

Using Cocoon 2.2

Using Cocoon 2.2

The Classic Way



If you use Cocoon 2.1:

- ⦿ In multiple software projects
- ⦿ Built multiple reusable, shared software modules
- ⦿ Have standardized directory structure, build, deployment processes...



If you think of Cocoon 2.2:

- ⦿ Ant has been replaced with Maven
- ⦿ LogKit has been replaced by Log4J
- ⦿ Avalon is being replaced by Spring Framework
- ⦿ 'Real Blocks' are almost here
- ⦿ No downloads (so far)



If you think of Cocoon 2.2:

- ⦿ Ant has been replaced with Maven
- ⦿ LogKit has been replaced by Log4J
- ⦿ Avalon is being replaced by Spring Framework
- ⦿ 'Real Blocks' are almost here
- ⦿ No downloads (so far)

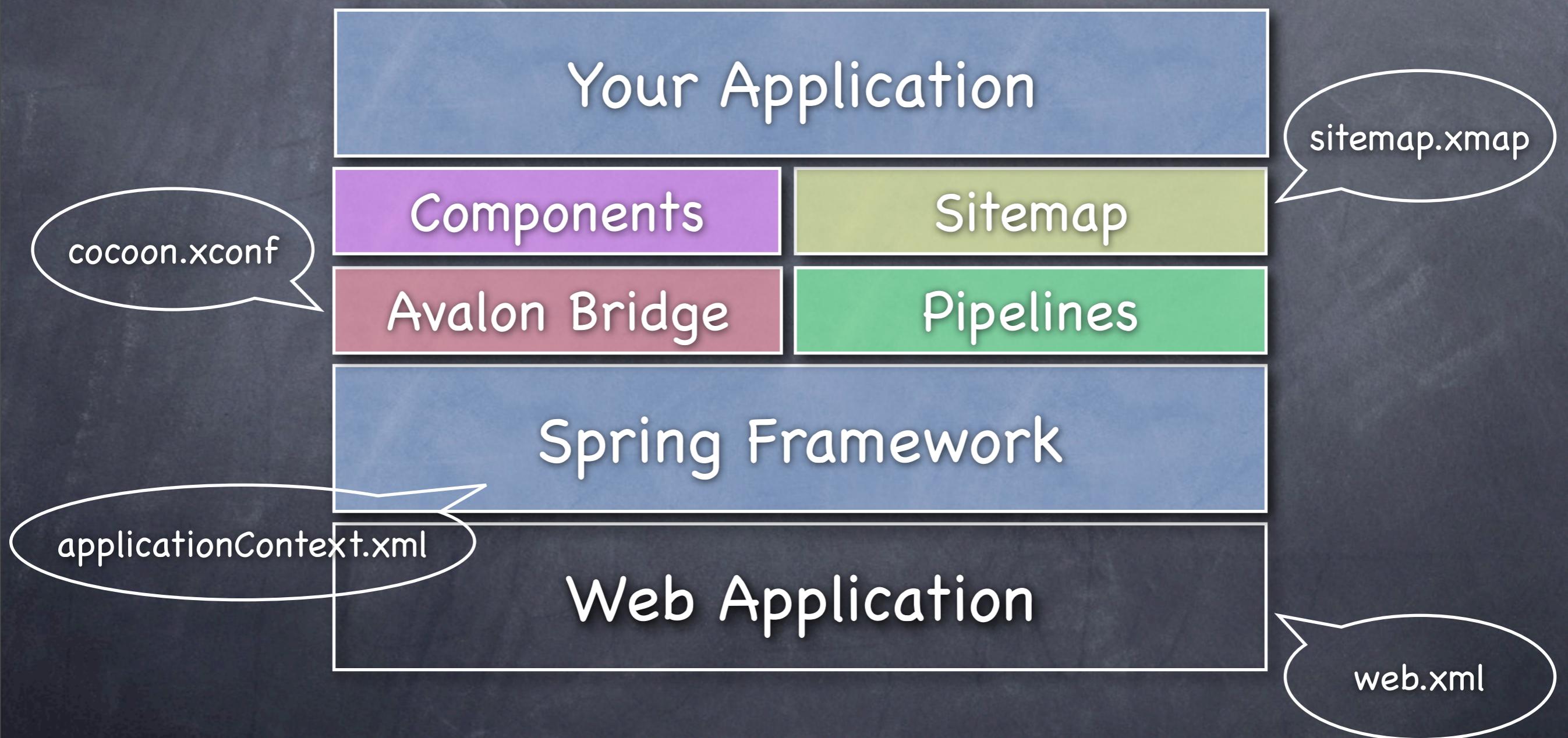


You don't have to start from scratch!

It's not that scary.

- ⦿ Ant has been replaced with Maven
 - ⦿ You don't have to use it.
- ⦿ LogKit has been replaced by Log4J
 - ⦿ You don't have to use it.
- ⦿ Avalon is being replaced by Spring Framework
 - ⦿ You don't have to use it.
- ⦿ No downloads (so far)
 - ⦿ Oops... You'd have to use Maven for that...

So how does it look?



Building Cocoon

```
$ svn co http://svn.apache.org/repos/asf/cocoon/trunk cocoon  
U    cocoon  
Checked out revision 577959.
```

```
$ mvn -P allblocks install  
java.lang.OutOfMemoryError: Java heap  
space
```

- ➊ Install latest maven 2
- ➋ Check out code from the svn repository
- ➌ Start maven build process
- ➍ Get coffee or beer (might take a while)

Building Cocoon

```
$ svn co http://svn.apache.org/repos/asf/cocoon/trunk cocoon  
U      cocoon  
Checked out revision 577959.  
  
$ mvn -P allblocks install  
java.lang.OutOfMemoryError: Java heap  
space  
  
$ export MAVEN_OPTS=-Xmx256m  
$ mvn -Dmaven.test.skip=true -P  
allblocks install  
[INFO] BUILD SUCCESSFUL
```

- ⦿ Install latest maven 2
- ⦿ Check out code from the svn repository
- ⦿ Start maven build process
