Adobe® PostScript®

Application Support for PostScript CJK Fonts

Adobe Developer Support

Technical Note #5640

15 December 1997

Adobe Systems Incorporated

Corporate Headquarters
345 Park Avenue
San Jose, CA 95110-2704
(408) 536-6000

Eastern Regional Office
24 New England
Executive Park
Burlington, MA 01803
(617) 273-2120

Adobe Systems Europe Limited
Adobe House, Mid New Cultins
Edinburgh EH11 4DU
Scotland, United Kingdom
+44-131-453-221

Adobe Systems Japan
Yebisu Garden Place Tower
4-20-3 Ebisu, Shibuya-ku
Tokyo 150 Japan
+81-3-5423-8100

PN LPS5640

All instances of the name PostScript in the text are references to the PostScript language as defined by Adobe Systems Incorporated unless otherwise stated. The name PostScript also is used as a product trademark for Adobe Systems' implementation of the PostScript language interpreter.

Any references to a “PostScript printer,” a “PostScript file,” or a “PostScript driver” refer to printers, files, and driver programs (respectively) which are written in or support the PostScript language. The sentences in this document that use “PostScript language” as an adjective phrase are so constructed to reinforce that the name refers to the standard language definition as set forth by Adobe Systems Incorporated.

This publication and the information herein is furnished AS IS, is subject to change without notice, and should not be construed as a commitment by Adobe Systems Incorporated. Adobe Systems Incorporated assumes no responsibility or liability for any errors or inaccuracies, makes no warranty of any kind (express, implied, or statutory) with respect to this publication, and expressly disclaims any and all warranties of merchantability, fitness for particular purposes, and noninfringement of third party rights.

Contents

Application Support for PostScript CJK Fonts

1 Introduction

This document provides an overview of how applications can support multiple font formats for Chinese, Japanese, and Korean (CJK) language applications in a manner that optimizes their use for PostScript® language printing.

Most applications use standard Windows® GDI or Macintosh QuickDraw system calls to display and print text. Applications that support more advanced text composition may need to use the Adobe® Type Manager® (ATM®) software Application Programmers Interface (API), and the AdobePS™ printer driver to handle font management and downloading.

Applications that generate their own PostScript language code often assume a single font format, when, in fact, there are several formats currently in use. Adobe originally introduced OCF (Original Composite Format) fonts, and has since introduced the CID font format. Future Adobe CJK font products will still use the CID format, but the CID font will be packaged as an OpenType™ font using the new font format being jointly developed by Adobe and Microsoft. In addition, there are double-byte TrueType CJK fonts in the market.

Applications must deal with these multiple font formats, or rely on the PostScript driver to handle them, or risk being a PostScript language “Print Criminal.” This document explains the need for applications to perform all font management and text operations in a format-independent manner. Also, it describes how developers can derive significant benefits by providing direct support for the CID font format.

Appendix A gives a brief overview of font formats, and Appendix B lists the Adobe Technical Notes that contain more technical information on the subjects discussed in this document.

1.1 Terminology

In this document, the term *CJK font* is used to mean any font in a format that is capable of encoding more than 256 characters at one time. The acronym *CJK* is used because most Chinese, Japanese, and Korean fonts generally use at least double-byte encodings, even though some instances of those fonts may use single, or even 3- or 4-byte encodings. Other examples of large fonts might include a Roman font with a large character set, or an Arabic or Indic language font with a large number of ligatures.

Applications that generate their own PostScript language code are called *pass-through applications* because they generally pass their PostScript code to the driver for inclusion in the file that gets downloaded to the printer.

2 CJK Fonts and Font Formats

Part of the challenge of supporting CJK fonts is the variety of font formats that must be supported, and the problem of how to do all necessary text operations in a manner that is not dependent on any single format.

The evolution of double-byte font formats for PostScript printing includes the OCF (Original Composite Format), CID (Character ID), and, in the near future, the OpenType format. In addition, users may have CID *sfnt* fonts or TrueType fonts on their host system. For more information on these font formats, see Appendix A, “Font Format Overview.”

When printing, outline fonts in any of the above formats may be downloaded as either an outline font, a bitmap font, bitmap images, or they may be converted to another format. For example, a TrueType font may be converted to a CID font containing Type 42 base fonts, or to other formats understood by the printer such as Type 1, Type 3, Type 32, or Type 42.

