

PDF

The **Portable Document Format (PDF)** is a file format developed by Adobe in the 1990s 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 was standardized as ISO 32000 in 2008, and no longer requires any royalties for its implementation.^[4]

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) and 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 ^[note 1]
Internet media type	application/pdf, ^[1] application/x-pdf application/x-bzpdf application/x-gzpdf
Type code	'PDF' ^[1] (including a single space)
Uniform Type Identifier (UTI)	com.adobe.pdf
Magic number	%PDF
Developed by	ISO Originally Adobe
Initial release	15 June 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	www.iso.org/ /standard/63534.html (https://www.iso.org/standard/63534.html)

Contents

History and standardization

Technical foundations

PostScript

Technical overview

File structure

Imaging model

Vector graphics

Raster images

Text

Fonts

Standard Type 1 Fonts (Standard 14 Fonts)

Encodings

Transparency

Interactive elements

AcroForms

Forms Data Format (FDF)

XML Forms Data Format (XFDF)

Adobe XML Forms Architecture (XFA)

Logical structure and accessibility

Optional Content Groups (layers)

Security and signatures

Usage rights

- File attachments
- Metadata
- Usage restrictions and monitoring
- Default display settings

Intellectual property

Technical issues

- Accessibility
- Viruses and exploits

Content

Software

- Conversions
- Annotation
- Other

See also

Notes

References

Further reading

External links

History and standardization

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 a variety of formats such as 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,^{[5][6]} at which time control of the specification passed to an ISO Committee of volunteer industry experts. In 2008, Adobe published a Public Patent License to ISO 32000-1 granting royalty-free rights for all patents owned by Adobe that are necessary to make, use, sell, and distribute PDF-compliant implementations.^[7]

PDF 1.7, the sixth edition of the PDF specification that became ISO 32000-1, includes some proprietary technologies defined only by Adobe, such as Adobe XML Forms Architecture (XFA) and JavaScript extension for Acrobat, which are referenced by ISO 32000-1 as normative and indispensable for the full implementation of the ISO 32000-1 specification. These proprietary technologies are not standardized and their specification is published only on Adobe's website.^{[8][9][10][11][12]} Many of them are also not supported by popular third-party implementations of PDF.

On July 28, 2017, ISO 32000-2:2017 (PDF 2.0) was published.^[13] ISO 32000-2 does not include any proprietary technologies as normative references.^[14]

Technical foundations

The PDF combines three technologies:

- A subset of the PostScript page description programming language, for generating the layout and graphics.
- A font-embedding/replacement system to allow fonts to travel with the documents.
- A structured storage system to bundle these elements and any associated content into a single file, with data compression where appropriate.

PostScript

PostScript is a page description language run in an interpreter to generate an image, a process requiring many resources. It can handle graphics and standard features of programming languages such as `if` and `loop` commands. PDF is largely based on PostScript but simplified to remove flow control features like these, while graphics commands such as `lineto` remain.

Often, the PostScript-like PDF code is generated from a source PostScript file. The graphics commands that are output by the PostScript code are collected and tokenized. Any files, graphics, or fonts to which the document refers also are collected. Then, everything is compressed to a single file. Therefore, the entire PostScript world (fonts, layout, measurements) remains intact.

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

- PDF contains tokenized and interpreted results of the PostScript source code, for direct correspondence between changes to items in the PDF page description and changes to the resulting page appearance.
- PDF (from version 1.4) supports graphic transparency; PostScript does not.
- PostScript is an interpreted programming language with an implicit global state, so instructions accompanying the description of one page can affect the appearance of any following page. Therefore, all preceding pages in a PostScript document must be processed to determine the correct appearance of a given page, whereas each page in a PDF document is unaffected by the others. As a result, PDF viewers allow the user to quickly jump to the final pages of a long document, whereas a PostScript viewer needs to process all pages sequentially before being able to display the destination page (unless the optional PostScript Document Structuring Conventions have been carefully complied and included).

Technical overview

File structure

A PDF file is a 7-bit ASCII file, except for certain elements that may have binary content. A PDF file starts with a header containing the magic number and the version of the format such as `%PDF-1.7`. The format is a subset of a COS ("Carousel" Object Structure) format.^[15] A COS tree file consists primarily of *objects*, of which there are eight types:^[16]

- Boolean values, representing *true* or *false*
- Numbers
- Strings, enclosed within parentheses (`((. . .))`), may contain 8-bit characters.
- Names, starting with a forward slash (`/`)
- Arrays, ordered collections of objects enclosed within square brackets (`[. . .]`)
- Dictionaries, collections of objects indexed by Names enclosed within double pointy brackets (`<< . . . >>`)

- Streams, usually containing large amounts of data, which can be compressed and binary, between the `stream` and `endstream` keywords, preceded by a dictionary
- The null object

Furthermore, there may be comments, introduced with the percent sign (%). Comments may contain 8-bit characters.

Objects may be either *direct* (embedded in another object) or *indirect*. Indirect objects are numbered with an *object number* and a *generation number* and defined between the `obj` and `endobj` keywords if residing in the document root. Beginning with PDF version 1.5, indirect objects (except other streams) may also be located in special streams known as *object streams* (marked `/Type /ObjStm`). This technique enables non-stream objects to have standard stream filters applied to them, reduces the size of files that have large numbers of small indirect objects and is especially useful for *Tagged PDF*. Object streams do not support specifying an object's *generation number* (other than 0).

