



- [Main page](#)
- [Contents](#)
- [Featured content](#)
- [Current events](#)
- [Random article](#)
- [Donate to Wikipedia](#)

- ▼ [Interaction](#)
 - [Help](#)
 - [About Wikipedia](#)
 - [Community portal](#)
 - [Recent changes](#)
 - [Contact Wikipedia](#)

- ▶ [Toolbox](#)
- ▶ [Print/export](#)

- ▼ [Languages](#)
 - [Alemannisch](#)
 - [العربية](#)
 - [বাংলা](#)
 - [Беларуская \(тарашкевіца\)](#)
 - [Български](#)
 - [Català](#)
 - [Česky](#)
 - [Cymraeg](#)
 - [Dansk](#)
 - [Deutsch](#)
 - [Eesti](#)
 - [Español](#)
 - [Esperanto](#)
 - [Euskara](#)
 - [فارسی](#)
 - [Français](#)
 - [Galego](#)
 - [ગુજરાતી](#)
 - [한국어](#)
 - [हिन्दी](#)
 - [Hrvatski](#)
 - [Bahasa Indonesia](#)
 - [Íslenska](#)

Portable Document Format

From Wikipedia, the free encyclopedia

"PDF" redirects here. For other uses, see [PDF \(disambiguation\)](#).

Portable Document Format (PDF) is an [open standard](#) for document exchange. The [file format](#) created by [Adobe Systems](#) in 1993 is used for representing two-dimensional documents in a manner independent of the application software, hardware, and operating system.^[2] Each PDF file encapsulates a complete description of a fixed-layout 2D document that includes the text, fonts, images, and 2D [vector graphics](#) which compose the documents. Today, 3D drawings can be embedded in PDF documents with Acrobat 3D using [U3D](#) or [PRC](#) and various other data formats.^{[3][4]}

In 1991 [Adobe Systems](#) co-founder [John Warnock](#) outlined a system called "Camelot"^[5] that evolved into the Portable Document Format (PDF).

Originally a [proprietary format](#), PDF was officially released as an [open standard](#) on July 1, 2008, and published by the [International Organization for Standardization](#) as ISO/IEC 32000-1:2008.^{[6][1]}

The ISO 32000-1 allows use of some specifications, which are not standardized (e.g. Adobe XML Forms Architecture^{[7][8]}). ISO 32000-1 does not specify methods for validating the conformance of PDF files or readers.^[1]

Portable Document Format



Filename extension	.pdf
Internet media type	application/pdf application/x-pdf application/x-bzpdf application/x-gzpdf
Type code	'PDF ' (including a single space)
Uniform Type Identifier	com.adobe.pdf
Magic number	%PDF
Developed by	Adobe Systems
Initial release	1993
Latest release	1.7
Standard(s)	ISO/IEC 32000-1:2008 ^[1]
Website	Adobe PDF Reference Archives 

Contents [\[hide\]](#)

- 1 History
 - 1.1 Adobe's PDF specifications
 - 1.1.1 Adobe's versions
 - 1.2 Standardization
 - 1.2.1 Specialized subsets of PDF
 - 1.2.2 Full function PDF
 - 1.2.2.1 PDF 1.7
 - 1.2.2.2 PDF 2.0
- 2 Technical foundations
 - 2.1 PostScript
- 3 Technical overview
 - 3.1 File structure
 - 3.2 Imaging model

Italiano
עברית
Basa Jawa
ಕನ್ನಡ
Kurdî
Latviešu
Lëtzebuergesch
Lietuvių
Lumbaart
Magyar
Македонски
മലയാളം
मराठी
Nederlands
日本語
Norsk (bokmål)
Nederlands
Norsk (nynorsk)
Polski
Português
Română
Русский
Simple English
Slovenčina
Српски / Srpski
Srpskohrvatski /
Српскохрватски
Suomi
Svenska
தமிழ்
Tarandíne
ไทย
Türkçe
Українська
Tiếng Việt
ꠘꠘꠘꠘꠘ
Yorùbá
粵語
中文

3.2.1 Vector graphics
3.2.2 Raster images
3.2.3 Text
 3.2.3.1 Fonts
 3.2.3.2 Encodings
3.2.4 Transparency
3.3 Interactive elements
 3.3.1 AcroForms
 3.3.1.1 Forms Data Format (FDF)
 3.3.1.2 XML Forms Data Format (XFDF)
 3.3.2 Adobe XML Forms Architecture (XFA)
3.4 Logical structure and accessibility
3.5 Security and signatures
 3.5.1 Usage rights
3.6 File attachments
3.7 Metadata
3.8 Subsets
3.9 Mars
4 Technical issues
 4.1 Accessibility
 4.2 Security
 4.3 Usage restrictions and monitoring
 4.4 Default display settings
5 Content
 5.1 Standard Type 1 Fonts
6 Implementations
7 Editing PDFs (structure)
8 Annotating PDFs
9 Other applications and functionalities
10 See also
11 References
12 Further reading
13 External links

History

[[edit](#)]

PDF's adoption in the early days of the format's history was slow.^[9] [Adobe Acrobat](#), Adobe's suite for reading and creating PDF files, was not freely available; early versions of PDF had no support for external hyperlinks, reducing its usefulness on the Internet; the larger size of a PDF document compared to plain text required longer download times over the slower [modems](#) common at the time; and rendering PDF files was slow on the less powerful machines of the day.

Additionally, there were competing formats such as [Envoy](#), [Common Ground Digital Paper](#), [Farallon Replica](#) and even Adobe's own [PostScript](#) format (.ps); in those early years, PDF was popular mainly in [desktop publishing workflows](#). In 1995, AT&T Labs commenced work on another electronic document standard targeted at libraries and archives for preserving their books and documents, [DjVu](#). This standard has evolved into the .djv/ .djvu format, which has had growing success and penetration in the online world for eBooks, catalogs, and image-sharing.

