Achieving Canonical PDF Validation

Duff Johnson
PDF Association
Neue Kantstrasse 14
14057 Berlin, Germany
+1 617 283 4226
duff.johnson@pdfa.org

ABSTRACT

While PDF is the best currently available option for archiving fixed-form electronic documents, low quality PDF files remain problematic throughout the document lifecycle, and can pose substantial challenges for memory institutions.

This paper proposes a model for realizing and promulgating PDF validation based on a canonical (i.e, accepted industry-wide as definitive) approach rather than focusing on preservation per se.

General Terms

strategic environment, preservation strategies and workflows, specialist content types, digital preservation marketplace

Keywords

PDF, PDF/A, software, validation, standard, canonical, adoption

1. INTRODUCTION

The Portable Document Format (PDF) was invented by Adobe Systems and first released with Adobe's Acrobat software in 1993. The value proposition was simple: reliability when shared. PDF has largely delivered on that promise - but not entirely.

21 years later PDF is an ISO standardized format. For electronic documents, PDF is an exemplar *de facto* standard as well [2].

This paper proposes development of a canonical (accepted industry-wide as definitive) validation model encompassing all PDF features and thus enforceable across the document lifecycle.

2. PDF RISES

For printing technology vendors PDF's popularity began to take off with the November 1996 release of PDF 1.2, but marketplace uptake was slower. The early PDF specification was too flexible; reliability was hard to guarantee. Workflows suffered when users encountered formally "valid" files too difficult (or impossible) to process [12]. The problems were serious and widely felt.

The industry's successful response was PDF/X, a subset of PDF designed to ensure reliable exchange in prepress workflows. PDF/X became the first ISO standard for PDF technology [17].

As PDF became popular for printing formal documents, the use of PDF for distribution and retention of electronic documents became commonplace in every functional area within business and government organizations, and as part of website content.

2.1 PDF/A (archive) and ISO standardization

PDF/X was not a general-purpose standard. Responding to industry and governmental requests for PDF files suitable for long-term retention, industry stakeholders and trade groups began development of a PDF subset for archival-grade electronic documents. In the PDF context, "archival-grade" means embedded fonts, no external dependencies and prohibition of certain functionality such as encryption and JavaScript. ISO 19005 (PDF/A-1) was published in 2005 and has been adopted by the US National Archives and Records Administration (NARA) [16], and by governments and businesses worldwide [18].

Other ISO-standardized subsets of PDF have followed: PDF/E for engineering, PDF/UA for accessibility and PDF/VT for variable data and transactional printing.

In the mid 2000s Adobe Systems realized that turning over PDF to the ISO was the right move to drive continued adoption of the format. While the PDF specification was freely downloadable and PDF viewer software is traditionally free, the fact that PDF was proprietary inhibited governments, engineering concerns and other preservation-minded institutions from comfortably standardizing their own publishing, accounting, enterprise content management (ECM) or line of business (LOB) systems on PDF.

With thousands of implementers and worldwide acceptance, PDF had become "too big to own". In the spirit of the company's original - and commercially brilliant - move to publish PDF's specification for free, Adobe offered PDF to the ISO for open, democratic management by a committee of volunteer experts. PDF 1.7 was thus standardized as ISO 32000-1 in 2008 [17].

2.2 A *de facto* standard for electronic paper

From the end user perspective PDF serves as electronic paper. Self-contained, reliable, flexible and resolution-independent, PDF is easy to make from any electronic source, and freely viewable on any platform. Emulating many key characteristics of paper has helped make PDF the most popular format worldwide for downloadable electronic documents [5] [17]. The format's nearly universal adoption makes it typical in common use-cases:

- Print or distribute finalized documents – "Post the PDF"
- Retain, share or manage draft documents - "PDF it"
- Annotation of 3rd party content – "Add a note to the PDF"
- Capture content from arbitrary source - "Scan to PDF"
- Collate from arbitrary sources – "Insert / replace PDF pages"
- End-user data capture – "Fill the PDF form"

Worldwide implementation and use of PDF technology shows no sign of abating; searches for PDF files continue to increase over time, and in contrast to other formats [7]. On the public internet, institutions communicating on formal terms tend to be heavy users of PDF [8], while privately held transactional and other documents in PDF are estimated to be in the billions [20].

