Updating Cocoon (2.1 legacy document)

Table of contents

Table of contents

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. Known Issues for 2.1.5

This is a list of known issues for Cocoon 2.1.5.

1.1. Store

Cocoon 2.1.5 uses different implementation of the store based on the JCS. This new implementation currently does not allow storing of not serializable objects, and it does not write out memory cache to the disk on shutdown. If you depend on this functionality, you can keep 2.1.4 store configuration.

Cocoon 2.1.4 used MRUMemoryStore as store implementation and JISPStore as persistent store implementation. JISPStore is known to corrupt storage file under some conditions.

1.2. Logging Configuration

Cocoon 2.1.5 uses a newer version of the Avalon Excalibur logging package than version 2.1.4. Unfortunately this logging package has an imcompatibility to the version used in Cocoon 2.1.4.

If you update your application from 2.1.4 to 2.1.5 make sure that you have a log definition for the empty category in your logkit.xconf. Add the following lines to your category definition:

<category name="" log-level="@loglevel@"> <log-target id-ref="core"/> <log-target id-ref="error"/> </category>

2. Updating Cocoon

Please take your time to read this document completely before trying to upgrade from a Cocoon 2.0.x installation to 2.1 (or above). You can also read it if you want to know what was going on in the development of Cocoon.

The Cocoon team took great care in making this new version as compatible as possible. However, in order to achieve even more flexibility, usability and performance, the internal architecure of Cocoon has been improved. Due to these improvements it has not been possible to be compatible in every little detail. If you follow the instructions of document closely, however, you should be able to quickly upgrade your Cocoon 2.0.x installation.

The Cocoon team has developed many Avalon components that are not specific to Cocoon and therefore have been donated to the Avalon Excalibur project and moved out of Cocoon. This has led to some configuration changes which are also described in this document.

3. Sitemap

There are some changes in the sitemap and in the configuration of some components in the sitemap. In general we recommend you to start with a new sitemap from 2.1 and to adapt it to your needs. But for manual migration we will list as many changes as possible.

3.1. Pipelines configuration in the sitemap

The configuration of the pipelines has moved from cocoon.xconf to the sitemap. To update your installation, you have to remove the "event-pipeline" and "stream-pipeline" section from your

cocoon.xconf (see also the cocoon.xconf section) and add the map:pipes section to the map:components section of your sitemap. You can find the pipelines components definition in the sample main sitemap of Cocoon. Here is an example:

<map:sitemap> <map:components> ... <map:pipes default="caching"> <map:pipe name="caching" src="org.apache.cocoon.components.pipeline.impl.CachingProcessingPipeline"/> <map:pipe name="noncaching" src="org.apache.cocoon.components.pipeline.impl.NonCachingProcessingPipeline"/> </map:pipes> </map:components> ... </map:sitemap>

You can choose these different pipeline implementations in the map:pipeline section by specifying their type attribute:

<map:sitemap> ... <map:pipelines> <map:pipeline type="noncaching"> <map:match pattern="welcome"> ... </map:match> .. </map:pipeline> </map:pipelines> </map:sitemap>

This is similar to choosing the type of a generator or any other sitemap component. If the type attribute is omitted, the default configuration from the map:components section is used.

3.2. FOP Serializer

Relative paths in FOP serializer's <user-config> are now resolved relatively to the directory that contains the sitemap.

All Cocoon URIs are supported too.

3.3. Sitemap components

Some of the sitemap components have been removed from Cocoon sources, others were renamed. If you have the old declaration in your sitemap, you will get ClassNotFoundExceptions. Trial and error will probably be the fastest way for removing them and getting a clean and working sitemap. Hopefully you are not using one of the removed components. The following components are known to be removed or renamed:

- o.a.c.XTTransformer - use the TraxTransformer instead.
- o.a.c.webapps.authentication.selection.MediaSelector - the full qualified class name has changed to o.a.c.webapps.session.selection.MediaSelector.

3.4. Error handling

The map:handle-errors section must now be a complete pipeline. This means the old form

<map:handle-errors> <map:transform src="stylesheets/system/error2html.xsl"/> <map:serialize status-code="404"/> </map:handle-errors>

is no longer valid, because the generator is missing. Therefore you can now describe explicitely the error handling. The replacement of the above looks like the following:

<map:handle-errors> <map:generate type="notifying"/> <map:transform src="stylesheets/system/error2html.xsl"/> <map:serialize status-code="404"/> </map:handle-errors>

For a more detailed example have a look into the default sitemap delivered with Cocoon sources or read the documentation on error handling.

4. Namespace changes

In order to have consistent namespaces, some transformers and generators (listed below) use new namespaces. If you use any of these components, you will need to use the new namespaces.

4.1. Request Generator

RequestGenerator changed its namespace from http://xml.apache.org/cocoon/requestgenerator/2.0 to http://apache.org/cocoon/request/2.0.

4.2. I18nTransformer

The I18nTransformer supports both http://apache.org/cocoon/i18n/2.0 and http://apache.org/cocoon/i18n/2.1 namespace.

5. Changes in logging interfaces require recompilation

Due to some interface changes in the Cocoon logging components, custom java components (generators, transformers or actions for example) compiled for Cocoon 2.0.x will not run under Cocoon 2.1 unless recompiled.

It's also advisable to change your implementations from using *Loggable* to *LogEnabled* when it comes to logging in your components.

6. Components

The Cocoon architecture has changed significantly. However, great care has been taken to preserve backwards compatibility.