Adobe soon started distributing its Acrobat Reader (now Adobe Reader) program at no cost, and continued supporting the original PDF, which eventually became the *de facto standard* for printable documents on the web (a standard *web document*).

Adobe's PDF specifications

[\[edit\]](#)

Adobe changed the PDF specification several times and continues to develop new specifications with new versions of Adobe Acrobat. There have been nine versions of PDF with corresponding Acrobat releases:^[10]

- (1993) – PDF 1.0 / Acrobat 1.0
- (1994) – PDF 1.1 / Acrobat 2.0
- (1996) – PDF 1.2 / Acrobat 3.0
- (1999) – PDF 1.3 / Acrobat 4.0
- (2001) – PDF 1.4 / Acrobat 5.0
- (2003) – PDF 1.5 / Acrobat 6.0
- (2005) – PDF 1.6 / Acrobat 7.0
- (2006) – PDF 1.7 / Acrobat 8.0
- (2008) – PDF 1.7, Adobe Extension Level 3 / Acrobat 9.0
- (2009) – PDF 1.7, Adobe Extension Level 5 / Acrobat 9.1

The ISO standard ISO 32000-1:2008 is equivalent to Adobe's PDF 1.7. Adobe declared that it is not producing a PDF 1.8 Reference. The future versions of the PDF Specification will be produced by ISO technical committees. However, Adobe published documents specifying what extended features for PDF, beyond ISO 32000-1 (PDF 1.7), are supported in its newly released products. This makes use of the extensibility features of PDF as documented in ISO 32000-1 in Annex E. Adobe declared all extended features in *Adobe Extension Level 3* and *5* have been accepted for a new proposal of ISO 32000-2 (a.k.a. PDF 2.0).^[11]

The specifications for PDF are backward inclusive. The PDF 1.7 specification includes all of the functionality previously documented in the Adobe PDF Specifications for versions 1.0 through 1.6. It should be noted that where Adobe removed certain features of PDF from their standard, they too are not contained in ISO 32000-1.^[1]

PDF documents conforming to ISO 32000-1 carry the PDF version number 1.7. Documents containing Adobe extended features still carry the PDF base version number 1.7 but also contain an indication of which extension was followed during document creation.^[11]

Adobe's versions

[\[edit\]](#)

Version	Edition ^[1]	Year of publication	New features	Supported by Reader version
1.0	First edition	1993		Acrobat Reader (Carousel)
1.1	First edition revised	1996	Passwords, encryption (MD5, RC4 40bit), device-independent color, threads and links ^[12]	Acrobat Reader 2.0
	First		Interactive page elements (such as radio buttons and checkboxes); interactive, fill-in forms (AcroForm); Forms Data Format (FDF) for interactive form data that can be imported,	

1.2	First edition revised	1996	exported, transmitted and received from the Web; mouse events; support for playing movies from external files; support for playing sounds, either embedded in the PDF file or from external files ; Unicode; advanced color features and image proxying ^[12]	Acrobat Reader 3.0
1.3	Second edition	2000	Digital signatures; ICC and DeviceN color spaces; JavaScript actions; Embedded file streams - embedding files of any type directly within the body of the PDF file itself (e.g. used for attachments); many new annotation types; new features of the Adobe imaging model embodied in PostScript LanguageLevel 3; Masked images; Alternate representations for a single image; Smooth shading; Enhanced page numbering; Web Capture - a facility for capturing information from World Wide Web and converting it to PDF; Logical structure - a facility for representing logical structure independently of its graphical structure; additional support for CIDFonts; data structures for mapping strings and numbers to PDF objects; Prepress Support - information useful in prepress production workflows; new functions - for several types of function object that represent parameterized classes of functions ^{[13][14]}	Acrobat Reader 4.0
1.4	Third edition	2001	JBIG2 ; transparency; RC4 encryption key lengths greater than 40bits (40-128bits); enhancements to interactive forms and Forms Data Format (FDF), support for XML form submissions, embedded FDF files, Unicode specification of field export values, support for remote collaboration and digital signatures in FDF files; support for accessibility to disabled users; metadata streams using XML - Extensible Metadata Platform (XMP); Tagged PDF ;Facilities for including printer's marks; Support for the display and preview of production-related page boundaries; New predefined CMaps; Alternate Presentations - alternate ways in which the document may be viewed; The ability to import content from one PDF document into another; EmbeddedFiles entry in the PDF document's name dictionary - a standard location for the embedded data ^{[14][15]} ; OCR text layer ^[citation needed]	Acrobat Reader 5.0
			JPEG 2000 ; enhanced support for embedding and playback of multimedia; object streams; cross reference streams; XML Forms Data	

1.5	Fourth edition	2003	<p>Format (XFDF) for interactive form submission (replaced the XML format in PDF 1.4); support for forms, rich text elements and attributes based on Adobe's XML Forms Architecture (XFA) 2.02; public-key security handlers using PKCS#7 (introduced in PDF 1.3 but not documented in the PDF Reference until PDF 1.5), public-key encryption, permissions - usage rights (UR) signatures (does not require document encryption), PKCS#7 with SHA-1, RSA up to 4096-bits; security handler can use its own encryption and decryption algorithms; Optional Content - sections of content in a PDF document that can be selectively viewed or hidden by document authors or consumers - for items such as CAD drawings, layered artwork, maps, and multi-language documents; Alternate Presentations - the only type is slideshow - invoked by means of JavaScript actions (Adobe Reader supports only SVG 1.0)^{[14][16][17]} Support for Windows 98 dropped. To view and print newer version PDFs, such as those at the IRS website ↗, with older versions of Reader requires downloading in Google Docs "Quick View" simplified PDF format.</p>	Adobe Reader 6.0
1.6	Fifth edition	2004	<p>3D artwork, e.g. support for Universal 3D file format; OpenType font embedding; support for XFA 2.2 rich text elements and attributes; AES encryption; PKCS#7 with SHA256, DSA up to 4096-bits; NChannel color spaces; additional support for embedded file attachments, including cross-document linking to and from embedded files; enhancements and clarifications to digital signatures related to usage rights and modification detection and prevention signatures^[14]</p>	Adobe Reader 7.0
1.7 (ISO 32000-1:2008 [11][1])	Sixth edition (ISO First Edition)	2006	<p>increased presentation of 3D artwork; support for XFA 2.4 rich text elements and attributes; presentation of multiple file attachments (portable collections); document requirements for a PDF consumer application; new string types - PDFDocEncoded string, ASCII string, byte string; PKCS#7 with SHA384, SHA512 and RIPEMD160</p>	Adobe Reader 8
1.7 Extension Level 3		2008	<p>256-bit AES encryption; incorporation of XFA Datasets into a file conforming PDF/A-2; RichMedia annotations - attaching Flash applications, video (including Flash video with H.264), audio, and other multimedia with expanded functionality, two-way scripting bridge</p>	Adobe Reader 9