An index table, also called the cross-reference table, is typically located near the end of the file and gives the byte offset of each indirect object from the start of the file.^[17] This design allows for efficient random access to the objects in the file, and also allows for small changes to be made without rewriting the entire file (*incremental update*). Before PDF version 1.5, the table would always be in a special ASCII format, be marked with the `xref` keyword, and follow the main body composed of indirect objects. Version 1.5 introduced optional *cross-reference streams*, which have the form of a standard stream object, possibly with filters applied. Such a stream may be used instead of the ASCII cross-reference table and contains the offsets and other information in binary format. The format is flexible in that it allows for integer width specification (using the `/W` array), so that for example a document not exceeding 64 KiB in size may dedicate only 2 bytes for object offsets.

At the end of a PDF file is a footer containing:

- The `startxref` keyword followed by an offset to the start of the cross-reference table (starting with the `xref` keyword) or the cross-reference stream object
- And the `%%EOF` end-of-file marker.

If a cross-reference stream is not being used, the footer is preceded by the `trailer` keyword followed by a dictionary containing information that would otherwise be contained in the cross-reference stream object's dictionary:

- A reference to the root object of the tree structure, also known as the *catalog* (`/Root`)
- The count of indirect objects in the cross-reference table (`/Size`)
- And other optional information.

There are two layouts to the PDF files: non-linear (not "optimized") and linear ("optimized"). Non-linear PDF files consume less disk space than their linear counterparts, though they are slower to access because portions of the data required to assemble pages of the document are scattered throughout the PDF file. Linear PDF files (also called "optimized" or "web optimized" PDF files) are constructed in a manner that enables them to be read in a Web browser plugin without waiting for the entire file to download, since they are written to disk in a linear (as in page order) fashion.^[18] PDF files may be optimized using Adobe Acrobat software or QPDF.

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 1.6) 24 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

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, 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 pattern of which the simplest are the *axial shade* (Type 2) and *radial shade* (Type 3).

Raster images

Raster images in PDF (called *Image XObjects*) are represented by dictionaries with an associated stream. The dictionary describes properties of the image, and the stream contains the image data. (Less commonly, a raster image 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 general purpose filters

- **ASCII85Decode** a filter used to put the stream into 7-bit ASCII
- **ASCIIHexDecode** similar to ASCII85Decode but less compact
- **FlateDecode** a commonly used filter based on the deflate algorithm defined in RFC 1951 (<http://tools.ietf.org/html/rfc1951>) (deflate is also used in the gzip, PNG, and zip file formats among others); introduced in PDF 1.2; it can use one of two groups of predictor functions for more compact zlib/deflate compression: *Predictor 2* from the TIFF 6.0 specification and predictors (filters) from the PNG specification (RFC 2083 (<https://tools.ietf.org/html/rfc2083>))
- **LZWDecode** a filter based on LZW Compression; it can use one of two groups of predictor functions for more compact LZW compression: *Predictor 2* from the TIFF 6.0 specification and predictors (filters) from the PNG specification
- **RunLengthDecode** a simple compression method for streams with repetitive data using the run-length encoding algorithm and the image-specific filters
- **DCTDecode** a lossy filter based on the JPEG standard
- **CCITTFaxDecode** a lossless bi-level (black/white) filter based on the Group 3 or Group 4 CCITT (ITU-T) fax compression standard defined in ITU-T T.4 and T.6

- **JBIG2Decode** a lossy or lossless bi-level (black/white) filter based on the JBIG2 standard, introduced in PDF 1.4
- **JPXDecode** a lossy or lossless filter based on the JPEG 2000 standard, introduced in PDF 1.5

Normally all image content in a PDF is embedded in the file. But PDF allows image data to be stored in external files by the use of *external streams* or *Alternate Images*. Standardized subsets of PDF, including PDF/A and PDF/X, prohibit these features.

Text

Text in PDF is represented by *text elements* in page content streams. A text element specifies that *characters* should be drawn at certain positions. The characters are specified using the *encoding* of a selected *font resource*.

Fonts

A font object in PDF is a description of a digital typeface. It may either describe the characteristics of a typeface, or it may include an embedded *font file*. The latter case is called an *embedded font* while the former is called an *unembedded font*. The font files that may be embedded are based on widely used standard digital font formats: **Type 1** (and its compressed variant **CFF**), **TrueType**, and (beginning with PDF 1.6) **OpenType**. Additionally PDF supports the **Type 3** variant in which the components of the font are described by PDF graphic operators.

Standard Type 1 Fonts (Standard 14 Fonts)

Fourteen typefaces, known as the *standard 14 fonts*, have a special significance in PDF documents:

- Times (v3) (in regular, italic, bold, and bold italic)
- Courier (in regular, oblique, bold and bold oblique)
- Helvetica (v3) (in regular, oblique, bold and bold oblique)
- Symbol
- Zapf Dingbats

These fonts are sometimes called the *base fourteen fonts*.^[19] These fonts, or suitable substitute fonts with the same metrics, should be available in most PDF readers, but they are not *guaranteed* to be available in the reader, and may only display correctly if the system has them installed.^[20] Fonts may be substituted if they are not embedded in a PDF.

Encodings

Within text strings, characters are shown using *character codes* (integers) that map to glyphs in the current font using an *encoding*. There are a number of predefined encodings, including *WinAnsi*, *MacRoman*, and many encodings for East Asian languages, and a font can have its own built-in encoding. (Although the *WinAnsi* 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).^[21] 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.

Transparency

The original imaging model of PDF was, like PostScript's, *opaque*: 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 the PDF 1.3 and earlier specifications. As a result, files that use a small amount of transparency might view acceptably in older viewers, but files making extensive use of transparency could be viewed incorrectly in an older viewer without warning.

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.^[22]

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.

Interactive elements

PDF files may contain interactive elements such as annotations, form fields, video, 3D and rich media.