- ⦿ Get coffee or beer (might take a while)

Building Cocoon

...but only if you want it!

```
$ svn co http://svn.apache.org/repos/asf/cocoon/trunk cocoon  
U      cocoon  
Checked out revision 577959.  
  
$ mvn -P allblocks install  
java.lang.OutOfMemoryError: Java heap  
space  
  
$ export MAVEN_OPTS=-Xmx256m  
$ mvn -Dmaven.test.skip=true -P  
allblocks install  
[INFO] BUILD SUCCESSFUL
```

- ☛ Install latest maven 2
- ☛ Check out code from the svn repository
- ☛ Start maven build process
- ☛ Get coffee or beer (might take a while)

Selecting Blocks

- No more local.build.properties
- No more local.blocks.properties
- Use maven pom.xml file to select blocks
- Need to set a version for each block
- More typing but similar in nature

Selecting Blocks

```
<?xml version="1.0" encoding="UTF-8"?>

<project xmlns="http://maven.apache.org/POM/4.0.0"
          xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
          xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/maven-v4_0_0.xsd">
    <modelVersion>4.0.0</modelVersion>
    <groupId>org.reverycodes</groupId>
    <artifactId>my-cocoon-webapp</artifactId>
    <version>1.0.1-SNAPSHOT</version>
    <packaging>war</packaging>
    <name>My Cocoon Webapp</name>

    <build>
        <finalName>my-cocoon-webapp</finalName>
    </build>

    <dependencies>
        <dependency>
            <groupId>org.apache.cocoon</groupId>
            <artifactId>cocoons-core</artifactId>
            <version>2.2.0-RC2-SNAPSHOT</version>
        </dependency>
        <dependency>
            <groupId>org.apache.cocoon</groupId>
            <artifactId>cocoons-template-impl</artifactId>
        </dependency>
    </dependencies>

```

- ➊ No more local.build.properties
- ➋ No more local.blocks.properties
- ➌ Use maven pom.xml file to select blocks
- ➍ Need to set a version for each block
- ➎ More typing but similar in nature