			between Flash and a conforming application; support for the rich text conventions described in XFA 2.5 and 2.6	
1.7 Extension Level 5		2009	XFA 3.0	Adobe Reader 9.1
				Adobe Reader X

Standardization

[\[edit\]](#)

Specialized subsets of PDF

[\[edit\]](#)

The following specialized subsets of PDF specification has been standardized as ISO standards (or are in standardization process):^{[1][18][19][20]}

- **PDF/X** (since 2001 - series of ISO 15929 and ISO 15930 standards) - a.k.a. "PDF for Exchange" - for the *Graphic technology - Prepress digital data exchange* - (working in ISO Technical committee 130)
- **PDF/A** (since 2005 - series of ISO 19005 standards) - a.k.a. "PDF for Archive" - *Document management - Electronic document file format for long-term preservation* (working in ISO Technical committee 171)
- **PDF/E** (since 2008 - ISO 24517) - a.k.a. "PDF for Engineering" - *Document management - Engineering document format using PDF* (working in ISO Technical committee 171)
- **PDF/VT** (since 2010 - ISO 16612) - a.k.a "PDF for exchange of variable data and transactional (VT) printing" - *Graphic technology - Variable data exchange* (working in ISO Technical committee 130)
- **PDF/UA** (under development in 2011 - ISO/DIS 14289-1) - a.k.a. "PDF for Universal Access" - *Document management applications - Electronic document file format enhancement for accessibility* (working in ISO Technical committee 171)

There is also the *PDF/H*, a.k.a. "PDF Healthcare", a Best Practices Guide (BPG), supplemented by an Implementation Guide (IG), published in 2008. PDF Healthcare is not a standard or proposed standard, but only a guide for use with existing standards and other technologies. It is supported by the standards development organizations [ASTM](#) and [AIIM](#).^{[21][22][23]}

Full function PDF

[\[edit\]](#)

PDF 1.7

[\[edit\]](#)

The final revised documentation for PDF 1.7 was approved by ISO Technical Committee 171 in January 2008 and published as ISO 32000-1:2008 on July 1, 2008. PDF is now a published ISO standard, titled *Document management—Portable document format—Part 1: PDF 1.7*.

ISO 32000-1:2008 is the first ISO standard for the full function PDF. The previous ISO PDF standards (PDF/A, PDF/X, etc.) are for more specialized uses. The ISO 32000-1 includes all of the functionality previously documented in the Adobe PDF Specifications for versions 1.0 through 1.6, but it should be noted that where Adobe removed certain features of PDF from their standard, they too are not contained in PDF 1.7.^{[1][vague]}

ISO 32000 document was prepared by Adobe Systems Incorporated based upon *PDF Reference, sixth edition, Adobe Portable Document Format version 1.7, November 2006*. It was reviewed, edited and adopted, under a special fast-track procedure, by *ISO Technical Committee 171 (ISO/TC 171), Document management application, Subcommittee SC 2, Application issues*, in parallel with its approval by the ISO member bodies.

According to the ISO PDF standard abstract:^[24]

ISO 32000-1:2008 specifies a digital form for representing electronic documents to enable users to exchange and view electronic documents independent of the environment in which they were created or the environment in which they are viewed or printed. It is intended for the developer of software that creates PDF files (conforming writers), software that reads existing PDF files and interprets their contents for display and interaction (conforming readers) and PDF products that read and/or write PDF files for a variety of other purposes (conforming products).

PDF 2.0

[edit]

A new version of PDF standard is under development under the name *ISO/NP 32000-2 - Document management—Portable document format—Part 2: PDF 2.0* (as of January 2011).^[25] Adobe has submitted the *Adobe Extension Level 5* and *Adobe Extension Level 3* specifications to ISO for inclusion into the next version of the ISO 32000 specification. Adobe declared they have all been accepted for part 2 of ISO 32000.^[11]

Technical foundations

[edit]

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.^[26]

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

[edit]

[PostScript](#) is a [page description language](#) run in an [interpreter](#) to generate an image, a process requiring many resources. It can handle not just graphics, but 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 `line` to 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 true [graphic transparency](#); PostScript does not.
- PostScript is an [interpretive 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 in order 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 with).

Technical overview

[edit]

File structure

[edit]

A PDF file consists primarily of *objects*, of which there are eight types:^[27]

- [Boolean](#) values, representing *true* or *false*
- Numbers
- [Strings](#)
- Names
- [Arrays](#), ordered collections of objects
- Dictionaries, collections of objects indexed by Names
- [Streams](#), usually containing large amounts of data
- The [null](#) object

Objects may be either *direct* (embedded in another object) or *indirect*. Indirect objects are numbered with an *object number* and a *generation number*. An index table called the *xref table* gives the byte offset of each indirect object from the start of the file.^[28] 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*). Beginning with PDF version 1.5, indirect objects may also be located in special streams known as *object streams*. This technique reduces the size of files that have large numbers of small indirect objects and is especially useful for *Tagged PDF*.

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.^[11] PDF files may be optimized using [Adobe Acrobat](#) software or [QPDF](#).