Rich Media PDF is a PDF file including interactive content that can be embedded or linked within the file.

Interactive Forms is a mechanism to add forms to the PDF file format.

PDF currently supports two different methods for integrating data and PDF forms. Both formats today coexist in the PDF specification.^{[23][24][25][26]}

- **AcroForms** (also known as **Acrobat forms**), introduced in the PDF 1.2 format specification and included in all later PDF specifications.
- Adobe XML Forms Architecture (XFA) forms, introduced in the PDF 1.5 format specification. Adobe XFA Forms are not compatible with AcroForms.^[27] XFA was deprecated from PDF with PDF 2.0.

AcroForms

AcroForms were introduced in the PDF 1.2 format. AcroForms permit using objects (*e.g.* text boxes, Radio buttons, *etc.*) and some code (*e.g.* JavaScript).

Alongside the standard PDF action types, interactive forms (AcroForms) support submitting, resetting, and importing data. The "submit" action transmits the names and values of selected interactive form fields to a specified uniform resource locator (URL). Interactive form field names and values may be submitted in any of the following formats, (depending on the settings of the action's ExportFormat, SubmitPDF, and XFDF flags):^[23]

- HTML Form format (HTML 4.01 Specification since PDF 1.5; HTML 2.0 since 1.2)
- Forms Data Format (FDF)
- XML Forms Data Format (XFDF) (external XML Forms Data Format Specification, Version 2.0; supported since PDF 1.5; it replaced the "XML" form submission format defined in PDF 1.4)
- 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.^{[28][29][30]} 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.^[23]

Forms Data Format (FDF)

The Forms Data Format (FDF) is based on PDF, it uses the same syntax and has essentially the same file structure, but is much simpler than PDF, since the body of an FDF document consists of only one required object. Forms Data Format is defined in the PDF specification (since PDF 1.2). The Forms Data Format can be used when submitting form data to a server, receiving the response, and incorporating into the interactive form. It can also be used to export form data to stand-alone files that can be imported back into the corresponding PDF interactive form.

XML Forms Data Format (XFDF)

XML Forms Data Format (XFDF) is the XML version of Forms Data Format, but the XFDF implements only a subset of FDF containing forms and annotations. Some entries in the FDF dictionary do not have XFDF equivalents – such as the Status, Encoding, JavaScript, Pages keys, EmbeddedFDFs, Differences and Target. In addition, XFDF does not allow the spawning, or addition, of new pages based on the given data; as can be done when using an FDF file. The XFDF specification is referenced (but not included) in PDF 1.5 specification (and in later versions). It is described separately in *XML Forms Data Format Specification*.^[29] The PDF 1.4 specification allowed form submissions in XML format, but this was replaced by submissions in XFDF format in the PDF 1.5 specification. XFDF conforms to the XML standard.

As of December 2016, XFDF 3.0 is an ISO/IEC standard under the formal name *ISO 19444-1:2016 – Document management – XML Forms Data Format – Part 1: Use of ISO 32000-2 (XFDF 3.0)*.^[34] This standard is a normative reference of ISO 32000-2.

XFDF can be used in the same way as FDF; e.g., form data is submitted to a server, modifications are made, then sent back and the new form data is imported in an interactive form. It can also be used to export form data to stand-alone files that can be imported back into

Forms Data Format (FDF)

Filename extension	.fdf
Internet media type	application/vnd.fdf ^[31]
Type code	'FDF'
Developed by	Adobe Systems
Initial release	1996 (PDF 1.2)
Extended from	PDF
Extended to	XFDF
Standard	ISO 32000-2:2017
Open format?	Yes

XML Forms Data Format (XFDF)

Filename extension	.xfdf
Internet media type	application/vnd.adobe.xfdf ^[32]
Type code	'XFDF'
Developed by	Adobe Systems
Initial release	July 2003 (referenced in PDF 1.5)
Latest release	3.0 (August 2009)
Extended from	PDF, FDF, XML
Standard	ISO 19444-1 ^[33]
Website	XFDF 3.0 specification (https://web.archive.org/web/20160408204348/https://partners.adobe.com/public/developer/en/xml/XFDF_Spec_3.0.pdf)

the corresponding PDF interactive form.

Adobe XML Forms Architecture (XFA)

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).

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. 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.^[35]

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 amongst consuming devices, including assistive technology (AT), is uneven at this time.^[36] 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.

Optional Content Groups (layers)

With the introduction of PDF version 1.5 (2003) came the concept of Layers. Layers, or as they are more formally known Optional Content Groups (OCGs), refer to sections of content in a PDF document that can be selectively viewed or hidden by document authors or consumers. This capability is useful in CAD drawings, layered artwork, maps, multi-language documents etc. Basically, it consists of an Optional Content Properties Dictionary added to the document root. This dictionary contains an array of Optional Content Groups (OCGs), each describing a set of information and each of which may be individually displayed or suppressed, plus a set of Optional Content Configuration Dictionaries, which give the status (Displayed or Suppressed) of the given OCGs.

Security and signatures

A PDF file may be encrypted for security, or digitally signed for authentication.

The standard security provided by Acrobat PDF consists of two different methods and two different passwords: a *user password*, which encrypts the file and prevents opening, and an *owner password*, which specifies operations that should be restricted even when the document is decrypted, which can include modifying, printing, or copying text and graphics out of the document, or adding or modifying text notes and AcroForm fields. The user password encrypts the file, while the owner password does not, instead relying on client software to respect these restrictions. An owner password can easily be removed by software, including some free online services.^[37] Thus, the use restrictions that a document author places on a PDF document are not secure, and cannot be assured once the file is distributed; this warning is displayed when applying such restrictions using Adobe Acrobat software to create or edit PDF files.

