Portable Document Format

"PDF" redirects here. For other uses, see PDF (disambiguation).

The **Portable Document Format** (**PDF**) is a file format used to present documents in a manner independent of application software, hardware, and operating systems.[3] Each PDF file encapsulates a complete description of a fixed-layout flat document, including the text, fonts, graphics, and other information needed to display it.

1 History and standardization

Main article: History and standardization of Portable Document Format

PDF was developed in the early 1990s[4] as a way to share computer documents, including text formatting and inline images.[5] It was among a number of competing formats such as DjVu, Envoy, Common Ground Digital Paper, Farallon Replica and even Adobe's own PostScript format. In those early years before the rise of the World Wide Web and HTML documents, PDF was popular mainly in desktop publishing workflows. Adobe Systems made the PDF specification available free of charge in 1993. PDF was a proprietary format controlled by Adobe, until it was officially 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]

However, there are still 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 application of the ISO 32000-1 specification. These proprietary technologies are not standardized and their specification is published only on Adobe's website.[9][10][11][12][13] Many of them are also not supported by popular third-party implementations of PDF. So when organizations publish PDFs which use these proprietary technologies, they present accessibility issues for some users.

On July 28, 2017, ISO 32000-2 was published by the ISO. [14]

2 Technical foundations

The PDF combines three technologies:

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

2.1 PostScript

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

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

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

- PDF contains tokenized and interpreted results of the PostScript source code, for direct correspondence between changes to items in the PDF page description and changes to the resulting page appearance.
- PDF (from version 1.4) supports graphic transparency; PostScript does not.
- PostScript is an interpreted programming language with an implicit global state, so instructions accompanying the description of one page can affect the

appearance of any following page. Therefore, all preceding pages in a PostScript document must be processed to determine the correct appearance of a given page, whereas each page in a PDF document is unaffected by the others. As a result, PDF viewers allow the user to quickly jump to the final pages of a long document, whereas a PostScript viewer needs to process all pages sequentially before being able to display the destination page (unless the optional PostScript Document Structuring Conventions have been carefully complied with).

3 Technical overview

3.1 File structure

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

- Boolean values, representing *true* or *false*
- Numbers
- Strings, enclosed within parentheses ((...)), may contain 8-bit characters.
- Names, starting with a forward slash (/)
- Arrays, ordered collections of objects enclosed within square brackets ([...])
- Dictionaries, collections of objects indexed by Names enclosed within double pointy brackets (<<...>>)
- Streams, usually containing large amounts of data, which can be compressed and binary
- 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. An index table, also called the cross-reference table and marked with the xref keyword, follows the main body and gives the byte offset of each indirect object from the start of the file.[17] This design allows for efficient random access to the objects in the file, and also allows for small changes to be made without rewriting the entire file (*incremental update*). 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*.

At the end of a PDF file is a trailer introduced with the trailer keyword. It contains

- A dictionary
- An offset to the start of the cross-reference table (the table starting with the xref keyword)
- And the %%EOF end-of-file marker.

The dictionary contains

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

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

3.2 Imaging model

The basic design of how graphics are represented in PDF is very similar to that of PostScript, except for the use of transparency, which was added in PDF 1.4.

PDF graphics use a device-independent Cartesian coordinate system to describe the surface of a page. A PDF page description can use a matrix to scale, rotate, or skew graphical elements. A key concept in PDF is that of the *graphics state*, which is a collection of graphical parameters that may be changed, saved, and restored by a *page description*. PDF has (as of version 1.6) 24 graphics state properties, of which some of the most important are:

- The *current transformation matrix* (CTM), which determines the coordinate system
- The *clipping path*
- The *color space*
- The *alpha constant*, which is a key component of transparency

3.2.1 Vector graphics

As in PostScript, vector graphics in PDF are constructed with *paths*. Paths are usually composed of lines and cubic Bézier curves, but can also be constructed from the outlines of text. Unlike PostScript, PDF does not allow a single path to mix text outlines with lines and curves. Paths can be stroked, filled, clipping. Strokes and fills can use any color set in the graphics state, including *patterns*.

PDF supports several types of patterns. The simplest is the *tiling pattern* in which a piece of artwork is specified to be drawn repeatedly. This may be a *colored tiling pattern*, with the colors specified in the pattern object, or an *uncolored tiling pattern*, which defers color specification to the time the pattern is drawn. Beginning with PDF 1.3 there is also a *shading pattern*, which draws continuously varying colors. There are seven types of shading pattern of which the simplest are the *axial shade* (Type 2) and *radial shade* (Type 3).

3.2.2 Raster images

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

- **ASCII85Decode** a filter used to put the stream into 7-bit ASCII
- **ASCIIHexDecode** similar to ASCII85Decode but less compact
- **FlateDecode** a commonly used filter based on the deflate algorithm defined in RFC 1951 (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)
- **LZWDecode** a filter based on LZW Compression; it can use one of two groups of predictor functions for more compact LZW compression: *Predictor 2* from the TIFF 6.0 specification and predictors (filters) from the PNG specification
- **RunLengthDecode** a simple compression method for streams with repetitive data using the run-length encoding algorithm and the image-specific filters
- **DCTDecode** a lossy filter based on the JPEG standard
- **CCITTFaxDecode** a lossless bi-level (black/white) filter based on the Group 3 or Group 4 CCITT (ITU-T) fax compression standard defined in ITU-T T.4 and T.6
- **JBIG2Decode** a lossy or lossless bi-level (black/white) filter based on the JBIG2 standard, introduced in PDF 1.4
- **JPXDecode** a lossy or lossless filter based on the JPEG 2000 standard, introduced in PDF 1.5

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

3.2.3 Text

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

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

Standard Type 1 Fonts (Standard 14 Fonts) Fourteen typefaces, known as the *standard 14 fonts*, have a special significance in PDF documents:

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

These fonts are sometimes called the *base fourteen fonts*.[19] These fonts, or suitable substitute fonts with the same metrics, should be available in most PDF readers. However, since Adobe Acrobat version 6, most of these fonts are not *guaranteed* to be available in the reader, and may only display correctly if the system has them installed.[20] Fonts may be substituted if they are not embedded in a PDF.

