I'm not a bot

Trasformare in pdf/a

Application for creating and editing PDFs PDF24 CreatorDeveloper(s)geek Software GmbHInitial releaseJuly 2006; 18 years ago (2006-07) [1]Stable release11.23.0[2] / January 13, 2025; 4 months ago (2025-01-13) Operating systemMicrosoft WindowsTypePDF printer/creatorLicensefreeware[3]Websiteen.pdf24.org PDF24 Creator is an application software by Geek Software GmbH for the creation of PDF files from any application and for converting files to the PDF format. The application is released under a proprietary freeware license. The software has been developed in Germany since 2006, originally under the name PDFDrucker,[1] and is actively developed. It is available in 32 languages,[4] including English and German. PDF24 Creator is installed as a virtual printer via a device driver in the operating system. This allows PDF files to be created directly from any application that provides a printing function. The commands sent are then used to create a PDF file. PDF24 Creator uses PDFBox, QPDF and Ghostscript, which are automatically installed as a private instances for the PDF24 Creator. After printing a document on the PDF printer, a wizard opens automatically, where the created PDF file can be edited or further worked on. The PDF24 Creator is also able to merge multiple documents to one PDF file and to extract pages. Compressing PDF files to shrink the file size is also possible. Since version 10.0.0 an added toolbox is present as well. Some features of the software include, but are not limited to:[5][6] Merge multiple PDF into one file Rotating, extracting, inserting pages Integrated preview for PDF editing PDF encryption, decryption and signing Change PDF information (author, title, etc.) Compress and shrink PDF files Add a watermark or stamp a PDF file Combine pages with a digital paper Convert to and from PDF Multiple PDF printers for different purposes since 7.7.0 Full featured and lightweight PDF reader since version 8.7.0 Tesseract OCR engine since version 8.8.0 Blackening of PDF files since version 10.0.0 PDF24 Creator is credited with more than 5 million downloads[7] within the top 3 (2017-01-17)[8] in the category "PDF Software" on the German site Chip.de. List of PDF software List of virtual printer software ^ a b "PDFDrucker". pdf24.org (in German). Archived from the original on 2006-07-02. Retrieved 2023-08-30. ^ "PDF24 Creator Changelog". en.pdf24.org. 2024-09-23. Retrieved 2024-02-16. ^ "License" (PDF). en.pdf24.org. Archived from the original (PDF) on August 10, 2018. Retrieved 2017-01-16. ^ "PDF24 Creator Language File Translation". en.pdf24.org. Retrieved 2017-01-16. ^ "Features of PDF24 PDF Creator". en.pdf24.org. Retrieved 2017-01-16. ^ "PDF24 PDF Creator". Gizmo's Freeware. 2016-04-23. Archived from the original on 2017-01-18. Retrieved 2017-01-17. ^ "PDF24 Creator". CHIP Online. Retrieved 2017-01-16. ^ "PDF Software Top 100 Downloads des Monats - CHIP". www.chip.de. Retrieved 2017-01-17. Official website Retrieved from " Portable Document Format, a digital file format For other uses, see PDF (disambiguation). Portable Document FormatAdobe PDF iconFilename extension.pdfInternet media type application/pdf,[1] application/x-pdf application/x-bzpdf application/x-gzpdf Type codePDF [1] (including a single trailing space)Uniform Type Identifier (UTI)com.adobe.pdfMagic number%PDFDeveloped byAdobe Inc. (1991–2008) ISO (2008–)Initial releaseJune 15, 1993; 31 years ago (1993-06-15)Latest release2.0 Extended toPDF/A, PDF/E, PDF/UA, PDF/VT, PDF/XStandardISO 32000-2Open format?YesWebsiteiso.org/standard/75839.html 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.[6] 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. Main article: History of PDF The development of PDF began in 1991 when John Warnock wrote a paper for a project then code-named Camelot, in which he proposed the creation of a simplified version of PostScript called Interchange PostScript (IPS).[7] Unlike traditional PostScript, which was tightly focused on rendering print jobs to output devices, IPS would be optimized for displaying pages to any screen and any platform.[7] Adobe Systems made the PDF specification available free of charge in 1993. In the early years PDF was popular mainly in desktop publishing workflows, and competed with several other formats, including DjVu, Envoy, Common Ground Digital Paper, Farallon Replica and even Adobe's own PostScript format. 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,[8][9] at which time control of the specification passed to an ISO Committee of volunteer industry experts. In 2008, Adobe published a Public Patent License to ISO 32000-1 granting royalty-free rights for all patents owned by Adobe necessary to make, use, sell, and distribute PDF-compliant implementations.