Imaging model

[edit]

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

[edit]

Vector graphics in PDF, as in PostScript, 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, 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 pattern of which the simplest are the *axial shade* (Type 2) and *radial shade* (Type 3).

Raster images

[\[edit\]](#)

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 deprecated 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](#) or Zip algorithm
- **LZWDecode** a deprecated filter based on [LZW](#) Compression
- **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](#) filter based on the [CCITT fax](#) compression standard
- **JBIG2Decode** a lossy or lossless 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 techniques.

Text

[\[edit\]](#)

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

[\[edit\]](#)

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.

Encodings

[\[edit\]](#)

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 a large number of 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).^[29] 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

[\[edit\]](#)

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 view completely wrongly 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.^[30]

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

[\[edit\]](#)



This section requires [expansion](#).

PDF files may contain interactive elements such as annotations and form fields.

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 PDF specification:^{[8][31][32][33][34]}

- **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. The XFA specification is not included in the PDF specification, it is only referenced as an optional feature. Adobe XFA Forms are not compatible with AcroForms.^{[8][35]}

AcroForms

[\[edit\]](#)

AcroForms were introduced in the PDF 1.2 format.^[36] AcroForms permit using objects (text boxes, radiobuttons, etc.) and some code ([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):^[31]

- 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). (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.^{[37][38][39]} 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.^[31]

Forms Data Format (FDF)

[\[edit\]](#)

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 format 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. Beginning in PDF 1.3, FDF can be used to define a container for annotations that are separate from the PDF document to which they apply. FDF is typically used to encapsulate information such as

Forms Data Format (FDF)

Filename extension	.fdf
Internet media type	application/vnd.fdf ^[40]
Type code	'FDF '
Developed by	Adobe Systems
Initial release	1996 (PDF 1.2)
Extended from	PDF
Extended to	XFDF
Standard(s)	ISO/IEC 32000-1:2008

X.509 certificates, requests for certificates, directory settings, timestamp server settings, and embedded PDF files for network transmission.^[39] The FDF uses the MIME content type application/vnd.fdf, filename extension .fdf and on Mac OS it uses file type 'FDF'.^[31] A support for importing and exporting FDF stand-alone files is not widely implemented in free or freeware PDF software. For example, there is no support in Evince, Okular, KPDF or Sumatra PDF. Import support for stand-alone FDF files is implemented in Adobe Reader; export and import support (including saving of FDF data in PDF) is for example implemented in Foxit Reader and PDF-XChange Viewer Free; saving of FDF data in a PDF file is also supported in pdftk.

XML Forms Data Format (XFDF)

[\[edit\]](#)

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. There are not XFDF equivalents for some entries in the FDF dictionary - such as the Status, Encoding, JavaScript, Pages keys, EmbeddedFDFs,

XML Forms Data Format (XFDF)

Filename extension	.xfdf
Internet media type	application/vnd.adobe.xfdf ^[41]
Type code	'XFDF'

Differences and Target. 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*^[38]. 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

Type Code	XFDF
Developed by	Adobe Systems
Latest release	2.0
Extended from	PDF, FDF, XML
Website	XFDF 2.0 specification 

the XML standard. XFDF can be used 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 the corresponding PDF interactive form. A support for importing and exporting FDF stand-alone files is not widely implemented in free or freeware PDF software. Import of XFDF is implemented in Adobe Reader 5 and later versions; import and export is implemented in PDF-XChange Viewer Free; embedding of XFDF data in PDF form is implemented in pdftk (pdf toolkit).

Adobe XML Forms Architecture (XFA)

[\[edit\]](#)

Main article: [XML Forms Architecture](#)

In the PDF 1.5 format, [Adobe Systems](#) introduced a new, proprietary format for forms, namely Adobe XML Forms Architecture (XFA) forms. The XFA 2.02 is referenced in the PDF 1.5 specification (and also in later versions) but is described separately in *Adobe XML Forms Architecture (XFA) Specification*, which has several versions.^{[36][42]} Adobe XFA Forms are not compatible with AcroForms. Creating XFA Forms for use in Adobe Reader requires Adobe LiveCycle Forms Designer or Avoka SmartForm Composer.^{[8][43]} Adobe Reader contains "disabled features" for use of XFA Forms, that will activate only when opening a PDF document that was created using enabling technology available only from Adobe.^{[44][45]} The XFA Forms are not compatible with Adobe Reader prior to version 6.^[8]

XFA forms can be created and used as PDF files or as XDP ([XML Data Package](#)) files. The format of an XFA resource in PDF is described by the XML Data Package Specification.^[31] The XDP may be a standalone document or it may in turn be carried inside a PDF document. XDP provides a mechanism for packaging form components within a surrounding XML container. An XDP can also package a PDF file, along with XML form and template data.^[42] PDF may contain XFA (in XDP format), but also XFA may contain PDF.^[42] When the XFA (XML Forms Architecture) grammars used for an XFA form are moved from one application to another, they must be packaged as an XML Data Package.^[46]

When the PDF and XFA are combined, the result is a form in which each page of the XFA form overlays a PDF background. This architecture is sometimes referred to as XFAF (XFA Foreground). The alternative is to express all of the form, including boilerplate, directly in XFA. It is sometimes called *full* XFA.^[46]

Starting with PDF 1.5, the text contents of variable text form fields, as well as markup annotations may include formatting information (style information). These rich text strings are XML documents that conform to the rich text conventions specified for the XML Forms Architecture specification 2.02, which is itself a subset of the XHTML 1.0 specification, augmented with a restricted set of CSS2 style attributes.^[31] In PDF 1.6, PDF supports the rich text elements and attributes specified in the XML Forms Architecture (XFA) Specification, 2.2. In PDF 1.7, PDF supports the rich text elements and attributes specified in the XML Forms Architecture (XFA) Specification, 2.4^[31]

Logical structure and accessibility

[edit]

 This section requires [expansion](#).