Encodings Within text strings, characters are shown using *character codes* (integers) that map to glyphs in the current font using an *encoding*. There are a number of predefined encodings, including *WinAnsi*, *MacRoman*, and 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).[21] The encoding mechanisms in PDF were designed for Type 1 fonts, and the rules for applying them to TrueType fonts are complex.

For large fonts or fonts with non-standard glyphs, the special encodings *Identity-H* (for horizontal writing) and *Identity-V* (for vertical) are used. With such fonts it is necessary to provide a *ToUnicode* table if semantic information about the characters is to be preserved.

3.2.4 Transparency

The original imaging model of PDF was, like PostScript's, *opaque*: each object drawn on the page completely replaced anything previously marked in the same location. In PDF 1.4 the imaging model was extended to allow transparency. When transparency is used, new objects interact with previously marked objects to produce blending effects. The addition of transparency to PDF was done by means of new extensions that were designed to be ignored in products written to the PDF 1.3 and earlier specifications. As a result, files that use a small amount of transparency might view acceptably in older viewers, but files making extensive use of transparency could be viewed incorrectly in an older viewer without warning.

The transparency extensions are based on the key concepts of *transparency groups*, *blending modes*, *shape*, and *alpha*. The model is closely aligned with the features of Adobe Illustrator version 9. The blend modes were based on those used by Adobe Photoshop at the time. When the PDF 1.4 specification was published, the formulas for calculating blend modes were kept secret by Adobe. They have since been published.[22]

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

3.3 Interactive elements

PDF files may contain interactive elements such as annotations, form fields, video and Flash animation.

Rich Media PDF is a term that is used to describe interactive content that can be embedded or linked to inside of a PDF. This content must be produced using the Flash file format. When Adobe bought Macromedia, the jewel of the company was Flash, and the Flash player was embedded inside Adobe Acrobat and Adobe Reader, removing the need for third-party plug-ins such as Flash, QuickTime, or Windows Media. Unfortunately, this caused a rift with Apple as QuickTime video was prohibited from PDF. Rich Media expert Robert Connolly believes this event triggered the war between Apple and Adobe over the Flash iPhone/iPad dispute. Rich Media PDF will not operate in Apple's iOS devices such as the iPad, and interactivity is limited.

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:[23][24][25][26]

- **AcroForms** (also known as **Acrobat forms**), introduced in the PDF 1.2 format specification and included in all later PDF specifications.
- **Adobe XML Forms Architecture (XFA)** forms, introduced in the PDF 1.5 format specification. 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.[27]

3.3.1 AcroForms

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

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

- HTML Form format (HTML 4.01 Specification since PDF 1.5; HTML 2.0 since 1.2)
- Forms Data Format (FDF)
- XML Forms Data Format (XFDF) (external XML Forms Data Format Specification, Version 2.0; supported since PDF 1.5; it replaced the "XML" form submission format defined in PDF 1.4)

- PDF (the entire document can be submitted rather than individual fields and values). (defined in PDF 1.4)

AcroForms can keep form field values in external stand-alone files containing key:value pairs. The external files may use Forms Data Format (FDF) and XML Forms Data Format (XFDF) files.[28][29][30] The usage rights (UR) signatures define rights for import form data files in FDF, XFDF and text (CSV/TSV) formats, and export form data files in FDF and XFDF formats.[23]

Forms Data Format (FDF) The Forms Data Format (FDF) is based on PDF, it uses the same syntax and has essentially the same file structure, but is much simpler than PDF, since the body of an FDF document consists of only one required object. Forms Data Format is defined in the PDF specification (since PDF 1.2). The Forms Data Format can be used when submitting form data to a server, receiving the response, and incorporating into the interactive form. It can also be used to export form data to stand-alone files that can be imported back into the corresponding PDF interactive form. Beginning in PDF 1.3, FDF can be used to define a container for annotations that are separate from the PDF document they apply to. FDF typically encapsulates information such as X.509 certificates, requests for certificates, directory settings, timestamp server settings, and embedded PDF files for network transmission.[30] The FDF uses the MIME content type application/vnd.fdf, filename extension .fdf and on Mac OS it uses file type 'FDF'.[23] Support for importing and exporting FDF stand-alone files is not widely implemented in free or freeware PDF software. For example, there is no import/export support in Evince, Okular, Poppler, KPDF or Sumatra PDF, however, Evince, Okular and Poppler support filling in of PDF Acroforms and saving filled data inside the PDF file. 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) 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, Differences and Target. In addition, XFDF does not allow the spawning, or addition, of new pages based on the given data; as can be done when using an FDF file. The XFDF specification is referenced (but not included) in PDF 1.5 specification (and in later versions). It is described separately in *XML Forms Data Format Specification*.[29] The PDF 1.4 specification allowed form submissions in XML format, but this was replaced by submissions in XFDF format in the PDF 1.5 specification. XFDF conforms to the XML standard.

As of December 2016, XFDF 3.0 is an ISO/IEC standard under the formal name *ISO 19444-1:2016 - Document management - XML Forms Data Format - Part 1: Use of ISO 32000-2 (XFDF 3.0)*.[34] This standard is listed in the *Normative references* of ISO/DIS 32000-2 and expected to be listed once that standard (PDF 2.0) will have been approved (probably 2017).

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 XFDF 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).