2.3 Beyond the printable page

While PDF is fundamentally a page description model for text, vector graphics and bitmap images, the technology includes many features that distinguish it from image formats such as TIFF and JPEG. Increased utilization of document-oriented features such as forms, annotations, XMP metadata, digital signatures, encryption, 3D, geospatial, video, embedded files, tagging and other advanced capabilities represents a growing challenge for the preservation community – a challenge that existing tools and workflows do not address in a cost-effective manner. Meanwhile, the volume of content that meets retention criteria is exploding [14].

3. CHALLENGE AND OPPORTUNITY

Although end users have enthusiastically adopted PDF the digital preservation community is more circumspect. Although research libraries prefer PDF/A to formats such as HTML or RTF they rate PDF itself as only slightly preferable to HTML [19].

In addition to concerns over PDF's complexity and reliability, some features that help make PDF compelling to end users complicate efforts to ensure electronic content remains accessible in the future. As a result, although PDF is generally extremely reliable and accepted in the marketplace, archivists have hesitated in trusting PDF as a long-term storage format [1].

An opportunity exists to harmonize industry's interest in promoting investment in PDF technologies with archivists' interest in reliable files, low-cost ingestion and maximum longevity. In the next section this paper provides an overview of the historical and technical reasons for archivists' concerns before moving on to discuss solutions as seen through an industry lens.

4. THE PROBLEMS

Compared to HTML PDF is a very complex file format. It includes 11 syntaxes, at least 20 native and 3rd party binary formats, 10 stream filters, 2 encryption algorithms, and more. Beyond PDF's rich imaging model the format includes interactive forms, encryption, digital signatures, annotations, embedded files, accessibility features and more [11].

The challenges PDF technology presents to archivists may be organized into five categories:

1. **Complexity.** Compared to plain text or TIFF, PDF is technically complex.
2. **PDF has changed.** While remaining backwards compatible, the PDF specification has changed (it has become more detailed and rigorous, as well as richer) over time. Even so:
    a. Old and "flaky" PDF files exist.
    b. Old software is still making flaky PDF files.
    c. Good files can be damaged by old or bad software.
3. **Varying degrees of support.** Few implementers claim to support all the functionality defined in PDF, which is fine. However, many implementers do not fully address the features they do claim to support.
4. **Fonts.** ISO 32000 does not require embedded fonts, so it is possible to inadvertently create unreliable PDF files.
5. **No canonical model for validation.** Today, developers must rely on their own tools or open-source applications lacking broad industry acceptance such as JHOVE to identify potential problems. Unfortunately, it is often difficult to determine with certainty whether or not a problem even exists. What should a digital preservation professional do if a PDF fails JHOVE, but passes Adobe's Preflight? Adobe Reader is not useful as a validator precisely because it is designed to accommodate very poor quality PDF files.

5. THE SOLUTIONS

The technical problems are significant, but the scope and scale of any given software development project may be the least of the barriers to addressing archivists' concerns about PDF.

Billions upon billions of PDF files already populate the world's desktops, shared-drives, ECM systems, SharePoint servers and websites. Obsolete software cannot be willed out of existence. Enforcement of policies, from embedded fonts to embedded files, will not occur spontaneously.

Let's review notionally and practically plausible responses in each problem category, looking for common threads.

5.1 Problem 1: PDF is technically complex

This problem is fundamentally ineradicable, since any other self-contained file format – even one designed only for rendering – would have to be similarly complex, at least in contrast to bitmap or ASCII-based formats that cannot replace PDF's functionality.

Solution: Developer education, ideally, via tools that deliver canonical information, analysis and advice about input PDF files.

5.2 Problem 2: PDF has changed over time

PDF was born as little more than a page description model, but it evolved through contact with the marketplace. Today's PDF (ISO 32000-1) has far more features compared to PDF 1.0, including support for rich content, transparency, new font types, support for color-management, accessibility features, and much more.