[10] 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.[11] These proprietary technologies are not standardized, and their specification is published only on Adobe's website.[12][13][14] Many of them are not supported by popular third-party implementations of PDF. ISO published version 2.0 of PDF, ISO 32000-2 in 2017, available for purchase, replacing the free specification provided by Adobe.[15] In December 2020, the second edition of PDF 2.0, ISO 32000-2:2020, was published, with clarifications, corrections, and critical updates to normative references[16] (ISO 32000-2 does not include any proprietary technologies as normative references).[17] In April 2023 the PDF Association made ISO 32000-2 available for download free of charge.[15] A PDF file is often a combination of vector graphics, text, and bitmap graphics. The basic types of content in a PDF are: Typeset text stored as content streams (i.e., not encoded in plain text); Vector graphics for illustrations and designs that consist of shapes and lines; Raster graphics for photographs and other types of images; and Other multimedia objects. In later PDF revisions, a PDF document can also support links (inside document or web page), forms, JavaScript (initially available as a plugin for Acrobat 3.0), or any other types of embedded contents that can be handled using plug-ins. PDF combines three technologies: An equivalent subset of the PostScript page description programming language but in declarative form, for generating the layout and graphics. A font-embedding/replacement system to allow fonts to travel with the documents. A structured storage system to bundle these elements and any associated content into a single file, with data compression where appropriate. PostScript is a page description language run in an interpreter to generate an image.[7] It can handle graphics and has standard features of programming languages such as branching and looping.[7] PDF is a subset of PostScript, simplified to remove such control flow features, while graphics commands remain.[7] PostScript was originally designed for a drastically different use case: transmission of one-way linear print jobs in which the PostScript interpreter would collect a series of commands until it encountered the showpage command, then execute all the commands to render a page as a raster image to a printing device.[18] PostScript was not intended for long-term storage and real-time interactive rendering of electronic documents to computer monitors, so there was no need to support anything other than consecutive rendering of pages.[18] If there was an error in the final printed output, the user would correct it at the application level and send a new print job in the form of an entirely new PostScript file. Thus, any given page in a PostScript file could be accurately rendered only as the cumulative result of executing all preceding commands to draw all previous pages—any of which could affect subsequent pages—plus the commands to draw that particular page, and there was no easy way to bypass that process to skip around to different pages.[18] Traditionally, to go from PostScript to PDF, a source PostScript file (that is, an executable program) is used as the basis for generating PostScript-like PDF code (see, e.g., Adobe Distiller). This is done by applying standard compiler techniques like loop unrolling, inlining and removing unused branches, resulting in code that is purely declarative and static.[18] The result is then packaged into a container format, together with all necessary dependencies for correct rendering (external files, graphics, or fonts to which the document refers), and compressed. Modern applications write to printer drivers that directly generate PDF rather than going through PostScript first. As a document format, PDF has several advantages over PostScript: PDF contains only static declarative PostScript code that can be processed as data, and does not require a full program interpreter or compiler.[18] This avoids the complexity and security risks of an engine with such a higher complexity level. Like Display PostScript, PDF has supported transparent graphics since version 1.4, while standard PostScript does not. PDF enforces the rule that the code for any particular page cannot affect any other pages.