Network Working Group
Internet-Draft
Intended status: Informational
Expires: January 22, 2015

T. Hansen, Ed.
AT&T Laboratories
L. Masinter
M. Hardy
Adobe

July 21, 2014

PDF for an RFC Series Output Document Format
draft-hansen-rfc-use-of-pdf-01

Abstract

<1>

This document discusses options and requirements for the PDF rendering of RFCs in the RFC Series, as outlined in RFC 6949. It also discusses the use of PDF for Internet Drafts, and available or needed software tools for producing and working with PDF.

Status of This Memo

This Internet-Draft will expire on January 22, 2015.

Copyright Notice

Table of Contents

1. Introduction

<2>

The RFC Series is evolving, as outlined in [RFC6949]. Future documents will use an archival format of XML with renderings in various formats, including PDF.

<3>

Because PDF has a wide range of capabilities and alternatives, not all PDFs are "equal". For example, visually similar documents could consist of scanned or rasterized images, include text layout options, hyperlinks, embedded fonts, and digital signatures. (See Appendix A for a brief history of PDF.)

<4>

This document explains some of the relevant options and makes recommendations, both for the RFC series and Internet Drafts.

<5>

The PDF format and the tools to manipulate it are not as well known as those for the other RFC formats, at least in the IETF community. This document discusses some of the processes for creating and using PDFs using both open source and commercial products.

<6>

NOTE: see [1] for XML source, related files, and an issue tracker for this document.

2. History and current use of PDF with RFCs and Internet Drafts

<7>

NOTE: this section is meant as an overview to give some background.

2.1. RFCs

<8>

The RFC series has for a long time accepted Postscript renderings of RFCs, either in addition to or instead of the text renderings of those same RFCs. These have usually been produced when there was a complicated figure or mathematics within the document. For example, consider the figures and mathematics found in RFCs 1119 and RFC 1142, and compare the figures found in the text version of RFC 3550 with those in the Postscript version. The RFC editor has provided a PDF rendering of RFCs. Usually, this has been a print of the text file that does not take advantage of any of the broader PDF functionality, unless there was a Postscript version of the RFC, which would then be used by the RFC editor to generate the PDF.

2.2. Internet Drafts

<9>

In addition to PDFs generated and published by the RFC editor, the IETF tools community has also long supported PDF for Internet Drafts. Most RFCs start with Internet Drafts, edited by individual authors. The Internet drafts submission tool at https://datatracker.ietf.org/submit/ accepts PDF and Postscript files in addition to the

(required) text submission and (currently optional) XML. If a PDF wasn't submitted for a particular version of an Internet Draft, the tools would generate one from the Postscript, HTML, or text.

3. Options and Requirements for PDF RFCs

<10>

This section lays out options and requirements for PDFs produced by the RFC editor for RFCs. There are two sections: "Visible" options are related to how the PDF appears when it is viewed with a PDF viewer. "Internal Structure" options affect the ability to process PDFs in other ways, but do not control the way the document appears. (Of course, a viewer UI might display processing capabilities, such as showing if a document has been digitally signed.)

<11>

In many cases, the choice of PDF requirements is heavily influenced by the capabilities of available tools to create PDFs. Most of the discussion of tooling is to be found in Section 4.

<12>

NOTE: each option in this section should outline the nature of the design choice, outline the pros and cons, and make a recommendation.

3.1. "Visible" requirements

<13>

PDF supports rich visible layout of fixed-sized pages.

3.1.1. General visible requirements

<14>

For a consistent 'look' of RFC and good style, the PDFs produced by the RFC editor should have a clear, consistent, identifiable and easy-to-read style. They should print well on the widest range of printers, and look good on displays of varying resolution.