Solution: A facility that promotes retirement of old software and drives adoption of common practices in handling PDF features that developers choose not to support.

5.3 Problem 3: PDF is feature-rich, but not all vendors want to be

Many PDF features are optional. For example, relatively few vendors as yet support digital signature or 3D features in PDF.

When a vendor chooses to support a given feature, it should do so as fully and correctly as the specification requires, and do no harm (whenever possible) to unsupported features. It should warn the user if harm is unavoidable. Today, however, some software fails to warn the user that it will destroy a part of their document!

Solution: A practical and potent means of promoting best practice in creating and processing PDF files. This solution is essentially the same as that identified in section 5.2.

5.4 Problem 4: Fonts need not be embedded for conformance with the specification

Unembedded fonts (permitted but usually inadvisable in PDF) are perhaps the single largest source of unrecoverable problems users encounter. Even in 2014, font problems are not unusual [11]. Although most modern software embeds font subsets by default, font programs remain some of PDF's most complex substructures. Mangled font encoding or a missing ToUnicode entry, for example, is not uncommon.

Solution: Recovering PDF files with missing or damaged font information (among other fatal errors) is sometimes possible. When it is not, providing definitive information about the error and supporting free, high-quality, interactive font substitution would mitigate support costs and enhance vendors' relationships with end-users and digital preservation professionals alike.

5.5 Problem 5: No model for validation

The PDF specification lacks a concept of validity. Neither PDF 1.4 nor ISO 32000 offers much guidance for getting it right, so "does it work in Adobe's Reader" became the fundamental real-world test for non-Adobe software developers (and Adobe's as well, for that matter).

In addition, PDF has a variety of subset specifications. It can be difficult to be sure which specification a file should be validated against, and how. For example, PDF/UA-1 requires the Scope attribute for standard structure type <TH>, but Scope was defined in PDF 1.5. Can a PDF/UA-1 file conform to PDF/A-1a, which is based on PDF 1.4? How do we get a ruling on that question?

It is possible to validate for PDF/A-1b conformance. The specifications for PDF's archival subset standard require specific resources and prohibit certain features. Even so, PDF/A is not obvious in certain cases, and itself relies on the PDF specification. The PDF Association's 2008 Isartor Test Suite [4], was a collaborative effort to resolve many of these problems for PDF/A-1b. Since publication, Isartor has garnered substantial acceptance well beyond the original participating vendors.

Solution: A canonical model for PDF validation would provide a framework for solving all the solvable problems related to PDF reliability and utility in both business and archival contexts. Archivists are aware of this possibility [15]. How do we get there?

6. CANONICAL PDF

The PDF Association has begun studying a concept tentatively named *VeraPDF* [9]. In the next section this paper discusses the concept, and what it could mean for digital preservationists.

6.1 If validators disagree, do they exist?

In the early days of PDF/A collisions between validators were not uncommon [14], which opened fundamental questions about their value. Ensuing customer disappointment prompted development of the Isartor Test Suite, which helped smooth disagreements between different software packages and enabled PDF/A's undeniable success in the marketplace.

Although it is possible that Isartor could be, in general terms, a model for validation of ISO 32000, the prospect is daunting. Isartor would be hard to scale. In itself it does little to promote implementation, and the Terms of Use prohibit using it to certify software products. It is not a solution for canonical validation.

Intended from the outset to serve as a canonical reference implementation, VeraPDF would address the need directly.

6.2 Why "canonical" matters

As mentioned in the introduction, "canonical" validation means a definitive (accepted industry-wide) understanding of compliance with the specification. Knowing that a given feature is implemented in a canonically valid manner it becomes possible to precisely assess the degree of accuracy and completeness with which a given piece of software creates or processes the feature.

In order to simplify matters for those presently concerned only with accurate rendering, for example, it might be argued that conformance with the formal specification is less important than attaining some relative, needs-specific measure of acceptability.