In addition to the problem of multiple font formats, applications often have problems determining which language or features are supported by a font. That is because current font formats do not support explicit specification of such features, but this situation will be improved with the coming of the OpenType font format.

2.1 The CID Font File Format

The CID font format is the only multiple-byte PostScript font format that is documented and fully supported by Adobe Systems. Currently, Adobe sells only CID fonts for users, and embeds them in the ROMs of new Adobe PostScript printer products. Adobe strongly encourages that all new font development and application support be focused on the CID font format.

Application developers who generate their own PostScript language code are urged to directly support the CID font format to take advantage of the added flexibility and enhanced performance that is possible with that format. For more information on the advantages of direct support for CID fonts, see section 8.

CID fonts either are, or will be, available in several forms. A single CID font file will work with ATM for either Windows or Macintosh, but it needs a PFM (printer font metrics) file for use in Windows, or a Suitcase file for the Macintosh. There are also CID fonts for the Macintosh in the *sfnt* font format (see Adobe Technical Note #5180, *The CID sfnt Font File Format*) which have been published by Adobe Systems and other font developers.

Lastly, the CID font format will be a part of the OpenType format, which is similar, but not identical, to the sfnt format (see section A.3 in Appendix A for more information). Developers for Macintosh sfnt fonts should find the transition to OpenType fairly easy. More information on OpenType fonts will be published when the specification is finalized.

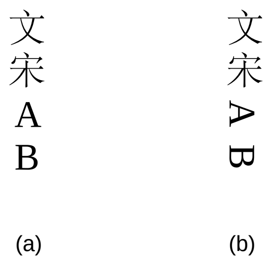
2.2 The Original Composite Format (OCF)

This proprietary format is no longer used or supported, and was never documented.While OCF fonts will work with PostScript printers, compatibility with future software products may be limited. Also, an increasing number of new software products do not support OCF fonts. For example, the AdobePS drivers do not support the downloading of OCF fonts from the host, nor are they supported by Adobe Acrobat® Distiller® 3.0J or Adobe Extreme™. This is because of the relative inflexibility of the format, which makes it difficult for both font and application developers to support it.

2.3 Fonts for Vertical Writing

Currently, CJK language fonts designed to support vertical writing generally contain CJK characters that have the correct orientation for vertical writing, but the Roman characters will appear as in Figure 1a. To be typographically correct, the Roman characters should appear as shown in Figure 1b.

FIGURE 1 *Vertical Writing Example*



(a) (b)

For the Macintosh, font menu names generally end with a “-V” for vertical fonts. For Windows, all CJK fonts have both horizontal and vertical versions, and the font menu name of the vertical font is prefaced with an *at-sign* (@) character. Applications format a page for vertical writing by laying out text in the standard way using horizontal character escapes. The page is then rotated before displaying or printing. Because ATM for Windows is aware of the writing direction, the bitmaps it generates for Roman characters are supplied in an orientation so they will be correct after the page is rotated.

For the Macintosh, there is no system convention for supporting the vertical writing mode for either display or printing. The application must format the page for the intended writing direction, and use the appropriate API call to ATM to obtain glyphs in the correct orientation.

2.4 Font Protection

Some CJK fonts may have various forms of copy protection, additional encryption, resolution restrictions, or disk-keyed protection applied to them. Typically, the fonts from some CJK font developers have extra encryption applied to the font's character descriptions, which means that the font dictionary can be copied and transformations applied to the entire font, but the characters cannot be converted to outlines in a drawing application on the host. Also, PostScript language operators such as cannot be used on the charstring descriptions in the print job.

2.5 The Need for Font Format Independence

The most important consideration is that all printing of text should be handled in a manner that is independent of the font format. For example, some applications that need to algorithmically manipulate fonts (rotate, embolden, outline, or oblique the characters) do so by assuming that the font has the OCF file structure. They use the **FDepVector** array in the font to locate the desired glyphs, but since there is no such entry in a CID font, the resulting PostScript code will not work. Applications that generate code that assumes only a single font format are considered PostScript language "Print Criminals."

3 Adobe Type Manager (ATM) Software

ATM software displays Type 1, CID, OCF, and, in the future, OpenType fonts, for both Windows and the Macintosh. It also enables printing of glyph bitmaps (for one- or two-byte fonts) for non-PostScript printers, and offers an Application Programmers Interface (API) that allows the creation of special effects and advanced text features.