Assembling Cocoon

```
$ ls  
pom.xml  
  
$ export MAVEN_OPTS=-Xmx256m  
$ mvn war:war  
[INFO] BUILD SUCCESSFUL  
  
$ ls target  
my-cocoon-webapp/  
my-cocoon-webapp.war
```

- ⦿ Install latest maven 2
- ⦿ Start maven build process
- ⦿ Get coffee or beer (might take a while)
- ⦿ Gets faster on subsequent runs

Assembling Cocoon

```
$ find . -type f  
./pom.xml  
./src/main/webapp/WEB-INF/web.xml  
  
$ export MAVEN_OPTS=-Xmx256m  
$ mvn war:war  
[INFO] BUILD SUCCESSFUL  
  
$ ls target  
my-cocoon-webapp/  
my-cocoon-webapp.war
```

- ⦿ Install latest maven 2
- ⦿ Start maven build process
- ⦿ Get coffee or beer (might take a while)
- ⦿ Gets faster on subsequent runs

Updating Webapp

- ⌚ web.xml
- ⌚ applicationContext.xml
- ⌚ core.xml
- ⌚ cocoon.xconf
- ⌚ sitemap.xmap
- ⌚ WEB-INF/lib

Updating web.xml

```
<!--  
 - Web application configuration.  
-->  
<web-app xmlns="http://java.sun.com/xml/ns/j2ee"  
 version="2.4">
```

Servlet 2.4

Updating web.xml

```
<!--  
 - Declare a filter for multipart MIME handling  
-->  
<filter>  
 <description>  
   Multipart MIME handling filter for Cocoon  
 </description>  
 <display-name>  
   Cocoon multipart filter  
 </display-name>  
 <filter-name>CocoonMultipartFilter</filter-name>  
 <filter-class>  
 org.apache.cocoon.servlet.multipart.MultipartFilter  
 </filter-class>  
</filter>
```

- Servlet 2.4
- MIME Multipart
Servlet Filter

Updating web.xml

```
<!--  
 - Use the Cocoon multipart filter together with  
the Cocoon servlet  
-->  
<filter-mapping>  
 <filter-name>CocoonMultipartFilter</filter-name>  
 <servlet-name>Cocoon</servlet-name>  
</filter-mapping>
```

- ➊ Servlet 2.4
- ➋ MIME Multipart
Servlet Filter
- ➌ Filter Mapping

Updating web.xml

```
<!--  
 - Spring request listener ...  
-->  
<listener>  
 <listener-class>  
org.springframework.web.context.request.RequestConte  
xtListener  
 </listener-class>  
</listener>  
  
<!--  
 - Spring context listener ...  
-->  
<listener>  
 <listener-class>  
org.springframework.web.context.ContextLoaderListene  
r  
 </listener-class>
```

- ➊ Servlet 2.4
- ➋ MIME Multipart
Servlet Filter
- ➌ Filter Mapping
- ➍ Context
Listeners

Updating web.xml

```
<!--  
 - Declare Classic Cocoon servlet  
-->  
<servlet>  
 <servlet-name>Cocoon</servlet-name>  
 <display-name>Cocoon</display-name>  
 <description>Classic Cocoon Servlet</description>  
 <servlet-class>  
 org.apache.cocoon.servlet.SitemapServlet  
 </servlet-class>  
 <load-on-startup>5</load-on-startup>  
</servlet>
```

- ➊ Servlet 2.4
- ➋ MIME Multipart
Servlet Filter
- ➌ Filter Mapping
- ➍ Context
Listeners
- ➎ and Cocoon
Servlet!