A PDF may contain structure information to enable better text extraction and accessibility. When published, [PDF/UA](#), now [ISO/AWI 14289](#), will provide definitive information on how the contents of PDF files are to be tagged with accurate structure information. Tagged PDFs also allow a page-limited [reflow of documents](#) for smaller devices, sometimes called "reflowable PDFs".^[47]

Security and signatures

[edit]

 This section requires [expansion](#).

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, "user password" and "owner password". A PDF document may be protected by password to open ('user' password) and the document may also specify operations that should be restricted even when the document is decrypted: printing, copying text and graphics out of the document, modifying the document, or adding or modifying text notes and [AcroForm](#) fields (using 'owner' password). However, all operations (except the document open password protection, if applicable) which are restricted by "owner" or "user" passwords are trivially circumvented by many commonly available "PDF cracking" software and even freely online,^[48] and if circumvented these restrictions no longer let the author control what can and cannot be done with the pdf file once distributed. This warning is also 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 will ignore the [digital rights management](#) "protections" and will allow the user to print or make copy of excerpts of the text as if the document were not limited by password protection.

Usage rights

[edit]

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 bonafide granting authority. For example, it can be used to allow a user:^[31]

- 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, export annotations

For example, Adobe Systems grants permissions to enable additional features in Adobe Reader, using public-key [cryptography](#). Adobe Reader will verify that the signature uses a [certificate](#) from an Adobe-[authorized](#) certificate authority. The PDF 1.5 specification declares that other PDF viewer applications are free to use this same mechanism for their own purposes.^[31]

File attachments

[edit]

 This section requires [expansion](#).

PDF files can have document-level and page-level file attachments, which the reader can access and open or save to their local filesystem. PDF attachments can be added to existing PDF files

for example using [pdftk](#). Adobe Reader provides support for attachments, and [poppler](#) based readers like [Evince](#) or [Okular](#) also have some support for document-level attachments.

Metadata

[edit]

PDF files can contain two types of metadata.^[49] 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.

Later, in PDF 1.4, support was added for the 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.

Subsets

[edit]

Proper subsets of PDF have been, or are being, standardized under [ISO](#) for several constituencies:

- [PDF/X](#) for the printing and graphic arts as ISO 15930 (working in ISO TC130)
- [PDF/A](#) for archiving in corporate/government/library/etc environments as ISO 19005 (work done in ISO TC171)
- [PDF/E](#) for exchange of engineering drawings (work done in ISO TC171)
- [PDF/UA](#) for universally accessible PDF files

A [PDF/H](#) variant (PDF for Healthcare) is being developed.^[50] However, it may consist more of a set of "best practices" than of a specific format or subset.

Mars

[edit]

See also: *[Page description markup language](#)*

Adobe is exploring an XML-based next-generation PDF [code named Mars](#).^[51] Information about the Mars file format is published by Adobe at <http://www.adobe.com/go/mars> and also <http://labs.adobe.com/wiki/index.php/Mars>.

The format of graphic elements of Mars is sometimes described simply as "[SVG](#)",^[citation needed] but according to the version 0.8 draft specification of November 2007 (§3 Mars SVG Support) the format is actually merely similar to SVG: it contains both additions to and subtractions from SVG, so it is in general neither viewable by nor creatable with standard SVG tools: some things will look noticeably different between SVG viewers and Mars viewers.

Technical issues

[edit]

Accessibility

[edit]

PDF files can be created specifically to be accessible for disabled people.^{[52][53][54][55][56]} Current PDF file formats can include tags ([XML](#)), text equivalents, captions, audio descriptions, et cetera. Some software can automatically produce tagged PDFs, however this feature is not always enabled by default. Leading [screen readers](#), including [JAWS](#), [Window-Eyes](#), Hal, and [Kurzweil 1000 and 3000](#) can read tagged PDFs; current versions of the Acrobat and Acrobat Reader programs can also read PDFs aloud.^{[57][58][59]} Moreover, tagged PDFs can be re-flowed and magnified for readers with visual impairments. Problems remain with adding tags to older PDFs and those that are generated from scanned documents. In these cases, accessibility tags and re-flowing are unavailable, and must be created either manually or with [OCR](#) techniques.

These processes are inaccessible to some disabled people. [PDF/UA](#), the PDF/Universal Accessibility Committee, an activity of [AIIM](#), is working on a specification for PDF accessibility based on ISO 32000.

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 read (useful for people with poor eyesight). The content view is displayed when the document is re-flowed to Acrobat (useful for people with mobility disability). For a PDF document to be accessible, the three views must be consistent with each other.^[60]

Security

[\[edit\]](#)

PDF attachments carrying viruses were first discovered in 2001. This 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.^[61]

From time to time, new vulnerabilities are discovered^[62] in various versions of Adobe Reader, prompting the company to issue security fixes. One aggravating factor is that Adobe 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 Adobe 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 Adobe Reader can help mitigate unknown exploits that may be discovered in the future, although it will not provide protection against exploits in other parts of the PDF viewing software. Security experts argue that JavaScript is a nonessential feature and that the security benefit that comes from disabling JavaScript outweighs any incompatibility issues that may arise from doing so.^[63] PDF file exploits can be mitigated to a much greater degree by having a third party web service convert them to another file format before viewing.^[64]

On March 30, 2010 security researcher Didier Stevens reported an Adobe Reader and Foxit Reader "exploit". The exploit results in an arbitrary executable being run if the user accepts a "launch executable" warning prompt.^[65]

Usage restrictions and monitoring

[\[edit\]](#)

PDFs may be [encrypted](#) so that a password is needed to view or edit the contents. The PDF Reference defines both 40-bit and 128-bit encryption, both making use of a complex system of [RC4](#) and [MD5](#). The PDF Reference also defines ways in which third parties can define their own encryption systems for use in PDF.

