

DebuggingTips

Introduction to Xmap debugging

This page is the result of a paragraph write by [AndreasHartmann](#) and a discussion on the Lenya ML (subject : [Debugging] see the stream anywhere in a sitemap).

With Lenya/Cocoon XMAP, debugging a complex process with multiple pipelines in multiple XMAPs can be a challenge.

But... Don't worry ! Here you will see 3 different techniques for debugging your Xmap.

1) Cut the stream and see it

When first creating an XMAP, serialize after each generator, aggregator, and transformer and save the XML from the browser to a file. Knowing the results of each process greatly helps with writing the next stage.

You may often discover namespaces in element names that need to be either dropped in the current process or handled by the next process. The saved output provides reference for later development. [1]

TODO : add an example with explanation of matcher and url address that we have to type.

2) Don't cut the stream and see it

After an XMAP is completed, debugging is more difficult. The final results are often combined from several XMAPs. Logging intermediate results to a file is a very good solution. [1]

1 – Create your own datetime

Datetime is create to prevent overwrites with multiple threads and to provide an history. A disadvantage is that many files are created and must be deleted manually.

A datetime is create because Lenya's default date format is "yyyy-M-dd HH:mm:ss Z" and most operating systems do not allow colons in filenames.

To configure a new datetime :

Open the build\lenya\webapp\WEB-INF\cocoon.xconf Inside the <input-modules> element, add:

```
<component-instance class="org.apache.cocoon.components.modules.input.DateInputModule" logger="core.modules.  
input" name="filedatetime">  
  <format>yyyy-MM-d-HH-mm-ss</format>  
</component-instance>
```

Restart your Lenya app.

2 – Create the log to file resource :

This resource groups some transformations that allow you to write the stream.

Copy-paste this in your module's sitemap for a local usage, or in the /lenya/sitemap.xmap for a global usage :

```
<map:resources>  
  <map:resource name="log-to-file">  
    <map:transform src="fallback://lenya/xslt/authoring/edit/addSourceTags.xsl">  
      <map:parameter name="source" value="context://lenya/pubs/debug/{fileprefix}-{filedatetime:currentDate}."  
xml"/>  
    </map:transform>  
    <map:transform type="write-source">  
      <map:parameter name="serializer" value="xml"/>  
    </map:transform>  
    <map:transform src="fallback://lenya/xslt/authoring/edit/removeSourceTags.xsl"/>  
  </map:resource>  
</map:resources>
```

The value in <map:parameter name="source"> can be configure as you want.

3 – Do your debug :

In your pipeline just adapt the value of fileprefix and write this :

```
<!-- DEV DEBUG 1 BEGIN -->
<map:call resource="log-to-file">
  <map:parameter name="fileprefix" value="AfterTransform1"/>
</map:call>
<!-- DEV DEBUG 1 END -->
```

3) Log SAX events as XML

For log this you can use the tee transformer :

```
<map:transform type="tee" src="context://WEB-INF/logs/tee/{date:HH_mm_ss_SSS}_{1}_{2}_{3}.xml"/>
```

Thanks to

- Solprovider [1]
- [AndreasHartmann](#)
- Thorsten Scherler