Complete web.xml

```
<!--
 - Web application configuration.
-->
<web-app xmlns="http://java.sun.com/xml/ns/j2ee" version="2.4">

<!--
 - Declare a filter for multipart MIME handling
-->
<filter>
<description>Multipart MIME handling filter for Cocoon</description>
<display-name>Cocoon multipart filter</display-name>
<filter-name>CocoonMultipartFilter</filter-name>
<filter-class>org.apache.cocoon.servlet.multipart.MultipartFilter</filter-class>
</filter>

<!--
 - Declare a filter for debugging incoming request
-->
<filter>
<description>Log debug information about each request</description>
<display-name>Cocoon debug filter</display-name>
<filter-name>CocoonDebugFilter</filter-name>
<filter-class>org.apache.cocoon.servlet.DebugFilter</filter-class>
</filter>

<!--
 - Use the Cocoon multipart filter together with the Cocoon servlet
-->
<filter-mapping>
<filter-name>CocoonMultipartFilter</filter-name>
<servlet-name>Cocoon</servlet-name>
</filter-mapping>

<!--
 - Declare Spring request listener which sets up the required RequestAttributes
 - to support Springs and Cocoon custom bean scopes like the request scope or the
 - session scope.
-->
<listener>
<listener-class>org.springframework.web.context.request.RequestContextListener</listener-class>
</listener>

<!--
 - Declare Spring context listener which sets up the Spring
 - Application Context containing all application components.
-->
<listener>
<listener-class>org.springframework.web.context.ContextLoaderListener</listener-class>
</listener>

<!--
 - Declare Classic Cocoon servlet
-->
<servlet>
<servlet-name>Cocoon</servlet-name>
<display-name>Cocoon</display-name>
<description>Classic Cocoon Servlet</description>
< servlet-class>org.apache.cocoon.servlet.SitemapServlet</servlet-class>
<load-on-startup>5</load-on-startup>
</servlet>

<!--
 - Cocoon handles all the URL space assigned to the webapp using its sitemap.
 - It is recommended to leave it unchanged. Under some circumstances though
 - (like integration with proprietary webapps or servlets) you might have
 - to change this parameter.
-->
< servlet-mapping>
< servlet-name>Cocoon</servlet-name>
< url-pattern>/*</url-pattern>
</servlet-mapping>

<!--
 - Some servlet engines (Tomcat) have defaults which are not overridden
 - by '' mapping, but must be overridden explicitly.
-->
```

⌚ Debug filter

⌚ Servlet mappings

⌚ Session timeout

⌚ Mime types

⌚ ...

Adding applicationContext.xml

```
<!--  
 - Spring Application Context definition  
-->  
<beans xmlns="http://www.springframework.org/schema/beans"  
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
       xsi:schemaLocation="http://www.springframework.org/schema/  
beans http://www.springframework.org/schema/beans/spring-  
beans-2.0.xsd">  
  
<!--  
 - Load Cocoon  
-->  
<import resource="cocoon/spring/core.xml"/>  
  
</beans>
```

- ➊ Main Spring configuration file
- ➋ Just loads Cocoon's core.xml

Adding core.xml

```
<beans xmlns="http://www.springframework.org/schema/beans"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:util="http://www.springframework.org/schema/util"
    xmlns:configurator="http://cocoon.apache.org/schema/
configurator"
    xmlns:avalon="http://cocoon.apache.org/schema/avalon"
    xmlns:aop="http://www.springframework.org/schema/aop"
    xsi:schemaLocation="http://www.springframework.org/schema/
beans http://www.springframework.org/schema/beans/spring-
beans-2.0.xsd
        http://www.springframework.org/schema/util
http://www.springframework.org/schema/util/spring-util-2.0.xsd
        http://cocoon.apache.org/schema/configurator
http://cocoon.apache.org/schema/configurator/cocoon-
configurator-1.0.1.xsd
        http://cocoon.apache.org/schema/avalon
http://cocoon.apache.org/schema/avalon/cocoon-avalon-1.0.xsd
        http://www.springframework.org/schema/aop
http://www.springframework.org/schema/aop/spring-aop-2.0.xsd">
```

- ➊ It's all Greek!
And a lot of it!
- ➋ Get used to it -
this is a
replacement for
cocoon.xconf!
- ➌ So, what's in it?