Even without removing the password, most freeware or open source PDF readers ignore the permission "protections" and allow the user to print or make copy of excerpts of the text as if the document were not limited by password protection.^{[38][39][40]}

Usage rights

Beginning with PDF 1.5, Usage rights (UR) signatures are used to enable additional interactive features that are not available by default in a particular PDF viewer application. The signature is used to validate that the permissions have been granted by a bona fide granting authority. For example, it can be used to allow a user:^[23]

- To save the PDF document along with modified form and/or annotation data
- Import form data files in FDF, XFDF, and text (CSV/TSV) formats
- Export form data files in FDF and XFDF formats
- Submit form data
- Instantiate new pages from named page templates
- Apply a digital signature to existing digital signature form field
- Create, delete, modify, copy, import, and export annotations

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.^[23]

File attachments

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

Metadata

PDF files can contain two types of metadata.^[41] The first is the Document Information Dictionary, a set of key/value fields such as author, title, subject, creation and update dates. This is stored in the optional Info 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. This allows metadata to be attached to any stream in the document, such as information about embedded illustrations, as well as the whole document (attaching to the document catalog), using an extensible schema.

Usage restrictions and monitoring

PDFs may be encrypted so that a password is needed to view or edit the contents. PDF 2.0 defines 256-bit AES encryption as standard for PDF 2.0 files. The PDF Reference also defines ways that third parties can define their own encryption systems for PDF.

PDF files may be digitally signed; complete details on implementing digital signatures in PDF is provided in ISO 32000-2.

PDF files may also contain embedded DRM restrictions that provide further controls that limit copying, editing or printing. These restrictions depend on the reader software to obey them, so the security they provide is limited.

Default display settings

PDF documents can contain display settings, including the page display layout and zoom level. Adobe Reader uses these settings to override the user's default settings when opening the document.^[42] The free Adobe Reader cannot remove these settings.

Intellectual property

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.^[43]

Technical issues

Accessibility

PDF files can be created specifically to be accessible for people with disabilities.^{[44][45][46][47][48]} 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.^{[49][50]} Leading screen readers, including JAWS, Window-Eyes, Hal, and Kurzweil 1000 and 3000 can read tagged PDF.^{[51][52]} 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 a 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.

PDF/UA, the International Standard for accessible PDF based on ISO 32000-1 was first published as ISO 14289-1 in 2012, and establishes normative language for accessible PDF technology.

Viruses and exploits

PDF attachments carrying viruses were first discovered in 2001. The virus, named *OUTLOOK.PDFWorm* or *Peachy*, uses Microsoft Outlook to send itself as an attachment to an Adobe PDF file. It was activated with Adobe Acrobat, but not with Acrobat Reader.^[53]

From time to time, new vulnerabilities are discovered in various versions of Adobe Reader,^[54] prompting the company to issue security fixes. Other PDF readers are also susceptible. One aggravating factor is that a PDF reader can be configured to start automatically if a web page has an embedded PDF file, providing a

vector for attack. If a malicious web page contains an infected PDF file that takes advantage of a vulnerability in the PDF reader, the system may be compromised even if the browser is secure. Some of these vulnerabilities are a result of the PDF standard allowing PDF documents to be scripted with JavaScript. Disabling JavaScript execution in the PDF reader can help mitigate such future exploits, although it does not protect against exploits in other parts of the PDF viewing software. Security experts say that JavaScript is not essential for a PDF reader, and that the security benefit that comes from disabling JavaScript outweighs any compatibility issues caused.^[55] One way of avoiding PDF file exploits is to have a local or web service convert files to another format before viewing.

On March 30, 2010 security researcher Didier Stevens reported an Adobe Reader and Foxit Reader exploit that runs a malicious executable if the user allows it to launch when asked.^[56]

Content

A PDF file is often a combination of vector graphics, text, and bitmap graphics. The basic types of content in a PDF are:

- Text stored as content streams (i.e., not encoded in plain text)
- Vector graphics for illustrations and designs that consist of shapes and lines
- Raster graphics for photographs and other types of image
- Multimedia objects in the document

In later PDF revisions, a PDF document can also support links (inside document or web page), forms, JavaScript (initially available as plugin for Acrobat 3.0), or any other types of embedded contents that can be handled using plug-ins.

PDF 1.6 supports interactive 3D documents embedded in the PDF – 3D drawings can be embedded using U3D or PRC and various other data formats.^{[57][58][59]}

Two PDF files that look similar on a computer screen may be of very different sizes. For example, a high-resolution raster image takes more space than a low-resolution one. Typically higher resolution is needed for printing documents than for displaying them on screen. Other things that may increase the size of a file is embedding full fonts, especially for Asiatic scripts, and storing text as graphics.

Software

PDF viewers are generally provided free of charge, and many versions are available from a variety of sources.

There are many software options for creating PDFs, including the PDF printing capabilities built into macOS, iOS,^[60] and most Linux distributions, LibreOffice, Microsoft Office 2007 (if updated to SP2) and later,^[61] WordPerfect 9, Scribus, numerous PDF print drivers for Microsoft Windows, the pdfTeX typesetting system, the DocBook PDF tools, applications developed around Ghostscript and Adobe Acrobat itself as well as Adobe InDesign, Adobe FrameMaker, Adobe Illustrator, Adobe Photoshop. Google's online office suite Google Docs also allows for uploading and saving to PDF.

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 rasterisation. RIPs capable of processing PDF directly include the Adobe PDF Print Engine^[62] from Adobe Systems and Jaws^[63] and the Harlequin RIP from Global Graphics.

