WIKIPEDIA
The Free Encyclopedia

PDF

Portable Document Format (**PDF**), standardized as **ISO 32000**, is a file format developed by Adobe in 1992 to present documents, including text formatting and images, in a manner independent of application software, hardware, and operating systems.[2][3] Based on the PostScript language, each PDF file encapsulates a complete description of a fixed-layout flat document, including the text, fonts, vector graphics, raster images and other information needed to display it. PDF has its roots in "The Camelot Project" initiated by Adobe co-founder John Warnock in 1991.[4] PDF was standardized as ISO 32000 in 2008.[5] It is maintained by ISO TC 171 SC 2 WG8, of which the PDF Association is the committee manager.[6] The last edition as ISO 32000-2:2020 was published in December 2020.[7]

PDF files may contain a variety of content besides flat text and graphics including logical structuring elements, interactive elements such as annotations and form-fields, layers, rich media (including video content), three-dimensional objects using U3D or PRC, and various other data formats. The PDF specification also provides for encryption and digital signatures, file attachments, and metadata to enable workflows requiring these features.

Portable Document Format

Adobe PDF icon

Filename extension	`.pdf`
Internet media type	`application/pdf`,[1] `application/x-pdf` `application/x-bzpdf` `application/x-gzpdf`
Type code	PDF [1] (including a single trailing space)
Uniform Type Identifier (UTI)	com.adobe.pdf
Magic number	%PDF
Developed by	Adobe Inc. (1991–2008) ISO (2008–)
Initial release	June 15, 1993
Latest release	2.0
Extended to	PDF/A, PDF/E, PDF/UA, PDF/VT, PDF/X
Standard	ISO 32000-2
Open format?	Yes
Website	iso.org/standard/75839.html (https://iso.org/standard/75839.html)

History

The development of PDF began in 1991 when John Warnock wrote a paper for a project then code-named Camelot, in which he proposed the creation of a simplified version of PostScript called Interchange PostScript (IPS).[8] Unlike traditional PostScript, which was tightly focused on rendering print jobs to output devices, IPS would be optimized for displaying pages to any screen and any platform.[8]

Adobe Systems made the PDF specification available free of charge in 1993. In the early years PDF was popular mainly in desktop publishing workflows, and competed with several other formats, including DjVu, Envoy, Common Ground Digital Paper, Farallon Replica and even Adobe's own PostScript format.

PDF was a proprietary format controlled by Adobe until it was released as an open standard on July 1, 2008, and published by the International Organization for Standardization as ISO 32000-1:2008,[9][10] at which time control of the specification passed to an ISO Committee of volunteer industry experts. In 2008, Adobe

application level and send a new print job in the form of an entirely new PostScript file. Thus, any given page in a PostScript file could be accurately rendered only as the cumulative result of executing all preceding commands to draw all previous pages—any of which could affect subsequent pages—plus the commands to draw that particular page, and there was no easy way to bypass that process to skip around to different pages.[19]

Traditionally, to go from PostScript to PDF, a source PostScript file (that is, an executable program) is used as the basis for generating PostScript-like PDF code (see, e.g., Adobe Distiller). This is done by applying standard compiler techniques like loop unrolling, inlining and removing unused branches, resulting in code that is purely declarative and static.[19] The result is then packaged into a container format, together with all necessary dependencies for correct rendering (external files, graphics, or fonts to which the document refers), and compressed. Modern applications write to printer drivers that directly generate PDF rather than going through PostScript first.

As a document format, PDF has several advantages over PostScript:

- PDF contains only static declarative PostScript code that can be processed as data, and does not require a full program interpreter or compiler.[19] This avoids the complexity and security risks of an engine with such a higher complexity level.
- Like Display PostScript, PDF has supported transparent graphics since version 1.4, while standard PostScript does not.
- PDF enforces the rule that the code for any particular page cannot affect any other pages.[19] That rule is strongly recommended for PostScript code too, but has to be implemented explicitly (see, e.g., the Document Structuring Conventions), as PostScript is a full programming language that allows for such greater flexibilities and is not limited to the concepts of pages and documents.
- All data required for rendering is included within the file itself, improving portability.[20]

Its disadvantages are:

- A loss of flexibility, and limitation to a single use case.
- A (sometimes much) larger file size.[21]

PDF since v1.6 supports embedding of interactive 3D documents: 3D drawings can be embedded using U3D or PRC and various other data formats.[22][23][24]

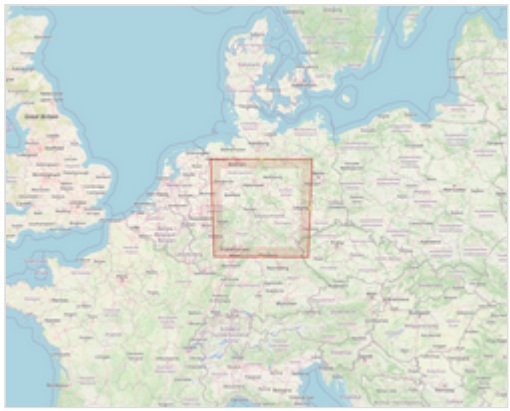
File format

A PDF file is organized using ASCII characters, except for certain elements that may have binary content. The file starts with a header containing a magic number (as a readable string) and the version of the format, for example `%PDF-1.7`. The format is a subset of a COS ("Carousel" Object Structure) format.[25] A COS tree file consists primarily of *objects*, of which there are nine types:[18]