What's in core.xml?

```
<!-- Activate Cocoon Spring Configurator -->
<configurator:settings readFromClasspath="false"/>

<!-- Configure Log4j -->
<bean name="org.apache.cocoon.spring.configurator.log4j"
      class="org.apache.cocoon.spring.configurator.log4j.Log4JConfigurator"
      scope="singleton">
    <property name="settings"
      ref="org.apache.cocoon.configuration.Settings"/>
    <property name="resource" value="/WEB-INF/cocoon/log4j.xml"/>
  </bean>

<!-- Activate Avalon Bridge -->
<avalon:bridge location="/WEB-INF/cocoon.xconf"/>
```

- ➊ Configurator loads Cocoon configuration and property files from file system
- ➋ Log4J Bean
- ➌ Avalon Bridge

What's in core.xml?

```
<!--+
 | Entity resolution catalogs
 +-->
<bean name="org.xml.sax.EntityResolver"
      class="org.apache.cocoon.core.xml.impl.DefaultEntityResolver"
      init-method="init" scope="singleton">
  <property name="verbosity" value="2"/>
</bean>

<!--+
 | DOM XML Parser
 +-->
<bean name="org.apache.cocoon.core.xml.DOMParser"
      class="org.apache.cocoon.core.xml.impl.JaxpDOMParser"
      scope="singleton">
  <property name="validate" value="false"/>
</bean>
```

Components

What's in core.xml?

```
<!--+
 | Entity resolution catalogs
 +-->
<entity-resolver logger="core.resolver">
  <parameter name="catalog" value="WEB-INF/entities/catalog"/>
  <parameter name="verbosity" value="1"/>
</entity-resolver>

<!--+
 | DOM XML Parser
 +-->
<xml-parser class="org.apache.excalibur.xml.impl.JaxpParser"
logger="core.xml-parser" pool-max="${xml-parser.pool-max}">
  <parameter name="validate" value="false"/>
</xml-parser>
```

- ⦿ Components
- ⦿ Deja Vu?

What's in core.xml?

```
<!--+
 | Entity resolution catalogs
 +-->
<bean name="org.xml.sax.EntityResolver"
      class="org.apache.cocoon.core.xml.impl.DefaultEntityResolver"
      init-method="init" scope="singleton">
  <property name="verbosity" value="2"/>
</bean>

<!--+
 | DOM XML Parser
 +-->
<bean name="org.apache.cocoon.core.xml.DOMParser"
      class="org.apache.cocoon.core.xml.impl.JaxpDOMParser"
      scope="singleton">
  <property name="validate" value="false"/>
</bean>
```

- ⦿ Components
- ⦿ Deja Vu?
- ⦿ Components are being migrated from Avalon to Spring – from cocoon.xconf into core.xml

Updating cocoon.xconf

```
<!--+
| Classic Cocoon configuration file
+-->
<cocoon version="2.2">
<!--+
| Include the core roles definitions.
+-->
<include src="resource://org/apache/cocoon/cocoon.roles"/>

<!--
- Allow other building blocks to supply components
- by dropping configuration file into the directory.
-->
<include dir="/WEB-INF/cocoon/avalon" pattern="*.xconf"/>

<!-- ===== Sitemap ===== -->
<sitemap check-reload="yes" file="context://sitemap.xmap"
logger="sitemap"/>
```

- Supports includes for modularity!
- Many components are moved to core.xml
- More will be moved in upcoming releases.

Updating sitemap.xmap

```
<map:sitemap xmlns:map="http://apache.org/cocoon/sitemap/1.0">
  <map:components>
    <map:generators default="file">
      <map:generator name="file"
        src="org.apache.cocoon.generation.FileGenerator"/>
      <map:generator name="jx"
        src="org.apache.cocoon.template.JXTemplateGenerator"/>
    </map:generators>

    <map:serializers default="html">
      <map:serializer name="html"
        src="org.apache.cocoon.serialization.HTMLSerializer"
        mime-type="text/html">
        <encoding>UTF-8</encoding>
        <doctype-public>-//W3C//DTD HTML 4.01 Transitional//EN</
        doctype-public>
        <doctype-system>http://www.w3.org/TR/html4/loose.dtd</
        doctype-system>
      </map:serializer>
    </map:serializers>
  </map:components>
</map:sitemap>
```

