the problem of wrapping long lines of text in source elements.

**Block Samples**

[Blocks with samples](#) - Functionality outside the core has been moved to units called "blocks". This will lead to a more modular Cocoon. Here you find the samples provided by the currently installed blocks.

**System Tools And Pages**

[Status Page](#) - This page shows the current internal status of cocoon, along with information on the cache and the object stores.

[Clear Cache](#) - Empties the Cocoon in-memory cache.

[Clear Store](#) - Empties the Cocoon persistent store.

[Link Status](#) - Verify that there are no broken hyperlinks in the local Cocoon documentation. This sample will start at the home page of the local documentation webapp and traverse each internal link, reporting its status in a colour-coded table. Be patient - this process will take a while. NOTE: this will work only if you have included the documentation when you built the web application

Copyright © 1999-2003 The Apache Software Foundation. All rights reserved.

## 2. WHO'S TALKING?

Bertrand Delacrétaz, Lausanne

[www.codeconsult.ch/bertrand](http://www.codeconsult.ch/bertrand)

ASF Member, Cocoon committer.

Web applications, architecture, programming, mentoring, teaching.



### 3. OUR EXAMPLE: WWW.NOUVO.CH

HTML + CSS

XSLT transforms

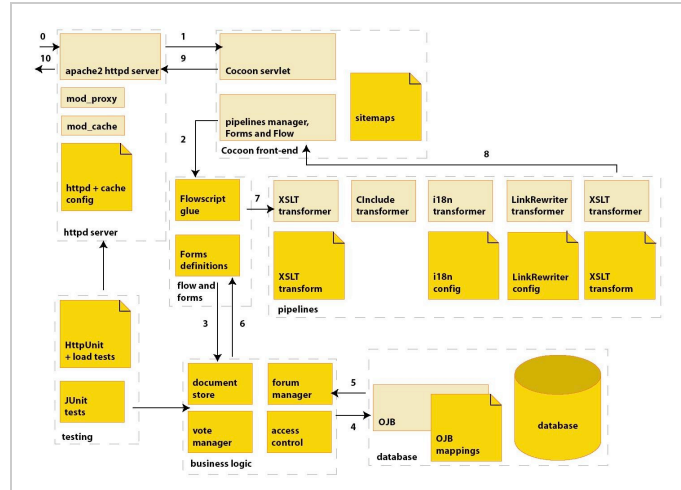
Flowscript for glue

Cocoon Forms

Java code for logic

OJB for data access, MySQL database

It's not the only way!



### 4. PRESENTATION LAYER

Sitemap configuration

XSLT

HTML + CSS

media students learn in 6 days

```
<map:match pattern="**/docs/*.html">
  <map:aggregate element="toc-and-content">
    <map:part src="{1}/docs/{2}.xml"/>
    <map:part src="intro/docs/toc.xml"/>
  </map:aggregate>

  <map:transform src="intro/presentation/insert-toc.xsl"/>
  <map:transform type="cinclude"/>
  <map:transform type="i18n"/>

  <map:serialize type="html"/>
</map:match>
```

### 5. PRESENTATION LAYER TOOLS

i18n Transformer maps keys to messages

LinkRewriter maps named links to the correct relative URLs.

```
<i18n:text key="a_key">article_text1</i18n:text>

<i18n:date-time pattern="FULL"/>

<a href="help-doc:28">help 28</a>
```

## 6. FLOWSCRIPT AS GLUE

Javascript syntax.

Transparent continuations.

Access Cocoon and own components.

Trigger pipelines.

```
docStore = cocoon.getComponent(DS_ROLE);
document =
  docStore.getDocumentById(documentId,forEditing);

if(document == null) {
  if(createEmptyIfNotFound) {
    document = docStore.createEmptyDocument();
    document.setDocumentId("XXX");
    document.setText("ZZZ");
  }
}

cocoon.sendPage(
  "xml-document",
  { "doc" : document }
);
```

## 7. COCOON FORMS

Driven by Flowscript

data model, data binding and form template.