- Boolean values, representing *true* or *false*
- Real numbers
- Integers
- Strings, enclosed within parentheses (`(...)`) or represented as hexadecimal within single angle brackets (`<...>`). Strings may contain 8-bit characters.
- Names, starting with a forward slash (`/`)
- Arrays, ordered collections of objects enclosed within square brackets (`[...]`)

that enables them to be read in a Web browser plugin without waiting for the entire file to download, since all objects required for the first page to display are optimally organized at the start of the file.[29] PDF files may be optimized using Adobe Acrobat software or QPDF.

Page dimensions are not limited by the format itself. However, Adobe Acrobat imposes a limit of 15 million by 15 million inches, or 225,000,000,000,000 square inches (145,161 km^2; 56,047 sq mi), an area slightly larger than Tajikistan.[2]:1129



The maximum size of an Acrobat PDF page, superimposed on a map of Europe.

Imaging model

The basic design of how graphics are represented in PDF is very similar to that of PostScript, except for the use of transparency, which was added in PDF 1.4.

PDF graphics use a device-independent Cartesian coordinate system to describe the surface of a page. A PDF page description can use a matrix to scale, rotate, or skew graphical elements. A key concept in PDF is that of the *graphics state*, which is a collection of graphical parameters that may be changed, saved, and restored by a *page description*. PDF has (as of version 2.0) 25 graphics state properties, of which some of the most important are:

- The *current transformation matrix* (CTM), which determines the coordinate system
- The *clipping path*
- The *color space*
- The *alpha constant*, which is a key component of transparency
- *Black point compensation* control (introduced in PDF 2.0)

Vector graphics

As in PostScript, vector graphics in PDF are constructed with *paths*. Paths are usually composed of lines and cubic Bézier curves, but can also be constructed from the outlines of text. Unlike PostScript, PDF does not allow a single path to mix text outlines with lines and curves. Paths can be stroked, filled, fill then stroked, or used for clipping. Strokes and fills can use any color set in the graphics state, including *patterns*. PDF supports several types of patterns. The simplest is the *tiling pattern* in which a piece of artwork is specified to be drawn repeatedly. This may be a *colored tiling pattern*, with the colors specified in the pattern object, or an *uncolored tiling pattern*, which defers color specification to the time the pattern is drawn. Beginning with PDF 1.3 there is also a *shading pattern*, which draws continuously varying colors. There are seven types of shading patterns of which the simplest are the *axial shading* (Type 2) and *radial shading* (Type 3).

Raster images

Raster images in PDF (called *Image XObjects*) are represented by dictionaries with an associated stream. The dictionary describes the properties of the image, and the stream contains the image data. (Less commonly, small raster images may be embedded directly in a page description as an *inline image*.) Images are typically *filtered* for compression purposes. Image filters supported in PDF include the following general-purpose filters:

- *ASCII85Decode*, a filter used to put the stream into 7-bit ASCII,

and MacRoman encodings are derived from the historical properties of the Windows and Macintosh operating systems, fonts using these encodings work equally well on any platform.) PDF can specify a predefined encoding to use, the font's built-in encoding or provide a lookup table of differences to a predefined or built-in encoding (not recommended with TrueType fonts).[2] The encoding mechanisms in PDF were designed for Type 1 fonts, and the rules for applying them to TrueType fonts are complex.

For large fonts or fonts with non-standard glyphs, the special encodings *Identity-H* (for horizontal writing) and *Identity-V* (for vertical) are used. With such fonts, it is necessary to provide a *ToUnicode* table if semantic information about the characters is to be preserved.

A text document which is scanned to PDF without the text being recognised by optical character recognition (OCR) is an image, with no fonts or text properties.

Transparency

The original imaging model of PDF was *opaque*, similar to PostScript, where each object drawn on the page completely replaced anything previously marked in the same location. In PDF 1.4 the imaging model was extended to allow transparency. When transparency is used, new objects interact with previously marked objects to produce blending effects. The addition of transparency to PDF was done by means of new extensions that were designed to be ignored in products written to PDF 1.3 and earlier specifications. As a result, files that use a small amount of transparency might be viewed acceptably by older viewers, but files making extensive use of transparency could be viewed incorrectly by an older viewer.

The transparency extensions are based on the key concepts of *transparency groups*, *blending modes*, *shape*, and *alpha*. The model is closely aligned with the features of Adobe Illustrator version 9. The blend modes were based on those used by Adobe Photoshop at the time. When the PDF 1.4 specification was published, the formulas for calculating blend modes were kept secret by Adobe. They have since been published.[32]

The concept of a transparency group in PDF specification is independent of existing notions of "group" or "layer" in applications such as Adobe Illustrator. Those groupings reflect logical relationships among objects that are meaningful when editing those objects, but they are not part of the imaging model.