ATM will intercept standard system calls to display text using PostScript language fonts. If the request is for a TrueType font, the request is passed back to the Macintosh or Windows operating system.

3.1 Versions and Compatibility Issues

For the Macintosh, ATM 3.8 was the first international version, but it was limited as to what encodings were supported for Chinese and Korean fonts (see Adobe Technical Note #5175, *CID-Keyed Font Installation with ATM*). ATM 3.9 and greater are fully international with respect to supporting OCF and CID fonts for any language.

For Windows, ATM 3.2J for Windows is the current double-byte version. ATM 4.0 for Windows does not support double-byte fonts, but future versions will be support double-byte fonts.

Although standard GDI or QuickDraw requests to display text using PostScript fonts are handled transparently by ATM, more advanced text operations such as rotation or obliquing requires use of the ATM API. The current versions of both ATM software for the Macintosh, and ATM software version 3.2J for Windows, have an API that supports double-byte CID fonts. An application can request a character's outline, apply a transformation, and use the result for display or printing. For more information, see Adobe Technical Notes #5072, *Adobe Type Manager Software API: Macintosh*, and #5074, *Adobe Type Manager Software API: Windows*.

3.2 ATM and Non-PostScript Printing

ATM version 4.0 for the Macintosh, and 3.2J for Windows, will also download bitmaps for CJK fonts for printing to a non-PostScript language printer. ATM intercepts calls from GDI or QuickDraw to the printer driver. If the request is for a PostScript font and the output device is not a PostScript language printer, ATM will return bitmap characters which the application or driver can then include in the print file. If the output device is a non-PostScript printer that contains a TrueType rasterizer, a TrueType outline font will be returned and downloaded. With newer versions of ATM for Windows, bitmap glyphs are sent as bitmap fonts, so the characters are cached and only one instance of each character is needed.

4 The AdobePS PostScript Printer Driver

In addition to handling all standard Windows or Macintosh system calls for printing, the AdobePS printer driver is an invaluable resource for applications to use, especially for managing and downloading fonts. It provides support for the printing of printer-resident CID and OCF PostScript language fonts, as well as for host-only TrueType or PostScript fonts. It can help the application operations such as enumerating which fonts are available, downloading the fonts to the printer, and in printing text with stylistic transformations (e.g. rotation, obliquing, or algorithmically emboldened text).

The most important benefit of using the driver is that it shields the application from the need to deal with the wide variety of font formats, platform-specific features, downloading options, and the variety of versions of PostScript interpreters.

Operations such as converting a TrueType font to a properly-formed Type 42 font can be difficult to engineer properly because of the way font implementation and printer interpreter versions can vary. System-level printer drivers, such as those by Adobe and Microsoft, often include code to work around common problems with major applications, fonts, and interpreters, where a smaller software developer would not have the resources to do similar development.

Application developers should be aware that on either the Windows or Macintosh platforms, either of two print drivers might be present. For Windows, either Microsoft's PScript driver or Adobe's AdobePS printer driver might be in use. For the Macintosh, either the Apple LaserWriter driver or the AdobePS driver (formerly called the PSPrinter driver) might be in use. While many features are shared between the drivers, it is important to not assume that features are identical. Adobe recommends that users should use the AdobePS printer driver.

Determining which fonts, in which formats and with which encodings, are on a particular printer, is complicated by the fact that for Windows, most communication with the printer is uni-directional. This means that a driver cannot query a printer in advance, and then decide what code to generate (the situation called "two-pass printing"). For the Macintosh, the Adobe PS driver is capable of operating in "two-pass" mode (that is, query the printer to be able to optimize the code it generates), but users have the option to switch to one-pass mode to improve performance.

With one-pass printing, the driver does have access on the host to the printer's PostScript Printer Definition (PPD) file. That file specifies the fonts in ROM or that were installed on the printer's hard disk when the printer shipped. However, the PPD file is not updated if the user permanently downloads additional fonts to the printer's hard disk. The result is that the print file must contain code that determines the format of the font on the printer, and treats the font accordingly. Future versions of the AdobePS driver are likely to maintain a data file that downloaders can modify to specify which fonts have been permanently downloaded to a printer's hard disk.

Appendix B lists Adobe Technical Notes that contain information on how applications can use the AdobePS driver.

