Ensuring long-term access: PDF validation with JHOVE?

JHOVE is an open source tool for identifying, characterizing and validating common formats such as pdf, tiff, jpeg, aiff and wave[1]. JHOVE includes validation modules for twelve different file formats, including PDF.

The PDF format is widely used by memory institutions and is one of the most commonly used long term archiving formats. However, because of the wide variety of potential source formats, conversion tools and PDF readers in use, a large percentage of PDF files fail to actually meet the PDF standard[2].

Naturally, many memory institutions use JHOVE's PDF module on a daily basis for digital long term archiving. I wish to discuss the extent to which JHOVE's PDF validation tools can be used for risk management and quality assurance when seeking to assure long term access to documents. Here, a "valid PDF" means that the PDF meets all criteria of the specification – that it is correct, in other words. "Invalid", on the other hand, means that the file fails to meet at least one point of the specification and is therefore incorrect.

Incorrectly classified as invalid: Page Tree

During the OPF (Open Preservation Foundation[3])'s PDF Hackathon[4] in Hamburg, together with the ZBW (the German National Library of Economics) and Goportis (the Leibniz Library Network for Research Information), the PDF Association's Olaf Drummer reported an error message from JHOVE which incorrectly identifies a breach of the PDF specification.

Pages within a PDF file are usually stored as a page tree, allowing the user to reach a given page as quickly as possible[5]. This is often represented as a balanced page tree[6]. Although the PDF standard references this option, it does not by any means prescribe it.

Pages can alternatively be stored as a simple array of pages, while still meeting the PDF standard. The only drawback is that this makes it less efficient to access a given page (making for slower navigation through the document), particularly when dealing with a PDF containing a particularly large number of pages. JHOVE, however, reports an error if pages are stored in a field (array) instead of a page tree. As this is not an error and does not present a risk to long-term archiving, the message can be ignored.

A digression: PDF/A validation with JHOVE?

Common advice for long-term archiving is to preferentially use the PDF/A format. However, this no longer matches to the day-to-day reality of many workflows which use JHOVE for validation tests.

JHOVE's PDF module is certainly capable of validating PDF/A files. According to JHOVE's developer, however, this feature was implemented late, is unstable, and does not work well[7]. Additionally, the

---

[1] URL: http://jhove.sourceforge.net/ (last accessed 12th November 2014)
[2] In the case of our document server, EconStor, the rate is around 20%. Other memory institutions have reported rates of up to 90%.
[3] URL: http://openpreservation.org/ (last accessed 21st November 2014)
[4] URL: http://openplanetsfoundation.org/blogs/2014-09-03-my-first-hackathon-hacking-pdf-files (last accessed 12th November 2014)
[5] Developing with PDF: Dive Into The Portable Document Format by Leonard Rosenthol, page 24
[6] URL: https://en.wikipedia.org/wiki/Self-balancing_binary_search_tree (last accessed 11th December 2014)

process does not analyze the content of the data streams, meaning that it cannot validate PDF/A compliance in line with ISO standards[8].

The PDF/A format differs from the standard PDF format in that it explicitly forbids or requires certain options considered essential for long-term archiving. A PDF/A-1b[9] file forbids the following:

- **Embedded audio or video files** (external players required)
- **LZW compression** (for patent protection reasons. Applies only to PDF/A-1; the patent has since expired and LZW compression has been available to PDF/A files since PDF/A-2.)
- **Encryption**
- **Transparency, JavaScript, executable files, functions (executable commands) and external links**

A PDF/A file also requires the following:

- **All data included in full**. This includes embedding all fonts and images used within the document, as well as metadata stored in XMP format.
- **Unambiguous reproduction** (e.g. font types and colors. For this reason, **layers** are also forbidden)

Additionally, PDF/A-1 is based on **PDF 1.4**. All functions introduced from PDF 1.5 onwards, including JPEG2000 compression and layering, are lost when converting to PDF/A-1 format. PDF/A-2, in contrast, is based on PDF 1.7.

The Isartor Test Suite[10] provided by the PDF Association consists of 204 invalid PDF/A files, only one of which was recognized by JHOVE as invalid. JHOVE failed to identify 51 of these as PDF/A files at all. As a result, JHOVE only checked to see whether these files met the PDF 1.4 standard[11].

These tests are more than sufficient to prove that JHOVE is not suited to PDF/A validation. Since JHOVE is a component in several out-of-the-box long term archiving solutions, however, it is still interesting to know what JHOVE does deliver in terms of PDF/A, to allow well informed decision making.