Additional features

Logical structure and accessibility

A **tagged PDF** (see clause 14.8 in ISO 32000) includes document structure and semantics information to enable reliable text extraction and accessibility.[33] Technically speaking, tagged PDF is a stylized use of the format that builds on the logical structure framework introduced in PDF 1.3. Tagged PDF defines a set of standard structure types and attributes that allow page content (text, graphics, and images) to be extracted and reused for other purposes.[34]

Tagged PDF is not required in situations where a PDF file is intended only for print. Since the feature is optional, and since the rules for tagged PDF were relatively vague in ISO 32000-1, support for tagged PDF among consuming devices, including assistive technology (AT), is uneven as of 2021.[35] ISO 32000-2, however, includes an improved discussion of tagged PDF which is anticipated to facilitate further adoption.

An ISO-standardized subset of PDF specifically targeted at accessibility, PDF/UA, was first published in 2012.

For example, Adobe Systems grants permissions to enable additional features in Adobe Reader, using public-key cryptography. Adobe Reader verifies that the signature uses a certificate from an Adobe-authorized certificate authority. Any PDF application can use this same mechanism for its own purposes.[40]

Under specific circumstances including non-patched systems of the receiver, the information the receiver of a digital signed document sees can be manipulated by the sender after the document has been signed by the signer.[41]

PAdES (*PDF Advanced Electronic Signatures*) is a set of restrictions and extensions to PDF and ISO 32000-1[42] making it suitable for advanced electronic signatures. This is published by ETSI as TS 102 778.[43]

File attachments

PDF files can have file attachments which processors may access and open or save to a local filesystem.[44]

Metadata

PDF files can contain two types of metadata.[2] The first is the Document Information Dictionary, a set of key/value fields such as author, title, subject, creation and update dates. This is optional and is referenced from an `Info` key in the trailer of the file. A small set of fields is defined and can be extended with additional text values if required. This method is deprecated in PDF 2.0.

In PDF 1.4, support was added for Metadata Streams, using the Extensible Metadata Platform (XMP) to add XML standards-based extensible metadata as used in other file formats. PDF 2.0 allows metadata to be attached to any object in the document, such as information about embedded illustrations, fonts, and images, as well as the whole document (attaching to the document catalog), using an extensible schema.

PDF documents can also contain display settings, including the page display layout and zoom level in a Viewer Preferences object. Adobe Reader uses these settings to override the user's default settings when opening the document.[45] The free Adobe Reader cannot remove these settings.

Accessibility

PDF files can be created specifically to be accessible to people with disabilities.[46][47][48][49][50] PDF file formats in use as of 2014 can include tags, text equivalents, captions, audio descriptions, and more. Some software can automatically produce tagged PDFs, but this feature is not always enabled by default.[51][52] Leading screen readers, including JAWS, Window-Eyes, Hal, and Kurzweil 1000 and 3000 can read tagged PDFs.[53][54] Moreover, tagged PDFs can be re-flowed and magnified for readers with visual impairments. Adding tags to older PDFs and those that are generated from scanned documents can present some challenges.

One of the significant challenges with PDF accessibility is that PDF documents have three distinct views, which, depending on the document's creation, can be inconsistent with each other. The three views are (i) the physical view, (ii) the tags view, and (iii) the content view. The physical view is displayed and printed (what most people consider a PDF document). The tags view is what screen readers and other assistive technologies use to deliver high-quality navigation and reading experience to users with disabilities. The content view is based on the physical order of objects within the PDF's content stream and may be displayed by software that does not fully support the tags' view, such as the Reflow feature in Adobe's Reader.

of August 2019, XFDF 3.0 is an ISO/IEC standard under the formal name *ISO 19444-1:2019 - Document management — XML Forms Data Format — Part 1: Use of ISO 32000-2 (XFDF 3.0)*.[60] This standard is a normative reference of ISO 32000-2.

PDF

The entire document can be submitted rather than individual fields and values, as was defined in PDF 1.4.

AcroForms can keep form field values in external stand-alone files containing key-value pairs. The external files may use Forms Data Format (FDF) and XML Forms Data Format (XFDF) files.[61][59][62] The usage rights (UR) signatures define rights for import form data files in FDF, XFDF, and text (CSV/TSV) formats, and export form data files in FDF and XFDF formats.[40]

In PDF 1.5, Adobe Systems introduced a proprietary format for forms; Adobe XML Forms Architecture (XFA). Adobe XFA Forms are not compatible with ISO 32000's AcroForms feature, and most PDF processors do not handle XFA content. The XFA specification is referenced from ISO 32000-1/PDF 1.7 as an external proprietary specification and was entirely deprecated from PDF with ISO 32000-2 (PDF 2.0).

