OpenOffice Documents with Lenya

Initial version

by Christian Egli

NOTICE:

This document describes the integration of Openoffice with Lenya CMS

1. Goals

This document describes the integration of Openoffice with Lenya CMS. The integration is guided by the following goals:

- Use OpenOffice as a content editor for static web pages
- Migrate OpenOffice document to a custom xml format

2. Prerequisites

In order to seamlessly integrate Openoffice into the publication process of Lenya/Cocoon the following prerequisites need to be met:

2.1. OpenOffice DTD

The DTDs for the OpenOffice documents has to be available on the system.

It's best to get them directly from your OpenOffice installation. They are located in the share directory of your installation. Copy the dtd's into your Lenya installation, e.g. as follows:

```
cp ~/Office/share/dtd/* ~/build/jakarta-tomcat-4.1.18-LE-jdk14/webapps/lenya/lenya/reso
```

FIXME (ce):

The DTDs should probably go into /usr/share/sgml/openoffice/*

Copyright © 2002-2003 The Apache Software Foundation. All rights reserved.

Note:

There's a bug in the xml parser. As a workaround we uncomment all the `draw:text-box` stuff.

2.2. XML Catalog

In order for Lenya/Cocoon to find the DTDs you need to setup an XML catalog as follows:

```
xmlcatalog --noout --create openoffice.cat
xmlcatalog --noout --add "public" \
  "-//OpenOffice.org//DTD OfficeDocument 1.0//EN"
  "file:///home/slide/build/jakarta-tomcat-4.1.18-LE-jdk14/webapps/lenya/lenya/resource
  openoffice.cat
```

Alternatively you can simply use the attached catalog.

Store this newly created catalog and edit CatalogManager.properties to make sure Cocoon finds this catalog and hence the OpenOffice DTDs.

Add the location of the OpenOffice catalog to Cocoon's CatalogManager.properties (which can be found in `~/build/jakarta-tomcat-4.1.18-LE-jdk14/webapps/lenya/WEB-INF/classes/Cata` by adding the following lines to this file:

```
#catalogs=/path/to/local/catalog
catalogs=/home/slide/build/jakarta-tomcat-4.1.18-LE-jdk14/webapps/lenya/lenya/resources
```

2.3. OpenOffice2HTML XSTL

In order to render the OpenOffice xml as html we need XSLT stylesheets to provide the necessary transformations.

A very good XSLT which is fairly complete can be fetched from zope.org (http://www.zope.org/Members/philikon/ZooDocument).

2.4. Slide

Slide is an Apache project which offers amongst other things a a WebDAV access module (implemented as a servlet). This will allow us to deploy the OpenOffice documents directly via WebDAV.

For a very basic installation the following changes need to be applied to a file named Domain.xml in the Slide webapp directory:

Copyright © 2002-2003 The Apache Software Foundation. All rights reserved.

- Change permissions
- ContentStore: set to parent dir of OpenOffice dir
- Replace folder "files" by OpenOffice dir name

The following patch will apply all changes you need:

```
diff -u Domain.xml.orig Domain.xml
--- Domain.xml.orig Thu Nov  1 15:47:52 2001
+++ Domain.xml         Thu Mar 20 16:44:09 2003
@@ -44,7 +44,7 @@
           <reference store="nodestore" />
         </revisiondescriptorstore>
         <contentstore classname="slidestore.reference.FileContentStore">
-          <parameter name="rootpath">contentstore</parameter>
+          <parameter name="rootpath">/home/slide/build/jakarta-tomcat-4.1.18-LE-jdk14/
           <parameter name="version">false</parameter>
           <parameter name="resetBeforeStarting">true</parameter>
         </contentstore>
@@ -136,7 +136,7 @@
       <!-- Paths configuration -->
       <userspath>/users</userspath>
       <guestpath>guest</guestpath>
-      <filespath>/files</filespath>
+      <filespath>/openoffice</filespath>
       <parameter name="dav">true</parameter>
       <parameter name="standalone">true</parameter>

@@ -245,13 +245,12 @@

         </objectnode>

-        <objectnode classname="org.apache.slide.structure.SubjectNode"
-         uri="/files">
+        <objectnode classname="org.apache.slide.structure.SubjectNode" uri="/openoffic

           <!-- ### Give read/write/manage permission to guest ###
                Uncomment the following line to give permission to do
                all actions on /files to guest (unauthenticated users) -->
-          <!-- <permission action="/actions" subject="/users/guest"/> -->
+          <permission action="/actions" subject="/users/guest"/>

           <permission action="/actions/manage" subject="/users/john"/>
           <permission action="/actions/write" subject="+/users/groupA"/>
```

3. Pipelines

In order for Lenya/Cocoon to be able to read the content of the OpenOffice document, a set of pipelines need to be set up.

Copyright © 2002-2003 The Apache Software Foundation. All rights reserved.

3.1. Read the zip/jar file

To read the OpenOffice documents we need to setup a simple reader which as follows:

```
<map:match pattern="**.sxw">
  <map:read src="content/{1}.sxw"/>
</map:match>
```

3.2. Unpack zip file and transform the OO xml to xhtml

OpenOffice documents are actually a zip file containing xml files for content and style plus other additional files such as jpg etc.

Zip is the same file format as jar. JDK supports jar unpacking natively with the jar protocol. The pipeline to read a jar file looks as follows:

```
<map:match pattern="**.oo">
  <map:generate src="jar:http://localhost:38080/lenya/computerworld/authoring/{1}.sxw!/
  <map:transform src="../../xslt/openoffice/ooo2html.xsl"/>
  <map:serialize/>
</map:match>
```

3.3. Aggregate with navigation

Additionaly we want to embed the OpenOffice document in the usual navigation, header and footer. The following is fairly specific to the Computerworld publication but can easily be adapted:

```
<map:match pattern="**.html">
  <map:aggregate element="lenya">
    <map:part src="cocoon:/menus/static/{1}.html"/>
    <map:part element="cmsbody" src="content/authoring/wrapper.html"/>
    <map:part src="cocoon:/{1}.oo" element="wrapper"/>
    <map:part src="content/authoring/small-preview.xml"/>
    <map:part src="content/authoring/sitetree.xml"/>
    <map:part src="cocoon:/today"/>
  </map:aggregate>

  <map:transform src="xslt/authoring/wrapper.xsl">
    <map:parameter name="id" value="/{1}"/>
    <map:parameter name="authoring" value="true"/>
  </map:transform>
  <map:transform src="xslt/authoring/images.xsl"/>
  <map:serialize type="html"/>
</map:match>
```


Copyright © 2002-2003 The Apache Software Foundation. All rights reserved.

4. Problems

- Caching prevents an update OO file (zip file) from being displayed.
- If you restart tomcat (slide) you lose the NodeContentStore so that WebDAV loses the nodes (documents and folders).
- xml parser cannot handle openoffice dtd's due to a parser bug

5. To do's

- Set permissions in tomcat/slide: authorization and autorisation
- Complete and improve OpenOffice2Html xslt (images, tables, etc.)
- Add pipelines for other files in zip like images
- Integration slide and lenya

© 200 wyona.org

Copyright © 2002-2003 The Apache Software Foundation. All rights reserved.