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

The PDF Reference has technical details or see ^[66] for an end-user overview. Like HTML files, PDF files may submit information to a web server. This could be used to track the [IP address](#) of the client PC, a process known as [phoning home](#). After update 7.0.5 to Acrobat Reader, the user will be notified "via a dialogue box that the author of the file is auditing usage of the file, and be offered the option of continuing."^[67]

Through its [LiveCycle Policy Server](#) product, Adobe provides a method to set security policies on specific documents. This can include requiring a user to authenticate and limiting the timeframe a document can be accessed or amount of time a document can be opened while offline. Once a PDF document is tied to a policy server and a specific policy, that policy can be changed or

revoked by the owner. This controls documents that are otherwise "in the wild." Each document open and close event can also be tracked by the policy server. Policy servers can be set up privately or Adobe offers a public service through Adobe Online Services. As with other forms of DRM, adherence to these policies and restrictions may or may not be enforced by the reader software being used.

Default display settings

[edit]

PDF documents can contain display settings, including the page display layout and zoom level. Adobe Reader will use these settings to override the user's default settings when opening the document.^[68] The free Adobe Reader cannot remove these settings.

Content

[edit]

A PDF file is often a combination of [vector graphics](#), text, and [raster graphics](#). The basic types of content in a PDF are:

- text stored as such
- vector graphics for illustrations and designs that consist of shapes and lines
- raster graphics for photographs and other types of image

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.^{[3][4]}

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.

Standard Type 1 Fonts

[edit]

There are fourteen typefaces that have a special significance to 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, sometimes referred to as the "base fourteen fonts"^[69] should always be present (actually present or a close substitute) and so need not be embedded in a PDF.^[70] PDF viewers must know about the metrics of these fonts. Other fonts may be substituted if they are not embedded in a PDF.

Implementations

[edit]

PDF-viewing software is generally provided free of charge, and many versions are available from a variety of sources ([List of PDF software](#)).

There are many software options for creating PDFs, including the PDF printing capabilities built in to [Mac OS X](#) and most [Linux](#) distributions, the multi-platform [OpenOffice.org](#), [Microsoft Office 2007](#) (if updated to [SP2](#)),^[71] [WordPerfect](#) since version 9, [Free PDF XP](#)^[72] and numerous PDF

print drivers for [Microsoft Windows](#), the [pdfTeX](#) typesetting system, the [DocBook](#) PDF tools, applications developed around [Ghostscript](#) and [Adobe Acrobat](#) itself. [Google's](#) online office suite [Google Docs](#) also allows for uploading, and saving to the Portable Document Format.

[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^[73] from [Adobe Systems](#) and [Jaws](#)^[74] and the [Harlequin RIP](#) from [Global Graphics](#).

Editing PDFs (structure)

[edit]

 This section requires [expansion](#).

There is also specialized software for editing PDF files, though the choices are much more limited and often expensive. As of version 0.46, [Inkscape](#) also allows PDF editing through an intermediate translation step involving [Poppler](#). A PDF editor most similar to word processor is called [Infix PDF Editor](#).^[75]

See [List of PDF software](#) for a more complete list of PDF editors.

Annotating PDFs

[edit]

See also: [Comparison of notetaking software](#) and [Category:Web 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](#). Another GPL-licensed application native to the Linux environment is [Xournal](#). Xournal allows for annotating in different fonts and colours, as well as a rule for quickly underlining and highlighting lines of text or paragraphs. Xournal also has a shape recognition tool for squares, rectangles and circles. In Xournal annotations may be moved, copied and pasted. The [freeware Foxit Reader](#) allows annotating but adds a watermark on each annotated page. The commercial version of the package does not have this limitation. Tracker Software's [PDF-XChange Viewer](#) allows annotations and markups without restrictions in its freeware alternative. [Apple's Mac OS X's](#) integrated PDF viewer, Preview, does also enable annotations. For mobile annotation, [iAnnotate PDF](#) for the [iPad](#) and [Aji Annotate](#) for the [iPhone](#), both produced by Aji, allow annotation of PDFs as well as exporting summaries of the annotations.

There are also [web annotation](#) systems which allow to annotate pdf and other documents formats, e.g. [A.nnotate](#), [crocodoc](#), [WebNotes](#).

In cases where PDFs are expected to have all of the functionality of paper documents, ink annotation is required. Many programs which accept ink input from the mouse are not responsive enough for handwriting using an input tablet or tablet PC. Existing solutions on the PC include [Bluebeam PDF Revu](#), and [PDF Annotator](#).

Other applications and functionalities

[edit]

Several applications embracing the PDF standard are now available as an online service including [Scribd](#) for viewing and storing, [Pdfvue](#) for online editing, and [Zamzar](#) for PDF Conversion.

In 1993 the [Jaws RIP](#) 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.^{[[citation needed](#)]}

[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.^[76] 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 [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 Print command. 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; in 10.4 and 10.5 the default behaviour is set to capture as a [PNG](#) file, though this behaviour can be set back to PDF if required.

Some desktop printers also support direct PDF printing, which can interpret PDF data without external help. Currently, all PDF capable printers also support PostScript, but most PostScript printers do not support direct PDF printing.

The [Free Software Foundation](#) considers 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."^{[77][78]} The [GNUpdf](#) library has, however, not been released yet, while [Poppler](#) has enjoyed wider use in applications such as [Evince](#), which comes with the [GNOME](#) desktop environment, at the expense of relying on the GPLv2-licensed [Xpdf](#)^{[79][80]} code base that can't be used by GPLv3 programs. 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](#).^[81]

See also

[\[edit\]](#)

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

References

[\[edit\]](#)

- ↑ **a** **b** **c** **d** **e** **f** **g** **h** "ISO 32000-1:2008 - Document management — Portable document format — Part 1: PDF 1.7" . Iso.org. 2008-07-01. Retrieved 2010-02-21.
- ↑ Adobe Systems Incorporated, [PDF Reference, Sixth edition, version 1.23 \(30 MB\)](#) , Nov 2006, p. 33.
- ↑ **a** **b** "3D supported formats" . Adobe. 2009-07-14. Retrieved 2010-02-21.
- ↑ **a** **b** "Acrobat 3D Developer Center" . Adobe. Retrieved 2010-02-21.
- ↑ **W** **arnock**, J. (1991). "The Camelot Project" (PDF). PlanetPDF. "This document describes the base technology and ideas behind the project named "Camelot." This project's goal is to solve a fundamental problem [...] there is no universal way to communicate and view ... printed information electronically."
- ↑ Orion, Egan (2007-12-05). "PDF 1.7 is approved as ISO 32000" . *The Inquirer*. The Inquirer.