Licensing

Anyone may create applications that can read and write PDF files without having to pay royalties to Adobe Systems; Adobe holds patents to PDF, but licenses them for royalty-free use in developing software complying with its PDF specification.[63]

Security

Changes to content

In November 2019, researchers from Ruhr University Bochum and Hackmanit GmbH published attacks on digitally signed PDFs.[64] They showed how to change the visible content in a signed PDF without invalidating the signature in 21 of 22 desktop PDF viewers and 6 of 8 online validation services by abusing implementation flaws. At the same conference, they additionally showed how to exfiltrate the plaintext of encrypted content in PDFs.[65] In 2021, they showed new so-called *shadow attacks* on PDFs that abuse the flexibility of features provided in the specification.[66] An overview of security issues in PDFs regarding denial of service, information disclosure, data manipulation, and arbitrary code execution attacks was presented by Jens Müller.[67][68]

Malware vulnerability

Some popular PDF readers have a history of security vulnerabilities that allows PDF files that have been infected with viruses, Trojans, and other malware to inflict damage. Such PDF files can have hidden JavaScript code that might exploit vulnerabilities in a PDF reader, hidden objects executed when the file that hides them is opened, and, less commonly, a malicious PDF can launch malware.[69]

PDF attachments carrying viruses were first discovered in 2001. The virus, named *OUTLOOK.PDFWorm* or *Peachy*, uses Microsoft Outlook to send itself as an attached Adobe PDF file. It was activated with Adobe Acrobat, but not with Acrobat Reader.[70]

The Apache PDFBox project of the Apache Software Foundation is an open source Java library, licensed under the Apache License, for working with PDF documents.[84]

Printing

Raster image processors (RIPs) are used to convert PDF files into a raster format suitable for imaging onto paper and other media in printers, digital production presses and prepress in a process known as rasterization. RIPs capable of processing PDF directly include the Adobe PDF Print Engine[85] from Adobe Systems and Jaws[86] and the Harlequin RIP from Global Graphics.

In 1993, the Jaws raster image processor from Global Graphics became the first shipping prepress RIP that interpreted PDF natively without conversion to another format. The company released an upgrade to its Harlequin RIP with the same capability in 1997.[87]

Agfa-Gevaert introduced and shipped Apogee, the first prepress workflow system based on PDF, in 1997.

Many commercial offset printers have accepted the submission of press-ready PDF files as a print source, specifically the PDF/X-1a subset and variations of the same.[88] The submission of press-ready PDF files is a replacement for the problematic need for receiving collected native working files.

In 2006, PDF was widely accepted as the standard print job format at the Open Source Development Labs Printing Summit. It is supported as a print job format by the Common Unix Printing System and desktop application projects such as GNOME, KDE, Firefox, Thunderbird, LibreOffice and OpenOffice have switched to emit print jobs in PDF.[89]

Some desktop printers also support direct PDF printing, which can interpret PDF data without external help.

Native display model

PDF was selected as the "native" metafile format for macOS (originally called Mac OS X), replacing the PICT format of the earlier classic Mac OS. The imaging model of the Quartz graphics layer is based on the model common to Display PostScript and PDF, leading to the nickname *Display PDF*. The Preview application can display PDF files, as can version 2.0 and later of the Safari web browser.[90][91] System-level support for PDF allows macOS applications to create PDF documents automatically, provided they support the OS-standard printing architecture. The files are then exported in PDF 1.3 format according to the file header. When taking a screenshot under Mac OS X versions 10.0 through 10.3, the image was also captured as a PDF; later versions save screen captures as a PNG file, though this behavior can be set back to PDF if desired.

Annotation

Adobe Acrobat is one example of proprietary software that allows the user to annotate, highlight, and add notes to already created PDF files. One UNIX application available as free software (under the GNU General Public License) is PDFedit. The freeware Foxit Reader, available for Microsoft Windows, macOS and Linux, allows annotating documents. Tracker Software's PDF-XChange Viewer allows annotations and markups without restrictions in its freeware alternative. Apple's macOS's integrated PDF viewer, Preview, does also enable annotations as does the open-source software Skim, with the latter supporting interaction with LaTeX, SyncTeX, and PDFSync and integration with BibDesk reference management software.

