PDF

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

PDF was standardized as ISO 32000 in 2008.[5] 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.

Portable Document Format

Adobe PDF icon

Filename extension	`.pdf`
Internet media type	`application/pdf`,[1] `application/x-pdf` `application/x-bzpdf` `application-gzpdf`
Type code	PDF [1] (including a single space)
Uniform Type Identifier (UTI)	com.adobe.pdf
Magic number	%PDF
Developed by	Adobe Inc. (1991–2008) 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,[6][7] 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.[8]

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.[9] These proprietary technologies are not standardized and their specification is published only on Adobe's website.[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 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

- 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:

- 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, 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 equivalent to `lineto` remain.

Historically, 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 and later 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 nine types:[14]

- Boolean values, representing *true* or *false*
- Real numbers
- Integers
- Strings, enclosed within parentheses (`(...)`) or represented as hexadecimal within single angle brackets (`<...>`). Strings may contain 8-bit characters.
- Names, starting with a forward slash (`/`)
- Arrays, ordered collections of objects enclosed within square brackets (`[...]`)
- 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 located near the end of the file and gives the byte offset of each indirect object from the start of the file.[19] 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

Within each page, there are one or multiple content streams that describe the text, vector and images being drawn on the page. The content stream is stack-based, similar to PostScript.[20]

There are two layouts to the PDF files: non-linearized (not "optimized") and linearized ("optimized"). Non-linearized 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. Linearized 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 all objects required for the first page to display are optimally organized at the start of the file.[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 2.0) 25 graphics state properties, of which some of the most important are:

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

Vector graphics

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

Raster images

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

- *ASCII85Decode*, a filter used to put the stream into 7-bit ASCII,
- *ASCIIHexDecode*, similar to ASCII85Decode but less compact,
- *FlateDecode*, a commonly used filter based on the deflate algorithm defined in RFC 1951 (ht tps://datatracker.ietf.org/doc/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://datatracke r.ietf.org/doc/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, and
- *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*.

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.

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*.[22] 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.[23] Fonts may be substituted if they are not embedded in a PDF.

Within text strings, characters are shown using *character codes* (integers) that map to glyphs in the current font using an *encoding*. There are several 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 <u>Windows</u> and <u>Macintosh</u> 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).[24] 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 PDF 1.3 and earlier specifications. As a result, files that use a small amount of transparency might view acceptably by older viewers, but files making extensive use of transparency could be viewed incorrectly by an older viewer.

The transparency extensions are based on the key concepts of *transparency groups*, *blending modes*, *shape*, and *alpha*. The model is closely aligned with the features of <u>Adobe Illustrator</u> version 9. The <u>blend modes</u> were based on those used by <u>Adobe Photoshop</u> 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.[25]

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

Additional features

Logical structure and accessibility

A "tagged" PDF (see clause 14.8 in ISO 32000) includes document structure and semantics information to enable reliable text extraction and <u>accessibility</u>. 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.[26]

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 <u>assistive technology</u> (AT), is uneven as of 2021.[27] 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 viewers. 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.

Encryption and signatures

A PDF file may be encrypted, for security, in which case 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, to provide secure authentication; 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.

The standard security provided by 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.[28] 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.[29][30][31]

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[32]

- To save the PDF document along with a 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.[32]

Under specific circumstances including non-patched systems of the receiver, the information the receiver of a digital signed document sees can be manipulated by the sender after the document has been signed by the signer.[33]

PAdES (*PDF Advanced Electronic Signatures*) is a set of restrictions and extensions to PDF and ISO 32000-1[34] making it suitable for advanced electronic signatures. This is published by ETSI as TS 102 778.[35]

File attachments

PDF files can have file attachments which processors may access and open or save to a local filesystem.[36]

Metadata

PDF files can contain two types of metadata.[37] The first is the Document Information Dictionary, a set of key/value fields such as author, title, subject, creation and update dates. This is optional and is referenced from `Info` key in the trailer of the file. A small set of fields is defined and can be extended with additional text values if required. This method is deprecated in PDF 2.0.

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

PDF documents can also contain display settings, including the page display layout and zoom level in a Viewer Preferences object. Adobe Reader uses these settings to override the user's default settings when opening the document.[38] The free Adobe Reader cannot remove these settings.

Accessibility

PDF files can be created specifically to be accessible for people with disabilities.[39][40][41][42][43] 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.[44][45] Leading screen readers, including JAWS, Window-Eyes, Hal, and Kurzweil 1000 and 3000 can read tagged PDF.[46][47] Moreover, tagged PDFs can be re-flowed and magnified for readers with visual impairments. Adding tags to older PDFs and those that are generated from scanned documents can present some challenges.