- Retrieved 2007-12-05.; "Adobe wins backing for PDF 1.7"  vnunet.com.
7. ^ "Gnu PDF - PDF Knowledge - Forms Data Format"  Retrieved 2010-02-19. ""Apparently Adobe introduced something newer called XFA (XML Forms Architecture) which doesn't seem standardized.""
 8. ^ [a b c d e](#) *A Quick Introduction to Acrobat Forms Technology* , 2006-05-14, retrieved 2010-02-19, ""Adobe's XFA Forms is a closed standard that competes with the fully open W3C Xforms standard.""
 9. ^ Laurens Leurs. "The history of PDF"  Retrieved 2007-09-19.
 10. ^ History of PDF Openness 
 11. ^ [a b c d e](#) "Adobe Developer Connection: PDF Reference and Adobe Extensions to the PDF Specification"  Adobe Systems Incorporated. Retrieved 2010-12-13.
 12. ^ [a b](#) Adobe Systems Incorporated (1996-11-12) (PDF), *Portable Document Format Reference Manual Version 1.2* , retrieved 2010-02-23
 13. ^ Adobe Systems Incorporated (2000) (PDF), *PDF Reference second edition - Adobe Portable Document Format Version 1.3* , retrieved 2010-02-23
 14. ^ [a b c d](#) Adobe Systems Incorporated. "Adobe PDF Reference Archives"  Retrieved 2010-02-23.
 15. ^ Adobe Systems Incorporated (2001) (PDF), *PDF Reference third edition - Adobe Portable Document Format Version 1.4* , retrieved 2010-02-23
 16. ^ Adobe Systems Incorporated (2003) (PDF), *PDF Reference fourth edition - Adobe® Portable Document Format Version 1.5* , retrieved 2010-02-23
 17. ^ "PDF compatibility levels"  Retrieved 2010-04-01.
 18. ^ "ISO standards by Technical committee - TC 171/SC 2 - Document management applications / Application issues"  Retrieved 2011-01-11.
 19. ^ "ISO standards by Technical committee - TC 130 - Graphic technology"  Retrieved 2011-01-11.
 20. ^ "AIIM to Facilitate ISO Standards Process for Leading Electronic Document Format"  2007-01-29. Retrieved 2011-01-11.
 21. ^ "PDF Healthcare Frequently Asked Questions"  2006-09-25. Retrieved 2011-01-11.
 22. ^ "PDF Healthcare (PDF/H) Committee"  Retrieved 2011-01-11.
 23. ^ "PDF/H Implementation Guide Materials"  Retrieved 2011-01-11.
 24. ^ ISO 32000-1:2008, Page 1, section "1 Scope"
 25. ^ "ISO/NP 32000-2 - Document management -- Portable document format -- Part 2: PDF 2.0"  2009-10-06. Retrieved 2010-02-24.
 26. ^ partners.adobe.com - Developer Resources 
 27. ^ Adobe Systems, PDF Reference, p. 51.
 28. ^ Adobe Systems, PDF Reference, pp. 39–40.
 29. ^ "PDF Referencem Sixth Edition, version 1.7, table 5.11" 
 30. ^ PDF Blend Modes Addendum 
 31. ^ [a b c d e f g h i](#) Adobe Systems Incorporated (2008-07-01), *Document Management – Portable Document Format – Part 1: PDF 1.7, First Edition* , retrieved 2010-02-19
 32. ^ "Gnu PDF - PDF Knowledge - Forms Data Format"  Retrieved 2010-02-19.
 33. ^ "About PDF forms"  Retrieved 2010-02-19.
 34. ^ "Convert XFA Form to AcroForm?"  2008. Retrieved 2010-02-19.
 35. ^ "Migrating from Adobe Acrobat forms to XML forms"  Retrieved 2010-02-22.
 36. ^ [a b](#) "What is the difference between Acroforms and XFA?"  2009-03-27. Retrieved 2010-02-19.
 37. ^ Adobe Systems Incorporated (2007-10-15). "Using Acrobat forms and form data on the web"  Retrieved 2010-02-19.
 38. ^ [a b](#) (PDF) *XML Forms Data Format Specification, version 2* , 2007-09, retrieved 2010-02-19
 39. ^ [a b](#) (PDF) *fdf Data Exchange Specification* , 2007-02-08, retrieved 2010-02-19
 40. ^ IANA Application Media Types - vnd.fdf  retrieved 2010-02-22
 41. ^ IANA Application Media Types - Vendor Tree - vnd.adobe.xfdf  retrieved 2010-02-22
 42. ^ [a b c](#) Adobe Systems Incorporated. "Adobe XML Forms Architecture (XFA)"  Retrieved 2010-02-19.
 43. ^ "Adobe LiveCycle Reader Extensions ES2"  Retrieved 2010-02-19.
 44. ^ (PDF) *Adobe Reader - Software license agreement* , retrieved 2010-02-19