3.1.2. Page size, margins, headers and footers

<15>

PDF files are laid out for a particular size of page, margins, and any headers and footers part of the layout. There are two paper sizes in common use: "US Letter" (8.5 x 11 inches, 216x279 mm, in popular use in North America) and "A4" (210x297 mm, 8.27x11.7 inches, standard for the rest of the world). Usually PDF printing software is used in a "shrink to fit" mode where the printing is adjusted to fit the paper in the printer. Whatever page size is chosen, the margins and header positioning will need to be chosen to look good on both paper sizes using common printing methods. In addition, for both Internet Drafts and RFCs, margins should be the smallest consistent with the above requirement.

<16>

Page headers and footers should contain similar information as the headings in the current text versions of documents, including page

numbers, short title, author, working group, but typeset in a lighter color, smaller typeface, so as to be inobtrusive.

3.1.3. Similarity to other outputs

<17>

There is some advantage to having the PDF files look like the text or HTML renderings of the same document. There are several options even so. The PDF

<18>

1. could look like the text version of the document, or

<19>

2. could look like the text version of the document but with pictures rendered as pictures instead of using their ASCII-art equivalent, or

<20>

3. could look like the HTML version.

<21>

Recommendation: the PDF rendition should look like the HTML rendition, at least in spirit. For example, some differences from the HTML rendition might include different typeface and size (chosen for printing), page numbers in the table of contents, page headers and footers and headers.

3.1.4. Typeface choices

<22>

A PDF may refer to a font by name, or it may use an embedded font. When a font is not embedded, a PDF viewer will attempt to locate a locally installed font of the same name. If it can not find an exact match, it will find a "close match". If a close match is not available, it will fallback to something implementation dependent and usually undesirable.

<23>

Recommendation: for consistent viewing, all fonts should be embedded.

<24>

In addition, if the HTML version of the document is being visually mimiced, the font(s) chosen should have both variable width and constant width components, as well as bold and italic representations.

<25>

The typefaces used by Internet Drafts and by RFCs need not be identical.

<26>

Few fonts have glyphs for the entire repertoire of Unicode characters; for this purpose, the PDF generation tool may need a set of fonts and a way of choosing them.

<27>

Typefaces are typically licensed and, in many cases, there is a fee

for use by PDF creation tools; however, not for display or print of the embedded fonts.

<28>

Recommendations:

<29>

- For readability when printed, the main body text should be in a serif font and the headings in a sans-serif font.

<30>

- Code, BNF, and other text could use a fixed-width font to aid in insuring alignment, e.g., in BNF.

<31>

- Type faces used by the xml2rfc application for Internet drafts should be freely available, and included with the xml2rfc application.

<32>

- The range of Unicode characters allowed in the XML source for Internet Drafts and RFCs may be bounded by the availability of embeddable fonts with appropriate glyphs.

3.1.5. Embedding of Fonts

<33>

The PDF/A standards mandate the embedding of fonts. Preferably, the software generating the files would produce PDF/A-conforming files directly, thus ensuring that all glyphs include Unicode mappings and embedded fonts from the outset.

3.1.6. Hyperlinks

<34>

PDF supports hyperlinks both to sections of the same document and to other documents.

<35>

The conversion to PDF can generate:

<36>

- hyperlinks within the document

<37>

- hyperlinks to external locations

<38>

- hyperlinks within a table of contents

<39>

Where should hyperlinks to RFCs point? to the info page for the RFC? to the PDF version of the RFC? (NOTE: the RFC Series Editor has stated a preference for them to point to the info page for the RFC.) Similar questions need to be answered on references to internet drafts: Where should hyperlinks to internet drafts point? To the datatracker entry? To the tools entry? To a PDF version of the internet draft?

<40>

Recommendations:

<41>

- o All hyperlinks available in the HTML rendition of the RFC should also be visible and active in the PDF produced.

<42>

- o Table of contents, including page numbers, are useful when printed. These should be hyperelinked.

<43>

- o Hyperlinks to RFCs and Internet drafts from the references section should point to a "landing" page which then links to the various formats available.