Such an approach, however, offers an unstable, unreliable target. A file may be acceptable in one viewer or when processed through one tool, but not acceptable in another, often as a function of features employed on specific files. This is not a recipe for reliable high-volume processing or long-term preservation.

The problem is especially acute when considering PDF features beyond basic rendering of text and graphics objects. For example, Apple's Preview may in most or all cases render PDF page content as accurately as Adobe's Reader, but as of April 2014, Preview ignores PDF/A, digital signatures and tagged PDF, and even destroys these features when saving a file [5].

A canonical approach sets clear performance expectations. In this context, even when they choose not to fully process a given feature, developers have concrete, impartial guidance at-hand. They are more likely to handle real-world PDF files in a consistent fashion. Open source and industry-accepted file-format validation is how we get there.

6.3 The PDF Reference, in action

VeraPDF would be an open source generic PDF parser similar to EpubCheck [3]. VeraPDF would process the entirety of PDF-defined structures and utilize extension mechanisms to facilitate processing of objects defined elsewhere: font programs, images, JavaScript and other features PDF files may include.

Architecture is always critical, but especially for a purpose-built, future-proofed validator. Ideally, VeraPDF would facilitate modules implemented in both Java and C++ environments and in various programming languages or using 3rd party protocols, and integrate unit-testing resources.

Error handling would allow processing deep into poorly constructed PDF files. Programmatically accessible and localizable reporting for developers would be complemented by industry-accepted "plain language" messages for end-users.

It is important to emphasize that generating useful results from real-world files is not a trivial task because PDF includes such a rich set of features and PDF files may be broken is so many creative ways. It will take an industry effort, but canonical validation offers substantial value to software developers from accelerated software development and reduced support costs.

VeraPDF libraries would be deployable from creation to curation across the entire document lifecycle. VeraPDF could operate as a service or integrate into PDF creation and processing applications including the ingest components of digital repository software.

Beyond establishing the parser's scope and framework the likely initial implementation objective would be validation of classical cross-reference tables, integrating selected grammars such as Adobe's Dictionary Validation Agent (DVA) plugin as potential sources for validation of primary PDF structures. The software can then evolve to meet feature-requests, cover distinct use-cases, highlight best practices, advise on optimization, and more.

One can readily imagine a fantastic open-source validator that understands every aspect of PDF and provides every desirable facility to developers who wish to contribute extensions for non-PDF objects found in PDF files. And yet, such software, if it existed, would not itself answer the key questions:

- How do we know it is canonical?
- What will drive its adoption?

6.4 What makes it canonical

Similar to other infrastructure technologies like plumbing or WiFi, a specific PDF validation model becomes canonical when the industry agrees to treat it as such. There is little question that developers would love canonical quality assurance (QA) tools. If and when the specification's remaining ambiguities and validator policy questions are resolved, and the software developed, then:

- PDF vendors will use it to distinguish conforming from non-conforming software, eventually displacing older or poorly-executed products from the market.
- End-users will use it to evaluate their software and understand (and hopefully, fix) their non-conforming files.

6.5 Adoption drivers

Solving the problems discussed above will require investment by both PDF software developers and those focused on ensuring long-term access to electronic data. The industry collaborations facilitated by the PDF Association's Competence Centers such as the Isartor Test Suite and the Matterhorn Protocol [13], show that for PDF, validation models can thrive in an industry-wide context.

Is VeraPDF achievable? The core value proposition of PDF is interoperability, and the PDF industry knows it. Recognizing the need, the EU created the PREFORMA project [21] to fund development of a purpose-built open source PDF/A implementation checker together with an institutional policy checker. PREFORMA's explicit objective is to become a generally adopted reference implementation.

Hosting the VeraPDF engine on a publically-accessible webserver akin to the W3C's HTML validator [22] with an appropriate interface could provide the functionality indicated in Table 1.