3.3.2 Adobe XML Forms Architecture (XFA)

In the PDF 1.5 format, Adobe Systems introduced a proprietary format for forms; Adobe XML Forms Architecture (XFA) forms. The XFA 2.02 is referenced in the PDF 1.5, 1.6 and 1.7 specifications, but was deprecated in PDF 2.0. The XFA specification was not included in ISO 32000-1 / PDF 1.7 and is only referenced as an external proprietary specification created by Adobe. Adobe XFA Forms are not compatible with AcroForms, and most PDF processors do not handle XFA content. XFA was not standardized as an ISO standard. XFA is deprecated in ISO 32000-2 (PDF 2.0).

3.4 Logical structure and accessibility

A “tagged” PDF (ISO 32000-1:2008 14.8) includes document structure and semantics information to enable reliable text extraction and accessibility. Technically speaking, tagged PDF is a stylized use of the format that builds on the logical structure framework introduced in PDF 1.3. Tagged PDF defines a set of standard structure types and attributes that allow page content (text, graphics, and images) to be extracted and reused for other purposes.[35]

Tagged PDF is not required in situations where a PDF file is intended only for print. Since the feature is optional, and since the rules for Tagged PDF as specified in ISO 32000-1 are relatively vague, support for tagged PDF amongst consuming devices, including assistive technology (AT), is uneven.[36]

An AIIM project to develop an ISO-standardized subset of PDF specifically targeted at accessibility began in 2004, eventually becoming PDF/UA.

3.5 Security and signatures

A PDF file may be encrypted for security, or digitally signed for authentication. However, since a SHA-1 collision was discovered making use of the PDF, digital signatures using SHA-1 are insecure.[37]

The standard security provided by Acrobat PDF consists of two different methods and two different passwords, *user password*, which encrypts the file and prevents opening, and *owner password*, which specifies operations that should be restricted even when the document is decrypted, which can include: printing, copying text and graphics out of the document, modifying the document, or adding or modifying text notes and AcroForm fields. The user password (controls opening) encrypts the file and requires password cracking to defeat, with difficulty depending on password strength and encryption method – it is potentially very secure (assuming good password and encryption method without known attacks). The owner password (controls operations) does not encrypt the file, and instead relies on client software to respect these restrictions, and is not secure. An "owner password" can be removed by many commonly available "PDF cracking" software, including some free online services.[38] 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.[39][40][41]

There are a number of commercial solutions including Adobe LiveCycle Rights Management and Locklizard PDF DRM[42] that are more robust means of information rights management. Not only can they restrict document access but they also reliably enforce permissions in ways that the standard security handler does not.[43]

3.5.1 Usage rights

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

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

For example, Adobe Systems grants permissions to enable additional features in Adobe Reader, using public-key cryptography. Adobe Reader verifies that the signature uses a certificate from an Adobe-authorized certificate authority. The PDF 1.5 specification declares that other PDF viewer applications are free to use this same mechanism for their own purposes.[23]

3.6 File attachments

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.

3.7 Metadata

PDF files can contain two types of metadata.[44] 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.

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

4 Intellectual property

Anyone may create applications that can read and write PDF files without having to pay royalties to Adobe Systems; Adobe holds patents to PDF, but licenses them for royalty-free use in developing software complying with its PDF specification.[45]

5 Technical issues

5.1 Accessibility

PDF files can be created specifically to be accessible for disabled people.[46][47][48][49][50] PDF file formats in use as of 2014 can include tags (XML), text equivalents, captions, audio descriptions, etc. Tagged PDF is required in the PDF/A−1a specification.[51][52] Some software can automatically produce tagged PDFs, but this feature is not always enabled by default.[53][54] Leading screen readers, including JAWS, Window-Eyes, Hal, and Kurzweil 1000 and 3000 can read tagged PDFs aloud, as can later versions of the Acrobat and Acrobat Reader programs.[55][56][57] 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.

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

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

5.2 Viruses and exploits

See also: Adobe Acrobat § Security

PDF attachments carrying viruses were first discovered in 2001. The virus, named *OUTLOOK.PDFWorm* or *Peachy*, uses Microsoft Outlook to send itself as an attachment to an Adobe PDF file. It was activated with Adobe Acrobat, but not with Acrobat Reader.[58]

From time to time, new vulnerabilities are discovered in various versions of Adobe Reader,[59] 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.[60] 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.[61]

5.3 Usage restrictions and monitoring

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 that third parties can define their own encryption systems for 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 for an end-user overview.[62] 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 is notified "... via a dialogue box that the author of the file is auditing usage of the file, and be offered the option of continuing."[63]

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 period during which 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.

5.4 Default display settings

PDF documents can contain display settings, including the page display layout and zoom level. Adobe Reader uses these settings to override the user's default settings when opening the document.[64] The free Adobe Reader cannot remove these settings.

6 Content

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

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

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

PDF 1.6 supports interactive 3D documents embedded in the PDF - 3D drawings can be embedded using U3D or PRC and various other data formats.[65][66]

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.

7 Software

For more details on this topic, see List of PDF software.

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

There are many software options for creating PDFs, including the PDF printing capabilities built into macOS 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 also allows for uploading and saving to PDF.

Raster image processors (RIPs) are used to convert PDF files into a raster format suitable for imaging onto paper and other media in printers, digital production presses and prepress in a process known as rasterisation. RIPs capable of processing PDF directly include the Adobe PDF Print Engine[68] from Adobe Systems and Jaws[69] and the Harlequin RIP from Global Graphics.

7.1 Editing

Adobe Illustrator reads and writes PDF as a semi-native format. With multipage documents, a dialog opens enabling the user to select a single page to edit. Editing paragraphs of text typically disturbs line justification and paragraph wrapping, as multiline text is converted to individual lines. In a multipage document, only the page being edited can be re-saved.

