

# Cocoon Plugin for Struts 1.1+

## 1. Introduction

This is a [Struts 1.1](#) plugin that integrates [Cocoon](#) into the Struts framework. Struts forwards are passed to Cocoon to be rendered in Cocoon XML pipelines. Struts is a familiar and proven application framework and Cocoon is a very powerful and flexible XML presentation technology. Rather than simply present the plugin, the Struts example web application has been slightly modified to show how easily Cocoon can be added to an existing Struts application.

## 2. Background

This plugin was created to attempt to provide a solution for those that need the familiar flow and structure of Struts but the presentation power of Cocoon. It is similar to [stxx](#), a project that replaces JSP's with XSLT in Struts, but not limited to simple transformation configurations. stxx's goal is to be a framework that makes 80% of the common uses for XML transformation really easy with a smooth migration path to the Cocoon plugin when the complexity requirements demand more functionality. Combining Struts and Cocoon provides a framework that has the familiarity and ease of Struts and allows the use the much improved JSP 2.0 and JSTL, but the separation of concerns, AOPishness, and features of Cocoon.

## 3. Features

### 3.1. Advantages

- **Familiarity and ease of use of Struts** - Struts has a number of books published of its use and is one of the most popular Java web application frameworks.
- **Complete separation of concerns** - Cocoon's XML pipelines can completely separate the data from its presentation and even separate layers within the presentation itself (for example theming).
- **Wide choice of presentation technologies** - Cocoon can support a large number of presentation technologies like JSP, Velocity, FreeMarker, PHP, and XSP.
- **Mimizes Cocoon complexity** - Much of Cocoon's complexity was due to the size and complexity of its sitemap, the file that describes the XML pipelines among other things. By letting Struts handle the actual request, the Struts forward can use any URI it likes.

Only a couple of Cocoon pipelines are needed to match the few types of URI's. For example, Struts could forward to "/simple/Foo.jsp" and Cocoon could match "simple/\*.jsp" so all URI's that start with "/simple" can be handled by one pipeline. Also, by letting Struts handle all the business logic, Cocoon can focus on what it is great at: XML publishing.

- **Cocoon is very configurable** - Lately, Cocoon has put a lot of effort into simplifying its core functionality and putting as much as possible into pluggable "blocks". The simple web app that wants to use JSP as the main presentation technology modified by XSLT's that create the theme needs very few jars (compared to Cocoon 2.0.x).
- **The Cocoon plugin can live side-by-side regular JSP forwards** - Not all requests have to be handled by Cocoon. If a cocoon pipeline cannot match the forward, it is handled normally by Struts.

### 3.2. Disadvantages

- **Slower** - Of course requests processed by Cocoon's XML pipeline will be slower than regular JSP processing, but this is minimized by Cocoon's caching pipelines and the ability to not use Cocoon for requests where performance is critical.
- **More dependencies** - Cocoon brings with it probably 2.5 megs of libraries (including itself), but it might be possible some of those are unnecessary.
- **Cannot be used side-by-side with the Tiles or Stxx plugin** - This is a limitation of Struts as it doesn't allow request processor chaining, however it is looking like it might be added with Struts 1.2.

## 4. What's New

### 4.1. 0.3 - June 11, 2004

- Updated to support Cocoon 2.1.4 (thanks Jeff Conrad)
- Removed a few unnecessary libraries and configuration files
- Cleaned up the example

### 4.2. 0.2 - July 6, 2003

- Updated Cocoon to 2.1M2 which contains a necessary path that allows requests that Cocoon doesn't handle to be processed by Struts normally
- Added plugin property "extension" to allow a specific path extension to be used to match requests that Cocoon should handle. Otherwise, all requests are handled by Cocoon but if not pipeline is found, they are then handled by Struts normally.
- Added more Javadocs and code cleanups
- JVM 1.4+ jars are included in the release. Please see the Cocoon distribution for the correct jars for previous versions of Java.

### 4.3. 0.1

- Released the plugin and modified Struts example web application.
- Required a patched version of Cocoon (bug [17612](#))

## 5. Usage Notes

To use the plugin, declare the plugin in `struts-config.xml`:

```
<plug-in className="org.twddata.struts.CocoonPlugin">
  <set-property property="configurations" value="/WEB-INF/cocoon.xconf" />
</plug-in>
```

Any `init-param` that would have been used to configure the Cocoon servlet can be declared as a plugin property. In addition, the plugin property "extension" will limit requests that are passed to Cocoon to only those that match the extension.

Finally, in the Cocoon pipelines, define pipelines that match Struts forwards. For example, a forward defined as:

```
<action path="/editRegistration"
        type="org.apache.struts.webapp.example.EditRegistrationAction"
        attribute="registrationForm"
        scope="request"
        validate="false">
  <forward name="success" path="/fancy/registration.jsp"/>
</action>
```

will be matched by a pipeline like this:

```
<map:match pattern="fancy/*.jsp">
  <map:generate type="jsp" src="{1}.jsp" />
  <map:transform src="fancy.xsl" />
  <map:serialize type="html" />
</map:match>
```

The Struts example application was modified to make use of the plugin. The only changes to the Struts example JSP and class files were to make sure it generated valid XML documents. Development took place on a box with JVM 1.4.0\_01, Struts 1.1 RC 1, and Cocoon 2.1M2. Note: there aren't any XML jars in the war. If you are using JDK 1.4, Cocoon requires more recent versions of Xalan and Xerces in your `$JAVA_HOME/jre/lib/endorsed`. See [Cocoon's installation page](#) for more information.

## 6. Struts Example Changes

- Added the `@xhtml="true"` attribute to the `html` element
- Corrected open elements like `<br>`

- Changed the Link\* classes to use & rather than & in URL's
- Added a CocoonPlugin plugin to struts-config.xml
- Modified one forward in struts-config.xml to use a different URI to better show how Cocoon handles them

As you can see, very little is needed to take an existing Struts app and use this plugin.

## **7. Contact**

Please contact [Don Brown](#) with comments, and suggestions.