45. ^ ["LiveCycle Reader Extensions ES features and benefits"](#) . Retrieved 2010-02-19.
46. ^ [a b](#) (PDF) *XML Forms Architecture (XFA) Specification Version 2.5* , 2007-06-08, retrieved 2010-02-19
47. ^ <http://books.google.com/books?id=dIJ94KZqwqcC&pg=PA379> 
48. ^ [FreeMyPDF - A website that removes PDF "user password" restrictions](#) 
49. ^ [Adobe PDF reference version 1.7](#) , section 10.2
50. ^ [AIIM \(2006-10-20\). "New Best Practices Guide Addresses Exchange of Healthcare Information"](#) . Retrieved 2007-03-09.
51. ^ [Jackson, Joab \(2006-12-07\). "Adobe plunges PDF into XML"](#) . Government Computer News. Retrieved 2008-01-12.
52. ^ ["PDF Accessibility"](#) . WebAIM. Retrieved 2010-04-24.
53. ^ [Joe Clark \(2005-08-22\). "Facts and Opinions About PDF Accessibility"](#) . Retrieved 2010-04-24.
54. ^ ["Accessibility and PDF documents"](#) . Web Accessibility Center. Retrieved 2010-04-24.
55. ^ ["PDF Accessibility Standards v1.2"](#) . Retrieved 2010-04-24.
56. ^ (PDF) *PDF Accessibility* , California State University, retrieved 2010-04-24
57. ^ ["Adobe Reader 8 - Read a PDF with Read Out Loud"](#) . Retrieved 2010-04-24.
58. ^ ["Tip of the Week: Adobe Reader's 'Read Aloud' Feature"](#) . *The New York Times*. 2009-04-10. Retrieved 2010-04-24.
59. ^ (PDF) *Accessing PDF documents with assistive technology: A screen reader user's guide* , Adobe, retrieved 2010-04-24
60. ^ [Chris Rusbridge \(2008-04-29\). "Why PDF is a Hamburger"](#) . Retrieved 2010-04-24.
61. ^ [Adobe Forums, Announcement: PDF Attachment Virus "Peachy"](#) , 15 August 2001.
62. ^ ["Security bulletins and advisories"](#) . Adobe. Retrieved 2010-02-21.
63. ^ [Steve Gibson - SecurityNow Podcast](#) 
64. ^ [PDFCleaner - PDF Exploit Sanitizer](#) 
65. ^ [PCmag.com blogs](#) 
66. ^ ["Create Adobe PDF Online - Security Settings Help"](#) . Createpdf.adobe.com. Retrieved 2010-02-21.
67. ^ [New features and issues addressed in the Acrobat 7.0.5 Update \(Acrobat and Adobe Reader for Windows and Mac OS\)](#) 
68. ^ ["Getting Familiar with Adobe Reader > Understanding Preferences"](#) . Retrieved 2009-04-22.
69. ^ [Adobe Acrobat Base 14 Fonts](#) 
70. ^ [The PDF Font Aquarium](#) 
71. ^ ["Description of 2007 Microsoft Office Suite Service Pack 2 \(SP2\)"](#) . Microsoft. Retrieved 2009-05-09.
72. ^ ["Free PDF XP"](#) . Retrieved 2010-07-27.
73. ^ www.adobe.com/products/pdfprintengine/overview.html 
74. ^ www.globalgraphics.com/products/jaws_rip/ 
75. ^ [PDF Editor Review and Comparison Chart](#) 
76. ^ [Press-Ready PDF Files](#)  "For anyone interested in having their graphic project commercially printed directly from digital files or PDFs." (last checked on 2009-02-10).
77. ^ [Current FSF High Priority Free Software Projects](#)  (last checked on 2009-02-10)
78. ^ [Goals and Motivations - GNUpdf](#) 
79. ^ [Poppler homepage](#)  "Poppler is a PDF rendering library based on the xpdf-3.0 code base." (last checked on 2009-02-10)
80. ^ [Xpdf license](#)  "Xpdf is licensed under the GNU General Public License (GPL), version 2." (last checked on 2009-02-10).
81. ^ [The Apache PDFBox project](#) . Retrieved 2009-09-19.

Further reading

[\[edit\]](#)

- [Hardy, M. R. B.; Brailsford, D. F. \(2002\). *Mapping and displaying structural transformations between XML and PDF* !\[\]\(d22f13df166c9ece0dad43bc81a8ec1f_img.jpg\). Proceedings of the 2002 ACM symposium on Document engineering. pp. 95-102. doi:10.1145/585058.585077 !\[\]\(fa3eb5ad50fe8ea332a7b11c79518f0c_img.jpg\). ISBN 1-58113-594-7. edit](#)

- [Adobe PDF 101: Quick overview of PDF](#)
- [Adobe: PostScript vs. PDF](#) – Official introductory comparison of PS, EPS vs. PDF.
- *[PDF Standards...transitioning the PDF specification from a de facto standard to a de jure standard](#)* – Information about PDF/E and PDF/UA specification for accessible documents file format
- [ISO 19005-1:2005](#) Document of the PDF/A-1 Standard at the [International Organization for Standardization](#) (chargeable)
- PDF Reference
 - [Latest PDF version specification \(including free version of the ISO standard\)](#)
 - Books: PDF 1.6 (ISBN 0-321-30474-8), PDF 1.4 (ISBN 0-201-75839-3), PDF 1.3 (ISBN 0-201-61588-6)
- [Portable Document Format: An Introduction for Programmers](#) – Quick one-page intro to PDF vs. PostScript and PDF internals (up to v1.3)
- [The Camelot Paper](#) – the paper in which John Warnock outlined the project that created PDF

v · d · e	Graphics file formats	[show]
v · d · e	Multi-purpose office document file formats	[show]
v · d · e	ISO standards	[show]
v · d · e	Electronic books	[show]

Categories: [Adobe Systems](#) | [Graphics file formats](#) | [Electronic documents](#) | [Open formats](#) | [Page description languages](#) | [Vector graphics](#) | [Computer file formats](#) | [Digital press](#) | [ISO standards](#) | [IEC standards](#)

This page was last modified on 11 February 2011 at 12:44.

Text is available under the [Creative Commons Attribution-ShareAlike License](#); additional terms may apply. See [Terms of Use](#) for details.

Wikipedia® is a registered trademark of the [Wikimedia Foundation, Inc.](#), a non-profit organization.

[Contact us](#)

[Privacy policy](#) [About Wikipedia](#) [Disclaimers](#)

