WIKIPEDIA

PDF

Portable Document Format (**PDF**), standardized as **ISO 32000**, is a file format developed by Adobe in 1993 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 therefore no longer requires 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`
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	Adobe Inc. (1993–2008) ISO (2008–)
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/75839.html (https://www.iso.org/standard/75839.html)

Contents

History

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.

In December, 2020, the second edition of PDF 2.0, ISO 32000-2:2020, was published, including clarifications, corrections and critical updates to normative references.[13] ISO 32000-2 does not include any proprietary technologies as normative references.[14]

Technical foundations

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 a plugin for Acrobat 3.0), or any other types of embedded contents that can be handled using plug-ins.

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 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` statements 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 transparent graphics; 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 compiled and included).

PDF 1.6 supports interactive 3D documents embedded in a PDF file: 3D drawings can be embedded using U3D or PRC and various other data formats.[15][16][17]

File format

A PDF file contains 7-bit ASCII characters, except for certain elements that may have binary content. The file starts with a header containing a magic number (as a readable string) and the version of the format, for example `%PDF-1.7`. The format is a subset of a COS ("Carousel" Object Structure) format.[18] A COS tree file consists primarily of *objects*, of which there are eight types:[19]

- Boolean values, representing *true* or *false*
- Numbers
- Strings, enclosed within parentheses (`(...)`). Strings 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 angle brackets (`<<...>>`)
- Streams, usually containing large amounts of optionally compressed binary data, preceded by a dictionary and enclosed between the `stream` and `endstream` keywords.
- 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.[20] 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, followed by
- 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`)
- Other optional information

There are two layouts to the PDF files: non-linear (not "optimized") and linear ("optimized"). Non-linear PDF files can be smaller 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 generated in a linear (as in page order) fashion.[21] 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: