

Network Working Group  
Internet-Draft  
Intended status: Informational  
Expires: January 23, 2015

T. Hansen, Ed.  
AT&T Laboratories  
L. Masinter  
M. Hardy  
Adobe  
July 22, 2014

**PDF for an RFC Series Output Document Format**  
**draft-hansen-rfc-use-of-pdf-02**

Abstract

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 is submitted in full conformance with the provisions of [BCP 78](#) and [BCP 79](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on January 23, 2015.

Copyright Notice

Copyright (c) 2014 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in [Section 4.e](#) of

the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

- [1. Introduction . . . . .](#) [3](#)
- [2. Options and Requirements for PDF RFCs . . . . .](#) [3](#)
  - [2.1. "Visible" requirements . . . . .](#) [3](#)
    - [2.1.1. General visible requirements . . . . .](#) [4](#)
    - [2.1.2. Page size, margins . . . . .](#) [4](#)
    - [2.1.3. headers and footers . . . . .](#) [4](#)
    - [2.1.4. paragraph numbering . . . . .](#) [4](#)
    - [2.1.5. Paged content layout quality . . . . .](#) [5](#)
    - [2.1.6. Similarity to other outputs . . . . .](#) [5](#)
    - [2.1.7. Typeface choices . . . . .](#) [6](#)
    - [2.1.8. Hyphenation and line breaks . . . . .](#) [7](#)
    - [2.1.9. Embedding of Fonts . . . . .](#) [7](#)
    - [2.1.10. Hyperlinks . . . . .](#) [7](#)
  - [2.2. "Invisible" options and requirements . . . . .](#) [8](#)
    - [2.2.1. Internal Text Representation . . . . .](#) [8](#)
    - [2.2.2. Unicode Support . . . . .](#) [9](#)
    - [2.2.3. Text description of images \(alt-text\) . . . . .](#) [9](#)
    - [2.2.4. Metadata Support . . . . .](#) [9](#)
    - [2.2.5. Document Structure Support . . . . .](#) [10](#)
    - [2.2.6. Tagged PDF . . . . .](#) [10](#)
    - [2.2.7. Embedded Files . . . . .](#) [10](#)
  - [2.3. Digital Signatures . . . . .](#) [11](#)
- [3. Tooling . . . . .](#) [11](#)
  - [3.1. PDF Viewers . . . . .](#) [11](#)
  - [3.2. Printers . . . . .](#) [12](#)
  - [3.3. PDF generation libraries . . . . .](#) [12](#)
  - [3.4. Typefaces . . . . .](#) [12](#)
  - [3.5. Other Tools . . . . .](#) [13](#)
- [4. Choosing PDF versions and standards . . . . .](#) [13](#)
- [5. References . . . . .](#) [14](#)
  - [5.1. References . . . . .](#) [14](#)
  - [5.2. Informative References . . . . .](#) [15](#)
  - [5.3. URIs . . . . .](#) [15](#)
- [Appendix A. History and current use of PDF with RFCs and Internet Drafts . . . . .](#) [16](#)
  - [A.1. RFCs . . . . .](#) [16](#)
  - [A.2. Internet Drafts . . . . .](#) [16](#)
- [Appendix B. A Synopsis of PDF Format History . . . . .](#) [16](#)
  - [B.1. PDF Profiles . . . . .](#) [17](#)
    - [B.1.1. PDF/A . . . . .](#) [17](#)
    - [B.1.2. PDF/UA . . . . .](#) [17](#)
  - [B.2. Additional Reading . . . . .](#) [18](#)
- [Appendix C. Acknowledgements . . . . .](#) [18](#)



Authors' Addresses . . . . . [18](#)

**[1. Introduction](#)**

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.

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 B](#) for a brief history of PDF.)

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

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.

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

**[2. Options and Requirements for PDF RFCs](#)**

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.)

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 3](#).

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

**[2.1. "Visible" requirements](#)**

PDF supports rich visible layout of fixed-sized pages.



### **2.1.1. General visible requirements**

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.

### **2.1.2. Page size, margins**

PDF files are laid out for a particular size of page and margins. 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.

Recommendation: There is some controversy, but the argument for choosing A4 over US Letter is appealing. 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.

### **2.1.3. headers and footers**

Page headers and footers are part of the page layout. There are a variety of options. Note that page headers and footers in PDF can be typeset in a way that the entire (longer) title might fit.