Version 0.46 and later of Inkscape allows PDF editing through an intermediate translation step involving Poppler.

Serif PagePlus can open, edit and save existing PDF documents, as well as publishing of documents created in the package.

Enfocus PitStop Pro, a plugin for Acrobat, allows manual and automatic editing of PDF files,[70] while the free Enfocus Browser makes it possible to edit the low-level structure of a PDF.[71]

Dochub, is a free online PDF editing tool that can be used without purchasing anything.[72]

7.2 Annotation

See also: Comparison of notetaking software

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 unix 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, 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 freeware

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.

For mobile annotation, iAnnotate PDF (from Branchfire) and GoodReader (from Aji) allow annotation of PDFs as well as exporting summaries of the annotations.

There are also web annotation systems that support annotation in 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. Some programs that accept ink input from the mouse may not be responsive enough for handwriting input on a tablet. Existing solutions on the PC include PDF Annotator and Qiqqa.

7.3 Other

Examples of PDF software as online services including Scribd for viewing and storing, Pdfvue for online editing, and Zamzar for conversion.

In 1993 the Jaws raster image processor from Global Graphics became the first shipping prepress RIP that interpreted PDF natively without conversion to another format. The company released an upgrade to their Harlequin RIP with the same capability in 1997.[73]

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

PDF was selected as the "native" metafile format for Mac OS X, replacing the PICT format of the earlier classic Mac OS. The imaging model of the Quartz graphics layer is based on the model common to Display PostScript and PDF, leading to the nickname *Display PDF*. The Preview application can display PDF files, as can version 2.0 and later of the Safari web browser. System-level support for PDF allows Mac OS X applications to create PDF documents automatically, provided they support the OS-standard printing architecture. The files are then exported in PDF 1.3 format according to the file header. When taking a screenshot under Mac OS X versions 10.0 through 10.3, the image was also captured as a PDF; later versions save screen captures as a PNG file, though this behaviour can be set back to PDF if desired.

In 2006 PDF was widely accepted as the standard print job format at the Open Source Development Labs Printing Summit. It is supported as a print job format by the Common Unix Printing System and desktop application projects such as GNOME, KDE, Firefox, Thunderbird, LibreOffice and OpenOffice have switched to emit print jobs in PDF.[75]

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 once considered 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."[76][77] In 2011, however, the GNU PDF project was removed from the list of "high priority projects" due to the maturation of the Poppler library,[78] which has enjoyed wider use in applications such as Evince with the GNOME desktop environment. Poppler is based on Xpdf[79][80] 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.[81]

8 See also

- Open XML Paper Specification
- Comparison of OpenXPS and PDF
- DjVu
- PAdES, PDF Advanced Electronic Signature
- Web document
- XSL Formatting Objects

9 References

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

[2] *The application/pdf Media Type, RFC 3778, Category: Informational*, 2004

[3] Adobe Systems Incorporated, PDF Reference, Sixth edition, version 1.23 (30 MB), Nov 2006, p. 33.

[4] "Adobe's Bob Wulff knows Acrobat and PDF -- inside and out".

[5] "The Camelot Project" (PDF).

[6] "ISO 32000-1:2008 - Document management – Portable document format – Part 1: PDF 1.7". Iso.org. 2008-07-01. Retrieved 2010-02-21.

[7] Orion, Egan (2007-12-05). "PDF 1.7 is approved as ISO 32000". *The Inquirer*. The Inquirer. Archived from the original on December 13, 2007. Retrieved 2007-12-05.

[8] Adobe Systems Incorporated (2008), *Public Patent License, ISO 32000-1: 2008 – PDF 1.7* (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". 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* (PDF), 2011-06-27, XFA is not to be ISO standard just yet. ... The Committee urges Adobe Systems to submit the XFA Specification, XML Forms Architecture (XFA), to ISO for standardization ... The Committee is concerned about the stability of the XFA specification ... Part 2 will reference XFA 3.1

[11] "Embedding and publishing interactive, 3-dimensional, scientificfigures in Portable Document Format (PDF) files". Retrieved 2013-10-20. ... the implementation of the U3D standard was not complete and proprietary extensions were used.

[12] Leonard Rosenthol, Adobe Systems (2012). "PDF and Standards" (PDF). Retrieved 2013-10-20.

[13] Duff Johnson (2010-06-10), *Is PDF an open standard? - Adobe Reader is the de facto Standard, not PDF*, retrieved 2014-01-19

[14] "ISO 32000-2 - Document management -- Portable document format -- Part 2: PDF 2.0". *www.iso.org*. Retrieved 2017-07-28.

[15] Jim Pravetz. "In Defense of COS, or Why I Love JSON and Hate XML". *jimpravetz.com*.

[16] Adobe Systems, PDF Reference, p. 51.

[17] Adobe Systems, PDF Reference, pp. 39–40.

[18] "Adobe Developer Connection: PDF Reference and Adobe Extensions to the PDF Specification". Adobe Systems. Retrieved 2010-12-13.

[19] "Desktop Publishing: Base 14 Fonts - Definition". *About.com Tech*.

[20] The PDF Font Aquarium

[21] "PDF Reference Sixth Edition, version 1.7, table 5.11" (PDF).

[22] PDF Blend Modes Addendum

[23] Adobe Systems Incorporated (2008-07-01), *Document Management – Portable Document Format – Part 1: PDF 1.7, First Edition* (PDF), retrieved 2010-02-19

[24] "Gnu PDF - PDF Knowledge - Forms Data Format". Archived from the original on 2013-01-01. Retrieved 2010-02-19.

[25] "About PDF forms". Retrieved 2010-02-19.

