

WebDAVAuthenticMetadata

First, for each XML file within the "repo" folder create an accompanying file with a ".meta" extension like the "step4" example is doing:

```
<?xml version="1.0" encoding="UTF-8"?>
<metapage>
  <author>me</author>
  <category>catb</category>
  <state>edited</state>
</metapage>
```

Add 2 additional match nodes to your pipeline:

```
<map:match pattern="page/**">
  <map:aggregate element="aggregation" label="content">
    <map:part src="cococon:/raw/{1}"/>
    <map:part src="cococon:/meta/{1}"/>
  </map:aggregate>
  <map:serialize type="xml"/>
</map:match>
<map:match pattern="meta/**">
  <map:generate src="{global:staging}repo/{1}.meta"/>
  <map:serialize type="xml"/>
</map:match>
```

Now create 2 additional files within the "authentic" folder, aggregation.xsd (the XML schema)

```

<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">
<xs:element name="aggregation">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="page" minOccurs="1" maxOccurs="1"/>
      <xs:element ref="metapage" minOccurs="1" maxOccurs="1"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="page">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="title"/>
      <xs:element ref="content"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="content">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="para" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="title" type="xs:string"/>
<xs:element name="para">
  <xs:complexType mixed="true">
    <xs:sequence minOccurs="0" maxOccurs="unbounded">
      <xs:element ref="emphasis" minOccurs="0" maxOccurs="unbounded"/>
      <xs:element ref="image" minOccurs="0" maxOccurs="unbounded"/>
      <xs:element ref="link" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="emphasis">
  <xs:complexType mixed="true"/>
</xs:element>
<xs:element name="image">
  <xs:complexType>
    <xs:attribute name="href" type="xs:string" use="required"/>
    <xs:attribute name="height" type="xs:long"/>
    <xs:attribute name="width" type="xs:long"/>
  </xs:complexType>
</xs:element>
<xs:element name="link">
  <xs:complexType mixed="true">
    <xs:attribute name="href" type="xs:string" use="required"/>
  </xs:complexType>
</xs:element>
<xs:element name="metapage">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="author" minOccurs="1" maxOccurs="1"/>
      <xs:element ref="category" minOccurs="1" maxOccurs="1"/>
      <xs:element ref="state" minOccurs="1" maxOccurs="1"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="author" type="xs:string"/>
<xs:element name="category" type="xs:string"/>
<xs:element name="state" type="xs:string"/>
</xs:schema>

```

and aggregation.sps (the authentic-specific stylesheet)

```

<?xml version="1.0" encoding="UTF-8"?>
<structure version="2" schemafile="aggregation.xsd" workingxmlfile="" templatexmlfile="">

```

```

<ns:prefix="xs" uri="http://www.w3.org/2001/XMLSchema"/>
<template>
  <match overridden="xs:match"="/"/>
  <children>
    <template>
      <match match="aggregation"/>
      <children>
        <template>
          <match match="metapage"/>
          <children>
            <xpath allchildren="1"/>
          </children>
        </template>
      </children>
    </template>
    <newline/>
    <template>
      <match match="page"/>
      <children>
        <template>
          <match match="title"/>
          <children>
            <xpath allchildren="1">
              <styles color="#0000A0" font-size="larger" font-weight="bold" text-decoration="underline"/>
            </xpath>
          </children>
        </template>
        <template>
          <match match="content"/>
          <children>
            <template>
              <match match="para"/>
              <children>
                <paragraph paragraphtag="p">
                  <children>
                    <xpath allchildren="1"/>
                  </children>
                </paragraph>
              </children>
            </template>
          </children>
        </template>
      </children>
    </template>
  </children>
</template>
<template>
  <match match="author"/>
  <children>
    <text fixtext="Author: "/>
    <field ownvalue="1">
      <properties size="19" value=""/>
    </field>
    <newline/>
  </children>
</template>
<template>
  <match match="category"/>
  <children>
    <text fixtext="Category: "/>
    <select ownvalue="1">
      <properties size="0"/>
      <selectoption description="Cat A" value="cata"/>
      <selectoption description="Cat B" value="catb"/>
      <selectoption description="Cat C" value="catc"/>
    </select>
    <newline/>
  </children>
</template>

```

```

</template>
<template>
  <match match="emphasis"/>
  <children>
    <xpath allchildren="1">
      <styles font-weight="bold"/>
    </xpath>
  </children>
</template>
<template>
  <match match="image"/>
  <children>
    <image>
      <properties border="0"/>
      <imagesource>
        <xpath value="@href"/>
      </imagesource>
    </image>
  </children>
</template>
<template>
  <match match="link"/>
  <children>
    <link>
      <hyperlink>
        <xpath value="@href"/>
      </hyperlink>
      <children>
        <xpath allchildren="1"/>
      </children>
    </link>
  </children>
</template>
<template>
  <match match="metapage"/>
  <children>
    <newline/>
    <template>
      <match match="author"/>
      <children>
        <xpath allchildren="1"/>
      </children>
    </template>
    <newline/>
    <template>
      <match match="category"/>
      <children>
        <xpath allchildren="1"/>
      </children>
    </template>
    <newline/>
    <template>
      <match match="state"/>
      <children>
        <xpath allchildren="1"/>
      </children>
    </template>
    <newline/>
  </children>
</template>
<template>
  <match match="page"/>
  <children>
    <xpath allchildren="1"/>
  </children>
</template>
<template>
  <match match="state"/>
  <children>
    <text fixtext="State: "/>
    <select ownvalue="1">