Page headers and footers should contain similar information as the headings in the current text versions of documents, including page numbers, title, author, working group, but typeset in a lighter color, smaller typeface, so as to be inobtrusive.

### **2.1.4. paragraph numbering**

One common feature of the new RFC output formats are visible paragraph numbers, to aid in discussions. In the PDF and thus printed rendition, it is possible to make paragraph numbers unobtrusive, and even to impinge on the margins.

Recommendation: show paragraph numbers in the right margin, not left, small font, light color, and, if possible, code them in a way as to not interfere with screen readers.



### **2.1.5. Paged content layout quality**

The process of creating a paged document from running text typically involves insuring related material is present on the same page together, and that artifacts of pagination don't interfere with easy reading of the document. Typical high-quality layout processors do several things:

widow and orphan management Insure a page break does not occur after the first line of a paragraph, or before the last line (unless the entire paragraph is only one line).

keep section heading contiguous Do not page break immediately after a section heading. If there isn't room on a page for the first (two) lines of a section after the section heading, page break before the heading.

avoid splitting artwork Figures should not be split from figure titles. If possible, keep the figure on the same page as the (first) mention of the figure.

headers for long tables after page breaks Another common option in producing paginated documents is to include the column headings of a table if the table cannot be displayed on a single page.

Layout engines differ in the quality of the algorithms used to automate these processes. In some cases, the automated processes require some manual assistance to insure, for example, that a text line intended as a heading is 'kept' with the text it is heading for.

Recommendation: Minimize requirements for authors to add specific markup to the XML source for assisting with with manually enhancing layout for Internet Drafts. Choose a layout engine for the RFC Editor so that manual intervention is minimized, and that widow and orphan processing, heading keep are automatic.

### **2.1.6. Similarity to other outputs**

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

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





3. could look like the HTML version.

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.

#### **2.1.7. Typeface choices**

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.

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

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.

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

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.

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.

Recommendations:

- o For readability when printed, the main body text should be in a serif font and the headings in a sans-serif font.
- o Code, BNF, and other text could use a fixed-width font to aid in insuring alignment, e.g., in BNF.
- o Type faces used by the xml2rfc application for Internet drafts should be freely available, and included with the xml2rfc application.



- o 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.

#### **2.1.8. Hyphenation and line breaks**

Typically, when doing page layout of running text, especially with narrow page width and long words, layout processors of English text often have the option of hyphenating words, or using existing hyphens as a place to introduce word breaks. However, line breaks mid-word where the 'word' is actually technically a sequence of characters representing a protocol element or protocol sequence is actively harmful.

Recommendation: do not introduce any hyphens into the visual display or any line breaks mid-word even for hyphens in the text.

#### **2.1.9. Embedding of Fonts**

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.

#### **2.1.10. Hyperlinks**

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

The conversion to PDF can generate:

- o hyperlinks within the document
- o hyperlinks to external locations
- o hyperlinks within a table of contents

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?

Recommendations:



- o All hyperlinks available in the HTML rendition of the RFC should also be visible and active in the PDF produced.
- o Table of contents, including page numbers, are useful when printed. These should be hyperlinked.
- 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.

## **2.2. "Invisible" options and requirements**

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.

### **2.2.1. Internal Text Representation**

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

- o as an image of the visual representation, such as a JPEG image of the word 'IETF'
- 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'
- o placing words in position on the page, such as keeping the word 'IETF' would be kept together, and
- 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.
- o A "role map" feature of PDF would allow mapping the logical tags found in the original XML into tags in the PDF.

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).

Recommendations:



- 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.)
- 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 2.2.7](#).

### **[2.2.2. Unicode Support](#)**

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

Recommendations:

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

Unicode normalization may occur.

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

Alt-text for images should also have Unicode.

### **[2.2.3. Text description of images \(alt-text\)](#)**

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

### **[2.2.4. Metadata Support](#)**

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.

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

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.





### **2.2.5. Document Structure Support**

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

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.

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.

### **2.2.6. Tagged PDF**

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.

### **2.2.7. Embedded Files**

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.

Embedded content may be compressed.

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

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

Useful possibilities:

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.

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.



#### Recommendations:

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

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.

### **2.3. Digital Signatures**

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.

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.

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.

## **3. Tooling**

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.

### **3.1. PDF Viewers**

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 [\[2\]](#).