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

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.

Multimedia

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

Forms

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:[32][48][49][50]

- AcroForms (also known as Acrobat forms), introduced in the PDF 1.2 format specification and included in all later PDF specifications.
- XML Forms Architecture (XFA) forms, introduced in the PDF 1.5 format specification. Adobe XFA Forms are not compatible with AcroForms.[51] XFA was deprecated from PDF with PDF 2.0.

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):[32]

HTML Form format
> HTML 4.01 Specification since PDF 1.5; HTML 2.0 since 1.2

Forms Data Format (FDF)
> based on PDF, 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 it 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. FDF was originally defined in 1996 as part of ISO 32000-2:2017.

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) 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, Page's 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*.[52] 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. 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 the corresponding PDF interactive form. As of August, 2019, XFDF 3.0 is an ISO/IEC standard under the formal name *ISO 19444-1:2019 - Document management — XML Forms Data Format — Part 1: Use of ISO 32000-2 (XFDF 3.0)*.[53] This standard is a normative reference of ISO 32000-2.

PDF

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

AcroForms can keep form field values in external stand-alone files containing key-value pairs. The external files may use Forms Data Format (FDF) and XML Forms Data Format (XFDF) files.[54][52][55] 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.[32]

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

Licensing

Anyone may create applications that can read and write PDF files without having to pay royalties to Adobe Systems; Adobe holds patents to PDF, but licenses them for royalty-free use in developing software complying with its PDF specification.[56]

Security

In November 2019, researchers from Ruhr University Bochum and Hackmanit GmbH published attacks on digitally signed PDFs .[57] They showed how to change the visible content in a signed PDF without invalidating the signature in 21 of 22 desktop PDF viewers and 6 of 8 online validation services by abusing implementation flaws. At the same conference, they additionally showed how to exfiltrate the plaintext of encrypted content in PDFs.[58] In 2021, they showed new so-called *shadow attacks* on PDFs that abuse the flexibility of features provided in the specification.[59] An overview of security issues in PDFs regarding denial of service, information disclosure, data manipulation, and Arbitrary code execution attacks was presented by Jens Müller.[60][61]

PDF attachments carrying viruses were first discovered in 2001. The virus, named *OUTLOOK.PDFWorm* or *Peachy*, uses Microsoft Outlook to send itself as an attached Adobe PDF file. It was activated with Adobe Acrobat, but not with Acrobat Reader.[62]

From time to time, new vulnerabilities are discovered in various versions of Adobe Reader,[63] 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.[64] 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.[65]

Software

Viewers and editors

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,[66] and most Linux distributions, LibreOffice, Microsoft Office 2007 (if updated to SP2) and later,[67] 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 allows for uploading and saving to PDF. Some web apps offer free PDF editing and annotation tools.

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]

Printing

Raster image processors (RIPs) are used to convert PDF files into a raster format suitable for imaging onto paper and other media in printers, digital production presses and prepress in a process known as rasterisation. RIPs capable of processing PDF directly include the Adobe PDF Print Engine[74] from Adobe Systems and Jaws[75] and the Harlequin RIP from Global Graphics.

In 1993, the Jaws raster image processor from Global Graphics became the first shipping prepress RIP that interpreted PDF natively without conversion to another format. The company released an upgrade to their Harlequin RIP with the same capability in 1997.[76]

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

In 2006, PDF was widely accepted as the standard print job format at the Open Source Development Labs Printing Summit. It is supported as a print job format by the Common Unix Printing System and desktop application projects such as GNOME, KDE, Firefox, Thunderbird, LibreOffice and OpenOffice have switched to emit print jobs in PDF.[78]

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

Native display model

PDF was selected as the "native" metafile format for 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 behavior can be set back to PDF if desired.

Annotation

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

Alternatives

The Open XML Paper Specification is a competing format used both as a page description language and as the native print spooler format for Microsoft Windows since Windows Vista.

Mixed Object: Document Content Architecture is a competing format. MO:DCA-P is a part of Advanced Function Presentation.