```

```

        <properties size="0"/>
        <selectoption description="New" value="new"/>
        <selectoption description="Edited" value="edited"/>
        <selectoption description="Reviewed" value="reviewed"/>
        <selectoption description="Released" value="released"/>
    </select>
    <newline/>
    <newline/>
</children>
</template>
</structure>

```

Now change your "file2html.xml" stylesheet to reference these files instead of "content.xsd" and "content.sps".

```

<?xml version="1.0"?>
<xsl:stylesheet version="1.0"
    xmlns:xsl="http://www.w3.org/1999/XSL/Transform">

<xsl:param name="file"></xsl:param>
<xsl:param name="sitemapURI"></xsl:param>
<xsl:param name="requestURI"></xsl:param>
<xsl:param name="serverName"></xsl:param>
<xsl:param name="serverPort"></xsl:param>

<xsl:template match="/page">
    <html>
        <body>
            <OBJECT id="objPlugIn"
                Classid="clsid:46987108-BA64-4fd1-A947-1BF7DA938FC0" width="700" height="500">
                <xsl:attribute name="CodeBase">http://<xsl:value-of select="$serverName"/>:<xsl:value-of
                    select="$serverPort"/><xsl:value-of select="substring-before($requestURI, $sitemapURI)"/>authentic
                    /AuthenticBrowserEdition.cab</xsl:attribute>
                <PARAM NAME="LicKey" VALUE="<YOUR LICENSE KEY>"/>
                <PARAM NAME="LicServer">
                    <xsl:attribute name="VALUE">http://<xsl:value-of select="$serverName"/></xsl:attribute>
                </PARAM>
                <PARAM NAME="XMLDataURL">
                    <xsl:attribute name="VALUE">http://<xsl:value-of select="$serverName"/>:<xsl:value-of
                        select="$serverPort"/><xsl:value-of select="substring-before($requestURI, $sitemapURI)"/>raw/<xsl:value-of
                        select="$file"/></xsl:attribute>
                </PARAM>
                <PARAM NAME="XMLDataSaveURL">
                    <xsl:attribute name="VALUE">http://<xsl:value-of select="$serverName"/>:<xsl:value-of
                        select="$serverPort"/><xsl:value-of select="substring-before($requestURI, $sitemapURI)"/>write/<xsl:value-of
                        select="$file"/></xsl:attribute>
                </PARAM>
                <PARAM NAME="SPSDataURL">
                    <xsl:attribute name="VALUE">http://<xsl:value-of select="$serverName"/>:<xsl:value-of
                        select="$serverPort"/><xsl:value-of select="substring-before($requestURI, $sitemapURI)"/>authentic/aggregation.
                        sps</xsl:attribute>
                </PARAM>
                <PARAM NAME="SchemaDataURL">
                    <xsl:attribute name="VALUE">http://<xsl:value-of select="$serverName"/>:<xsl:value-of
                        select="$serverPort"/><xsl:value-of select="substring-before($requestURI, $sitemapURI)"/>authentic/aggregation.
                        xsd</xsl:attribute>
                </PARAM>
                <PARAM NAME="EntryHelpersEnabled" VALUE="TRUE"/>
                <PARAM NAME="EntryHelperWindows" VALUE="3"/>
            </OBJECT>
            <br/>
            <input type="button" name="submit1" value="Save" onClick="objPlugIn.SavePOST()"/>
        </body>
    </html>
</xsl:template>

</xsl:stylesheet>

```

Now you already can edit your meta data together with your XML document. Adjust your "stream2write.xml" stylesheet to separate the data and metadata down the pipeline again:

```

<?xml version="1.0"?>
<xsl:stylesheet version="1.0"
    xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
    xmlns:req="http://apache.org/cocoon/request/2.0"
    xmlns:source="http://apache.org/cocoon/source/1.0">

<xsl:param name="file"></xsl:param>

<xsl:template match="/aggregation">
<page>
    <xsl:apply-templates/>
</page>
</xsl:template>

<xsl:template match="page">
    <source:write create="true">
        <source:source><xsl:value-of select="$file"/></source:source>
        <source:fragment>
            <xsl:copy-of select="."/>
        </source:fragment>
    </source:write>
</xsl:template>

<xsl:template match="metapage">
    <source:write create="true">
        <source:source><xsl:value-of select="$file"/>.meta</source:source>
        <source:fragment>
            <xsl:copy-of select="."/>
        </source:fragment>
    </source:write>
</xsl:template>

</xsl:stylesheet>

```

Adjust your sitemap to exclude your ".meta" files from folder traversal like the "step4" sample is doing.