Table 1. Objectives for a canonical PDF validation service

Problem	The VeraPDF Public Validator
1. PDF is complex	Canonical developer education using language accepted by the vendor community
2. Bad PDF software	Collects bad files, identifies the software producer and provides definitive problem identification and corrective information. When possible the server also fixes the file
3. Incomplete support	Drives adoption of best practice via warnings and advisories
4. Problems with fonts	As with Problem 2, provides a mechanism for pooling corrective information
5. No model for validation	Provides developer-centric features to accelerate development and reduce support costs as well as delivering authoritative 3rd party conformance information to end users

6.6 Is canonical validation realistic?

Beyond their protean nature, PDF documents may include a rich mixture of complex, variegated features. It might thus be argued that developing an open-source canonical validator is unrealistic due to the effort required. Adobe has doubtless invested hundreds of man-years in the Adobe Reader, so why would development of a validator be any less daunting? There are three basic reasons:

- A substantial proportion of Adobe's development effort is focused on handling and fixing corrupt or malformed PDF files. Although a useful validator must be able to parse deeply into corrupted files, it need only report its findings.
- Adobe's efforts must meet diverse end user needs and deliver an end user UI and attractive features in a myriad of contexts. By contrast, a validator is a purpose-built developer tool with minimal UI requirements.
- Although the required development effort certainly exceeds the resources readily available to the preservation community, as previously noted, a truly canonical validator has strong appeal to the commercial software world. Such a project will not depend on preservation community resources at all; commercial software interests can drive it.

6.7 How the preservation community can help

The development of a canonical PDF validator will not be trivial, either as a technical matter or in terms of mustering the required collaboration. Since industry acceptance is critical, adoption of the project is most likely to succeed if it is industry-led. The digital preservation community can help make it happen in several ways:

- **Ask for it.** The new NARA Transfer Guidance requires file formats be "valid" according to the format's specification. Encourage procurement entities to require specific assurances from vendors as to the validity of their output.
- **Lobby for it.** The PDF software space is broad and deep, ranging from Microsoft, Google, Apple and Adobe to one-developer shops. Digital preservation professionals know many of the people who develop software and set policy in these vendor organizations. Let them know your priorities.
- **Be a part of it.** From code contributions (for example, to the PREFORMA project) to discussion forums to writing informative error messages and serving on management or policy boards there will be a variety of ways for developers and preservation policy experts to join the effort.

7. CONCLUSION

As ISO 32000, PDF is openly and democratically managed; a *de facto* public trust. Reliability is the bottom line for PDF (and even more so for PDF/A), but ISO committees cannot write software.

While PDF is undeniably the best currently-available format for fixed-form self-contained documents, it is not yet as reliable as it should be. Developers, authors, consumers and archivists alike will all benefit from a concept of valid PDF. Working with commercial software developers the digital preservation community can take a leading role in helping to move PDF from the best available option to the ideal format for now and forever.

8. REFERENCES

[1] Arms, C., Chalfant, D., DeVorsey, K., Dietrich, C., Fleischhauer, C., Lazorchak, B., Morrissey, S., Murray, K. The benefits and risks of the PDF/A-3 file format for archival institutions. NDSA Standards and Practices Working Group 2014-02-20. Retrieved 2014-02-28 from the Library of Congress: http://www.digitalpreservation.gov/ndsa/working_groups/documents/NDSA_PDF_A3_report_final022014.pdf

[2] *De facto* standard, Wikipedia. Retrieved 2014-04-04 from Wikipedia: https://en.wikipedia.org/wiki/De_facto_standard#Examples

[3] EpubCheck, IDPF. Retrieved 2014-04-04 from Github: https://github.com/IDPF/epubcheck

[4] Isartor Test Suite, PDF Association 2011-08-03. Retrieved 2014-03-29 from PDF Association: http://www.pdfa.org/2011/08/isartor-test-suite/

[5] Johnson, D. Apple's Preview: Still not safe for work, Duff Johnson Strategy & Communications 2014-04-07. Retrieved 2014-07-21 from Duff Johnson Strategy & Communications: http://duff-johnson.com/2014/04/07/apples-preview-still-not-safe-for-work/

[6] Johnson, D. The 8 most popular document formats on the web, Duff Johnson Strategy & Communications 2014-02-17. Retrieved 2014-03-26 from Duff Johnson Strategy & Communications:

http://duff-johnson.com/2014/02/17/the-8-most-popular-document-formats-on-the-web/