See also

- Web document

- XSL Formatting Objects

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. "What is a PDF? Portable Document Format | Adobe Acrobat DC" (https://www.adobe.com/acrobat/about-adobe-pdf.html). *www.adobe.com*. Retrieved 2021-09-17.
5. "ISO 32000-1:2008" (http://wwwimages.adobe.com/www.adobe.com/content/dam/acom/en/devnet/pdf/pdfs/PDF32000_2008.pdf) (PDF).
6. "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.
7. 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.
8. 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
9. "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)."
10. *ISO/TC 171/SC 2/WG 8 N 603 – Meeting Report* (https://web.archive.org/web/20121126013025/http://pdf.editme.com/files/pdfREF-meetings/ISO-TC171-SC2-WG8_N0603_SC2WG8_MtgRept_SLC.pdf) (PDF), 2011-06-27, archived from the original (http://pdf.editme.com/files/pdfREF-meetings/ISO-TC171-SC2-WG8_N0603_SC2WG8_MtgRept_SLC.pdf) (PDF) on 2012-11-26, "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"
11. "Embedding and publishing interactive, 3-dimensional, scientific figures in Portable Document Format (PDF) files" (https://doi.org/10.1371%2Fjournal.pone.0069446.s001). *PLoS One*. **8** (9). 2013. doi:10.1371/journal.pone.0069446.s001 (https://doi.org/10.1371%2Fjournal.pone.0069446.s001). "the implementation of the U3D standard was not complete and proprietary extensions were used."
12. Leonard Rosenthol, 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.

13. "ISO 32000-2:2020 is now available" (https://www.pdfa.org/iso-32000-22020-is-now-availabl e/). *www.pdfa.org*. 14 December 2020. Retrieved 2021-02-03.
14. "ISO 32000-2 – Document management -- Portable document format -- Part 2: PDF 2.0" (http s://www.iso.org/standard/75839.html). *www.iso.org*. Retrieved 2021-02-03.
15. "3D supported formats" (https://web.archive.org/web/20100212072951/http://www.adobe.co m/manufacturing/resources/3dformats/). Adobe. 2009-07-14. Archived from the original (http s://www.adobe.com/manufacturing/resources/3dformats/) on 2010-02-12. Retrieved 2010-02-21.
16. "Supported file formats in Acrobat and Reader" (https://helpx.adobe.com/acrobat/kb/support ed-file-formats-acrobat-reader.html#main_2D_and_3D_formats__Acrobat_9_Pro_Extended __Adobe_3D_Reviewer_). *helpx.adobe.com*. Retrieved 2020-03-22.
17. "JavaScript for Acrobat 3D | Adobe Acrobat Developer Center" (https://www.adobe.com/dev net/acrobat/javascript_acrobt_3d.html). *www.adobe.com*. Retrieved 2020-03-22.
18. Pravetz, Jim. "In Defense of COS, or Why I Love JSON and Hate XML" (https://web.archive. org/web/20140502013429/http://jimpravetz.com/blog/2012/12/in-defense-of-cos/). *jimpravetz.com*. Archived from the original on 2014-05-02.
19. Adobe Systems, PDF Reference, pp. 39–40.
20. "Working with content streams" (https://pikepdf.readthedocs.io/en/latest/topics/content_strea ms.html).pikepdf documentation.
21. "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.
22. Howard, Jacci. "Desktop Publishing: Base 14 Fonts – Definition" (https://web.archive.org/we b/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.ht m) on June 14, 2016.
23. "The PDF Font Aquarium" (http://www.planetpdf.com/planetpdf/pdfs/pdf2k/03e/merz_fontaqu arium.pdf) (PDF).
24. "PDF Reference Sixth Edition, version 1.7, table 5.11" (https://www.adobe.com/devnet/acrob at/pdfs/pdf_reference_1-7.pdf) (PDF).
25. "PDF Blend Modes Addendum" (https://www.adobe.com/content/dam/Adobe/en/devnet/pdf/ pdfs/pdf_reference_archives/blend_modes.pdf) (PDF).
26. Duff Johnson, April 22, 2004 What is Tagged PDF? (https://web.archive.org/web/200408071 32851/http://www.planetpdf.com/enterprise/article.asp?ContentID=6067)
27. "Is PDF accessible?" (https://www.washington.edu/doit/pdf-accessible?1002=). *DO-IT - Disabilities, Opportunities, Internetworking, and Technology*. washington.edu. 8 April 2021.
28. "FreeMyPDF.com – Removes passwords from viewable PDFs" (http://freemypdf.com/). *freemypdf.com*.
29. Jeremy Kirk (December 5, 2008). "Adobe admits new PDF password protection is weaker" (http://www.macworld.com/article/1137343/pdf.html). *Macworld*.
30. Bryan Guignard. "How secure is PDF" (https://www.cs.cmu.edu/~dst/Adobe/Gallery/PDFsec urity.pdf) (PDF).
31. "PDF Security Overview: Strengths and Weaknesses" (http://www.planetpdf.com/planetpdf/p dfs/pdf2k/01W/merz_securitykeynote.pdf) (PDF).
32. 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/PDF320 00_2008.pdf) (PDF), retrieved 2010-02-19
33. "PDF Insecurity Website" (https://pdf-insecurity.org/signature-shadow/shadow-attacks.html). *pdf-insecurity.org*.

