

# JSP Generator (2.1 legacy document)

## Table of contents

1 Comments.....	4
-----------------	---

## Table of contents

1 JSP Generator.....	3
2 JSPEngine.....	3
2.1 JSPEngineImpl.....	3
2.2 JSPEngineImplWLS.....	4
2.3 JSPEngineImplNamedDispatcherInclude.....	4

**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. JSP Generator

The JspGenerator selects a JSPEngine component. The JSPEngine component launches a JSP servlet engine of your servlet container, feeds the HttpRequest into the JSP servlet engine, and pipes the jsp response as SAX events into Cocoon. The JSP page is specified by the HttpRequest.

This way you can continue to use your JSP pages. Your migration from JSP to XSP may be done step by step. You may specify your JSP pages either as JSP scriptlets or as JSP-XML. But keep in mind that your JSP output should be valid XML.

- Name : jsp
- Class: org.apache.cocoon.generation.JspGenerator
- Cacheable: no

```
<map:generate type="jsp"/>
```

## 2. JSPEngine

As JSP servlet engines are implemented differently, you may have to select the appropriate JSPEngine component. The default is a JSPEngine working with Tomcat's JSP servlet engine Jasper. You may override the cocoon.roles by your own my.roles, as described in the [FAQs](#).

The JSPEngine component of Tomcat's JSPEngine is implemented in JSPEngineImpl. If you want to use another JSPEngine component, you may specify it in a my.roles file. The following sample specify in file WEB-INF/my.roles a JSPEngine working with WebLogicServer:

```
<?xml version="1.0"?> <role-list> <role
name="org.apache.cocoon.components.jsp.JSPEngine" shorthand="jsp-engine"
default-class="org.apache.cocoon.components.jsp.JSPEngineImplWLS"/> </role-list>
```

Defining the file my.roles this way you must ensure that your cocoon.xconf references my.roles, like that:

```
... <cocoon version="2.0" user-roles="WEB-INF/my.roles"> ...
```

Currently there are tree JSPEngine components available:

JSPEngine	ServletEngine
JSPEngineImpl	Tomcat, generic jsp servlet class
JSPEngineImplWLS	WebLogic 5.1, 6.0(?)
JSPEngineImplNamedDispatchInclude	Generic JSP Servlet

The next sections describe the settings of the JSPEngine components.

### 2.1. JSPEngineImpl

This JSPEngine is the default engine selected in cocoon.roles. By default it uses Tomcats' JASPER JSP servlet engine.

Running under a different JSP servlet engine, you can try to change the settings in cocoon.xconf, by modifying parameter name servlet-class to your needs.

```
<jsp-engine> <parameter name="servlet-class"
value="my.servlet.MyJspServletOfMyServletEngine"/> </jsp-engine>
```

JSPEngineImpl instances directly the JSP servlet engine class, and services HttpRequest to this instance.

JSPEngineImplNamedDispatcherInclude delegates the selection of a JSP servlet engine instance to the servlet engine. It selects by servlet-name, and not by servlet-class. This is the key differences of these two implementations.

## 2.2. JSPEngineImplWLS

This JSPEngine is implemented especially for WebLogic 5.1. WebLogic 6.0, and WebLogic 6.1 may work, too. JSPEngineImplWLS finds the named request dispatch for jsp, the jsp response is piped into Cocoon.

The name of the JSP servlet is by default set to '\*.jsp'. This is the default servlet name of the JSP servlet engine under WLS. You may adopt the parameter servlet-name to your needs.

If you want to specify a different JSP servlet name, you can change the settings in cocoon.xconf, by modifying the parameter servlet-name.

```
<jsp-engine> <parameter name="servlet-name"
value="MyNameOfMyJspServletOfMyServletEngine"/> </jsp-engine>
```

## 2.3. JSPEngineImplNamedDispatcherInclude

This JSPEngine is implemented like JSPEngineImplWLS without using any WebLogic classes. You may try to use this JSPEngine if JSPEngineImpl does not meet your requirements.

The name of the JSP servlet is by default set to '\*.jsp'. This is the default servlet name of the jsp servlet engine under WLS. You may adopt the parameter servlet-name to your needs.

If you want to specify a different JSP servlet name, you can change the settings in cocoon.xconf, by modifying the parameter servlet-name.

```
<jsp-engine> <parameter name="servlet-name"
value="MyNameOfMyJspServletOfMyServletEngine"/> </jsp-engine>
```

## 1. Comments

add your comments