[26] "Convert XFA Form to AcroForm?". 2008. Retrieved 2010-02-19.

[27] "Migrating from Adobe Acrobat forms to XML forms". Retrieved 2010-02-22.

[28] Adobe Systems Incorporated (2007-10-15). "Using Acrobat forms and form data on the web". Retrieved 2010-02-19.

[29] *XML Forms Data Format Specification, version 2* (PDF), September 2007, retrieved 2010-02-19

[30] *FDF Data Exchange Specification* (PDF), 2007-02-08, retrieved 2010-02-19

[31] *IANA Application Media Types - vnd.fdf*, retrieved 2010-02-22

[32] *IANA Application Media Types - Vendor Tree - vnd.adobe.xfdf*, retrieved 2010-02-22

[33] *ISO/CD 19444-1 - Document management - XML Forms Data Format - Part 1: Use of ISO 32000-2 (XFDF 3.0)*, retrieved 2017-05-28

[34] "ISO 19444-1:2016 - Document management -- XML Forms Data Format -- Part 1: Use of ISO 32000-2 (XFDF 3.0)". *www.iso.org*. Retrieved 2017-02-28.

[35] What is Tagged PDF?

[36] "Is PDF accessible?". *washington.edu*.

[37] "SHAttered - We have broken SHA-1 in practice".

[38] "FreeMyPDF.com - Removes passwords from viewable PDFs". *freemypdf.com*.

[39] Jeremy Kirk. "Adobe admits new PDF password protection is weaker".

[40] Bryan Guignard. "How secure is PDF" (PDF).

[41] "PDF Security Overview: Strengths and Weaknesses" (PDF).

[42] "LockLizard Develops Zero Footprint Solution for PDF Security".

[43] "PDF DRM Security Software for Adobe Document Protection".

[44] Adobe PDF reference version 1.7, section 10.2

[45] "Developer Resources". *adobe.com*.

[46] "PDF Accessibility". WebAIM. Retrieved 2010-04-24.

[47] Joe Clark (2005-08-22). "Facts and Opinions About PDF Accessibility". Retrieved 2010-04-24.

[48] "Accessibility and PDF documents". Web Accessibility Center. Retrieved 2010-04-24.

[49] "PDF Accessibility Standards v1.2". Retrieved 2010-04-24.

[50] *PDF Accessibility* (PDF), California State University, retrieved 2010-04-24

[51] *Frequently Asked Questions (FAQs) – ISO 19005-1:2005 – PDF/A-1, Date: July 10, 2006* (PDF), 2006-07-10, retrieved 2011-07-06

[52] "PDF/A – A Look at the Technical Side" (PDF). Retrieved 2011-07-06.

[53] *LibreOffice Help - Export as PDF*, retrieved 2012-09-22

[54] *Exporting PDF/A for long-term archiving*, 2008-01-11

[55] "Adobe Reader 8 - Read a PDF with Read Out Loud". Retrieved 2010-04-24.

[56] Biersdorfer, J.D. (2009-04-10). "Tip of the Week: Adobe Reader's 'Read Aloud' Feature". *The New York Times*. Retrieved 2010-04-24.

[57] *Accessing PDF documents with assistive technology: A screen reader user's guide* (PDF), Adobe, retrieved 2010-04-24

[58] Adobe Forums, Announcement: PDF Attachment Virus "Peachy", 15 August 2001.

[59] "Security bulletins and advisories". Adobe. Retrieved 2010-02-21.

[60] Steve Gibson - SecurityNow Podcast

[61] "Malicious PDFs Execute Code Without a Vulnerability". *PCMAG*.

[62] "Create Adobe PDF Online - Security Settings Help". Createpdf.adobe.com. Retrieved 2010-02-21.

[63] New features and issues addressed in the Acrobat 7.0.5 Update (Acrobat and Adobe Reader for Windows and Mac OS)

[64] "Getting Familiar with Adobe Reader > Understanding Preferences". Retrieved 2009-04-22.

[65] "3D supported formats". Adobe. 2009-07-14. Retrieved 2010-02-21.

[66] "Acrobat 3D Developer Center". Adobe. Retrieved 2010-02-21.

[67] "Description of 2007 Microsoft Office Suite Service Pack 2 (SP2)". Microsoft. Retrieved 2009-05-09.

[68] "Adobe PDF Print Engine". *adobe.com*.

[69] "Jaws® 3.0 PDF and PostScript RIP SDK". *globalgraphics.com*.

[70] "Preflight and edit PDF files in Acrobat". *enfocus.com*.

[71] "Enfocus product overview - online store". *enfocus.com*.

[72] "DocHub". *DocHub*. Retrieved 2015-12-12.

[73] "Harlequin MultiRIP". Retrieved 2014-03-02.

[74] 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).

[75] "PDF as Standard Print Job Format". *The Linux Foundation*. Linux Foundation. Retrieved 21 June 2016.

[76] On 2014-04-02, a note dated 2009-02-10 referred to Current FSF High Priority Free Software Projects as a source. Content of the latter page, however, changes over time.

[77] GNUpdf contributors (2007-11-28). "Goals and Motivations". *gnupdf.org*. *GNUpdf*. Retrieved 2014-04-02.

[78] Lee, Matt (2011-10-06). "GNU PDF project leaves FSF High Priority Projects list; mission complete!". *fsf.org*. Free Software Foundation. Retrieved 2014-04-02.

[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 or 3." (last checked on 2012-09-23).

[81] The Apache PDFBox project . Retrieved 2009-09-19.