[18] That rule is strongly recommended for PostScript code too, but has to be implemented explicitly (see, e.g., the Document Structuring Conventions), as PostScript is a full programming language that allows for such greater flexibilities and is not limited to the concepts of pages and documents. All data required for rendering is included within the file itself, improving portability.[19] Its disadvantages are: A loss of flexibility, and limitation to a single use case.[citation needed] A (sometimes much) larger file size.[20] PDF since v1.6 supports embedding of interactive 3D documents: 3D drawings can be embedded using U3D or PRC and various other data formats.[21][22][23] A PDF file is organized using ASCII characters, except for certain elements that may have binary content. The file starts with a header containing a magic number (as a readable string) and the version of the format, for example %PDF-1.7. The format is a subset of a COS ("Carousel" Object Structure) format.[24] A COS tree file consists primarily of objects, of which there are nine types:[17] 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 Comments using 8-bit characters prefixed with the percent sign (%) may be inserted. 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.[25] 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.[26] The maximum size of an Acrobat PDF page, superimposed on a map of Europe. 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.[27] PDF files may be optimized using Adobe Acrobat software or QPDF. Page dimensions are not limited by the format itself. However, Adobe Acrobat imposes a limit of 15 million by 15 million inches, or 225 trillion in2 (145,161 km2).[2]:1129 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) 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 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 (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, 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 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.[28] 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.[29] 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 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).[2] 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. A text document which is scanned to PDF without the text being recognised by optical character recognition (OCR) is an image, with no fonts or text properties. The original imaging model of PDF was opaque, similar to PostScript, where 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 be viewed 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 Adobe Illustrator version 9. The blend modes were based on those used by Adobe Photoshop at the time. When the PDF 1.4 specification was published, the formulas for calculating blend modes were kept secret by Adobe. They have since been published.[30] The concept of a transparency group in PDF specification is independent of existing notions of "group" or "layer" in applications such as Adobe Illustrator. Those groupings reflect logical relationships among objects that are meaningful when editing those objects, but they are not part of the imaging model. See also: PDF/A-1 and PDF/UA A tagged PDF (see clause 14.8 in ISO 32000) includes document structure and semantics information to enable reliable text extraction and accessibility.[31] 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.[32] 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 among consuming devices, including assistive technology (AT), is uneven as of 2021.[33] 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. With the introduction of PDF version 1.5 (2003) came the concept of Layers. Layers, more formally known as 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. 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 the 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 are 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.[34] 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 copies of excerpts of the text as if the document were not limited by password protection.[35][36][37] 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:[38] To save the PDF document along with a modified form 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.[38] 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.[39] PAdES (PDF Advanced Electronic Signatures) is a set of restrictions and extensions to PDF and ISO 32000-1[40] making it suitable for advanced electronic signatures. This is published by ETSI as TS 102 778.[41] PDF files can have file attachments which processors may access and open or save to a local filesystem.[42] PDF files can contain two types of metadata.[2] 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 an 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, and 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.