6.1. Cocoon Configuration (cocoon.xconf)

In order to reflect the new version, the version information in the *cocoon.xconf* has changed from *2.0* to *2.1*.

To update *cocoon.xconf*, we recommend that you start with the new cocoon.xconf from V2.1 and incorporate your changes in it, instead of trying to migrate your old configuration file.

The SAXConnectors have been removed, so if you upgrade manually you have to remove the *sax-connectors* configuration from *cocoon.xconf*.

6.2. Source Resolver

Every sitemap component gets a SourceResolver from the sitemap processor. For all other components that need access to a SourceResolver, the SourceResolver is now an Avalon component that can be accessed using *cocoon.manager.lookup(SourceResolver.ROLE).*. The package name of the component is *org.apache.excalibur.source*.

6.3. XSLT Processor

There are some issues related to JDK 1.4.

6.3.1. XML/XSLT with JDK 1.4

Another serious issue is the presence of the Xalan and Xerces package in the JDK 1.4. For general

information on this please read the Xalan FAQ and our own EndorsedLibsProblem wiki page.

Basically, you have to update your libraries in the endorsed dirs of the JDK or the servlet containers with every new version of Xalan and Xerces delivered with Cocoon. Strange errors can occur if you have different versions of these packages in the classpath (independent of those in the JDK).

6.4. XML Parser

The XML parser component has been moved to Excalibur. In cocoon.xconf, the hint name has therefore changed from *parser* to *xml-parser*. The configuration has not changed, so changing the hint names is sufficent.

Java code should not use *org.apache.cocoon.components.parser.Parser.ROLE* anymore; use *org.apache.excalibur.xml.sax.SAXParser.ROLE* instead.

6.5. XML Entity Resolver

Similarly, the XML entity resolver component has been moved to Excalibur. In cocoon.xconf the hint name has therefore changed from *resolver* to *entity-resolver*. The configuration has not changed, so changing the hint names is sufficent.

Java code should not use *org.apache.cocoon.components.resolver.Resolver.ROLE* anymore; use *org.apache.excalibur.xml.EntityResolver.ROLE* instead.

The default entities (DTDs, entity sets, etc.) have moved to the WEB-INF/ directory.

6.6. Caching

Although the basic caching mechanism is still the same (each sitemap component in the pipeline is queried), the interface for a component have been improved as well.

The old interface *Cacheable* is deprecated in favour of the new *CacheableProcessingComponent* interface. The basic behaviour of this interface has been preserved, however the method names and the signatures have changed a little bit.

Some other interfaces, like the validity of the cached information has moved to the source packacke in Avalon Excalibur.

The old interfaces are still support but deprecated, so it's advisable to update your components. However, you can support both interfaces at the same time, making your component runable in old and new Cocoon installations at the same time.

6.7. Stores

The Store and StoreJanitor components and implementations have been moved to Avalon Excalibur.

To make upgrading easier, the class attributes of the store janitor component have been removed in *cocoon.xconf* as the class names have changed. The *cache-transient* and *cache-persistent* components do not exist anymore, so any reference to them must be removed from cocoon.xconf. Use the *persistent-store* and *transient-store* components instead.

In general the package names changed from *org.apache.cocoon.components.store* to *org.apache.excalibur.store* (and *org.apache.excalibur.store.impl*). So if you have custom java code using these components, you have to change your imports.

The roles *PERSISTENT_CACHE* and *TRANSIENT_CACHE* have been renamed to *PERSISTENT_STORE* and *TRANSIENT_STORE*. The hold() method has been removed from the Store interface.

6.8. SAXConnectors, Stream and Event Pipeline

This is the only real incompatible change (But don't panic, this will not affect you, or maybe just a little bit..).

The internal architecture of Cocoon has changed: previously, the processing pipeline - consisting of a generator, the transformers and a serializer - was represented by two components, called *stream* and *event pipeline*.

For a simpler architecture, enhanced functionality and improved performance, these components have been combined into one: the *processing pipeline*. The *SAXConnectors*, which were rarely used, have been removed to avoid overcomponentization.

6.9. File Upload

The class name for file upload has changed from *org.apache.cocoon.components.request.multipart.FilePart* to *org.apache.cocoon.servlet.multipart.Part*, and the *getFilePath()* has been renamed *Part.getUploadName()*.

6.10. RequestLifeCycleComponent

If you are using the marker interface *RequestLifeCycleComponent* for your own components, you have to make sure that your implementations still implement the *Component* interface. The *RequestLifeCycleComponent* does no longer extend the *Component* interface, so you have to declare it in your own components together with *RequestLifeCycleComponent*. Otherwise you will get a *ClassCastException* as soon as you access your component.

7. Components from the scratchpad

Cocoon 2.0.x had some components in the scratchpad area that have now moved into the main trunk as blocks. With this move some things have changed.

7.1. Session, Authentication and Portal

The session framework (sunShine), the authentication framework (sunRise) and the portal framework (sunSpot) are now blocks (session-fw, authentication-fw and portal-fw).

The *sunShine transformer* has been renamed to *session transformer*. All sitemap components starting with *sunrise-* have been changed to start now with *auth-*. The *sunrise-auth* action has been renamed to *auth-protect*.

The transformer namespace has changed from *http://cocoon.apache.org/sunshine/1.0* to *http://apache.org/cocoon/session/1.0* and the context names have changed from *sunshine* to *session* and from *sunrise* to *authentication*.

1. Comments

add your comments