Conversions

Documents saved in PDF (Portable Document Format, Portable Document Format for Archiving, Forms Data Format, any printable document) could be converted back and forth from many other formats such as:^[64]

- Microsoft Office formats,
- Rich Text Format,
- Open office,
- WordPerfect,
- Microsoft Windows Bitmap,
- Windows Metafile,
- Enhanced Metafile,
- Graphics Interchange Format,
- Microsoft HD Photo File,
- JPEG File Interchange,
- JPEG 2000 File Format and Code Stream,
- Portable Network Graphics,
- Tagged Image File,
- HyperText Markup Language,
- MIME Encapsulation of Aggregate HTML Documents,
- Scalable Vector Graphics,
- Microsoft Outlook, Text,
- XML Paper Specification,
- Extensible Markup Language,
- Silverlight XPS Document,
- Extensible Application Markup Language

Companies such as Adobe Acrobat provide SDK libraries for developers to add and create PDF features in any software. Besides the Adobe PDF Library, there are other development libraries available.

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. Freeware Qiqqa can create an annotation report that summarizes all the annotations and notes one has made across their library of PDFs. The Text Verification Tool exports differences in documents as annotations and markups.

There are also web annotation systems that support annotation in pdf and other documents formats. In cases where PDFs are expected to have all of the functionality of paper documents, ink annotation is required.

Other

Examples of PDF software as online services including [Scribd](#) for viewing and storing, [Pdfvue](#) for online editing, and [Thinkfree](#), [Zamzar](#) for conversion.

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 their [Harlequin RIP](#) with the same capability in 1997.^[65]

[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.^[66] The submission of press-ready PDF files are a replacement for the problematic need for receiving collected native working files.

PDF was selected as the "native" [metafile](#) format for [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. System-level support for PDF allows [Mac OS X](#) 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 behaviour can be set back to PDF if desired.

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.^[67]

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

The [Free Software Foundation](#) once thought of as one of their [high priority projects](#) to be "developing a free, high-quality and fully functional set of libraries and programs that implement the PDF file format and associated technologies to the ISO 32000 standard."^{[68][69]} In 2011, however, the **GNU PDF** project was removed from the list of "high priority projects" due to the maturation of the [Poppler library](#),^[70] which has enjoyed wider use in applications such as [Evince](#) with the [GNOME](#) desktop environment. Poppler is based on [Xpdf](#)^{[71][72]} code base. There are also commercial development libraries available as listed in [List of PDF software](#).

The [Apache PDFBox](#) project of the [Apache Software Foundation](#) is an open source Java library for working with PDF documents. PDFBox is licensed under the [Apache License](#).^[73]

See also

- [Open XML Paper Specification](#)
- [Comparison of OpenXPS and PDF](#)
- [DjVu](#)
- [PAdES](#), PDF Advanced Electronic Signature
- [pdfrecycle](#)
- [Web document](#)
- [XSL Formatting Objects](#)

Notes

1. Before Adobe Acrobat and Portable Document Format, file extension .pdf was used by a word processor named WordStar, which used this extension for *printer definition files*.

References

1. *The application/pdf Media Type* (<http://tools.ietf.org/html/rfc8118>), 2017, RFC 8118 (<https://tools.ietf.org/html/rfc8118>)
2. Adobe Systems Incorporated, *PDF Reference, Sixth edition, version 1.23 (53 MB)* (https://www.adobe.com/devnet/acrobat/pdfs/pdf_reference_1-7.pdf), Nov 2006, p. 33.
3. "The Camelot Project" (http://www.planetpdf.com/planetpdf/pdfs/warnock_camelot.pdf) (PDF).
4. "ISO 32000-1:2008" (http://www.images.adobe.com/www.adobe.com/content/dam/acom/en/devnet/pdf/pdfs/PDF32000_2008.pdf) (PDF).
5. "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.org. 2008-07-01. Retrieved 2010-02-21.
6. Orion, Egan (2007-12-05). "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*. 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 2007-12-05.
7. Adobe Systems Incorporated (2008), *Public Patent License, ISO 32000-1: 2008 – PDF 1.7* (<https://www.adobe.com/pdf/pdfs/ISO32000-1PublicPatentLicense.pdf>) (PDF), retrieved 2011-07-06
8. "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" (<http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=SWD:2013:0224:FIN:EN:PDF>). European Commission. 2013-06-25. Retrieved 2013-10-20. "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)."
9. *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_MtgReport_SLC.pdf) (PDF), 2011-06-27, "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"
10. "Embedding and publishing interactive, 3-dimensional, scientific figures in Portable Document Format (PDF) files" (<http://www.plosone.org/article/fetchSingleRepresentation.action?uri=info:doi/10.1371/journal.pone.0069446.s001>). Retrieved 2013-10-20. "the implementation of the U3D standard was not complete and proprietary extensions were used."
11. Leonard Rosenthal, Adobe Systems (2012). "PDF and Standards" (https://web.archive.org/web/20130902000323/http://cdn.parleys.com/p/5148922a0364bc17fc56c6e5/iSUM2012_00_LRO_presentation.pdf) (PDF). Archived from the original (http://cdn.parleys.com/p/5148922a0364bc17fc56c6e5/iSUM2012_00_LRO_presentation.pdf) (PDF) on 2013-09-02. Retrieved 2013-10-20.
12. Duff Johnson (2010-06-10), *Is PDF an open standard? - Adobe Reader is the de facto Standard, not PDF* (https://web.archive.org/web/20160416014955/http://www.planetpdf.com/enterprise/article.asp?ContentID=Is_PDF_an_open_standard&page=1), retrieved 2014-01-19
13. "The worldwide standard for electronic documents is evolving" (<https://www.iso.org/news/ref2199.html>). *www.iso.org*. Retrieved 2018-06-29.