10 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* (PDF). Proceedings of the 2002 ACM symposium on Document engineering. pp. 95–102. ISBN 1-58113-594-7. doi:10.1145/585058.585077.
- Standards
  - PDF 1.7
  - PDF 1.6 (ISBN 0-321-30474-8)
  - PDF 1.4 (ISBN 0-201-75839-3)
  - PDF 1.3 (ISBN 0-201-61588-6)

11 External links

- How was the PDF created? Quora
- PDF Association - The PDF Association is the industry association for software developers producing or processing PDF files.
- Adobe PDF 101: Summary 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 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)
- ISO 19005-1:2005 the PDF/A-1 ISO standard published by the International Organization for Standardization (chargeable)
- PDF Reference and Adobe Extensions to the PDF Specification
- Portable Document Format: An Introduction for Programmers – Introduction 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
- Everything you wanted to know about PDF but were afraid to ask - recording of talk by Leonard Rosenthol (Adobe Systems) at TUG 2007
- How to produce PDF with XSL-FO
- PDF To Excel Converter

12 Text and image sources, contributors, and licenses

12.1 Text

- **Portable Document Format** *Source:* https://en.wikipedia.org/wiki/Portable_Document_Format?oldid=795970897 *Contributors:* Mav, The Anome, Stephen Gilbert, LA2, Enchanter, Jdpipe, Dwheeler, Leandrod, Edward, Nealmcb, Patrick, RTC, Michael Hardy, Nixdorf, Liftarn, Menchi, TakuyaMurata, Miciah, CesarB, Ahoerstemeier, Mac, Ronz, Samuelsen, ?alabio, Randywombat, Andrewa, Aarchiba, Salsa Shark, Glenn, Stefan-S, Dpol, Marknew, Jonik, GRAHAMUK, Ehn, Hashar, Emperorbma, Crissov, Andrevan, Reddi, Viajero, Dysprosia, Zoicon5, Dougjih, Kierant, AndrewKepert, Bevo, Traroth, Nickshanks, AnonMoos, Ldo, Shantavira, Wilinckx~enwiki, Donarreiskoffer, Robbot, Mazin07, Pigsonthewing, Fredrik, R3m0t, RedWolf, Chocolateboy, Dittaeva, Nurg, Modulatum, Nbarr, Rfc1394, TimR, Kielsky, Hemanshu, Timrollpickering, Benc, Mattflaschen, Tea2min, David Gerard, Honta, Giftlite, DocWatson42, Kerttie, Nick8325, Elf, ShaunMacPherson, Inter, Luis Dantas, Lupin, Mark Richards, Bfinn, Tsca, Jdavidb, Thetorpedodog, Danio, Vunzmstr, Zhen Lin, AlistairMcMillan, Pne, Uzume, Thewikipedian, Chowbok, Saxsux, Coldacid, Uranographer, Toytoy, Quadell, Antandrus, Beland, OverlordQ, Kusunose, Mamizou, Jossi, Exigentsky, Taka, Oneiros, MFNickster, Buchs, DragonflySixtyseven, Wilhelmtell, Supadawg, Marc Mongenet, Fusion, Hellisp, Ojw, Neale Monks, Ukexpat, Addicted, Squash, Neschek, Grunt, Danc, Corti, Mormegil, Poccil, Lubaf, Discospinster, Rich Farmbrough, FT2, Kdammers, Pmsyyz, Qutezuce, GeoEvan, Notinasnaid, Joeclark, Horkana, Fluzwup, LeeHunter, Mani1, Hhielscher, Bender235, Thebrid, Plugwash, Danakil, Nabla, Perspective, Art LaPella, Vdm, NetBot, Whosyourjudas, John Vandenberg, Shenme, Blakkandekka, Polluks, Redquark, Shlomital, Boredzo, Larryv, Minghong, SilasM, Hagerman, Pearle, Justinc, Landroni, Frodet, Beinsane, DavidCiani, Gary, Apatterno, Carrasco, Quiggles, Guy Harris, CyberSkull, Jeltz, Yamla, Water Bottle, Ayeroxor, Velella, Ronark, ProhibitOnions, Evil Monkey, Gpvos, Zoohouse, VoluntarySlave, Versageek, MIT Trekkie, Netkinetic, Feline1, Kitch, Forderud, Amitpagarwal, Stemonitis, Simetrical, Woohookitty, RHaworth, Brazil4Linux, Georgia guy, Guy M, Deeahbz, Jacobolus, Armando, Michael R Bax, MattGiuca, Pol098, Josephw, Trapolator, Tabletop, Firien, GregorB, Jetpeach, Joke137, Toussaint, KoRnholio8, PhilippWeissenbacher, DaveTheRed, Kesla, Graham87, Zerblatt, BD2412, FreplySpang, Snafflekid, Mlewan, Sjö, Demian12358, Koavf, Vary, Hiberniantears, Gudeldar, Datapharmer, Ttwaring, Amelio Vázquez, Utuado, Tvhuang, Boris Alexeev, FlaBot, Weihao.chiu~enwiki, Juliank, Nihiltres, Alexjohnc3, Intgr, Lmatt, Davidbrake, Malhonen, Srleffler, Coolhawks88, Chobot, Cbrandlehner, Tene, Bgwhite, Mrnatural, Shervinafshar, YurikBot, Noclador, Wavelength, Jim.Liu, Eirik, Klingoncowboy4, Hawaiian717, RobotE, Todd Vierling, Pmg, J. M., Cliffb, KSmrq, SpuriousQ, Hydrargyrum, Chensiyuan, Gaius Cornelius, Canageek, Akhristov, Member, Rkrater, Wiki alf, Joshf, Grafen, ErkDemon, D. F. Schmidt, Chichui, Alex43223, Bota47, Eriktown, D-Day, Ziggiz, Mholland, Zzuuzz, Ninly, Reyk, JahJah, TimOertel, David Biddulph, ViperSnake151, NeilN, GrinBot~enwiki, Owl-syme, Finell, A bit iffy, SmackBot, Timoey, Aido2002, BeeBot, Brianyoumans, KnowledgeOfSelf, Gigs, Unyoyega, Solaris smoke, Verne Equinox, Pandion auk, Eskimbot, AnOddName, Norz, Scott Paeth, Septegram, Yamaguchi□□, Gilliam, SylvainBeucler, Ohnoitsjamie, Simon123, Thumperward, Oli Filth, BrandonCsSanders, EncMstr, SentientSoul241, Deli nk, Jerome Charles Potts, Nbarth, InfinityDUCK, DHN-bot~enwiki, CDV, Can't sleep, clown will eat me, Frap, Jacob Poon, Richardcrocker1, Behlal, GeorgeMoney, Bennose, Cybercobra, Dream out loud, Markhobley, Mwtoews, Kfoss, RayGates, Daniel.Cardenas, Jóna Þórunn, J.smith, □□□, Ugur Basak Bot~enwiki, Michaelejahn, Spiritia, AThing, KingpinE7, Soap, Mpbailey, Copeland.James.H, Sundström, Adam7davies, BladeHamilton, IronGargoyle, JWaters, Mr. Vernon, Jec, Randomtime, Fisel, Hotmop, EdC~enwiki, MTSbot~enwiki, SirJohn~enwiki, Darry2385, Hgrobe, Mdanh2002, Lee Carre, Hu12, Pjrm, Iridescent, Dreftymac, Colonel Warden, GMcGath, IvanLanin, Rreeves, Ouzo~enwiki, MGlosenger, Gligi~enwiki, Chovain, DaCropduster, FairuseBot, Cornlad, INkubusse, Mstorer, The