Load-edit-validate-save cycle is transparent.

```
var form = new Form("document-model.xml");
form.createBinding("document-binding.xml");
form.load(document);
form.showForm("/internal/docedit-form");
form.save(document);
```

## 8. BUSINESS LOGIC

Plain old java code.

Isolated from the front-end.

Robust and testable.

```
Criteria crit = new Criteria();
crit.addGreaterThan("ref_date",doc.getRefDate());
crit.addEqualTo("role",doc.getRole());

QueryByCriteria q =
  new QueryByCriteria(DocStoreDocument.class, crit);
q.addOrderBy("ref_date",isNext);
q.setEndAtIndex(1);

setResult(broker.getObjectByQuery(q));
```

## 9. DATABASE ACCESS

OJB rocks ('xcept the name)!

Simple for simple things, powerful when needed.

Stores POJOs, simple XML mapping file.

JDBC/SQL skills needed for setup and debugging.

```
public static void storeProduct(Product product)
{
    PersistenceBroker broker = null;
    try
    {
        broker = PersistenceBrokerFactory.defaultPersistenceBroker();
        broker.beginTransaction();
        broker.store(product);
        broker.commitTransaction();
    }
    catch(PersistenceBrokerException e)
    {
        if(broker != null) broker.abortTransaction();
        // do more exception handling
    }
    finally
    {
        if (broker != null) broker.close();
    }
}
```

## 10. COCOON BLOCKS?

52 blocks today.

www.nouvo.ch uses...two blocks.

apples	jms	qdox
asciart	jsp	repository
authentication-fw	linkrewriter	scratchpad
axis	linotype	serializers
batik	lucene	session-fw
bsf	mail	slide
chaperon	midi	slop
cron	naming	stx
databases	ojb	taglib
deli	paranoid	tour
eventcache	petstore	velocity
fop	php	web3
forms	poi	webdav
hsqldb	portal	woody
html	portal-fw	xmldb
itext	profiler	xsp
javaflow	proxy	
ifor	ovthon	

## 11. SKILLS - INSTALLATION

Life at the command-line.

JVM 1.4 (officially still 1.3.x).

INSTALL.TXT - very easy self-contained  
install: unpack, select blocks, run.

Don't be shy: look at the logs.

```
install JVM

tar zvxf cocoon...

./build.sh

./cocoon.sh servlet
```

## 12. WHERE TO START?

Supersonic Tour block, quick tour with  
live examples (need live Cocoon  
instance).

Documentation Tracks at <http://cocoon.apache.org/2.1>: exploring the  
samples.

Look around.

Be curious.

Leave no stone unturned?

Supersonic Tour of Apache Cocoon : Pipelines  
Pipelines overview Basic pipelines **Generating graphics** XML news source Copyright 1999-2004 The Apache Software Foundation

### Generating graphics

---

#### SVG output

At the end of the pipeline, different Serializers can generate different output formats, provided they're fed the right data (i.e. the required XML elements and namespaces).

Let's generate a simple graph out of our request parameters. If you have an SVG plugin in your browser you can view the result: [red/request.svg](#) or [blue/request.svg](#). If you don't have an SVG plugin see below for the bitmapped version of the same image.

#### Sitemap excerpt

```
<map:match id="svgrequest" pattern="*/request.svg">
  <map:generate type="request"/>
  <map:transform src="xsl/request-to-svg.xsl">
    <map:parameter name="fillColor" value="{1}"/>
  </map:transform>
  <map:serialize type="svgxml"/>
</map:match>
```

#### XSL transformation

Here we show only the main template wich generates the SVG skeleton

```
<xsl:template id="main" match="/">
  <svg width="600" height="500">
    <defs>
      <filter id="blur1">
        <feGaussianBlur stdDeviation="3"/>
      </filter>
      <filter id="blur2">
        <feGaussianBlur stdDeviation="1"/>
      </filter>
    </defs>
    <g title="this is a tooltip">
      <rect
```