JHOVE's PDF module checks to see whether a file is a PDF/A, in which case it uses the PDF/A profile subcategory[12]. In this case, JHOVE tests whether a file identified as a PDF/A contains:

- Annotation types: movie, sound and file-attachments[13]

---

[7] URL: http://www.garymcgath.com/jhovenote.html(last accessed 12th November 2014)
[8] URL: Does JHOVE validate PDF/A files? http://anjackson.github.io/keeping-codes/experiments/does-jhove-validate-pdfa-files.html (last accessed 12th November 2014)
[9] JHOVE only validates against the PDF/A-1 standard; PDF/A-2 and -3 have not been implemented. This is because JHOVE has not been updated since PDF 1.6 (JHOVE for Developers, Gary McGath, page 17) and PDF/A-2 is based on PDF 1.7. JHOVE can validate against PDF/A-1a, but in my experience, this format is seldom if ever used due to the extensive post-processing required. I will therefore not discuss it here.

[10] URL: http://www.pdfa.org/2011/08/isartor-test-suite/ (last accessed 12th November 2014)
[11] URL: http://anjackson.github.io/keeping-codes/experiments/does-jhove-validate-pdfa-files.html (last accessed 12th November 2014)
[12] JHOVE for Developers, Gary McGath (2012), page 14

- Action Types: e.g. JavaScript and links[14]
- LZW encoding[15]
- Encryption[16]

If so, the file is identified as invalid. JHOVE also checks whether the file contains:

- Metadata in a readable format[17]
- Sufficient font information within the PDF dictionary[18]

If not, then the PDF/A file will again be recognized as invalid.

A quick test of 670 invalid PDF/A files as identified by PDF Box[19] shows that JHOVE identifies only five files as not well formed (4) or invalid (1). These are all breaches of the standard PDF specification, however: in these cases, JHOVE failed to identify the PDF/A files correctly and therefore only tested their validity as standard PDFs.

Modern conversion tools make it difficult to integrate components into a PDF/A file which JHOVE's PDF module would test and identify as errors. In fact, I am not aware of any currently available PDF/A conversion tool which would even permit it. For example, the protection applied to encrypted PDF files means that either they cannot be converted to PDF/A at all, or the encryption is identified and removed by the tool during the conversion process. The same applies for other components which are forbidden in PDF/A and checked by JHOVE.

JHOVE's PDF/A validation tools are therefore so rudimentary that I am unsure if any PDF/A files at all would break its rules and be detected by JHOVE as invalid.

Conclusion

JHOVE is not suited to PDF/A validation. I know of no alternative to JHOVE for validating standard PDFs. As many memory institutions primarily use the PDF format and the quality of their files is not always enough of an argument for converting them to PDF/A, I believe that a standard PDF validator remains as necessary as it always has been. In general, JHOVE will continue to be used, despite its limitations, and decisions regarding the archivability of a given file will be dependent on the results JHOVE gives.

---

[13] URL: https://github.com/gmcgath/jhove/blob/master/src/main/java/edu/harvard/hul/ois/jhove/module/pdf/AProfile.java (last accessed 12th November 2014) Mavenized JHOVE Source Code on Github; Line 43 &46 in source code

[14] Mavenized JHOVE Source Code on Github; Line 64 in source code

[15] Mavenized JHOVE Source Code on Github; Line 69 in source code

[16] Mavenized JHOVE Source Code on Github; Line 128, 160 & 164 in source code

[17] Line 198 in source code

[18] Is mentioned in several functions in the source code; some only affect PDF/A-1a conformance, like Unicode, but others are mandatory for PDF/A-1b as well

[19] URL: https://pdfbox.apache.org/ (last accessed 12th November 2014)

JHOVE can still be useful, provided users understand its error reports[20] and are aware of ways to resolve them. So far there is not a great deal of documentation on this issue. Both nestor (AG Format Recognition[21]) and the Open Preservation Foundation[22] aim to do their part to improve this situation soon.

Some JHOVE errors, such as "Invallid PDF trailer", have proven to be a very useful part of day to day work. PDF files which trigger this error can usually not be opened. This is because the file has not been fully up- or downloaded, resulting in an incomplete file. The ability to automatically recognize such serious errors and report them to the data provider for correction is very useful when working with large archives. Other errors affecting a PDF's structure can also be identified using JHOVE and easily fixed.

Even before taking into account the meaning and the effects of a JHOVE error report – to say nothing of the resolution options – JHOVE remains an excellent option for providing initial guidance. Since JHOVE developer Gary McGath himself warns against using JHOVE as the final word on the matter, we will continue to avoid making our decisions and workflows dependent on JHOVE alone.

[20] I have personally encountered 52 different JHOVE errors to date.
[21] URL: https://wiki.dnb.de/display/NESTOR/AG+Formaterkennung (last accessed 13th November 2014)
[22] URL: http://www.openplanetsfoundation.org/ (last accessed 13th November 2014)