4. "What is a PDF? Portable Document Format | Adobe Acrobat DC" (https://www.adobe.com/acrobat/about-adobe-pdf.html). Adobe Systems Inc. Archived (https://web.archive.org/web/20230130032548/https://www.adobe.com/acrobat/about-adobe-pdf.html) from the original on January 30, 2023. Retrieved January 12, 2023.
5. "ISO 32000-1:2008" (https://web.archive.org/web/20180726064724/http://wwwimages.adobe.com/www.adobe.com/content/dam/acom/en/devnet/pdf/pdfs/PDF32000_2008.pdf) (PDF). Archived from the original (http://wwwimages.adobe.com/www.adobe.com/content/dam/acom/en/devnet/pdf/pdfs/PDF32000_2008.pdf) (PDF) on July 26, 2018.
6. "TC171 SC2 US WG8 PDF – PDF Association" (https://pdfa.org/community/tc171-sc2-us-wg8-pdf/). Retrieved June 25, 2025.
7. "ISO 32000-2 – PDF Association" (https://pdfa.org/resource/iso-32000-2/). Retrieved January 27, 2025.
8. Pfiffner, Pamela (2003). *Inside the Publishing Revolution: The Adobe Story*. Berkeley: Peachpit Press. p. 137. ISBN 0-321-11564-3.
9. "ISO 32000-1:2008 – Document management – Portable document format – Part 1: PDF 1.7" (http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=51502). ISO. July 1, 2008. Archived (https://web.archive.org/web/20101206175751/http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=51502) from the original on December 6, 2010. Retrieved February 21, 2010.
10. Orion, Egan (December 5, 2007). "PDF 1.7 is approved as ISO 32000" (https://web.archive.org/web/20071213004627/http://www.theinquirer.net/gb/inquirer/news/2007/12/05/pdf-approved-iso-32000). *The Inquirer*. Archived from the original (http://www.theinquirer.net/gb/inquirer/news/2007/12/05/pdf-approved-iso-32000) on December 13, 2007. Retrieved December 5, 2007.
11. "Public Patent License, ISO 32000-1: 2008 – PDF 1.7" (https://www.adobe.com/pdf/pdfs/ISO32000-1PublicPatentLicense.pdf) (PDF). Adobe Systems Inc. 2008. Archived (https://web.archive.org/web/20090618144613/http://www.adobe.com/pdf/pdfs/ISO32000-1PublicPatentLicense.pdf) (PDF) from the original on June 18, 2009. Retrieved January 12, 2023.
12. "Guide for the procurement of standards-based ICT – Elements of Good Practice, Against lock-in: building open ICT systems by making better use of standards in public procurement" (https://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=SWD:2013:0224:FIN:EN:PDF). European Commission. June 25, 2013. Archived (https://web.archive.org/web/20200919174545/https://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=SWD:2013:0224:FIN:EN:PDF) from the original on September 19, 2020. Retrieved January 12, 2023. "Example: ISO/IEC 29500, ISO/IEC 26300 and ISO 32000 for document formats reference information that is not accessible by all parties (references to proprietary technology and brand names, incomplete scope or dead web links)."
13. "ISO/TC 171/SC 2/WG 8 N 603 – Meeting Report" (https://web.archive.org/web/20121126013025/http://pdf.editme.com/files/pdfREF-meetings/ISO-TC171-SC2-WG8_N0603_SC2WG8_MtgRept_SLC.pdf) (PDF). *Edit me*. June 27, 2011. Archived from the original (http://pdf.editme.com/files/pdfREF-meetings/ISO-TC171-SC2-WG8_N0603_SC2WG8_MtgRept_SLC.pdf) (PDF) on November 26, 2012 – via Archive. "XFA is not to be ISO standard just yet. The Committee urges Adobe Systems to submit the XFA Specification, XML Forms Architecture (XFA), to ISO for standardization The Committee is concerned about the stability of the XFA specification Part 2 will reference XFA 3.1"
14. "Embedding and publishing interactive, 3-dimensional, scientific figures in Portable Document Format (PDF) files" (https://doi.org/10.1371%2Fjournal.pone.0069446.s001). *PLOS ONE*. **8** (9). 2013. doi:10.1371/journal.pone.0069446.s001 (https://doi.org/10.1371%2Fjournal.pone.0069446.s001). "the implementation of the U3D standard was not complete and proprietary extensions were used."
15. Rosenthol, Leonard (2012). "PDF and Standards" (https://web.archive.org/web/20130902000323/http://cdn.parleys.com/p/5148922a0364bc17fc56c6e5/iSUM2012_00_LRO_presentation.pdf) (PDF). Adobe Systems. Archived from the original (http://cdn.parleys.com/p/5148922a0364bc17fc56c6e5/iSUM2012_00_LRO_presentation.pdf) (PDF) on September 2, 2013. Retrieved October 20, 2013 – via Parleys.