### 13. SKILLS - PRESENTATION

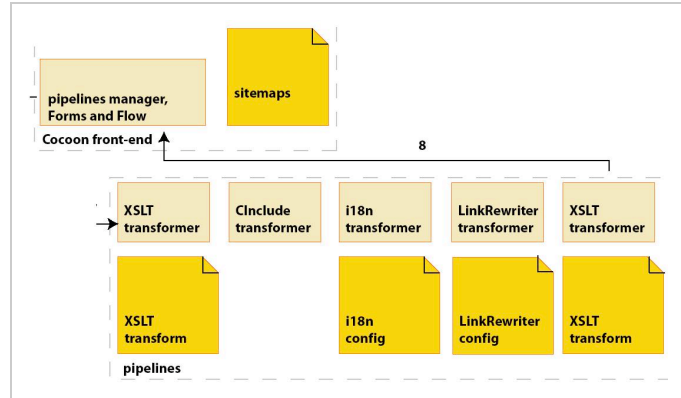
XML, XSLT + presentation markup  
(HTML, CSS, WML, ...)

Cocoon Sitemap configuration

CInclude is very useful (but simple)

i18n and LinkRewriter

HTTPUnit testing?

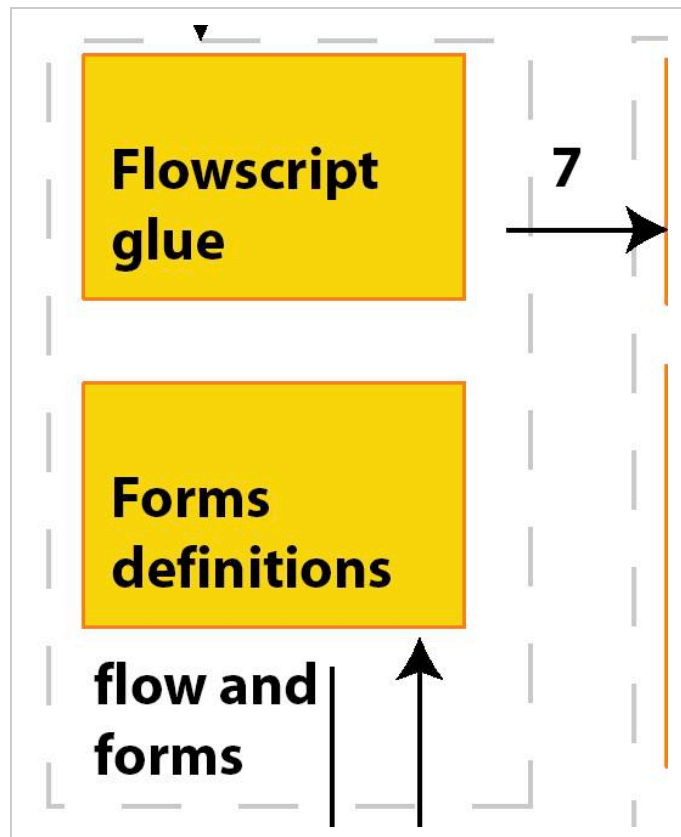


### 14. SKILLS - FORMS

Like presentation + Flowscript and  
Cocoon Forms

Forms concepts: model, binding,  
template. simple for simple problems,  
powerful for complex ones.

Customization requires java skills - but  
standard Forms components do a lot  
already.



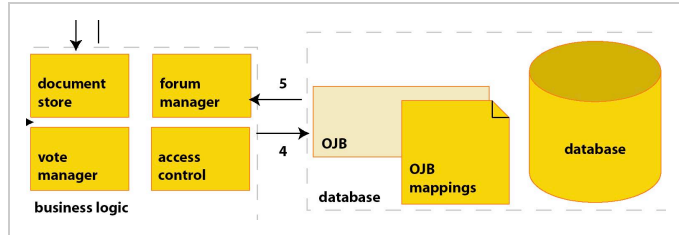
## 15. SKILLS - BACKEND DEVELOPMENT

Java coding, OJB, some SQL

Avalon components (or Spring? Ugo?)

Lucene tends to replace DB searching.