34. "ISO 32000-1:2008 Document management -- Portable document format -- Part 1: PDF 1.7" (http://www.iso.org/iso/catalogue_detail.htm?csnumber=51502). International Organization for Standardization ISO. Retrieved 22 March 2016.

35. "ETSI TS 102 778-1 V1.1.1 (2009-07): Electronic Signatures and Infrastructures (ESI); PDF Advanced Electronic Signature Profiles; Part 1: PAdES Overview - a framework document for PAdES" (http://www.etsi.org/deliver/etsi_ts%5C102700_102799%5C10277801%5C01.01.01_60%5Cts_10277801v010101p.pdf) (PDF). European Telecommunications Standards Institute ETSI. Retrieved 22 March 2016.

36. "Links and attachments in PDFs" (https://helpx.adobe.com/uk/acrobat/using/links-attachments-pdfs.html).

37. Adobe PDF reference version 1.7 (https://www.adobe.com/devnet/acrobat/pdfs/pdf_reference_1-7.pdf), section 10.2

38. "Getting Familiar with Adobe Reader > Understanding Preferences" (http://www.adobepress.com/articles/article.asp?p=412914). Retrieved 2009-04-22.

39. "PDF Accessibility" (http://www.webaim.org/techniques/acrobat/). WebAIM. Retrieved 2010-04-24.

40. Joe Clark (2005-08-22). "Facts and Opinions About PDF Accessibility" (http://www.alistapart.com/articles/pdf_accessibility). Retrieved 2010-04-24.

41. "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.

42. "PDF Accessibility Standards v1.2" (https://www.bbc.co.uk/guidelines/futuremedia/accessibility/accessible_pdf.shtml). Retrieved 2010-04-24.

43. 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

44. LibreOffice Help – Export as PDF (http://help.libreoffice.org/Common/Export_as_PDF#PDF.2FA-1a), retrieved 2012-09-22

45. Exporting PDF/A for long-term archiving (http://www.oooninja.com/2008/01/generating-pdfa-for-long-term-archiving.html), 2008-01-11

46. 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.

47. 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

48. "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.

49. "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.

50. Demling, Peter (July 1, 2008). "Convert XFA Form to AcroForm?" (http://forums.adobe.com/thread/301733). Retrieved 2010-02-19.

51. "Migrating from Adobe Acrobat forms to XML forms" (https://web.archive.org/web/201010061 51011/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.

52. *XML Forms Data Format Specification, version 2* (https://web.archive.org/web/20180730100 811/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 2018-07-30, retrieved 2010-02-19

53. "ISO 19444-1:2019(en)" (https://www.iso.org/obp/ui/#iso:std:iso:19444:-1:ed-2:v1:en). *www.iso.org*. Retrieved 3 December 2020.

54. 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.

55. *FDF Data Exchange Specification* (https://www.adobe.com/devnet/acrobat/pdfs/fdf_data_ex change.pdf) (PDF), 2007-02-08, retrieved 2010-02-19

56. "Developer Resources" (https://web.archive.org/web/20160227041714/http://partners.adob e.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.

57. *1 Trillion Dollar Refund: How To Spoof PDF Signatures* (https://dl.acm.org/doi/10.1145/3319 535.3339812). CCS '19. ACM Digital Library, ACM SIGSAC Conference on Computer and Communications Security. 6 November 2019. pp. 1–14. doi:10.1145/3319535.3339812 (http s://doi.org/10.1145%2F3319535.3339812). ISBN 9781450367479. S2CID 199367545 (http s://api.semanticscholar.org/CorpusID:199367545).

58. *Practical Decryption exFiltration: Breaking PDF Encryption* (https://dl.acm.org/doi/10.1145/3 319535.3354214). CCS '19. ACM Digital Library, ACM SIGSAC Conference on Computer and Communications Security. 6 November 2019. pp. 15–29. doi:10.1145/3319535.3354214 (https://doi.org/10.1145%2F3319535.3354214). ISBN 9781450367479. S2CID 207959243 (https://api.semanticscholar.org/CorpusID:20795 9243).