[43] The free Adobe Reader cannot remove these settings. PDF files can be created specifically to be accessible to people with disabilities.[44][45][46][47][48] PDF file formats in use as of 2014[update] 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.[49][50] Leading screen readers, including JAWS, Window-Eyes, Hal, and Kurzweil 1000 and 3000 can read tagged PDFs.[51][52] 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. Rich Media PDF is a PDF file including interactive content that can be embedded or linked within the file. It can contain images, audio, video content, or buttons. For example, if the interactive PDF is a digital catalog for an E-commerce business, products can be listed on the PDF pages and can be added with images and links to the website and buttons to order directly from the document. 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:[38][53][54][55] 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.[56] XFA was deprecated from PDF with PDF 2.0. AcroForms were introduced in the PDF 1.2 format. AcroForms permit the uses of 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):[38] 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.[citation needed] 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.[57] 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).[58] 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.[59][57][60] 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.[38] 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). 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.[61] See also: Adobe Acrobat § Security In November 2019, researchers from Ruhr University Bochum and Hackmanit GmbH published attacks on digitally signed PDFs.[62] 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.[63] In 2021, they showed new so-called shadow attacks on PDFs that abuse the flexibility of features provided in the specification.[64] 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.[65][66] PDF files can be infected with viruses, Trojans, and other malware. They can have hidden JavaScript code that might exploit vulnerabilities in a PDF, hidden objects executed when the file that hides them is opened, and, less commonly, a malicious PDF can launch malware.[67] 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.[68] From time to time, new vulnerabilities are discovered in various versions of Adobe Reader,[69] 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.[70] 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.[71] For a more comprehensive list, see List of PDF software. Many PDF viewers are provided free of charge from a variety of sources. Programs to manipulate and edit PDF files are available, usually for purchase. There are many software options for creating PDFs, including the PDF printing capabilities built into macOS, iOS,[72] and most Linux distributions. Much document processing software including LibreOffice, Microsoft Office 2007 (if updated to SP2) and later,[73] WordPerfect 9, and Scribus can export documents in PDF. There are many 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, that allow a "PDF printer" to be set up, which when selected sends output to a PDF file instead of a physical printer. Google's online office suite Google Docs allows uploading and saving to PDF. Some web apps offer free PDF editing and annotation tools. The Free Software Foundation was "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", as one of its high priority projects.[74][75] In 2011, however, the GNU PDF project was removed from the list of "high priority projects" due to the maturation of the Poppler library,[76] which has enjoyed wider use in applications such as Evince with the GNOME desktop environment. Poppler is based on Xpdf[77][78] 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, licensed under the Apache License, for working with PDF documents.[79] 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 rasterization. RIPs capable of processing PDF directly include the Adobe PDF Print Engine[80] from Adobe Systems and Jaws[81] 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 its Harlequin RIP with the same capability in 1997.[82] 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.[83] 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.