Font, BeenAroundAWhile, Engelec, NaBUru38, Matthew T, Signo, Aubrey Jaffer, Michael B. Trausch, Krauss, Frzl, Gogo Dodo, GuyIncognito, Richard.shakeshaft, Emmanuel Vallois, Tawkerbot4, Zian, Teratornis, Kozuch, Shrish, Thijs!bot, Egriffin, Anne226, JukkaZitting, Al Lemos, Bobblehead, Eatrains, Pltrgyst, Escarbot, Ratherthanlater, Gioto, Kenahoo, Lovibond, Jj137, Dylan Lake, Sjledet, Pseudo account, JustJake, Rbb l181, JAnDbot, Dereckson, Nannus, Barek, MER-C, Skomorokh, NE2, Fetchcomms, Rothorpe, HAl, SteveSims, AshishKulkarni, Magioladitis, VoABot II, Ishikawa Minoru, Lotusv82, Spartan26, Cic, Indon, Ensign beedrill, PatialaPeg, Allstarecho, Martynas Patasius, Vssun, Shijualex, Rezd, JaGa, B9 hummingbird hovering, Xoid, AVRS, Conquerist, Mbufkin, Bobcat hokie, Skyhawker666, Zeus, Lapis.br, Panarchy, Rettetast, Drewmutt, Keith D, Glrx, CommonsDelinker, Ramesh Chandra, Dudley Miles, Ssolbergj, Pharaoh of the Wizards, MikeSims, Numbo3, Jesant13, Mrhsj, Laurusnobilis, Zero3Cool, LordAnubisBOT, Nemo bis, Theo148, Luke1138, SJP, Duffjohnson, Jmcw37, Zanzibarfiction, Ansiansiansi, Raspalchima, Tbhartman, Jevansen, ACBest, Treisijs, Fuenfundachtzig, Bonadea, TWCarlson, Mikedesmith, Boombaard, Pborenstein, RJASE1, Funandtrvl, Priceman86, Crisperdue, TreasuryTag, ABF, Jeff G., Johnny Au, Senachie, MrSerif, Bortek, Barneca, TXiKiBoT, Oshwah, A4bot, Joe2832, Tuanese, Scde, Qxz, Jeanhaney, DaBest1, Anna Lincoln, Rich Janis, AndrewJD, LeaveSleaves, Jorophose, PDFbot, Wykypydya, Shaileshpatel1234546, Pascal sa, Webeasy5, Meters, Lamro, Falcon8765, Enviroboy, LittleBenW, WIKIPEDIA DESTROYER 4, Melrayes, Colm.mcmullan, Ponyo, SieBot, Dexterj, TJRC, YonaBot, Malcolmxl5, MJaggard, Xenophon777, Happysailor, Toddst1, EeshEesh, Yerpo, Oxymoron83, Callidior, Hobartimus, Int21h, Ctxppc, Svick, Silvergoat, Dezwitser, Maralia, Khecaji, Pinkadelica, Denisarona, ShelleyAdams, Aidnieci, ImageRemovalBot, Martarius, ClueBot, Smart Viral, Ebatalha, Rrider, Acrobatfiles, Cochonfou, Boing! said Zebedee, Timberframe, ChandlerMapBot, Vofficeware, Sv1xv, Excirial, Anonymous101, Alexbot, Serviendo, Eujin16, PixelBot, Bvconway, Rhododendrites, NuclearWarfare, Arjayay, Jotterbot, Botsjeh, TobiasPersson, Revotfel, Nukeless, Minterior, Thingg, Versus22, Stevenrasnick, Humanengr, Tdslk, DumZiBoT, XLinkBot, Jamalofski, Stickee, SJ Morg, Little Mountain 5, Poikiloid, Skarebo, RP459, Addbot, Mwahaj, Ghettoblaster, Ph wiki edit, Yoenit, Nepfan91, LockLizard, Mabdul, JoeLibrarian, AttoRenato, Reasoft, Mootros, Ronhjones, Jncraton, MagnusA.Bot, MrOllie, Lihaas, Jasper Deng, Sandrarossi, Maria.yudina, Sabrina07080236, F Notebook, Gail, Zorrobot, Fiftyquid, CountryBot, Rav3naiah, Chiborg, Waltloc, Frehley, Legobot, Luckas-bot, Yobot, Themfromspace, EdwardFlach, KamikazeBot, TestEditBot, AnomieBOT, Morris lin, Rubinbot, Jim1138, Galoubet, Tucoxn, Player82, Mike McL, Materialscientist, Abelardia, Dirick, Maxis ftw, Quebec99, OlgaVN, TheRealNightRider, Keevie, Xqbot, Portable d, Gilo1969, Quillaja, Flamingo bob, Nasa-verve, Mich5482, JohnFlood, RibotBOT, Stoufville, Doulos Christos, Kurapix, Soystuff, Josemanimala, Prepressx, EdHubertson, Dohocheol, Jc3s5h, X7q, Kjvr, Joker1984, Brentbucci, NyBroadway, BenzolBot, Tamariki, Ndayakar, Calmer Waters, Xiaoshuang, Serols, SpaceFlight89, Dougk7, Mehanika, TobeBot, DixonDBot, Rlanman, Dinamik-bot, Lrosenthol, Milan292208, 7H4 6U713RR3Z, DARTH SIDIOUS 2, Mean as custard, ArwinJ, RjwilmsiBot, Bentleyd, Ripchip Bot, Hannahe2009, Beleg Tâl, Rollins83, Bernd Schwabe in Hannover, EmausBot, John of Reading, WikitanvirBot, Racerx11, Sumsum2010, Japs 88, GoingBatty, The Mysterious El Willstro, Your Lord and Master, Dcirovic, Albert.meronyo, Its snowing in East Asia, Don-hide, Josve05a, AlexTheHorrible, The Nut, Idhodwiq, SporkBot, Otzithecaveman, Aerotee965, Donner60, Tijfo098, ChuispastonBot, Atmalla, Senator2029, AzertyFab, 5gXhlp2fd, Alan-hicks-london, Yerffog2, ClueBot NG, Bridgeway04, O.Koslowski, Widr, Antiqueight, Xyrph, Johnny C. Morse, Ffirmin, Ndavidow, Titodutta, Yuan2085, Brianeversole, BG19bot, Wz0911, Vagobot, WikiTryHardDieHard, Bogdan.nemteanu, ShawnaMcAlearney, Jherbach, Nsda, Markgavin pa, Mark Arsten, Fitoblok, Supernerd11, NGIGrace, Sunnyamirr, Toccata quarta, Whitegum, Sparthorse, Rahul0001, Hohohohobo360, BattyBot, Sarabutton, Justincheng12345-bot, Yeshizena, Kayo Katsuragi, Tonyxc600, Cyberbot II, Talghatuly, CF259, Dexbot, Rezonansowy, Codename Lisa, Mogism, Shrutipatankar, Sammaz, Zziccardi, Plangrid, Shambhusharan, C5st4wr6ch, Mark viking, Aura707, Tentinator, Rybec, Prncsngh, Kmbremer, Uwe Lück, Kalsira, Sarr X, Spike0xff, BerkeleyArchivist, Masterofgolf, Noobletzz, Manfromcoimbatore, JaconaFrere, Edison Biadora, Yvesmitt, Grohn, Phleg1, Magenta1984, Filedelinkerbot, Alexoett, Annastacy, Wikimaster62, WanderingLost, A Great Catholic Person, Ashlayclark, RK Vemula, KH-1, TheMagikCow, Zppix, Stevenjhodgin, 7degreedarkness, SweetCanadianMullet, Cbelth, Davidnotcoulthard, Scarfacedude, [illegible], Amccann421, King756, Mohd Arif Warsi, JJMC89, DavOsk, Daosk, EnCo, CAPTAIN RAJU, Babymissfortune, Prap2015, JayWH2, Marianna251, GreenC bot, MissDyne, Aristos20, KinjiNyoba16, Celsogouveia1, Akiplatao, Do not sack this wiki., Nerd1a4i, Joshbell2008, Monty358, Magic links bot, Gsrinivasan071142, ManDan and Anonymous: 892

