

CHAPTER 1

Introduction

The Adobe Portable Document Format (PDF) is the native file format of the Adobe® Acrobat® family of products. The goal of these products is to enable users to exchange and view electronic documents easily and reliably, independently of the environment in which they were created. PDF relies on the same imaging model as the PostScript® page description language to describe text and graphics in a device-independent and resolution-independent manner. To improve performance for interactive viewing, PDF defines a more structured format than that used by most PostScript language programs. PDF also includes objects, such as annotations and hypertext links, that are not part of the page itself but are useful for interactive viewing and document interchange.

1.1 About This Book

This book provides a description of the PDF file format and is intended primarily for developers of *PDF producer* applications that create PDF files directly. It also contains enough information to allow developers to write *PDF consumer* applications that read existing PDF files and interpret or modify their contents.

Although the *PDF Reference* is independent of any particular software implementation, some PDF features are best explained by describing the way they are processed by a typical application program. In such cases, this book uses the Acrobat family of PDF viewer applications as its model. (The prototypical viewer is the fully capable Acrobat product, not the limited Adobe Reader® product.) Appendix C discusses some implementation limits in the Acrobat viewer applications, even though these limits are not part of the file format itself. Appendix H provides compatibility and implementation notes that describe how Acrobat viewers behave when they encounter newer features they do not understand and specify areas in which the Acrobat products diverge from the specification presented in

this book. Implementors of PDF producer and consumer applications can use this information as guidance.

This edition of the *PDF Reference* describes version 1.7 of PDF. (See implementation note 1 in Appendix H.) Throughout the book, information specific to particular versions of PDF is marked with indicators such as (*PDF 1.3*) or (*PDF 1.4*). Features so marked may be new or substantially redefined in that version. Features designated (*PDF 1.0*) have generally been superseded in later versions; unless otherwise stated, features identified as specific to other versions are understood to be available in later versions as well. (PDF consumer applications designed for a specific PDF version generally ignore newer features they do not recognize; implementation notes in Appendix H point out exceptions.)

Note: *In this edition, the term consumer is generally used to refer to PDF processing applications; viewer is reserved for applications that implement features that interact with users. This distinction is not always clear, however, since non-interactive applications may process objects in PDF documents (such as annotations) that represent interactive features.*

The rest of the book is organized as follows:

- Chapter 2, “Overview,” briefly introduces the overall architecture of PDF and the design considerations behind it, compares it with the PostScript language, and describes the underlying imaging model that they share.
- Chapter 3, “Syntax,” presents the syntax of PDF at the object, file, and document level. It sets the stage for subsequent chapters, which describe how that information is interpreted as page descriptions, interactive navigational aids, and application-level logical structure.
- Chapter 4, “Graphics,” describes the graphics operators used to describe the appearance of pages in a PDF document.
- Chapter 5, “Text,” discusses PDF’s special facilities for presenting text in the form of character shapes, or glyphs, defined by fonts.
- Chapter 6, “Rendering,” considers how device-independent content descriptions are matched to the characteristics of a particular output device.
- Chapter 7, “Transparency,” discusses the operation of the transparent imaging model, introduced in PDF 1.4, in which objects can be painted with varying degrees of opacity, allowing the previous contents of the page to show through.

- Chapter 8, “Interactive Features,” describes those features of PDF that allow a user to interact with a document on the screen by using the mouse and keyboard.
- Chapter 9, “Multimedia Features,” describes those features of PDF that support embedding and playing multimedia content, including video, music and 3D artwork.
- Chapter 10, “Document Interchange,” shows how PDF documents can incorporate higher-level information that is useful for the interchange of documents among applications.
- Appendix A, “Operator Summary,” lists all the operators used in describing the visual content of a PDF document.
- Appendix B, “Operators in Type 4 Functions,” summarizes the PostScript operators that can be used in PostScript calculator functions, which contain code written in a small subset of the PostScript language.
- Appendix C, “Implementation Limits,” describes typical size and quantity limits imposed by the Acrobat viewer applications.
- Appendix D, “Character Sets and Encodings,” lists the character sets and encodings that are assumed to be predefined in any PDF consumer application.
- Appendix E, “PDF Name Registry,” discusses a registry, maintained for developers by Adobe Systems, that contains private names and formats used by PDF producers or Acrobat plug-in extensions.
- Appendix F, “Linearized PDF,” describes a special form of PDF file organization designed to work efficiently in network environments.
- Appendix G, “Example PDF Files,” presents several examples showing the structure of actual PDF files, ranging from one containing a minimal one-page document to one showing how the structure of a PDF file evolves over the course of several revisions.
- Appendix H, “Compatibility and Implementation Notes,” provides details on the behavior of Acrobat viewer applications and describes how consumer applications should handle PDF files containing features that they do not recognize.
- Appendix I, “Computation of Object Digests,” describes in detail an algorithm for calculating an object digest (discussed in Section 8.7, “Digital Signatures”).