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 2](#)).



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:

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

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

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.

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].

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

### **3.2. Printers**

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.

### **3.3. PDF generation libraries**

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

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.

### **3.4. Typefaces**

This section is intended to discuss available typefaces that might satisfy requirements. Some openly available fixed-width typefaces (without extensive Unicode support, however) include:

- o Source Sans [5]
- o Source Serif Pro [6]



- o Source Code Pro [7]

### 3.5. Other Tools

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.

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.

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 adherence to expected profiles would mainly be useful to insure `xml2rfc` has functioned properly.

Recommendations:

- 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 submitted, with an opportunity to verify the conversion.
- o The RFC editor should create PDF files from the XML rather than accepting PDFs from the author.

## 4. Choosing PDF versions and standards

PDF has gone through several revisions, primarily addition of features, as noted in in [Appendix B](#). 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.

As PDF has evolved 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.

Two additional standards families are important to the RFC format, though: long-term preservation (PDF/A), and user accessibility (PDF/UA). These then have sub-profiles (PDF/A-1, PDF/A-2, PDF/A-3), each





of which have conformance levels. These standards are then supported by various software libraries and tools.

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.

#### Recommendations:

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

Require PDF/A-3, conformance level 'u' for RFCs. This requirement captures the archivability and long-term stability of PDF 1.7 files, and many of the requirement features.

Use PDF/A-3 for embedding additional data (including the source files) in RFCs and Internet Drafts.

Require PDF/UA for RFCs.

## 5. References

### 5.1. 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., [\[8\]](#)

[PDF/A2] ISO, "Electronic document file format for long-term preservation -- Part 2: Use of ISO 32000-1 (PDF/A-2).", ISO 19005-2, 2011.

[PDF/A3] 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.



## 5.2. 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.

## 5.3. URIs

- [1] <https://github.com/masinter/pdfRFC>
- [2] [http://en.wikipedia.org/wiki/List\\_of\\_PDF\\_software#Viewers](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] <https://sourceforge.net/projects/sourcesans.adobe/?source=directory>
- [6] <https://sourceforge.net/projects/sourceserifpro.adobe/?source=directory>
- [7] <https://sourceforge.net/projects/sourcecodepro.adobe/?source=directory>
- [8] [http://www.adobe.com/devnet/pdf/pdf\\_reference\\_archive.html](http://www.adobe.com/devnet/pdf/pdf_reference_archive.html)
- [9] <http://en.wikipedia.org/wiki/PDF>
- [10] <http://www.pdfLib.com/fileadmin/pdfLib/pdf/whitepaper/Whitepaper-Technical-Introduction-to-PDFA.pdf>
- [11] [http://www.pdfa.org/wp-content/uploads/2011/08/tn0003\\_metadata\\_in\\_pdfa-1\\_2008-03-128.pdf](http://www.pdfa.org/wp-content/uploads/2011/08/tn0003_metadata_in_pdfa-1_2008-03-128.pdf)
- [12] [http://www.pdfa.org/wp-content/uploads/2011/08/PDFA-in-a-Nutshell\\_1b.pdf](http://www.pdfa.org/wp-content/uploads/2011/08/PDFA-in-a-Nutshell_1b.pdf)
- [13] <http://www.pdfa.org/2011/08/pdfa-%E2%80%93-a-look-at-the-technical-side/>
- [14] <http://pdf.editme.com/pdfa>



## **Appendix A. History and current use of PDF with RFCs and Internet Drafts**

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

### **A.1. RFCs**

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 [RFC 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.

### **A.2. Internet Drafts**

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.

## **Appendix B. A Synopsis of PDF Format History**

[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.

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. [\[9\]](#)

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 [[10](#)] has a nice summary table going into further details.

## **[B.1.](#) PDF Profiles**

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.

### **[B.1.1.](#) PDF/A**

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.

### **[B.1.2.](#) PDF/UA**

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).





## **B.2. Additional Reading**

[11] [[12](#)] [[13](#)] [[14](#)] [15]

## **Appendix C. Acknowledgements**

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](mailto:tony+rfc2pdf@maillennium.att.com)

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

Email: [masinter@adobe.com](mailto:masinter@adobe.com)  
URI: <http://larry.masinter.net>

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

Email: [mahardy@adobe.com](mailto:mahardy@adobe.com)

