

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

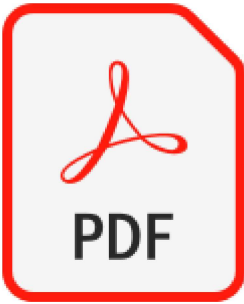
PDF files may contain a variety of content besides flat text and graphics including logical structuring elements, interactive elements such as annotations and form-fields, layers, rich media (including video content), three-dimensional objects using U3D or PRC, and various other data formats. The PDF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

## History

The development of PDF began in 1991 when John Warnock wrote a paper for a project then code-named Camelot, in which he proposed the creation of a simplified version of PostScript called Interchange PostScript (IPS).<sup>[6]</sup> Unlike traditional PostScript, which was tightly focused on rendering print jobs to output devices, IPS would be optimized for displaying pages to any screen and any platform.<sup>[6]</sup>

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

### Portable Document Format



Adobe PDF icon

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

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,<sup>[7][8]</sup> 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 necessary to make, use, sell, and distribute PDF-compliant implementations.<sup>[9]</sup>

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.<sup>[10]</sup> These proprietary technologies are not standardized, and their specification is published only on Adobe's website.<sup>[11][12][13]</sup> Many of them are not supported by popular third-party implementations of PDF.

ISO published ISO 32000-2 in 2017, available for purchase, replacing the free specification provided by Adobe.<sup>[14]</sup> In December 2020, the second edition of PDF 2.0, ISO 32000-2:2020, was published, with clarifications, corrections, and critical updates to normative references<sup>[15]</sup> (ISO 32000-2 does not include any proprietary technologies as normative references).<sup>[16]</sup> In April 2023 the PDF Association made ISO 32000-2 available for download free of charge.<sup>[14]</sup>

## Technical details

---

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

- Typeset 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 images; and
- Other multimedia objects.

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:

- An equivalent subset of the PostScript page description programming language but in declarative form, 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 language

PostScript is a page description language run in an interpreter to generate an image.<sup>[6]</sup> It can handle graphics and has standard features of programming languages such as branching and looping.<sup>[6]</sup> PDF is a subset of PostScript, simplified to remove such control flow features, while graphics commands remain.<sup>[6]</sup>

PostScript was originally designed for a drastically different use case: transmission of one-way linear print jobs in which the PostScript interpreter would collect a series of commands until it encountered the `showpage` command, then execute all the commands to render a page as a raster image to a printing device.<sup>[17]</sup> PostScript was not intended for long-term storage and real-time interactive rendering of electronic documents to computer monitors, so there was no need to support anything other than consecutive rendering of pages.<sup>[17]</sup> If there was an error in the final printed output, the user would correct it at the application level and send a new print job in the form of an entirely new PostScript file. Thus, any given page in a PostScript file could be accurately rendered only as the cumulative result of executing all preceding commands to draw all previous pages—any of which could affect subsequent pages—plus the commands to draw that particular page, and there was no easy way to bypass that process to skip around to different pages.<sup>[17]</sup>

Traditionally, to go from PostScript to PDF, a source PostScript file (that is, an executable program) is used as the basis for generating PostScript-like PDF code (see, e.g., Adobe Distiller). This is done by applying standard compiler techniques like loop unrolling, inlining and removing unused branches, resulting in code that is purely declarative and static.<sup>[17]</sup> The end result is then packaged into a container format, together with all necessary dependencies for correct rendering (external files, graphics, or fonts to which the document refers), and compressed. Modern applications write to printer drivers which directly generate PDF rather than going through PostScript first.

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

- PDF contains only static declarative PostScript code that can be processed as data, and does not require a full program interpreter or compiler.<sup>[17]</sup> This avoids the complexity and security risks of an engine with such a higher complexity level.
- Like Display PostScript, PDF has supported transparent graphics since version 1.4, while standard PostScript does not.
- PDF enforces the rule that the code for any particular page cannot affect any other pages.<sup>[17]</sup> That rule is strongly recommended for PostScript code too, but has to be implemented explicitly (see, e.g., the Document Structuring Conventions), as PostScript is a full programming language that allows for such greater flexibilities and is not limited to the concepts of pages and documents.
- All data required for rendering is included within the file itself, improving portability.<sup>[18]</sup>

Its disadvantages are:

- A loss of flexibility, and limitation to a single use case.
- A (sometimes much) larger file size.<sup>[19]</sup>

PDF since v1.6 supports embedding of interactive 3D documents: 3D drawings can be embedded using U3D or PRC and various other data formats.<sup>[20][21][22]</sup>

## File format

A PDF file is organized using ASCII characters, except for certain elements that may have binary content. The file starts with a header containing a magic number (as a readable string) and the version of the format, for example `%PDF-1.7`. The format is a subset of a COS ("Carousel" Object Structure) format.<sup>[23]</sup> A COS tree file consists primarily of *objects*, of which there are nine types:<sup>[16]</sup>

- Boolean values, representing *true* or *false*