A color plate section provides illustrations of some of PDF's color-related features. References in the text of the form "see Plate 1" refer to the contents of this section.

The book concludes with a Bibliography and an Index.

1.2 Introduction to PDF 1.7 Features

Several features have been introduced or modified in PDF 1.7. The following is a list of the most significant additions, along with references to the primary sections where those additions are discussed:

1.2.1 Presentation of 3D Artwork

PDF 1.7 introduces new features that increase the control the PDF viewing application has over the appearance and behavior of 3D artwork:

- More control over the appearance of 3D artwork, without having to change the original artwork and without the use of embedded JavaScript. Specific views of 3D artwork can specify how that artwork should be rendered, colored, lit, and cross-sectioned. They can also specify which nodes (three-dimensional areas) of 3D artwork should be included in a view, where those nodes should be placed in the view, and whether they should be transparent. These features can expose areas of geometry that would otherwise be difficult to view.
- The ability to place markup annotations on specific views of 3D artwork. This ensures that markups applied to 3D artwork can later be shown properly with respect to both the artwork as a whole and individual elements within the artwork. Markup annotations applied to 3D artwork provide a means of ensuring the artwork has not changed since the markup annotation was applied.
- Control over the user interfaces and toolbars presented on activation of 3D artwork.
- Control over the timeframe, repetition, and style of play of keyframe animations. The styles of play are linear repetition (as in a walking character) and a cosine-based repetition (as in an exploding-contracting image).

1.2.2 Interactive Features

Several additions to markup annotations make them more suitable for technical communication and review, or for use in a legal setting.

Interactive Features That Aid Technical Communication

Several additions to markup annotations aid technical communication and review:

- The addition of dimension intents for polyline and polygon markup annotations. Dimension intent supports the association of user-provided dimension information with the line segments that compose polyline and polygon markup annotations. This feature is similar to the dimension intent introduced for line markup annotations in PDF 1.6.
- The ability to specify units and scaling for the dimension intents of line, polyline, and polygon markup annotations. This feature enables users to measure distances in the document, such as the width of an architectural diagram or the diameter of a 3D cross section.
- The ability to place markup annotations on specific views of 3D artwork
- The ability to lock the contents of an annotation

Interactive Feature for Use in a Legal Setting

One addition to markup annotations is intended for use in a legal setting, especially banking. The addition of new viewer preference settings that specify print characteristics, such as paper selection and handling, page range, copies, and scaling. When a user prints a PDF document with those viewer preference settings, the print dialog is pre-populated as specified in those settings. This capability increases the predictability of how PDF documents are printed, which can make PDF documents more suitable for use in a legal setting.

1.2.3 Accessibility Related Features

Additions to TaggedPDF identify the roles of more types of page content:

- The ability to identify the roles of form fields in non-interactive PDF documents. This change identifies button fields (pushbuttons, check boxes and radio buttons) and text fields (populated or unpopulated).

- The ability to provide table summaries associated with table structures. This feature can help a visually impaired person understand the purpose and structure of a table without having to read the content in that table.
- The ability to identify background page artifacts, which can be important to document reflowing. Background artifacts are collections of objects that do not contribute to the meaning of the author's original content, such as a colored rectangle behind a sidebar or a full-page background image. Such page backgrounds may not correlate to any logical structure, but they may be useful in reproducing the appearance of original document.
- The ability to differentiate the pagination artifacts: watermarks, headers and footers.