[84] Some desktop printers also support direct PDF printing, which can interpret PDF data without external help. This section does not cite any sources. Please help improve this section by adding citations to reliable sources. Unsourced material may be challenged and removed. (November 2023) (Learn how and when to remove this message) PDF was selected as the "native" metafile format for macOS (originally called 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 macOS 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. This section does not cite any sources. Please help improve this section by adding citations to reliable sources. Unsourced material may be challenged and removed. (November 2023) (Learn how and when to remove this message) See also: Comparison of note-taking 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. 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 document formats. In cases where PDFs are expected to have all of the functionality of paper documents, ink annotation is required. PDF's emphasis on preserving the visual appearance of documents across different software and hardware platforms poses challenges to the conversion of PDF documents to other file formats and the targeted extraction of information, such as text, images, tables, bibliographic information, and document metadata. Numerous tools and source code libraries support these tasks. Several labeled datasets to test PDF conversion and information extraction tools exist and have been used for benchmark evaluations of the tool's performance.[85] Main article: Open XML Paper Specification § Comparison with PDF See also: EPUB 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. ebook Web page XSL Formatting Objects Page margin PDF portfolio ^ a b Hardy, M.; Masinter, L.; Markovic, D.; Johnson, D.; Bailey, M. (March 2017). The application/pdf Media Type. IETF. doi:10.17487/RFC8118. RFC 8118. ^ a b c d Adobe Systems Incorporated (November 2006). "PDF Reference" (PDF). 1.7 (6th ed.). Archived from the original (PDF) on October 1, 2008. Retrieved January 12, 2023. ^ Warnock, J. (October 14, 2004) [Original date 5 May 1995]. "The Camelot Project" (PDF). Archived (PDF) from the original on July 18, 2011. ^ "What is a PDF? Portable Document Format | Adobe Acrobat DC". Adobe Systems Inc. Archived from the original on January 30, 2023. Retrieved January 12, 2023. ^ "ISO 32000-1:2008" (PDF). Archived from the original (PDF) on July 26, 2018. ^ "ISO 32000-2 – PDF Association". Retrieved January 27, 2025. ^ a b c d e Pfiffner, Pamela (2003). Inside the Publishing Revolution: The Adobe Story. Berkeley: Peachpit Press. p. 137. ISBN 0-321-11564-3. ^ "ISO 32000-1:2008 – Document management – Portable document format – Part 1: PDF 1.7". ISO. July 1, 2008. Archived from the original on December 6, 2010. Retrieved February 21, 2010. ^ Orion, Egan (December 5, 2007). "PDF 1.7 is approved as ISO 32000". The Inquirer. Archived from the original on December 13, 2007. Retrieved December 5, 2007. ^ "Public Patent License, ISO 32000-1: 2008 – PDF 1.7" (PDF). Adobe Systems Inc. 2008. Archived (PDF) from the original on June 18, 2009. Retrieved January 12, 2023. ^ "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. June 25, 2013. Archived from the original on September 19, 2020. Retrieved January 12, 2023. 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). ^ "ISO/TC 171/SC 2/WG 8 N 603 – Meeting Report" (PDF). Edit me. June 27, 2011. Archived from the original (PDF) on November 26, 2012 – via Archive. 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 ^ "Embedding and publishing interactive, 3-dimensional, scientific figures in Portable Document Format (PDF) files". PLOS ONE. 8 (9). 2013. doi:10.1371/journal.pone.0069446.s001. the implementation of the U3D standard was not complete and proprietary extensions were used. ^ Rosenthol, Leonard (2012). "PDF and Standards" (PDF). Adobe Systems. Archived from the original (PDF) on September 2, 2013. Retrieved October 20, 2013 – via Parleys. ^ a b "Announcing no-cost access to the latest PDF standard: ISO 32000-2 (PDF 2.0)" (Press release). PDF Association. June 16, 2023 [Updated; originally published 5 April 2023]. Archived from the original on September 23, 2023. Retrieved October 6, 2023. ^ "ISO 32000-2:2020 is now available". PDFA. December 14, 2020. Archived from the original on December 4, 2022. Retrieved February 3, 2021. ^ a b "ISO 32000-2 – Document management — Portable document format — Part 2: PDF 2.0". ISO. January 5, 2021. Archived from the original on January 28, 2021. Retrieved February 3, 2021. ^ a b c d e f Pfiffner, Pamela (2003). Inside the Publishing Revolution: The Adobe Story. Berkeley: Peachpit Press. p. 139. ISBN 0-321-11564-3. ^ "PostScript Language Reference" (PDF). Archived from the original (PDF) on July 24, 2021. ^ Anton Ertl, Martin. "What is the