

# Digital Preservation at Oxford and Cambridge

A collaborative research project to evaluate and provide sustainable recommendations for our digital preservation programmes

## Customizable JHOVE TIFF output handler anyone?

Posted on [25 May, 2017](#) by [James Mooney](#)

*Technical Fellow, James, talks about the challenges with putting JHOVE's full XML output into a reporting tool and how he found a work around. We would love feedback about how you use JHOVE's TIFF output. What workarounds have you tried to extract the data for use in reporting tools and what do you think about having a customizable TIFF output handler for JHOVE?*

As mentioned in my last blog post, I've been looking to validate a reasonably large collection of TIFF master image files from a digitization project. On a side note from that, I would like to talk about the output from JHOVE's TIFF module.



The JHOVE TIFF module allows you to specify an output handler as either Text, a XML audit, or a full XML output format.

Text provides a straight forward line by line breakdown of the various characteristics and properties of each TIFF processed. But not being a structured document means that processing the output when many files are characterized is not ideal.

The XML audit output provides a very minimal XML document which will simply report if the TIFF files were valid and well formed or not; this is great to a quick check, but lacks some other metadata properties that I was looking for.

The full XML output provides the same information that was provided in text output format, but with the advantage of being a structural document. However, I've found some of the additional metadata structuring in the full XML rather cumbersome to process with further reporting tools.

As result, I've been struggling a bit to extract all of the properties I would like from the full XML output into a reporting tool. I then started to wonder about having a more customizable output handler which would simply report the the properties I required in a neat and easier to parse XML format.

I had looked at using an XSLT transformation on the XML output but, as mentioned, I found it rather complicated to extract some of the metadata property values I wanted due to the excessive nesting of these and the property naming structure. I think I need to brush up on my XSLT skills perhaps?

In the short term, I've converted the XML output to a CSV file, using a little freeware program called XML2CSV from A7Soft. Using the tool, I selected the various fields required (filename, last modified date, size, compression scheme, status, TIFF version, image width & height, etc) for my reporting. Then, the conversion program extracted the selected values, which provided a far simpler and smaller document to process in the reporting tool.

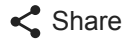
I would be interested to know what others have done when confronted with the XML output and wonder if there is any mileage in a more customizable output handler for the TIFF module...

**Update 31st May 2017**

Thanks to Ross Spencer, Martin Hoppenheit and others from Twitter. I've now created a basic [JHOVE XML to CSV XSLT stylesheet](#). Draft version on my [GitHub](#) should anyone want to do something similar.

---

#### SHARE THIS:



This entry was posted in [technology](#), [tools](#) and tagged [jhove](#), [output handler](#), [tiff](#), [validation](#), [xml](#), [xslt](#) by [James Mooney](#). Bookmark the [permalink](#) [<http://www.dpoc.ac.uk/2017/05/25/customizable-jhove-tiff-output-handler-anyone/>].

5 THOUGHTS ON “CUSTOMIZABLE JHOVE TIFF OUTPUT HANDLER ANYONE?”

Liz

on [1 October, 2018 at 22:41](#) said:

I wanted to include the MD5 checksum. I am sure there is a better way to do this, but I tried to add it the xslt, although I have to admit, I have no idea if I used the syntax correctly. Anyways, I forked stylesheet and added one line that would maybe display the MD5 checksum?  
<https://github.com/archivist-liz/miscellaneous-scripts> Thanks for sharing in any case, it's given me ideas, even if I might not be able to realize them yet.

James Mooney

on [1 June, 2017 at 09:58](#) said:

Nice to hear it's not just me Yvonne!

I'm now piping the JHOVE XML output into a XSLT transform which then generates a CSV format line per file.

Let me know how the CSV output handler discussions go, I would be happy to contribute into the project in anyway I can.

Yvonne Tunnat  
on **1 June, 2017 at 09:49** said:

Dear James,

I have the same issues since years. The JHOVE output is just cumbersome.

2 or 3 years ago I have started to use the JHOVE java library to have my own xml & xslt-output with the metadata I needed (usually validation findings and error messages + used module only).

As I am not a developer, the performance of my tool is not as good as JHOVE, so I have written a small java tool to collect from the JHOVE output what I need. This still is pretty cumbersome, as I have to use two steps instead of just one (like with my own java tool which uses the JHOVE library), but it is more robust this way.

2 days ago I learned how to transform the JHOVE xml in Excel. Of course there are too many columns and I had to delete a lot, but at least it looks good and I did not need to write & develop sth. on my own.

I think it would be great if JHOVE would have a csv-output on top to the possibilities it has now to work with. I will suggest that and maybe even put some work in it myself, as my institution is an OPF-member.

Best, Yvonne

James Mooney  
on **30 May, 2017 at 15:58** said:

Thanks Martin I will take a look at that.

I've been writing up a XSLT transformation today, having reviewed the FITS XSLT used for the JHOVE module as mentioned by the twitter comments. Once I have something will put it into GitHub as it maybe useful for others.

**Martin Hoppenheit**

on **30 May, 2017 at 13:24** said:

I often use xmlstarlet, a CLI XML processor, to extract the interesting properties on the fly in a Unix Pipeline. Of course, this is only feasible when I'm interested in no more than two or three properties of the JHOVE output. Plus, using xmlstarlet in this way again requires some XSLT proficiency (not much, but you need to know the concepts). For example, to extract a list of file names (uri attribute of replInfo element) and validation status (status child element):

```
jhove -h xml image.tif | xmlstarlet sel -N  
'j=http://hul.harvard.edu/ois/xml/ns/jhove' -t -m '//j:replInfo' -v  
'@uri' -o ',' -v 'j:status' -n
```

It looks a bit weird, but it all makes sense when you read the -t, -m and -v options like XSLT's template, match and value-of constructs (plus -o for literal text output and -n for a newline).

This site uses Akismet to reduce spam. [Learn how your comment data is processed.](#)