12.2 Images

- **File:Commons-logo.svg** *Source:* https://upload.wikimedia.org/wikipedia/en/4/4a/Commons-logo.svg *License:* PD *Contributors:* ? *Original artist:* ?
- **File:Crystal_Clear_device_cdrom_unmount.png** *Source:* https://upload.wikimedia.org/wikipedia/commons/1/10/Crystal_Clear_device_cdrom_unmount.png *License:* LGPL *Contributors:* All Crystal Clear icons were posted by the author as LGPL on kde-look; *Original artist:* Everaldo Coelho and YellowIcon;
- **File:EBookreal.jpg** *Source:* https://upload.wikimedia.org/wikipedia/commons/e/eb/EBookreal.jpg *License:* CC-BY-SA-3.0 *Contributors:* Own work *Original artist:* Jorghex
- **File:Folder_Hexagonal_Icon.svg** *Source:* https://upload.wikimedia.org/wikipedia/en/4/48/Folder_Hexagonal_Icon.svg *License:* Cc-by-sa-3.0 *Contributors:* ? *Original artist:* ?
- **File:Symbol_neutral_vote.svg** *Source:* https://upload.wikimedia.org/wikipedia/en/8/89/Symbol_neutral_vote.svg *License:* Public domain *Contributors:* ? *Original artist:* ?
- **File:Wiki_letter_w_cropped.svg** *Source:* https://upload.wikimedia.org/wikipedia/commons/1/1c/Wiki_letter_w_cropped.svg *License:* CC-BY-SA-3.0 *Contributors:* This file was derived from Wiki letter w.svg:
  Original artist: Derivative work by Thumperward

12.3 Content license