14. "ISO 32000-2 – Document management -- Portable document format -- Part 2: PDF 2.0" (<http://www.iso.org/standard/63534.html>). *www.iso.org*. Retrieved 2017-07-28.
15. Pravetz, Jim. "In Defense of COS, or Why I Love JSON and Hate XML" (<http://jimpravetz.com/blog/2012/12/in-defense-of-cos/>). *jimpravetz.com*.
16. Adobe Systems, PDF Reference, p. 51.
17. Adobe Systems, PDF Reference, pp. 39–40.
18. "Adobe Developer Connection: PDF Reference and Adobe Extensions to the PDF Specification" (https://www.adobe.com/devnet/pdf/pdf_reference.html). Adobe Systems. Retrieved 2010-12-13.
19. 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.
20. The PDF Font Aquarium (http://www.planetpdf.com/planetpdf/pdfs/pdf2k/03e/merz_fontaquarium.pdf)
21. "PDF Reference Sixth Edition, version 1.7, table 5.11" (https://www.adobe.com/devnet/acrobat/pdfs/pdf_reference_1-7.pdf) (PDF).
22. PDF Blend Modes Addendum (https://www.adobe.com/content/dam/Adobe/en/devnet/pdf/pdfs/pdf_reference_archives/blend_modes.pdf)
23. Adobe Systems Incorporated (2008-07-01), *Document Management – Portable Document Format – Part 1: PDF 1.7, First Edition* (https://www.adobe.com/devnet/acrobat/pdfs/PDF3200_0_2008.pdf) (PDF), retrieved 2010-02-19
24. "Gnu PDF – PDF Knowledge – Forms Data Format" (https://web.archive.org/web/20130101054615/http://www.gnupdf.org/Forms_Data_Format). Archived from the original (http://gnupdf.org/Forms_Data_Format) on 2013-01-01. Retrieved 2010-02-19.
25. "About PDF forms" (https://web.archive.org/web/20110429032948/http://livedocs.adobe.com/coldfusion/8/htmldocs/help.html?content=formsPDF_02.html). Archived from the original (http://livedocs.adobe.com/coldfusion/8/htmldocs/help.html?content=formsPDF_02.html) on 2011-04-29. Retrieved 2010-02-19.
26. Demling, Peter (July 1, 2008). "Convert XFA Form to AcroForm?" (<http://forums.adobe.com/thread/301733>). Retrieved 2010-02-19.
27. "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 2010-10-06. Retrieved 2010-02-22.
28. Adobe Systems Incorporated (2007-10-15). "Using Acrobat forms and form data on the web" (<http://kb2.adobe.com/cps/325/325874.html>). Retrieved 2010-02-19.
29. *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 2009-12-23, retrieved 2010-02-19
30. *FDX Data Exchange Specification* (https://www.adobe.com/devnet/acrobat/pdfs/fdf_data_exchange.pdf) (PDF), 2007-02-08, retrieved 2010-02-19
31. *IANA Application Media Types – vnd.fdf* (<http://www.iana.org/assignments/media-types/application/>), retrieved 2010-02-22
32. *IANA Application Media Types – Vendor Tree – vnd.adobe.xfdf* (<http://www.iana.org/assignments/media-types/application/vnd.adobe.xfdf>), retrieved 2010-02-22
33. *ISO/CD 19444-1 – Document management – XML Forms Data Format – Part 1: Use of ISO 32000-2 (XPDF 3.0)* (http://www.iso.org/iso/home/store/catalogue_ics/catalogue_detail_ics.htm?ics1=35&ics2=240&ics3=30&csnumber=64911), retrieved 2017-05-28