PDF format good for?". complang.tuwien.ac.at. Vienna University of Technology. Archived from the original on April 4, 2024. Retrieved April 8, 2024. ^ "3D supported formats". Adobe Systems Inc. July 14, 2009. Archived from the original on February 12, 2010. Retrieved February 21, 2010. ^ "Supported file formats in Acrobat and Reader". Adobe Systems Inc. November 11, 2022. Archived from the original on December 21, 2022. Retrieved January 12, 2023. ^ "JavaScript for Acrobat 3D | Adobe Acrobat Developer Center". Adobe Systems Inc. Archived from the original on November 12, 2009. Retrieved January 12, 2023. ^ Pravetz, Jim. "In Defense of COS, or Why I Love JSON and Hate XML". jimpravetz.com. Archived from the original on May 2, 2014. ^ Adobe Systems, PDF Reference, pp. 39–40. ^ PikePdf documentation. "Working with content streams". Archived from the original on July 5, 2022. Retrieved May 8, 2022. ^ "Adobe Developer Connection: PDF Reference and Adobe Extensions to the PDF Specification". Adobe Systems Inc. Archived from the original on November 15, 2006. Retrieved December 13, 2010. ^ Howard, Jacci. "Desktop Publishing: Base 14 Fonts – Definition". About.com Tech. Archived from the original on June 14, 2016. ^ Merz, Thomas (June 2003). "The PDF Font Aquarium" (PDF). Archived from the original on July 18, 2011. ^ "PDF Blend Modes Addendum" (PDF). Archived from the original (PDF) on October 14, 2011. Retrieved January 12, 2023. ^ "Tagged PDF Best Practice Guide: Syntax" (PDF). pdfa.org. PDF Association. June 2019. Retrieved June 24, 2024. ^ Johnson, Duff (April 22, 2004). "What is Tagged PDF?". Archived from the original on August 7, 2004. ^ "Is PDF accessible?". DO-IT - Disabilities, Opportunities, Internetworking, and Technology. University of Washington. October 4, 2022. Archived from the original on February 10, 2023. Retrieved January 12, 2023. ^ "FreeMyPDF.com – Removes passwords from viewable PDFs". freemypdf.com. Archived from the original on February 20, 2021. Retrieved June 23, 2009. ^ Kirk, Jeremy (December 4, 2008). "Adobe admits new PDF password protection is weaker". Macworld. IDG Communications Inc. Archived from the original on January 17, 2017. Retrieved September 14, 2016. ^ Guignard, Bryan. "How secure is PDF" (PDF). Carnegie Mellon University. Archived from the original (PDF) on October 24, 2005. ^ Merz, Thomas (November 2001). PDF Security Overview: Strengths and Weaknesses (PDF). PDF 2001 conference. Scottsdale/Arizona. Archived from the original on October 11, 2010. ^ a b c d e Adobe Systems Inc. (July 1, 2008). "Document Management – Portable Document Format – Part 1: PDF 1.7, First Edition" (PDF). Archived from the original (PDF) on December 3, 2008. Retrieved January 12, 2023. ^ "PDF Insecurity Website". pdf-insecurity.org. Archived from the original on March 26, 2023. Retrieved January 12, 2023. ^ "ISO 32000-1:2008 Document management -- Portable document format -- Part 1: PDF 1.7". International Organization for Standardization ISO. Archived from the original on February 10, 2017. Retrieved March 22, 2016. ^ "ETSI TS 102 778-1 - Electronic Signatures and Infrastructures (ESI); PDF Advanced Electronic Signature Profiles; Part 1: PAdES Overview - a framework document for PAdES" (PDF). 1.1.1. European Telecommunications Standards Institute ETSI. July 2009. Archived (PDF) from the original on March 8, 2023. Retrieved January 12, 2023. ^ "Links and attachments in PDFs". Archived from the original on April 23, 2021. Retrieved April 23, 2021. ^ "Getting Familiar with Adobe Reader > Understanding Preferences". Adobe Press. Pearson. September 2, 2005. Archived from the original on October 23, 2012. Retrieved January 12, 2023. ^ "PDF Accessibility". WebAIM. Archived from the original on January 12, 2023. Retrieved January 12, 2023. ^ Clark, Joe (August 22, 2005). "Facts and Opinions About PDF Accessibility". Archived from the original on January 24, 2013. Retrieved January 12, 2023. ^ "Accessibility and PDF documents". Web Accessibility Center. The Ohio State University. Archived from the original on April 27, 2010. Retrieved January 12, 2023. ^ "PDF Accessibility Standards". 1.2. BBC. Archived from the original on May 29, 2010. Retrieved January 12, 2023. ^ "PDF Accessibility" (PDF). California State University. 