30. Howard, Jacci. "Desktop Publishing: Base 14 Fonts – Definition" (https://web.archive.org/web/20160614134144/http://desktoppub.about.com/od/glossary/g/base14fonts.htm). *About.com Tech*. Archived from the original (http://desktoppub.about.com/od/glossary/g/base14fonts.htm) on June 14, 2016.
31. Merz, Thomas (June 2003). "The PDF Font Aquarium" (https://web.archive.org/web/20110718231502/http://www.planetpdf.com/planetpdf/pdfs/pdf2k/03e/merz_fontaquarium.pdf) (PDF). Archived from the original on July 18, 2011.
32. "PDF Blend Modes Addendum" (https://web.archive.org/web/20111014100004/https://www.adobe.com/content/dam/Adobe/en/devnet/pdf/pdfs/pdf_reference_archives/blend_modes.pdf) (PDF). Archived from the original (https://www.adobe.com/content/dam/Adobe/en/devnet/pdf/pdfs/pdf_reference_archives/blend_modes.pdf) (PDF) on October 14, 2011. Retrieved January 12, 2023.
33. "Tagged PDF Best Practice Guide: Syntax" (https://pdfa.org/wp-content/uploads/2019/06/TaggedPDFBestPracticeGuideSyntax.pdf) (PDF). *pdfa.org*. PDF Association. June 2019. Retrieved June 24, 2024.
34. Johnson, Duff (April 22, 2004). "What is Tagged PDF?" (https://www.talkingpdf.org/what-is-tagged-pdf/). Archived (https://web.archive.org/web/20040807132851/http://www.planetpdf.com/enterprise/article.asp?ContentID=6067) from the original on August 7, 2004.
35. "Is PDF accessible?" (https://www.washington.edu/doit/pdf-accessible?1002=). *DO-IT - Disabilities, Opportunities, Internetworking, and Technology*. University of Washington. October 4, 2022. Archived (https://web.archive.org/web/20230210114239/https://www.washington.edu/doit/pdf-accessible?1002=) from the original on February 10, 2023. Retrieved January 12, 2023.
36. "FreeMyPDF.com – Removes passwords from viewable PDFs" (http://freemypdf.com/). *freemypdf.com*. Archived (https://web.archive.org/web/20210220014724/https://freemypdf.com/) from the original on February 20, 2021. Retrieved June 23, 2009.
37. Kirk, Jeremy (December 4, 2008). "Adobe admits new PDF password protection is weaker" (http://www.macworld.com/article/1137343/pdf.html). *Macworld*. IDG Communications Inc. Archived (https://web.archive.org/web/20170117225255/http://www.macworld.com/article/1137343/pdf.html) from the original on January 17, 2017. Retrieved September 14, 2016.
38. Guignard, Bryan. "How secure is PDF" (https://web.archive.org/web/20051024235303/https://www.cs.cmu.edu/~dst/Adobe/Gallery/PDFsecurity.pdf) (PDF). Carnegie Mellon University. Archived from the original (https://www.cs.cmu.edu/~dst/Adobe/Gallery/PDFsecurity.pdf) (PDF) on October 24, 2005.
39. Merz, Thomas (November 2001). *PDF Security Overview: Strengths and Weaknesses* (https://web.archive.org/web/20101011050457/http://www.planetpdf.com/planetpdf/pdfs/pdf2k/01W/merz_securitykeynote.pdf) (PDF). PDF 2001 conference. Scottsdale/Arizona. Archived from the original on October 11, 2010.
40. Adobe Systems Inc. (July 1, 2008). "Document Management – Portable Document Format – Part 1: PDF 1.7, First Edition" (https://web.archive.org/web/20081203002256/https://www.adobe.com/devnet/acrobat/pdfs/PDF32000_2008.pdf) (PDF). Archived from the original (https://www.adobe.com/devnet/acrobat/pdfs/PDF32000_2008.pdf) (PDF) on December 3, 2008. Retrieved January 12, 2023.
41. "PDF Insecurity Website" (https://pdf-insecurity.org/signature-shadow/shadow-attacks.html). *pdf-insecurity.org*. Archived (https://web.archive.org/web/20230326024850/https://pdf-insecurity.org/signature-shadow/shadow-attacks.html) from the original on March 26, 2023. Retrieved January 12, 2023.
42. "ISO 32000-1:2008 Document management -- Portable document format -- Part 1: PDF 1.7" (http://www.iso.org/iso/catalogue_detail.htm?csnumber=51502). International Organization for Standardization ISO. Archived (https://web.archive.org/web/20170210072051/http://www.iso.org/iso/catalogue_detail.htm?csnumber=51502) from the original on February 10, 2017. Retrieved March 22, 2016.

