

ReadWriteToSVN

Reading and writing to SVN using SSL.

This is how to set up your system so that you can read content directly from a Subversion repository, retrieving the latest version and also write to the repository, updating the current version, using a Cocoon Source.

We needed to take the following steps with Cocoon 2.1.7-dev to make all of this work (not tested in the 2.1.6 release).

Install the CA cert in the JVM

If you are using a self-signed certificate, you first need to install your Certificate Authority certificate into the JVM of the machine running Cocoon.

For this sample, I used the same .pem file that was installed in /etc/ssl/cacert.pem in our installation. Your certificate may have a different name and location.

```
$sudo keytool -import -trustcacerts -alias "Our CA" -file cacert.pem \  
-keystore $JAVA_HOME/lib/security/cacerts -storepass *****
```

NB. depending on your system, your keystores may be at a different location. This works with MacOSX.

NB. Theoretically, we only needed to perform this step because we were using a self-signed certificate.

More information about self-signed certificates can be found [here](#)

Turn on auto versioning in httpd

Auto versioning needs to be turned on if you want to use a Cocoon Source to write changes back to the repository, for example, if you were implementing an online editor.

Add the following instruction to the appropriate place in your Apache httpd.conf file:

```
SVNAutoversioning on
```

Revert Slide Libs

There are currently bugs in the Slide WebDAV jar that are distributed with Cocoon (as of 7th Feb 2005).

We needed to replace WEB-INF/lib/jakarta-slide-webdavlib-2.1.jar with WEB-INF/lib/jakarta-slide-webdavlib-2.1M1.jar.

Define webdavs:// source

You will be using an extension of the webdav:// pseudo protocol, webdavs://, for accessing the secured Subversion repository, this needs to be configured in cocoon.xconf.

Add the following to your cocoon.xconf, a good place is right after the existing definition of the webdav:// source.

```
<component-instance class="org.apache.cocoon.components.source.impl.WebDAVSourceFactory" name="webdavs">  
  <parameter name="secure" value="true"/>  
</component-instance>
```

Use the webdavs source

Now you are in a position to access content from the Subversion repository.

To generate content from a resource in the repository, we used a pipeline like this:

```
<map:generate src="webdavs://username:password@svn.mydomain.org/myrepository/myfile.xml" />
```

We could also generate a collection listing from the repository, using one of the traversable generators like this:

```
<map:generate src="webdav://username:password@svn.mydomain.org/myrepository/" type="traversable">
  <map:parameter name="depth" value="1" />
</map:generate>
```

The exact parameters depend on which traversable generator you are using.

We were able to write updated content to the repository by using typical techniques in Flow Script, resolving the source, streaming a DOM to it, the same that you would use to write to any modifiable source.

Many thanks for all of those people who gave us the advice we needed to get all of this working !!!