[7] Johnson, D. Interest in PDF vs. other formats, Duff Johnson Strategy & Communications. Retrieved 2014-03-26 from Duff Johnson Strategy & Communications: http://duff-johnson.com/articles/interest-in-pdf-vs-other-formats/

[8] Johnson, D. 98% of .com is HTML, but 38% of .gov is PDF, Duff Johnson Strategy and Communications 2014-03-10. Retrieved 2014-03-26 from Duff Johnson Strategy & Communications: http://duff-johnson.com/2014/03/10/98-percent-of-dot-com-is-html-but-38-percent-of-dot-gov-is-pdf/

[9] Johnson, D. PDF Validation, Dream or Yawn?, Duff Johnson Strategy & Communications. Retrieved 2014-03-29 from Duff Johnson Strategy & Communications: http://duff-johnson.com/wp-content/uploads/2014/01/PDFValidationDreamOrYawn.pdf

[10] King, J., Introduction to the Insides of PDF, Adobe Systems 2005-04-26. Retrieved 2014-03-29 from Adobe Systems: http://www.adobe.com/content/dam/Adobe/en/technology/pdfs/PDF_Day_A_Look_Inside.pdf

[11] Knijff, J. van der. Identification of PDF preservation risks: analysis of Govdocs selected corpus, Open Planets Foundation 2014-01-27. Retrieved 2014-03-20 from Open Planets Foundation: http://www.openplanetsfoundation.org/blogs/2014-01-27-identification-pdf-preservation-risks-analysis-govdocs-selected-corpus

[12] Leurs, L. The history of PDF, Prepressure.com 2013-08-09. Retrieved 2014-03-27 from Prepressure.com: http://www.prepressure.com/pdf/basics/history

[13] Matterhorn Protocol, PDF Association 2014-02-11. Retrieved 2014-03-29 from PDF Association: http://www.pdfa.org/publication/the-matterhorn-protocol-1/

[14] Moore, R., and Evans, T. Preserving the Grey Literature Explosion: PDF/A and the Digital Archive. Information Standards Quarterly, Fall 2013, 25(3): 20-27 http://dx.doi.org/10.3789/isqv25no3.2013.04

[15] Morrissey, S., "The Network is the Format: PDF and the Long-term Use of Digital Content", *Archiving 2012*, (2012): pp. 200-203. Retrieved 2014-03-23 from Portico: http://www.portico.org/digital-preservation/wp-content/uploads/2012/11/TheNetworkIsTheFormat.pdf

[16] NARA Transfer Guidance, National Archives and Records Administration. Retrieved 2014-04-03 from NARA: http://www.archives.gov/records-mgmt/policy/transfer-guidance.html

[17] PDF (Portable Document Format) Family, National Digital Information Infrastructure and Preservation Program 2014-02-08. Retrieved 2014-03-22 from the Library of Congress: http://www.digitalpreservation.gov/formats/fdd/fdd000030.shtml

[18] PDF/A Competence Center, Nearly all archives projects use PDF/A, PDF/A Competence Center 2009-03-31. Retrieved 2014-04-04 from the PDF Association: http://www.pdfa.org/2009/03/nearly-all-archiving-projects-use-pdfa/

[19] Rimkus, K., Padilla, T., Popp, T. and Martin, G. Digital Preservation File Format Policies of ARL Member Libraries: An Analysis. *D-Lib Magazine*, 20 (3/4): http://dx.doi.org/doi:10.1045/march2014-rimkus

[20] Rosenthol, L., ISO 32000 Document management Portable document format PDF 1.7, Inside PDF Blog 2008-01-28. Retrieved 2014-03-22 from Adobe Systems: http://blogs.adobe.com/insidepdf/2008/01

[21] Tender, PREFORMA. Retrieved 2014-08-03 from PREFORMA Project: http://www.preforma-project.eu/tender.html

[22] W3C Markup Validation Service, W3C. Retrieved 2014-03-29 from W3C: http://validator.w3.org/