56. "About PDF forms" (https://web.archive.org/web/20110429032948/http://livedocs.adobe.com/coldfusion/8/htmldocs/help.html?content=formsPDF_02.html). Adobe Systems Inc. Archived from the original (http://livedocs.adobe.com/coldfusion/8/htmldocs/help.html?content=formsPDF_02.html) on April 29, 2011. Retrieved February 19, 2010.
57. Demling, Peter (July 1, 2008). "Convert XFA Form to AcroForm?" (https://community.adobe.com/t5/acrobat-sdk-discussions/convert-xfa-form-to-acroform/td-p/1175365). Archived (https://web.archive.org/web/20230112153241/https://community.adobe.com/t5/acrobat-sdk-discussions/convert-xfa-form-to-acroform/td-p/1175365) from the original on January 12, 2023. Retrieved January 12, 2023.
58. "Migrating from Adobe Acrobat forms to XML forms" (https://web.archive.org/web/20101006151011/http://partners.adobe.com/public/developer/tips/topic_tip2.html). Archived from the original (http://partners.adobe.com/public/developer/tips/topic_tip2.html) on October 6, 2010. Retrieved January 12, 2023.
59. "XML Forms Data Format Specification, version 2" (https://web.archive.org/web/20180730100811/https://www.immagic.com/eLibrary/ARCHIVES/TECH/ADOBE/A070914X.pdf) (PDF). September 2007. Archived from the original (https://www.immagic.com/eLibrary/ARCHIVES/TECH/ADOBE/A070914X.pdf) (PDF) on July 30, 2018. Retrieved February 19, 2010.
60. "ISO 19444-1:2019(en)" (https://www.iso.org/obp/ui/#iso:std:iso:19444:-1:ed-2:v1:en). The International Organization for Standardization. Archived (https://web.archive.org/web/20160617031837/https://www.iso.org/obp/ui/#iso:std:iso:19444:-1:ed-2:v1:en) from the original on June 17, 2016. Retrieved December 3, 2020.
61. Adobe Systems Incorporated (September 20, 2022). "Using Acrobat forms and form data on the web" (https://helpx.adobe.com/acrobat/kb/acrobat-forms-form-data-web.html). Archived (https://web.archive.org/web/20230112153235/https://helpx.adobe.com/acrobat/kb/acrobat-forms-form-data-web.html) from the original on January 12, 2023. Retrieved January 12, 2023.
62. "FDF Data Exchange Specification" (https://web.archive.org/web/20081203041943/https://www.adobe.com/devnet/acrobat/pdfs/fdf_data_exchange.pdf) (PDF). February 8, 2007. Archived from the original (https://www.adobe.com/devnet/acrobat/pdfs/fdf_data_exchange.pdf) (PDF) on December 3, 2008. Retrieved January 12, 2023.
63. "Developer Resources" (https://web.archive.org/web/20160227041714/http://partners.adobe.com/public/developer/support/topic_legal_notices.html). Adobe Systems Inc. Archived from the original (http://partners.adobe.com/public/developer/support/topic_legal_notices.html) on February 27, 2016.
64. Mladenov, Vladislav; Mainka, Christian; Meyer Zu Selhausen, Karsten; Grothe, Martin; Schwenk, Jörg (November 6, 2019). "1 Trillion Dollar Refund: How to Spoof PDF Signatures". *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security* (https://dl.acm.org/doi/10.1145/3319535.3339812). CCS '19. ACM Digital Library, ACM SIGSAC Conference on Computer and Communications Security. pp. 1–14. doi:10.1145/3319535.3339812 (https://doi.org/10.1145%2F3319535.3339812). ISBN 9781450367479. S2CID 199367545 (https://api.semanticscholar.org/CorpusID:199367545). Archived (https://web.archive.org/web/20210426223722/https://dl.acm.org/doi/10.1145/3319535.3339812) from the original on April 26, 2021. Retrieved April 6, 2021.
65. Müller, Jens; Ising, Fabian; Mladenov, Vladislav; Mainka, Christian; Schinzel, Sebastian; Schwenk, Jörg (November 6, 2019). "Practical Decryption exFiltration: Breaking PDF Encryption". *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security* (https://dl.acm.org/doi/10.1145/3319535.3354214). CCS '19. ACM Digital Library, ACM SIGSAC Conference on Computer and Communications Security. pp. 15–29. doi:10.1145/3319535.3354214 (https://doi.org/10.1145%2F3319535.3354214). ISBN 9781450367479. S2CID 207959243 (https://api.semanticscholar.org/CorpusID:207959243). Archived (https://web.archive.org/web/20210426223415/https://dl.acm.org/doi/10.1145/3319535.3354214) from the original on April 26, 2021. Retrieved April 6, 2021.