34. "ISO 19444-1:2016 – Document management -- XML Forms Data Format -- Part 1: Use of ISO 32000-2 (XFDf 3.0)" (<https://www.iso.org/standard/64911.html>). *www.iso.org*. Retrieved 2017-02-28.
35. Duff Johnson, April 22, 2004 [What is Tagged PDF?](https://web.archive.org/web/20040807132851/http://www.planetpdf.com/enterprise/article.asp?ContentID=6067) (<https://web.archive.org/web/20040807132851/http://www.planetpdf.com/enterprise/article.asp?ContentID=6067>)
36. "Is PDF accessible?" (<http://www.washington.edu/doi/Stem/articles?1002>). *washington.edu*.
37. "FreeMyPDF.com – Removes passwords from viewable PDFs" (<http://freemypdf.com/>). *freemypdf.com*.
38. Jeremy Kirk (December 5, 2008). "Adobe admits new PDF password protection is weaker" (<http://www.macworld.com/article/1137343/pdf.html>). *Macworld*.
39. Bryan Guignard. "How secure is PDF" (<https://www.cs.cmu.edu/~dst/Adobe/Gallery/PDFsecurity.pdf>) (PDF).
40. "PDF Security Overview: Strengths and Weaknesses" (http://www.planetpdf.com/planetpdf/pdfs/pdf2k/01W/merz_securitykeynote.pdf) (PDF).
41. Adobe PDF reference version 1.7 (https://www.adobe.com/devnet/acrobat/pdfs/pdf_reference_1-7.pdf), section 10.2
42. "Getting Familiar with Adobe Reader > Understanding Preferences" (<http://www.adobepress.com/articles/article.asp?p=412914>). Retrieved 2009-04-22.
43. "Developer Resources" (https://web.archive.org/web/20160227041714/http://partners.adobe.com/public/developer/support/topic_legal_notices.html). *adobe.com*. Archived from the original (http://partners.adobe.com/public/developer/support/topic_legal_notices.html) on 2016-02-27.
44. "PDF Accessibility" (<http://www.webaim.org/techniques/acrobat/>). WebAIM. Retrieved 2010-04-24.
45. Joe Clark (2005-08-22). "Facts and Opinions About PDF Accessibility" (http://www.alistapart.com/articles/pdf_accessibility). Retrieved 2010-04-24.
46. "Accessibility and PDF documents" (<https://web.archive.org/web/20100427062242/http://wac.osu.edu/pdf/>). Web Accessibility Center. Archived from the original (<http://wac.osu.edu/pdf/>) on 2010-04-27. Retrieved 2010-04-24.
47. "PDF Accessibility Standards v1.2" (https://www.bbc.co.uk/guidelines/futuremedia/accessibility/accessible_pdf.shtml). Retrieved 2010-04-24.
48. *PDF Accessibility* (https://web.archive.org/web/20100527215445/http://www.csus.edu/training/handouts/workshops/creating_accessible_pdfs.pdf) (PDF), California State University, archived from the original (http://www.csus.edu/training/handouts/workshops/creating_accessible_pdfs.pdf) (PDF) on 2010-05-27, retrieved 2010-04-24
49. *LibreOffice Help – Export as PDF* (http://help.libreoffice.org/Common/Export_as_PDF#PDF.2FA-1a), retrieved 2012-09-22
50. *Exporting PDF/A for long-term archiving* (<http://www.ooninja.com/2008/01/generating-pdf-a-for-long-term-archiving.html>), 2008-01-11
51. Biersdorfer, J.D. (2009-04-10). "Tip of the Week: Adobe Reader's 'Read Aloud' Feature" (<http://gadgetwise.blogs.nytimes.com/2009/04/10/tip-of-the-week-adobe-readers-read-aloud-feature/>). *The New York Times*. Retrieved 2010-04-24.
52. *Accessing PDF documents with assistive technology: A screen reader user's guide* (<https://www.adobe.com/accessibility/pdfs/accessing-pdf-sr.pdf>) (PDF), Adobe, retrieved 2010-04-24
53. Adobe Forums, Announcement: PDF Attachment Virus "Peachy" (<https://forums.adobe.com/thread/302989>), 15 August 2001.
54. "Security bulletins and advisories" (<https://www.adobe.com/support/security/#readerwin>). Adobe. Retrieved 2010-02-21.
55. Steve Gibson – SecurityNow Podcast (<https://www.grc.com/sn/sn-187.txt>)

56. "Malicious PDFs Execute Code Without a Vulnerability" (https://web.archive.org/web/20100404034752/http://blogs.pcmag.com/securitywatch/2010/03/malicious_pdfs_execute_code_wi.php). PCMAG. Archived from the original (http://blogs.pcmag.com/securitywatch/2010/03/malicious_pdfs_execute_code_wi.php) on 4 April 2010.
57. "3D supported formats" (<https://web.archive.org/web/20100212072951/http://www.adobe.com/manufacturing/resources/3dformats/>). Adobe. 2009-07-14. Archived from the original (<https://www.adobe.com/manufacturing/resources/3dformats/>) on 2010-02-12. Retrieved 2010-02-21.
58. "Supported file formats in Acrobat and Reader" (https://helpx.adobe.com/acrobat/kb/supported-file-formats-acrobat-reader.html#main_2D_and_3D_formats_Acrobat_9_Pro_Extended_Adobe_3D_Reviewer). *helpx.adobe.com*. Retrieved 2020-03-22.
59. "JavaScript for Acrobat 3D | Adobe Acrobat Developer Center" (https://www.adobe.com/devnet/acrobat/javascript_acrobt_3d.html). *www.adobe.com*. Retrieved 2020-03-22.
60. Pathak, Khamosh (October 7, 2017). "How to Create a PDF from Web Page on iPhone and iPad in iOS 11" (<http://www.iphonehacks.com/2017/10/how-to-create-pdf-web-page-safari-iphone-ipad-ios-11.html>). *iPhone Hacks*. Retrieved February 2, 2018.
61. "Description of 2007 Microsoft Office Suite Service Pack 2 (SP2)" (<http://support.microsoft.com/kb/953195>). Microsoft. Archived (<https://web.archive.org/web/20090429212434/http://support.microsoft.com/kb/953195>) from the original on 29 April 2009. Retrieved 2009-05-09.
62. "Adobe PDF Print Engine" (<https://www.adobe.com/products/pdfprintengine/overview.html>). *adobe.com*.
63. "Jaws® 3.0 PDF and PostScript RIP SDK" (http://www.globalgraphics.com/products/jaws_rip/). *globalgraphics.com*.
64. "Supported File Formats" (<https://www.pdftron.com/pdf-sdk/supported-file-formats>). PDFTron Systems.
65. "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 2014-02-09. Retrieved 2014-03-02.
66. Press-Ready PDF Files (<https://web.archive.org/web/20090205151505/http://www.prepressx.com/>) "For anyone interested in having their graphic project commercially printed directly from digital files or PDFs." (last checked on 2009-02-10).
67. "PDF as Standard Print Job Format" (http://www.linuxfoundation.org/collaborate/workgroups/openprinting/pdf_as_standard_print_job_format). *The Linux Foundation*. Linux Foundation. Retrieved 21 June 2016.
68. On 2014-04-02, a note dated 2009-02-10 referred to [Current FSF High Priority Free Software Projects](http://www.fsf.org/campaigns/priority.html) (<http://www.fsf.org/campaigns/priority.html>) as a source. Content of the latter page, however, changes over time.
69. GNUpdf contributors (2007-11-28). "Goals and Motivations" (http://gnupdf.org/Goals_and_Motivations). *gnupdf.org*. GNUpdf. Retrieved 2014-04-02.
70. Lee, Matt (2011-10-06). "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. Retrieved 2014-04-02.
71. Poppler homepage (<http://poppler.freedesktop.org/>) "Poppler is a PDF rendering library based on the xpdf-3.0 code base." (last checked on 2009-02-10)
72. Xpdf license (<http://cgkit.freedesktop.org/poppler/poppler/tree/README-XPDF>) "Xpdf is licensed under the GNU General Public License (GPL), version 2 or 3." (last checked on 2012-09-23).
73. The Apache PDFBox project (<http://pdfbox.apache.org/>) . Retrieved 2009-09-19.