- ⦿ Cocoon 2.1 Sitemaps are completely supported.
- ⦿ No changes necessary!
- ⦿ Springified Generators, etc must be in core.xml (ex: POI)

About Blocks

```
$ unzip -l target/my-cocoon-webapp/WEB-INF/lib/cocoon-...
...
/META-INF/cocoon/spring/cocoon-some-block.xml
/META-INF/cocoon/avalon/cocoon-some-block.xconf
...
$ # Edit core.xml
<import resource="classpath://META-INF/cocoon/spring/cocoon-
poi.xml"/>
...
$ # Edit cocoon.xconf
<include src="resource://META-INF/cocoon/avalon/cocoon-
template.xconf"/>
...
$ # Or use config directory
```

- ⦿ Where are all xpatch files?
- ⦿ Spring XML files and Cocoon xconf files inside block JARs
- ⦿ Include them or copy, edit, paste
- ⦿ Or add config directory

About Forms Block

- ⦿ If you are using resources shipped with forms block
- ⦿ There are couple of references to “servlet:” protocol in resource files
- ⦿ Replace “servlet:” with path to resources

In `cocoon-forms-impl`:

```
/org/apache/cocoon/forms/resources/js/manifest.js  
/org/apache/cocoon/forms/resources/js/MultiValueEditorWithSuggestion.js
```

In `cocoon-forms-sample`:

```
/src/main/resources/COB-INF/resources/forms-samples-styling.xsl  
...
```

Config Directory

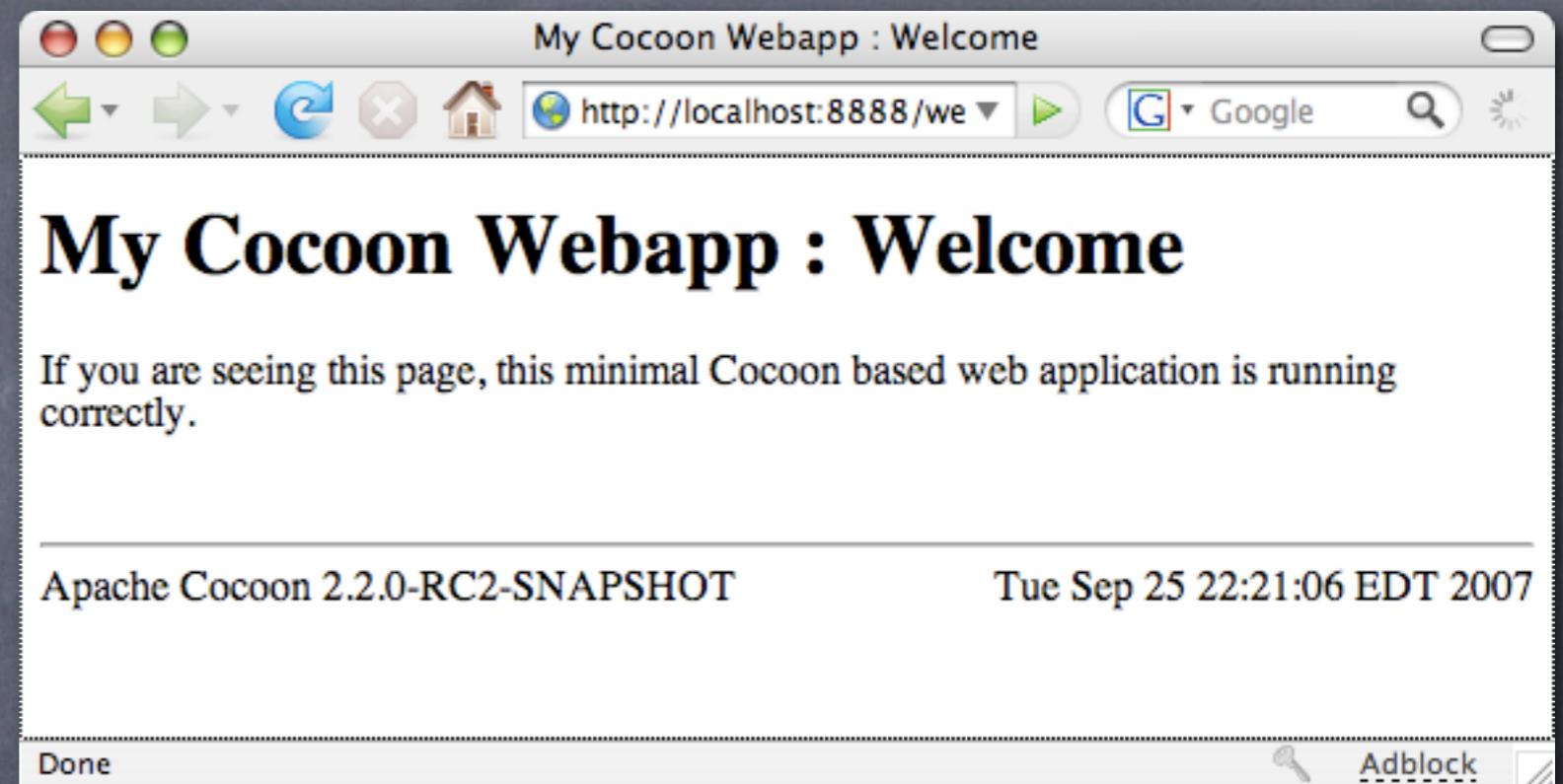
```
$ find . -type f  
./sitemap.xmap  
./config/avalon/custom.xconf  
./config/spring/custom.xml  
./config/properties/custom.properties  
...  
  
$ # Example: Deploy POI Spring components into sitemap  
$ unzip -p WEB-INF/lib/cocoon-poi-impl-1.0.0-SNAPSHOT.jar META-INF/cocoon/spring/cocoon-poi.xml > config/spring/cocoon-poi.xml
```

