17

This book describes the Portable Document Format (PDF), the native file format of the Adobe ® Acrobat ® family of products. The goal of these products is to enable users to easily and reliably exchange and view electronic documents independent of the environment in which they were created. PDF relies on the imaging model of the PostScript ® language to describe text and graphics in a device- 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.

PDF files are built from a sequence of numbered objects similar to those used in the PostScript language. The text, graphics, and images that make up the contents of a page are represented using operators that are based on those in the PostScript language and that closely follow the Adobe Illustrator ® 3.0 page description operators.

A PDF file is not a PostScript language program and cannot be directly interpreted

This line uses a PostScript outline font.
This line uses an OpenType font.
This line uses a fully-embedded OpenType font.
This line uses a subset-embedded OpenType font.

This line uses an unembedded font.
This line uses an embedded & subset Japanese font. 𠀋

by a PostScript interpreter. However, the page descriptions in a PDF file can be converted into a PostScript language program.

1.1 About this book

This book provides a description of the PDF file format, as well as suggestions for producing efficient PDF files. It is intended primarily for application developers who wish to produce PDF files directly. This book also contains enough information to allow developers to write applications that read and modify PDF files. While PDF is independent of any particular application, occasionally PDF features are best explained by the actions a particular application takes when it encounters that feature in a file. Similarly, Appendix D discusses some implementation limits in the Acrobat viewer applications, even though these limits are not part of the file format itself.

This book consists of two sections. The first section describes the file format and the second lists techniques for producing efficient PDF files. In addition, appendices provide example files, detailed descriptions of several predefined font encodings, and a summary of PDF page marking operators.

Readers are assumed to have some knowledge of the PostScript language, as described in the *PostScript Language Reference Manual, Second Edition* [1]. In addition, some understanding of fonts, as described in the *Adobe Type 1 Font Format* [6], is useful.

The first section of this book, Portable Document Format, includes Chapters 2 through 7 and describes the PDF file format.

Chapter 2 describes the motivation for creating the PDF file format and provides an overview of its architecture. PDF is compared to the PostScript language.

Chapter 3 discusses the coordinate systems and transformations used in PDF files. Because the coordinate systems used in PDF are very much like those used in the PostScript language, users with substantial background in the PostScript language may wish to read this chapter only as a review.

Chapter 4 describes the types of objects used to construct documents in PDF files. These types are similar to those used in the PostScript language. Readers familiar with the types of objects present in the PostScript language may wish to read this chapter quickly as a reminder.

Chapter 5 provides a description of the format of PDF files, how they are organized on disk, and the mechanism by which updates can be appended to a PDF file.

Chapter 6 describes the way that a document is represented in a PDF file, using the object types presented in Chapter 4.

Chapter 7 defines the resources used in a PDF file, including fonts, color spaces, images, and others.

Chapter 8 discusses the page marking operators used in PDF files. These are the operators that actually make marks on a page. Many are similar to one or more

This line uses a PostScript outline font.

This line uses an OpenType font.

This line uses a fully-embedded OpenType font.

This line uses a subset-embedded OpenType font.

This line uses an unembedded font.

This line uses an embedded & subset Japanese font. 丈

PostScript language operators. Readers with PostScript language experience will quickly see the similarities.

The second section of this book, Optimizing PDF Files, includes Chapters 10 through 14 and describes techniques for producing efficient PDF files. Many of the techniques presented can also be used in the PostScript language. The techniques are broken down into four areas: text, graphics, images, and general techniques.

Chapter 10 discusses general optimizations that may be used in a wide variety of situations in PDF files.

Chapter 11 discusses optimizations for text.

Chapter 12 discusses graphics optimizations.

Chapter 13 discusses optimizations that may be used on sampled images.

Finally, Chapter 14 contains techniques for using clipping paths to restrict the region in which drawing occurs and a technique using images to make efficient blends.

This line uses a PostScript outline font.
This line uses an OpenType font.
This line uses a fully-embedded OpenType font.
This line uses a subset-embedded OpenType font.

This line uses an unembedded font.
This line uses an embedded & subset Japanese font. 弋