3.2. "Invisible" options and requirements

<44>

PDF offers a number of features which improve the utility of PDF files in a variety of workflows, at the cost of extra effort in the xml2rfc conversion process; the tradeoffs may be different for the RFC editor production of RFCs and for Internet Drafts.

3.2.1. Internal Text Representation

<45>

The contents of a PDF file can be represented in many ways. The PDF file could be generated:

<46>

- o as an image of the visual representation, such as a JPEG image of the word 'IETF'

<47>

- o placing individual characters in position on the page, such as saying "put an 'F' here", then "put an 'T' before it", then "put an 'E' before that", then "put an 'I' before that" to render the word 'IETF'

<48>

- o placing words in position on the page, such as keeping the word 'IETF' would be kept together, and

<49>

- o insuring that the running order of text in the content stream matches logical reading order, e.g., keeping the sentence 'The Internet Engineering Task Force (IETF) supports the Internet.' together as a sentence.

<50>

- o A "role map" feature of PDF would allow mapping the logical tags found in the original XML into tags in the PDF.

<51>

All of these end up with essentially the same visual representation of the output. However, each level has tradeoffs for auxiliary uses, such as searching or indexing, commenting and annotation, and accessibility (text-to-speech).

<52>

Recommendations:

<53>

- o Text in content streams should follow the XML document's logical order (in the order of tags) to the extent possible. This will provide optimal reuse by software that does not understand Tagged PDF. (PDF/UA requires this.)

<54>

- o We should investigate the use of role-maps to capture more of the xml2rfc source structure, to the point where it might even be possible to reconstruct much or all of the source. However, there is not a compelling use case over embedding the original XML, as described in Section 3.2.7.

3.2.2. Unicode Support

<55>

PDF itself does not require use of Unicode. Text is represented as a sequence of glyphs which then can be mapped to Unicode.

<56>

Recommendations:

<57>

PDF files generated must have the full text, as it appears in the original XML.

<58>

Unicode normalization may occur.

<59>

Text within SVG for SVG images should also have Unicode mappings.

<60>

Alt-text for images should also have Unicode.

3.2.3. Text description of images (alt-text)

<61>

NOTE: This section should describe how alt-text for images is presented in PDF....TBD

3.2.4. Metadata Support

<62>

Metadata encodes information about the document authors, the document series, date created, etc. using the RDF Dublin core (and other elements). Having this metadata within the PDF file allows it to be used by search engines, viewers and other reuse tools.

<63>

PDF supports embedded metadata using XMP [XMP], the Extensible Metadata Platform (XMP).

<64>

Recommendation: The PDFs generated should have all of the metadata from the XML version embedded directly as XMP metadata, including the author and date information, set the document series, and a URL for where the document can be retrieved.

3.2.5. Document Structure Support

<65>

PDF supports an 'outline' feature where sections of the document are marked; this oould be used in addition to the table of contents as a navigation aid.

<66>

The section structure of an RFC can be mapped into the PDF elements for the document structure. This will allow the bookmark feature of PDF readers to be used to quickly access sections of the document.

<67>

Requirement: The section structure of an RFC should be mapped into the PDF elements for the document structure. This would include section headings for the boilerplate sections such as the Abstract, Status of the Document, Table of Contents, and Author Addresses.

3.2.6. Tagged PDF

<68>

NOTE: say more about the use of alternative texts for images, tagging text spans, and providing replacement texts for symbols and images. A role-map could be provided here to map the logical tags found in the RFC XML to the standard tagset for PDF. This could be included in the generated PDF.

3.2.7. Embedded Files

<69>

PDF has the capability of including other files; the files may be labeled both by a media type and a role, the AFRelationship key [PDFA3]. In this way, the PDF file acts also as a container.

<70>

Embedded content may be compressed.

<71>

Many PDF viewers support the ability to view and extract embedded files, although this capability is not universal.

<72>

Embedding content in the PDF file allows the PDF to act as a complete package, which can be transformed, archived, digitally signed. Useful possibilities:

<73>

Embed the source XML input file itself within the PDF. If the source SVG and images for illustrations are also embedded, this would make the PDF file totally self-referential.