59. "Shadow Attacks: Hiding and Replacing Content in Signed PDFs" (https://www.ndss-sympo sium.org/ndss-paper/shadow-attacks-hiding-and-replacing-content-in-signed-pdfs/). Internet Society, The Network and Distributed System Security Symposium.

60. "Processing Dangerous Paths – On Security and Privacy of the Portable Document Format" (https://www.ndss-symposium.org/ndss-paper/processing-dangerous-paths-on-security-and-privacy-of-the-portable-document-format/). Internet Society, The Network and Distributed System Security Symposium.

61. "Portable Document Flaws 101" (https://www.blackhat.com/us-20/briefings/schedule/#portab le-document-flaws--20387). Blackhat.

62. Adobe Forums, Announcement: PDF Attachment Virus "Peachy" (https://forums.adobe.com/t hread/302989), 15 August 2001.

63. "Security bulletins and advisories" (https://www.adobe.com/support/security/#readerwin). Adobe. Retrieved 2010-02-21.

64. "Steve Gibson – SecurityNow Podcast" (https://www.grc.com/sn/sn-187.txt).

65. "Malicious PDFs Execute Code Without a Vulnerability" (https://web.archive.org/web/20100 404034752/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.

66. 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-ip hone-ipad-ios-11.html). *iPhone Hacks*. Retrieved February 2, 2018.

67. "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.

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) 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. `{{cite web}}`: |last= has generic name (help)

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://cgit.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.

74. "Adobe PDF Print Engine" (https://www.adobe.com/products/pdfprintengine/overview.html). *adobe.com*.

75. "Jaws® 3.0 PDF and PostScript RIP SDK" (https://web.archive.org/web/20160305090728/http://globalgraphics.com/products/jaws_rip). *globalgraphics.com*. Archived from the original (http://www.globalgraphics.com/products/jaws_rip/) on 2016-03-05. Retrieved 2010-11-26.

76. "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.

77. 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).

78. "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.

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* (https://www.cs.nott.ac.uk/~psadb1/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. S2CID 9371237 (https://api.semanticscholar.org/CorpusID:9371237).

- PDF 2.0 "ISO 32000-2:2020(en), Document management — Portable document format — Part 2: PDF 2.0" (https://www.iso.org/standard/75839.html). *www.iso.org*. Retrieved 2020-12-16.

- 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 (ISO 32000-1:2008) (https://opensource.adobe.com/dc-acrobat-sdk-docs/pdfstandards/PDF32000_2008.pdf)

- PDF 1.7 (https://opensource.adobe.com/dc-acrobat-sdk-docs/pdfstandards/pdfreference1.7old.pdf) and errata to 1.7 (https://web.archive.org/web/20220306202833/https://www.adobe.com/content/dam/acom/en/devnet/pdf/pdf_reference_archive/pdf_17_errata.pdf)
- PDF 1.6 (https://opensource.adobe.com/dc-acrobat-sdk-docs/pdfstandards/pdfreference1.6.pdf) (ISBN 0-321-30474-8) and errata to 1.6 (https://web.archive.org/web/20220306152230/https://www.adobe.com/content/dam/acom/en/devnet/pdf/pdf_reference_archive/PDF16Errata.pdf)
- PDF 1.5 (https://opensource.adobe.com/dc-acrobat-sdk-docs/pdfstandards/pdfreference1.5_v6.pdf) and errata to 1.5 (https://web.archive.org/web/20211222122128/https://www.adobe.com/content/dam/acom/en/devnet/pdf/pdf_reference_archive/errata.txt)
- PDF 1.4 (https://opensource.adobe.com/dc-acrobat-sdk-docs/pdfstandards/pdfreference1.4.pdf) (ISBN 0-201-75839-3) and errata to 1.4 (https://web.archive.org/web/20220306152229/https://www.adobe.com/content/dam/acom/en/devnet/pdf/pdf_reference_archive/PDF14errata.txt)
- PDF 1.3 (https://opensource.adobe.com/dc-acrobat-sdk-docs/pdfstandards/pdfreference1.3.pdf) (ISBN 0-201-61588-6) and errata to 1.3 (https://web.archive.org/web/20220306152234/https://www.adobe.com/content/dam/acom/en/devnet/pdf/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://web.archive.org/web/20210116133007/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 was 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 a talk by Leonard Rosenthol (https://www.youtube.com/watch?v=poc9PVmFzpc) (45 mins) (Adobe Systems) at TUG 2007