- ➊ Sitemap can have
 - ➋ Properties
 - ➋ Sitemap level components
 - ➋ Sitemap level Spring beans

Updating Webapp

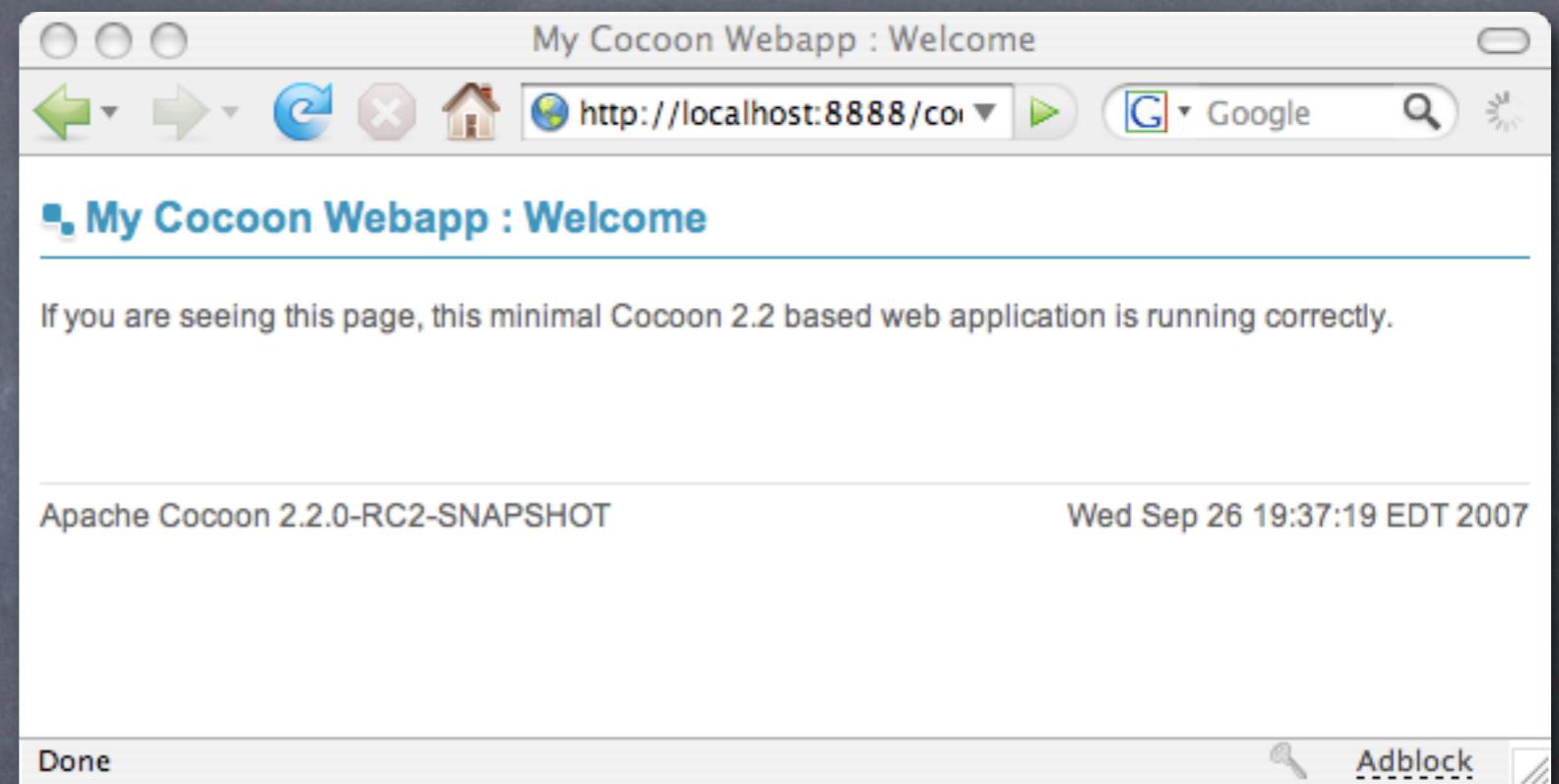
- ⌚ web.xml
- ⌚ applicationContext.xml
- ⌚ core.xml
- ⌚ cocoon.xconf
- ⌚ sitemap.xmap
- ⌚ WEB-INF/lib