Further reading

- Hardy, M. R. B.; Brailsford, D. F. (2002). "Mapping and displaying structural transformations between XML and PDF". *Proceedings of the 2002 ACM symposium on Document engineering – DocEng '02* (<http://www.cs.nott.ac.uk/~dfb/Publications/Download/2002/Hardy02.pdf>) (PDF). Proceedings of the 2002 ACM symposium on Document engineering. pp. 95–102. doi:10.1145/585058.585077 (<https://doi.org/10.1145%2F585058.585077>). ISBN 1-58113-594-7.

Standards

- PDF 2.0 "ISO 32000-2:2017(en), Document management — Portable document format — Part 2: PDF 2.0" (<https://www.iso.org/standard/63534.html>). *www.iso.org*. Retrieved 2019-01-31.
- PDF 1.7 (https://www.adobe.com/content/dam/acom/en/devnet/acrobat/pdfs/pdf_reference_1-7.pdf) and errata to 1.7 (https://www.adobe.com/content/dam/acom/en/devnet/pdf/pdf_reference_archive/pdf_17_errata.pdf)
- PDF 1.6 (https://www.adobe.com/content/dam/acom/en/devnet/pdf/pdf_reference_archive/PDFReference16.pdf) (ISBN 0-321-30474-8) and errata to 1.6 (https://www.adobe.com/content/dam/acom/en/devnet/pdf/pdf_reference_archive/PDF16Errata.pdf)
- PDF 1.5 (https://www.adobe.com/content/dam/acom/en/devnet/pdf/pdf_reference_archive/PDFReference15_v6.pdf) and errata to 1.5 (https://www.adobe.com/content/dam/acom/en/devnet/pdf/pdf_reference_archive/errata.txt)
- PDF 1.4 (https://www.adobe.com/content/dam/acom/en/devnet/pdf/pdf_reference_archive/PDFReference.pdf) (ISBN 0-201-75839-3) and errata to 1.4 (https://www.adobe.com/content/dam/acom/en/devnet/pdf/pdf_reference_archive/PDF14errata.txt)
- PDF 1.3 (<https://www.adobe.com/content/dam/acom/en/devnet/pdfs/PDFReference13.pdf>) (ISBN 0-201-61588-6) and errata to 1.3 (<https://www.adobe.com/content/dam/acom/en/devnet/pdfs/PDFerrata.txt>)

External links

- PDF Association (<https://www.pdfa.org/>) – The PDF Association is the industry association for software developers producing or processing PDF files.
- Adobe PDF 101: Summary of PDF (https://web.archive.org/web/20101007220449/http://partners.adobe.com/public/developer/tips/topic_tip31.html)
- Adobe: PostScript vs. PDF (<https://web.archive.org/web/20160413212438/https://www.adobe.com/print/features/psvspdf/>) – Official introductory comparison of PS, EPS vs. PDF.
- PDF Standards....transitioning the PDF specification from a de facto standard to a de jure standard (<https://web.archive.org/web/20110424013530/http://www.aiim.org/Resources/Archive/Magazine/2007-Jul-Aug/33448>) at the [Wayback Machine](#) (archived April 24, 2011) – Information about PDF/E and PDF/UA specification for accessible documents file format (archived by [The Wayback Machine](#))
- PDF/A-1 ISO standard (<https://www.iso.org/standard/38920.html>) published by the International Organization for Standardization (with corrigenda)
- PDF Reference and Adobe Extensions to the PDF Specification (https://www.adobe.com/devnet/pdf/pdf_reference.html)
- Portable Document Format: An Introduction for Programmers (<http://www.mactech.com/articles/mactech/Vol.15/15.09/PDFIntro/>) – Introduction to PDF vs. PostScript and PDF internals (up to v1.3)
- The Camelot Paper (<https://web.archive.org/web/20190422013101/http://www.planetpdf.com/enterprise/article.asp?ContentID=6519>) – the paper in which John Warnock outlined the project that created PDF

- [Everything you wanted to know about PDF but were afraid to ask \(https://web.archive.org/web/20160118105015/http://river-valley.zeeba.tv/everything-you-wanted-to-know-about-pdf-but-were-afraid-to-ask/\)](https://web.archive.org/web/20160118105015/http://river-valley.zeeba.tv/everything-you-wanted-to-know-about-pdf-but-were-afraid-to-ask/) – recording of talk by Leonard Rosenthal (<https://www.youtube.com/watch?v=poc9PvmFzpc>) (45 mins) (Adobe Systems) at TUG 2007
-

Retrieved from "<https://en.wikipedia.org/w/index.php?title=PDF&oldid=957966377>"

This page was last edited on 21 May 2020, at 09:50 (UTC).

Text is available under the [Creative Commons Attribution-ShareAlike License](#); additional terms may apply. By using this site, you agree to the [Terms of Use](#) and [Privacy Policy](#). Wikipedia® is a registered trademark of the [Wikimedia Foundation, Inc.](#), a non-profit organization.