JUnit testing

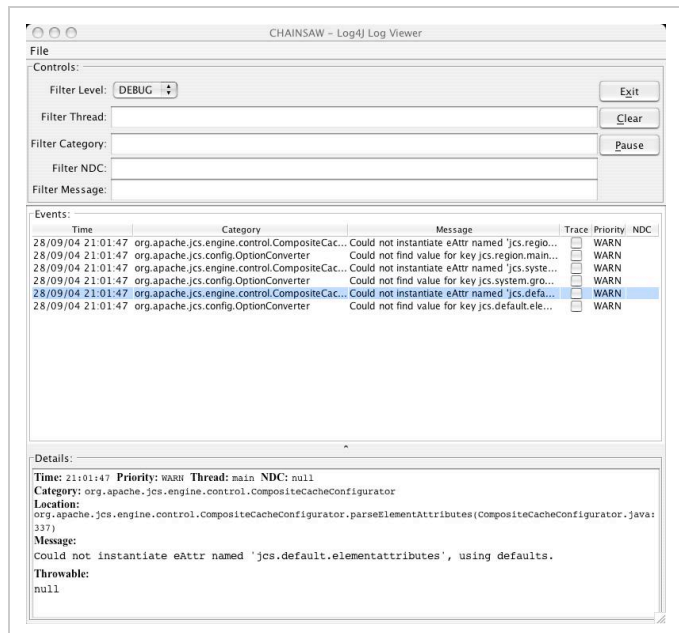


## 16. SKILLS - DEBUGGING

Log configuration and analysis.

Unix tools: grep, sed, awk, tail.

Java code debugging, very useful to explore internals.

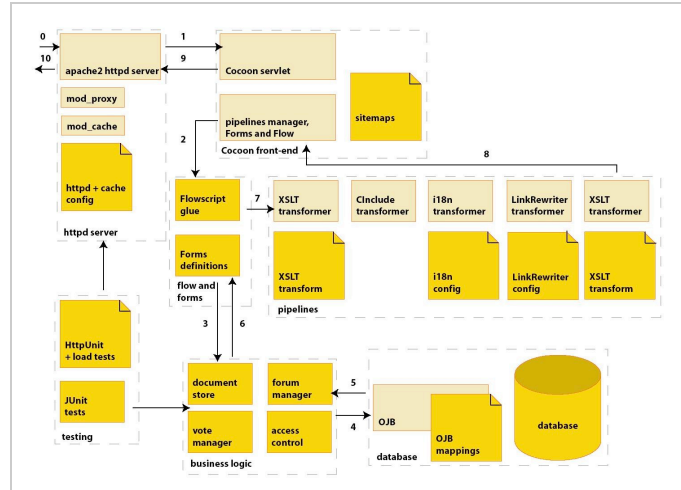




## 17. SKILLS - INTEGRATION

SQL, OJB, Java, JUnit, Flowscript, Cocoon Forms, XSLT, CInclude, i18n, LinkRewriter, Apache httpd configuration, Load testing, HttpUnit, ...

Need a guru?



## 18. WILL YOU MAKE IT?

No one knows everything about Cocoon...too big!

Need to know a little about many things.

Learn as you go, work in small concrete steps.

Need strong integrator / architect skills.

(agora app by Stefano Mazzocchi)





## **19. WILL IT BE WORTH IT?**

YES if you're comfortable with a command line.

YES if you take the time to explore.

YES if your vision is not a short-term one.

YES if yours is a heavy-duty application.

...and many other possible ways. Be yourself..

## **20. CONCLUSIONS**

Apache Cocoon is like the toolbox of a very rich man: lots of tools, lots of options, lots of power...but one can get lost.

XML-based publishing/data conversion is easy.

Full-blown applications need a wide range of skills.

Complete control: the sky is the limit.

Need an explorer's state of mind!

## **21. CODA**

Starting up? Have a look at the Supersonic Tour.

Got the skills? Take some time to explore.

You might like what you see \*a lot\* ;-)

Thanks to the Cocoon community for a great framework!

## 22. REFERENCES

This talk, <http://codeconsult.ch/gt2004>

Cocoon, <http://cocoon.apache.org/2.1>

Supersonic Tour, in the "blocks with samples" menu of the Cocoon samples.

MySQL, <http://www.mysql.com/>

OJB, <http://db.apache.org/ojb>

JUnit, <http://junit.org>

HttpUnit, <http://httpunit.sourceforge.net/>

agora, <http://nagoya.apache.org/~stefano/>