Demo



```
mybook ~/Projects/ReveryCodes/MyCocoonWebapp $ ./run.sh
2007-09-24 23:58:36.326::INFO: Logging to STDERR via org.mortbay.log.StderrLog
2007-09-24 23:58:37.519::INFO: jetty-6.1.5
2007-09-24 23:58:37.905::INFO: Initializing Spring root WebApplicationContext
log4j:WARN No appenders could be found for logger (org.springframework.web.context.ContextLoader).
log4j:WARN Please initialize the log4j system properly.
2007-09-24 23:58:38.666::INFO: Apache Cocoon Spring Configurator v1.0.1-SNAPSHOT is running in mode 'prod'.
2007-09-24 23:58:39.785::INFO: WebApp@13222953 at http://0.0.0:8888/cocoon
2007-09-24 23:58:39.797::INFO: Started SelectChannelConnector@0.0.0:8888
```

Demo



```
mybook ~/Projects/ReveryCodes/MyCocoonWebapp $ ./run.sh
2007-09-24 23:58:36.326::INFO: Logging to STDERR via org.mortbay.log.StderrLog
2007-09-24 23:58:37.519::INFO: jetty-6.1.5
2007-09-24 23:58:37.905::INFO: Initializing Spring root WebApplicationContext
log4j:WARN No appenders could be found for logger (org.springframework.web.context.ContextLoader).
log4j:WARN Please initialize the log4j system properly.
2007-09-24 23:58:38.666::INFO: Apache Cocoon Spring Configurator v1.0.1-SNAPSHOT is running in mode 'prod'.
2007-09-24 23:58:39.785::INFO: WebApp@13222953 at http://0.0.0:8888/cocoon
2007-09-24 23:58:39.797::INFO: Started SelectChannelConnector@0.0.0:8888
```

Questions?

- ⦿ Ready to use application template is available at:
<http://reverycodes.com/gt/MyCocoonWebapp.zip>
- ⦿ Includes Jetty configuration, build and run scripts.