

Morpho - Feature #6557

Add rendered metadata as pdf file in Morpho export

05/21/2014 06:52 PM - Matt Jones

Status:	Resolved	Start date:	05/21/2014
Priority:	Normal	Due date:	
Assignee:	Lauren Walker	% Done:	0%
Category:		Estimated time:	0.00 hour
Target version:	1.10.3	Spent time:	0.00 hour
Bugzilla-Id:			

Description

Morpho currently exports metadata both as xml and html. Users have requested the rendered metadata also be provided in PDF format. Add stylesheets for rendering as PDF and include this in the export file.

The stylesheets for generating PDF can probably be general enough to be included in the EML package for many to use, and then simply imported and used within Morpho.

See related issue [#6053](#) in Metacat for delivering Bagit packages. Ideally, Morpho's export would produce a Bagit compatible zip file equivalent to what one gets from Metacat.

History

#1 - 05/21/2014 07:10 PM - ben leinfelder

A couple comments/questions:

Using Apache FOP would be nice since we are going from XML->PDF. There's a ton in the FO spec for laying out the document format and the sky is kind of the limit on how we want it to look.

Do we want it to be exactly like the existing HTML metadata outputs?

Should it be skinnable with the ability to add header graphics, change fonts, etc?

#2 - 05/22/2014 03:16 PM - ben leinfelder

- File *metadata.pdf* added

- Status changed from *New* to *In Progress*

Looked into FOP, but would take a lot of coding.

Looked at HTML -> PDF options and there is "flying saucer" that uses iText. I tried it out and it's very promising. We do need to edit the existing EML XSLTs to make better "printable" HTML before we convert to PDF, but this is more tractable than writing FO XSLTs from scratch.

To clean up our nasty non-XHTML: <http://jtidy.sourceforge.net/howto.html>

To generate the PDF from the tidy XHTML: <https://today.java.net/pub/a/today/2007/06/26/generating-pdfs-with-flying-saucer-and-itext.html>

#3 - 05/22/2014 03:17 PM - ben leinfelder

Super simple code to do the transformation:

```
public void export(String inputFile, String outputFile) throws IOException, DocumentException {
    OutputStream os = new FileOutputStream(outputFile);

    String tidyFile = inputFile + ".tidy";
    OutputStream tidyOut = new FileOutputStream(tidyFile);

    Tidy tidy = new Tidy();
    tidy.setXHTML(true);
    tidy.parse(new FileInputStream(inputFile), tidyOut);

    String url = new File(tidyFile).toURI().toURL().toString();

    ITextRenderer renderer = new ITextRenderer();
    renderer.setDocument(url);
    renderer.layout();
    renderer.createPDF(os);
    os.close();
}
```

}

#4 - 05/23/2014 08:29 AM - ben leinfelder

I've added a class and corresponding test to the EML project that renders a sample EML file as both HTML and PDF using the default CSS. 'ant runonetest' will allow you to run it (HtmlToPdfTest is the default class to run). The output will be in build/tests/eml-sample.xml.html and build/tests/eml-sample.xml.pdf

#5 - 05/23/2014 08:33 AM - ben leinfelder

- Project changed from Morpho to EML
- Assignee changed from ben leinfelder to Lauren Walker

Hoping Lauren can do a bit of work on the layout to make it narrow enough to fit on a page.

#6 - 05/27/2014 03:34 PM - Lauren Walker

- Status changed from In Progress to Resolved

I styled the EML -> HTML output a bit to make it more modern and simple, and made sure that it converts to a PDF without running off the page.

#7 - 05/27/2014 03:44 PM - Lauren Walker

- File eml-sample.pdf added

Attached is a test EML->HTML->PDF that was generated using the ant runonetest HtmlToPdfTest

#8 - 11/12/2014 09:25 AM - ben leinfelder

- Project changed from EML to Morpho
- Target version set to 1.10.3

Moving to Morpho release for feature tracking even though it is implemented in utilities project.

Files

metadata.pdf	12.3 KB	05/22/2014	ben leinfelder
eml-sample.pdf	14 KB	05/27/2014	Lauren Walker