<74>

Embed directly extractable components that are useful for independent processing, including ABNF, MIBs, source code for reference implementations. This capability might be supported through other mechanisms from the XML source files, but could also be supported within the PDF.

<75>

Recommendations:

<76>

Embed the XML source and all illustrations, for both RFCs and Internet Drafts, as a standard feature for xml2rfc's PDF output.

<77>

Finding, extracting and embedding other components will require additional markup to clearly identify them, and additional review to insure the correctness of embedded files which are not visible.

3.3. Digital Signatures

<78>

PDF has supported digital signatures since PDF 1.2. There are multiple methods for signing PDF files. The signature is intended to apply not only to the bits in the file (that they haven't been modified) but also to lock down the visual presentation as well.

<79>

Normally, the authenticity of RFC files is not an issue, since the RFC editor maintains a repository of all RFCs which is widely replicated. However, the RFC Editor and staff are at times called to provide evidence that a particular RFC is the 'original' and has not been visually modified, and there may be other use cases.

<80>

Recommendation: The use cases for digital signatures need further review, including management of certificates for the RFC editor function. PDFs produced by the RFC editor likely SHOULD be signed. As signatures also apply to embedded content, this will provide a way of signing the source XML as well. There is no need for digital signatures on Internet Drafts.

4. Tooling

<81>

This section discusses tools for viewing, comparing, creating, manipulating, transforming PDF files, including those currently in use by the RFC editor and Internet drafts, as well as outlining available PDF tools for various processes.

4.1. PDF Viewers

<82>

As with most file formats, PDF files are experienced through a reader or viewer of PDF files, and there are numerous viewers. One partial list of PDF viewers can be found at [2].

<83>

PDF viewers vary in capabilities, and it is important to note which PDF viewers support the features utilized in PDF RFCs and Internet drafts (features such as links, digital signatures, Tagged PDF and others mentioned in Section 3).

<84>

A survey of the IETF community might broaden the list of viewers in common use, but an initial list to consider include some that are currently maintained and supported viewers and legacy systems. Maintained viewers include:

<85>

Adobe Reader Multiple platforms. Supports all of the features on most platforms.

<86>

Google Chrome Multiple platforms. Web browser which includes PDF support. Rapidly moving target, open source.

<87>

PDF.js Multiple platforms. A JavaScript library to convert PDF files into HTML5, usable as a web-based viewer that can be included in web browsers. Used by Mozilla Firefox. Also rapidly moving target.

<88>

Foxit Reader Multiple platforms. PDF Viewer / Reader for Desktop computer and Mobile Devices. Recently licensed by Google, and the code for this purpose was made open source; see [3].

<89>

Several 'legacy' viewers to consider include: Ghostview, Xpdf.

4.2. Printers

<90>

While almost all viewers also support printing of PDF files, printing is one of the most important use cases for PDFs. Some printers have direct PDF support.

4.3. PDF generation libraries

<91>

Because the xml2rfc format is a unique format, software for converting XML source documents to the various formats will be needed, including PDF generation.

<92>

One promising direction is suggested in [4]: using XSLT to generate XSL-FO which is then processed by a formatting object processor such as Apache FOP.

4.4. Other Tools

<93>

In addition to generating and viewing PDF, other categories of PDF tools are available and may be useful both during specification development and for published RFCs.

<94>

These include tools for comparing two PDFs, checkers that could be used to validate the results of conversion, review and comment tools

which attach annotations to PDF files, digital signature creation and validation.

<95>

Validation of an arbitrary author-generated PDF file would be quite difficult; there are few PDF validation tools. However, if internet drafts and RFCs are generated by conversion from XML via xml2rfc, then explicit validation of PDF and adherance to expected profiles would mainly be useful to insure xml2rfc has functioned properly.