2009. Archived from the original (PDF) on May 27, 2010. Retrieved January 12, 2023. ^ "LibreOffice Help – Export as PDF". Archived from the original on January 12, 2023. Retrieved January 12, 2023. ^ Z., Andrew (January 11, 2008). "Exporting PDF/A for long-term archiving". Archived from the original on February 24, 2021. Retrieved September 22, 2012. ^ Biersdorfer, J.D. (April 10, 2009). "Tip of the Week: Adobe Reader's 'Read Aloud' Feature". The New York Times. Archived from the original on November 22, 2020. Retrieved January 12, 2023. ^ "Accessing PDF documents with assistive technology: A screen reader user's guide" (PDF). Adobe Systems Inc. Archived from the original (PDF) on July 28, 2008. Retrieved January 12, 2023. ^ "Gnu PDF – PDF Knowledge – Forms Data Format". Archived from the original on January 1, 2013. Retrieved January 12, 2023. ^ "About PDF forms". Adobe Systems Inc. Archived from the original on April 29, 2011. Retrieved February 19, 2010. ^ Demling, Peter (July 1, 2008). "Convert XFA Form to AcroForm?". Archived from the original on January 12, 2023. Retrieved January 12, 2023. ^ "Migrating from Adobe Acrobat forms to XML forms". Archived from the original on October 6, 2010. Retrieved January 12, 2023. ^ a b "XML Forms Data Format Specification, version 2" (PDF). September 2007. Archived from the original (PDF) on July 30, 2018. Retrieved February 19, 2010. ^ "ISO 19444-1:2019(en)". The International Organization for Standardization. Archived from the original on June 17, 2016. Retrieved December 3, 2020. ^ Adobe Systems Incorporated (September 20, 2022). "Using Acrobat forms and form data on the web". Archived from the original on January 12, 2023. Retrieved January 12, 2023. ^ "FDF Data Exchange Specification" (PDF). February 8, 2007. Archived from the original (PDF) on December 3, 2008. Retrieved January 12, 2023. ^ "Developer Resources". Adobe Systems Inc. Archived from the original on February 27, 2016. ^ Mladenov, Vladislav; Mainka, Christian; Meyer Zu Selhausen, Karsten; Grothe, Martin; Schwenk, Jörg (November 6, 2019). "1 Trillion Dollar Refund: How to Spoof PDF Signatures". Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security. CCS '19. ACM Digital Library, ACM SIGSAC Conference on Computer and Communications Security. pp. 1–14. doi:10.1145/3319535.3339812. ISBN 9781450367479. S2CID 199367545. Archived from the original on April 26, 2021. Retrieved April 6, 2021. ^ Müller, Jens; Ising, Fabian; Mladenov, Vladislav; Mainka, Christian; Schinzel, Sebastian; Schwenk, Jörg (November 6, 2019). "Practical Decryption exFiltration: Breaking PDF Encryption". Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security. CCS '19. ACM Digital Library, ACM SIGSAC Conference on Computer and Communications Security. pp. 15–29. doi:10.1145/3319535.3354214. ISBN 9781450367479. S2CID 207959243. Archived from the original on April 26, 2021. Retrieved April 6, 2021. ^ "Shadow Attacks: Hiding and Replacing Content in Signed PDFs". Internet Society, The Network and Distributed System Security Symposium. Archived from the original on April 21, 2021. Retrieved April 6, 2021. ^ "Processing Dangerous Paths – On Security and Privacy of the Portable Document Format". Internet Society, The Network and Distributed System Security Symposium. Archived from the original on April 21, 2021. Retrieved April 6, 2021. ^ "Portable Document Flaws 101". Blackhat. Archived from the original on April 9, 2021. Retrieved April 6, 2021. ^ "Can PDFs have viruses? Keep your files safe". Adobe. Archived from the original on October 4, 2023. Retrieved October 3, 2023. ^ Adobe Forums, Announcement: PDF Attachment Virus "Peachy" Archived September 4, 2015, at the Wayback Machine, August 15, 2001. ^ "Security bulletins and advisories". Adobe Systems Inc. January 10, 2023. Archived from the original on April 6, 2010. Retrieved January 12, 2023. ^ Gibson, Steve; Laporte, Leo (March 12, 2009). "Steve Gibson – SecurityNow Podcast". Archived from the original on May 8, 2020. Retrieved January 11, 2011. ^ "Malicious PDFs Execute Code Without a Vulnerability". PCMAG. Archived from the original on April 4, 2010. ^ Pathak, Khamosh (October 7, 2017). "How to Create a PDF from Web Page on iPhone and iPad in iOS 11". iJunkie. Archived from the original on January 12, 2023. Retrieved January 12, 2023. ^ "Description of 2007 Microsoft Office Suite Service Pack 2 (SP2)". Microsoft. Archived from the original on April 29, 2009. Retrieved January 12, 2023. ^ On 2014-04-02, a note dated February 10, 2009 referred to Current FSF High Priority Free Software Projects Archived August 10, 2007, at the Wayback Machine as a source. Content of the latter page, however, changes over time. ^ "Goals and Motivations". gnupdf.org. GNUpdf. November 28, 2007. Archived from the original on July 4, 2014. Retrieved April 2, 2014. ^ Lee, Matt (October 6, 2011). "GNU PDF project leaves FSF High Priority Projects list; mission complete!". fsf.org. Free