1.2.4 Document Navigation Feature

Additions to document navigation specify the viewing and organizational characteristics of *portable collections*, in which multiple file attachments are displayed within a single window. Portable collections are used to present, sort, and search collections of related documents, such as email archives, photo collections, and engineering bid sets.

1.2.5 Security-Related Features

Additions to PDF introduced in 1.7 increase the control the document author can impose upon digital signatures and over requirements PDF consumer applications must satisfy:

- Additional digital signature constraints, which are enforced at the time the signature is applied. These constraints include preferred digest methods, revocation checking of the certificate used in a signature, and flags that clarify the interpretation of other parameters.
- Additional constraints regarding the certificate to be used when signing. These constraints include Subject Distinguished Name (DN) dictionaries that must be present in the certificate, KeyUsage extensions that must be present in the signing certificate, and flags that clarify the interpretation of other parameters that specify certificate constraints.
- The ability to specify requirement handlers that verify some requirement that the PDF consumer applications must satisfy before processing or displaying a PDF document. This feature provides an approach that ensures backward com-

patibility with PDF documents that may include JavaScript segments to verify a requirement. Before this feature was added, JavaScript was the only way to perform such requirement-checking. The feature ensures that either the JavaScript segment verifies the requirement or a named handler verifies the requirement.

1.2.6 General Features

Additions to PDF 1.7 provide more cross-platform and cross-application stability, by providing encoding information for strings and file names:

- The clarification of string types to describe the encodings used for strings. Throughout the entire PDF Reference, any uses of the string type are replaced with one of the more specific string types. This clarification does not require changes to PDF consumer applications. Instead, it provides a clearer understanding of the encoding supported by each PDF string entry. This understanding can be especially important when comparing strings in a PDF document to strings in an external source, such as an XML document or 3D artwork.
- The ability to specify file names using Unicode in addition to specifying file names using the standard encoding for the platform on which the document is being viewed. This feature reduces problems in decoding file path names that have been encoded on a different platform or in a different language.

1.2.7 PDF Reference Changes

This release of the *PDF Reference* includes clarifications not related to new features or additional capabilities:

- A description of the formulas for all blend modes.
- An explanation of the TaggedPDF representation of nested table of contents entries or list entries.

1.3 Related Publications

PDF and the PostScript page description language share the same underlying Adobe imaging model. A document can be converted straightforwardly between PDF and the PostScript language; the two representations produce the same output when printed. However, PostScript includes a general-purpose programming language framework not present in PDF. The *PostScript Language Reference* is the comprehensive reference for the PostScript language and its imaging model.

PDF and PostScript support several standard formats for font programs, including Adobe Type 1, CFF (Compact Font Format), TrueType, OpenType and CID-keyed fonts. The PDF manifestations of these fonts are documented in this book. However, the specifications for the font files themselves are published separately, because they are highly specialized and are of interest to a different user community. A variety of Adobe publications are available on the subject of font formats. The Bibliography lists these publications, as well as additional documents related to PDF and the contents of this book.

1.4 Intellectual Property

Adobe owns copyrights in the *PDF Reference*. Adobe will enforce its copyrights. One reason Adobe must retain its copyrights in the *PDF Reference* is to maintain the integrity of the Portable Document Format standard and ensure that the public can distinguish between the Portable Document Format and other interchange formats for electronic documents. Nonetheless, Adobe desires to promote the use of the Portable Document Format for information interchange among diverse products and applications. Accordingly, Adobe gives permission to everyone under its copyrights to copy, modify, and distribute any example code in the written specification, to the extent necessary to implement the Portable Document Format in a manner compliant with the *PDF Reference*.¹

Adobe Systems Incorporated and its subsidiaries own a number of patents covering technology disclosed in the *PDF Reference*. Nothing in the *PDF Reference* itself grants rights under any patent. Nonetheless, Adobe desires to encourage implementation of the PDF computer file format on a wide variety of devices and platforms, and for this reason offers certain royalty-free patent licenses to PDF implementors worldwide. To review those licenses, please visit http://www.adobe.com/go/developer_legalnotices.

1. This example code includes, but is not limited to, the copyrighted list of data structures, operators, and PostScript language function definitions, that were referenced in PDF Reference, fifth edition, version 1.6, Section 1.5 (Intellectual Property).