<96>

Recommendations:

<97>

- o Discourage (but allow) submission of a PDF representation for Internet Drafts. In most cases, the PDF for an Internet draft should be produced automatically when XML is suhmitted, with an opportunity to verify the conversion.

<98>

- o The RFC editor should create PDF files from the XML rather than accepting PDFs from the author.

5. Choosing PDF versions and standards

<99>

PDF has gone through several revisions, primarily addition of features, as noted in in Appendix A. PDF features have generally been added in a way that older viewers 'fail gracefully', but even so, the older the PDF version produced, the more legacy viewers will support that version, but the fewer features will be enabled.

<100>

As PDF has evoled a broad set capabilities, additional standards for PDF files are applicable. These standards establish ground rules that are important for specific applications. For example PDF/X was specifically designed for Prepress digital data exchange, with careful attention to color management and printing instructions, while PDF/E standard was designed for engineering documents.

<101>

Two additional standards families are important to the RFC format, though: long-term preservation (PDF/A), and user acessibility (PDF/UA). These standards are then supported by various software libraries and tools.

<102>

It is effective and useful to use these standards to capture PDF for RFC requirements, and they will make the PDF files useful in workflows that expect them.

<103>

Recommendations:

<104>

Choose PDF 1.7; although relatively recent, it is well supported by widely available viewers.

<105>

Require PDF/A3 for RFCs. It captures the archivability and long-term stability of PDF 1.7 files. Use PDF/A3 for embedding additional data (incluing the source files) in RFCs and Internet Drafts.

<106>

Require PDF/UA for RFCs.

6. References

6.1. References

[PDF] ISO, "Portable document format -- Part 1: PDF 1.7", ISO 32000-1, 2008.

Also available free from Adobe.

[XMP] ISO, "Extensible metadata platform (XMP) specification -- Part 1: Data model, serialization and core properties", ISO 16684-1, 2012.

Not available free, but there are a number of descriptive resources, e.g., [5]

[PDFA2] ISO, "Electronic document file format for long-term preservation -- Part 2: Use of ISO 32000-1 (PDF/A-2).", ISO 19005-2, 2011.

[PDFA3] ISO, "Electronic document file format for long-term preservation -- Part 3: Use of ISO 32000-1 with support for embedded files (PDF/A-3)", ISO 19005-3, 2012.

[PDFUA] ISO, "Electronic document file format enhancement for accessibility -- Part 1: Use of ISO 32000-1 (PDF/UA-1)", ISO 19005-3, 2012.

6.2. Informative References

[RFC3778] Taft, E., Pravetz, J., Zilles, S., and L. Masinter, "The application/pdf Media Type", RFC 3778, May 2004.

[RFC6949] Flanagan, H. and N. Brownlee, "RFC Series Format Requirements and Future Development", RFC 6949, May 2013.

6.3. URIs

[1] https://github.com/masinter/pdfrfc

[2] http://en.wikipedia.org/wiki/List_of_PDF_software#Viewers

[3] http://www.i-programmer.info/news/136-open-source/7433-google-open-sources-pdf-software-library.html

[4] http://greenbytes.de/tech/webdav/rfc2629xslt/rfc2629xslt.html#output.pdf.fop

[5] http://www.adobe.com/devnet/pdf/pdf_reference_archive.html

[6] http://en.wikipedia.org/wiki/PDF

[7] http://www.pdflib.com/fileadmin/pdflib/pdf/whitepaper/Whitepaper-Technical-Introduction-to-PDFA.pdf

[8] http://www.pdfa.org/wp-content/uploads/2011/08/tn0003_metadata_in_pdfa-1_2008-03-128.pdf

[9] http://www.pdfa.org/wp-content/uploads/2011/08/PDFA-in-a-Nutshell_1b.pdf

[10] http://www.pdfa.org/2011/08/pdfa-%E2%80%93-a-look-at-the-technical-side/

[11] http://pdf.editme.com/pdfa

Appendix A. A Synopsis of PDF Format History

<107>
[RFC3778] contains some history of PDF. This is a capsule view, plus additional information on events that have occurred since the publication of [RFC3778]. NOTE: currently doesn't talk about the handoff of change control to ISO and the evolution as an ISO standard 32000. Plans are to update the application/pdf MIME registration to include this information, and then point to that.