Software Foundation. Archived from the original on December 28, 2014. ^ "Poppler Homepage". Archived from the original on January 8, 2015. Retrieved January 12, 2023. Poppler is a PDF rendering library based on the xpdf-3.0 code base. ^ "Xpdf License". Archived from the original on April 14, 2013. Retrieved January 12, 2023. Xpdf is licensed under the GNU General Public License (GPL), version 2 or 3. ^ "The Apache PDFBox project- Apache PDFBox 3.0.0 released". August 17, 2023. Archived from the original on January 7, 2023. Updated for new releases. ^ "Adobe PDF Print Engine". Adobe Systems Inc. Archived from the original on August 22, 2013. Retrieved August 20, 2014. ^ "Jaws® 3.0 PDF and PostScript RIP SDK". globalgraphics.com. Archived from the original on March 5, 2016. Retrieved November 26, 2010. ^ "Harlequin MultiRIP". Archived from the original on February 9, 2014. Retrieved March 2, 2014. ^ "Press-Ready PDF Files". Archived from the original on February 5, 2009. Retrieved January 12, 2023. For anyone interested in having their graphic project commercially printed directly from digital files or PDFs. ^ "PDF as Standard Print Job Format". The Linux Foundation. Linux Foundation. October 23, 2009. Archived from the original on November 14, 2009. Retrieved January 12, 2023. ^ Meuschke, Norman; Jagdale, Apurva; Spinde, Timo; Mitrović, Jelena; Gipp, Bela (2023), Sserwanga, Isaac; Goulding, Anne; Moulaison-Sandy, Heather; Du, Jia Tina (eds.), "A Benchmark of PDF Information Extraction Tools Using a Multi-task and Multi-domain Evaluation Framework for Academic Documents", Information for a Better World: Normality, Virtuality, Physicality, Inclusivity, vol. 13972, Cham: Springer Nature Switzerland, pp. 383–405, arXiv:2303.09957, doi:10.1007/978-3-031-28032-0_31, ISBN 978-3-031-28031-3 ISO Standards PDF 2.0 "ISO 32000-2:2020(en), Document management — Portable document format — Part 2: PDF 2.0". International Organization for Standardization. Retrieved December 16, 2020. PDF 2.0 "ISO 32000-2:2017(en), Document management — Portable document format — Part 2: PDF 2.0". International Organization for Standardization. August 3, 2017. Retrieved January 31, 2019. Adobe open source standards PDF 1.7 (ISO 32000-1:2008) PDF 1.7 and errata to 1.7 at the Wayback Machine (archived March 6, 2022) PDF 1.6 (ISBN 0-321-30474-8) and errata to 1.6 at the Wayback Machine (archived March 6, 2022) PDF 1.5 and errata to 1.5 at the Wayback Machine (archived December 22, 2021) PDF 1.4 (ISBN 0-201-75839-3) and errata to 1.4 at the Wayback Machine (archived March 6, 2022) PDF 1.3 (ISBN 0-201-61588-6) and errata to 1.3 at the Wayback Machine (archived March 6, 2022) PDF 1.2 PDF 1.0 (ISBN 0-201-62628-4) Conference papers Hardy, M. R. B.; Brailsford, D. F. (2002). "Mapping and displaying structural transformations between XML and PDF" (PDF). Proceedings of the 2002 ACM symposium on Document engineering – DocEng '02. pp. 95–102. doi:10.1145/585058.585077. ISBN 1-58113-594-7. S2CID 9371237. Archived from the original (PDF) on March 24, 2017.[relevant?] Wikimedia Commons has media related to Portable Document Format. PDF Association – The PDF Association is the industry association for software developers producing or processing PDF files. PDF Specification Index at the PDF Association PDF Cheat Sheets, 2nd edition by the PDF Association (last updated October 29, 2024) Sponsored free access to the ISO 32000-2 (PDF 2.0) bundle, including the latest core PDF specification and five ISO standardized extensions to the core specification Format description of the PDF family, PDF/A, PDF/X from Library of Congress Tech notes from Adobe Adobe PDF 101: Summary of PDF at the Wayback Machine (archived 2010-10-07) Adobe: PostScript vs. PDF at the Wayback Machine (archived 2016-04-13) – Official introductory comparison of PS, EPS vs. PDF. PDF Reference and Adobe Extensions to the PDF Specification at the Wayback Machine (archived 2021-01-16) Articles Portable Document Format: An Introduction for Programmers (1999) from MacTech – Introduction to PDF vs. PostScript and PDF internals (up to v1.3) John Warnock's 'Camelot' signalled birth of PDF (2002) at the Wayback Machine (archived 2019-04-22) from Planet PDF that describes the paper in which John Warnock outlined the project that created PDF Videos Video: Everything you wanted to know about PDF but was afraid to ask — Recording of a talk by Leonard Rosenthol (Adobe Systems) at TUG 2007 Retrieved from " Free and easy to use PDF solutions to solve PDF problems. PDF24 provides tools to merge PDF, compress PDF, edit PDF, PDF converter, PDF editor and much more.