80. "Goals and Motivations" (https://web.archive.org/web/20140704114405/http://www.gnupdf.org/Goals_and_Motivations). *gnupdf.org*. GNUpdf. November 28, 2007. Archived from the original on July 4, 2014. Retrieved April 2, 2014.
81. Lee, Matt (October 6, 2011). "GNU PDF project leaves FSF High Priority Projects list; mission complete!" (http://www.fsf.org/blogs/community/gnu-pdf-project-leaves-high-priority-projects-list-mission-complete). *fsf.org*. Free Software Foundation. Archived (https://web.archive.org/web/20141228050435/http://www.fsf.org/blogs/community/gnu-pdf-project-leaves-high-priority-projects-list-mission-complete) from the original on December 28, 2014.
82. "Poppler Homepage" (http://poppler.freedesktop.org/). Archived (https://web.archive.org/web/20150108235708/http://poppler.freedesktop.org/) from the original on January 8, 2015. Retrieved January 12, 2023. "Poppler is a PDF rendering library based on the xpdf-3.0 code base."
83. "Xpdf License" (http://cgit.freedesktop.org/poppler/poppler/tree/README-XPDF). Archived (https://archive.today/20130414194348/http://cgit.freedesktop.org/poppler/poppler/tree/README-XPDF) from the original on April 14, 2013. Retrieved January 12, 2023. "Xpdf is licensed under the GNU General Public License (GPL), version 2 or 3."
84. "The Apache PDFBox project- Apache PDFBox 3.0.0 released" (https://pdfbox.apache.org/). August 17, 2023. Archived (https://web.archive.org/web/20230107234923/https://pdfbox.apache.org/) from the original on January 7, 2023. Updated for new releases.
85. "Adobe PDF Print Engine" (https://www.adobe.com/products/pdfprintengine/overview.html). Adobe Systems Inc. Archived (https://web.archive.org/web/20130822034446/http://www.adobe.com/products/pdfprintengine/overview.html) from the original on August 22, 2013. Retrieved August 20, 2014.
86. "Jaws® 3.0 PDF and PostScript RIP SDK" (https://web.archive.org/web/20160305090728/http://globalgraphics.com/products/jaws_rip). *globalgraphics.com*. Archived from the original (http://www.globalgraphics.com/products/jaws_rip/) on March 5, 2016. Retrieved November 26, 2010.
87. "Harlequin MultiRIP" (https://web.archive.org/web/20140209215413/http://www.globalgraphics.com/products/harlequin-multi-rip/). Archived from the original (http://www.globalgraphics.com/products/harlequin-multi-rip) on February 9, 2014. Retrieved March 2, 2014.
88. "Press-Ready PDF Files" (https://web.archive.org/web/20090205151505/http://www.prepressx.com/). Archived from the original on February 5, 2009. Retrieved January 12, 2023. "For anyone interested in having their graphic project commercially printed directly from digital files or PDFs."
89. "PDF as Standard Print Job Format" (https://web.archive.org/web/20091114130224/https://www.linuxfoundation.org/collaborate/workgroups/openprinting/pdf_as_standard_print_job_format). *The Linux Foundation*. Linux Foundation. October 23, 2009. Archived from the original (http://www.linuxfoundation.org/collaborate/workgroups/openprinting/pdf_as_standard_print_job_format) on November 14, 2009. Retrieved January 12, 2023.
90. "View PDFs and images in Preview on Mac" (https://support.apple.com/en-gb/guide/preview/prvw11470/mac). *Apple Support*. Retrieved July 16, 2025.
91. "See a PDF in Safari on Mac" (https://support.apple.com/en-gb/guide/safari/ibrw1090/18.0/mac/15.0). *Apple Support*. Retrieved July 16, 2025.
92. Meuschke, Norman; Jagdale, Apurva; Spinde, Timo; Mitrović, Jelena; Gipp, Bela (2023), Sserwanga, Isaac; Goulding, Anne; Moulaison-Sandy, Heather; Du, Jia Tina (eds.), "A Benchmark of PDF Information Extraction Tools Using a Multi-task and Multi-domain Evaluation Framework for Academic Documents" (https://link.springer.com/10.1007/978-3-031-28032-0_31), *Information for a Better World: Normality, Virtuality, Physicality, Inclusivity*, vol. 13972, Cham: Springer Nature Switzerland, pp. 383–405, arXiv:2303.09957 (https://arxiv.org/abs/2303.09957), doi:10.1007/978-3-031-28032-0_31 (https://doi.org/10.1007%2F978-3-031-28032-0_31), ISBN 978-3-031-28031-3

External links

- PDF Association (https://pdfa.org/) – The PDF Association is the industry association for software developers producing or processing PDF files.
  - PDF Specification Index (https://pdfa.org/resource/pdf-specification-index/) at the PDF Association
  - PDF Cheat Sheets, 2nd edition (https://pdfa.org/resource/pdf-cheat-sheets/) by the PDF Association (last updated October 29, 2024)
  - Sponsored free access to the ISO 32000-2 (PDF 2.0) bundle (https://www.pdfa-inc.org/product/iso-32000-2-pdf-2-0-bundle-sponsored-access/), including the latest core PDF specification and five ISO standardized extensions to the core specification
- Format description of the PDF family (https://www.loc.gov/preservation/digital/formats/fdd/fdd000030.shtml), PDF/A (https://www.loc.gov/preservation/digital/formats/fdd/fdd000318.shtml), PDF/X (https://www.loc.gov/preservation/digital/formats/fdd/fdd000124.shtml) from Library of Congress

Tech notes from Adobe

- Adobe PDF 101: Summary of PDF (https://web.archive.org/web/20101007220449/http://partners.adobe.com/public/developer/tips/topic_tip31.html) at the Wayback Machine (archived 2010-10-07)
- Adobe: PostScript vs. PDF (https://web.archive.org/web/20160413212438/https://www.adobe.com/print/features/psvspdf/) at the Wayback Machine (archived 2016-04-13) – Official introductory comparison of PS, EPS vs. PDF.
- PDF Reference and Adobe Extensions to the PDF Specification (https://web.archive.org/web/20210116133007/https://www.adobe.com/devnet/pdf/pdf_reference.html) at the Wayback Machine (archived 2021-01-16)

Articles

- Portable Document Format: An Introduction for Programmers (1999) (http://preserve.mactech.com/articles/mactech/Vol.15/15.09/PDFIntro/index.html) from MacTech – Introduction to PDF vs. PostScript and PDF internals (up to v1.3)
- John Warnock's 'Camelot' signalled birth of PDF (2002) (https://web.archive.org/web/20190422013101/http://www.planetpdf.com/enterprise/article.asp?ContentID=6519) at the Wayback Machine (archived 2019-04-22) from Planet PDF that describes the paper in which John Warnock outlined the project that created PDF

Videos

- Video: Everything you wanted to know about PDF but was afraid to ask (https://www.youtube.com/watch?v=poc9PVmFzpc) — Recording of a talk by Leonard Rosenthol (Adobe Systems) at TUG 2007