<108>
The Portable Document Format (PDF) family of document formats was invented by Adobe Systems in the early 1990s. At the time, it was a proprietary format that underwent a variety of revisions that matched the release of different versions of the Adobe Acrobat products. For example, Acrobat 1 supported PDF version 1.0, Acrobat 2 supported PDF version 1.1, Acrobat 5 supported PDF version 1.4, etc. [6]

<109>
Each release (and extension level) introduced new features. For example, (1.0) character, word and image rendering, externally-

referenced or embedded fonts, (1.1) passwords, encryption, device-independent color, (1.2) interactive forms, unicode, signatures, compression, (1.3) web semantic capture, embedded files, Adobe javascript, (1.4) metadata streams, tagged PDF, (1.5) controllable hiding of sections, slideshows, (1.6) 3D artwork, OpenType font embedding, linking into embedded files, and (1.7) video and audio support. After release 1.7, additional Extension Levels have been introduced. Each release also provided enhancements to the previous support. For example, encryption was introduced in 1.1, but AES encryption wasn't supported until 1.7 extension level 3. A PDF reader for PDF 1.1 is not able to read and display a PDF 1.7 file, but a PDF reader for PDF 1.7 can also handle all previous versions of PDF. The wikipedia page at [7] has a nice summary table going into further details.

A.1. PDF Profiles

<110>

Certain profiles or subsets of PDF have been standardized. PDF/X (X for Exchange), PDF/A (A for Archive), PDF/E (E for Engineering), PDF/VT (VT for Variables and Transactions), and PDF/UA (UA for Universal Access) all have ISO standards associated with them. Of particular potential interest to the RFC community are PDF/A and PDF/UA.

A.1.1. PDF/A

<111>

PDF/A in turn has nuances, as there have been a couple updates to it and conformance levels within each version. PDF/A-1 was based on PDF release 1.4. PDF/A-2 was based on PDF release 1.7, and PDF/A-3 adds embedded arbitrary files. PDF/A is considered a profile because it mandates that certain optional features be used. At a high level, the conformance levels are B (basic), U (mandatory unicode mapping [not in PDF/A-1]) and A (accessible). The requirements for conformance level A are that: the document structure must be represented within the PDF (e.g., section headings, table cells, paragraph divisions), tagged PDF is used (e.g., element anchors) and that language tags be used where appropriate. When referring to PDF/A, you would refer to the version and conformance level. So PDF/A-1A would be the profile for the Accessible conformance level of version 1 of PDF/A, which was based on PDF 1.4.

A.1.2. PDF/UA

<112>

The PDF/UA (Universal Access) profile is orthogonal to the other profiles, specifying user accessibility requirements. It places some restrictions on the other profiles, such as requiring the use of higher-level constructs for the textual representation and adds additional requirements for programatic access (think automatic readers for the blind).

A.2. Additional Reading

<113>

[8] [9] [10] [11] [12]

Appendix B. Acknowledgements

<114>

The input of the following people is gratefully acknowledged: Brian Carpenter, Chris Dearlove, Martin Duerst, Joe Hildebrand, Duff Johnson, Leonard Rosenthal, ...

Authors' Addresses

Tony Hansen (editor)
AT&T Laboratories
200 Laurel Ave. South
Middletown, NJ 07748
USA

Email: tony+rfc2pdf@maillennium.att.com

Larry Masinter
Adobe
345 Park Ave
San Jose, CA 95110
USA

Email: masinter@adobe.com
URI: http://larry.masinter.net

Matthew Hardy
Adobe
345 Park Ave
San Jose, CA 95110
USA

Email: mahardy@adobe.com