4.1 Application/Driver Interaction

Using the AdobePS driver to handle some or all of the task of managing fonts and generating PostScript language code for printing allows applications to save valuable development time and achieve consistent results for both the Macintosh and Windows platform. It also allows fonts to be handled in a consistent manner that is independent of the font format, which should be one of the primary goals of all applications and printer drivers.

Applications that take advantage of the PostScript language imaging model to go beyond what is available from GDI or QuickDraw, must generate at least some of their own PostScript code. They may use the driver to handle all or part of the printing process. For example, some major applications use the driver to handle all font operations, but generate their own code for everything else. Other applications use the driver to handle device setup, and to handle the Document Structuring Conventions (DSC) operations.

Most major pass-through applications use the AdobePS driver to handle all font operations. In addition to font format independence, the AdobePS driver for Windows, for example, will create Type 32 and Type 42 fonts when needed, and subset and do incremental downloading of large CID fonts (currently available in the Windows version of AdobePS). These are operations that would be fairly difficult for most applications to develop on their own.

The main reason for an application to generate its own PostScript code is for platforms where the driver is not available, such as for UNIX®. In the past, some applications have had to generate their own PostScript code because the driver did not support PostScript LanguageLevel 2, or because it was necessary for cross-platform compatibility. However, these reasons are becoming less compelling as the AdobePS driver is constantly improving, and as it evolves to be more consistent for both the Macintosh and Windows environment. Cross-platform applications that must generate their own code for a UNIX version are likely to find that it is still a big advantage to use AdobePS driver for Macintosh and Windows.

4.2 AdobePS Driver Software for Windows

The current Adobe PostScript print driver for Windows is version 4.2.2, which handles both single- and double-byte fonts.

Note: *The previous CJK driver for Windows, AdobePS driver 4.1.1 CJK, did not support proportional CJK characters such as Japanese language kana, and did not support the outline style of characters— those with a black stroke around the contour, with the interior unfilled).*

The three modes in which an application can interact with the AdobePS driver for Windows are shown in Table 1.

TABLE 1 AdobePS for Windows: Driver Modes

MODE	OPERATION
Normal GDI Mode	Application makes standard text calls to display and print; driver services requests. Application is limited by the GDI imaging model. Examples: Microsoft® Word; WordPerfect®.
Min Header	Application calls the driver to download a header that does device setup and handles downloaded fonts; all other operations and generation of PostScript code are generally handled by the application. Examples include Adobe Illustrator®, Adobe Photoshop®, and CorelDraw!™.
Open Channel	Application takes full responsibility for printer control and generates *all* PostScript code; may use driver to download CJK TrueType fonts (currently, host-based PostScript fonts are not downloaded). Application may also use GDI calls. Open Channel applications typically produce multi-page documents which may have complex graphics. Examples include Adobe PageMaker® and QuarkXPress®.

Table 2 shows how three basic text handling operations are supported (or not) by each of the modes.

TABLE 2 Supported Operations for Each Driver Mode

DRIVER FEATURES	NORMAL OS Mode	MIN HEADER	OPEN CHANNEL
Font downloading	Yes	Yes	Yes
Application ShowText calls	Yes	No	Yes
Device Setup	Yes	Yes	No

An application can download a single-byte font in Windows by using the system call *ESCAPE (DownLoadFace)*. This call can be used to download a font, or to query whether a specified font is printer resident or needs to be downloaded. Printer resident fonts include those in printer ROM, or those permanently downloaded to a printer's hard disk. The PPD file on the host specifies the fonts in ROM or on the hard disk, as supplied by the printer vendor. The AdobePS driver determines the name of ROM-resident fonts from the printer's PPD file, and those of fonts on a printer's hard disk by querying a data structure that is stored as part of the driver's resources, but which is updated only by a font downloader.

There are a wide variety of font format options for font downloading. There are two main variables: the charstring (the character drawing procedures) format and the file organization format. For example, a CID font file might be chosen as the file organization used to package characters that might be in Type 1, Type 3, Type 32, or TrueType charstring format. Both the charstring and file format must be carefully chosen for their particular strengths as well as for a variety of other factors such as the printer capabilities, user preferences, and the amount of available printer memory.

Current printer drivers do not support subsetting (including only the glyphs used in the document) or incremental downloading of CJK fonts. The application may request bitmaps from ATM, and include them in the print file using the CID Type 32 format.

Because of the lack of bi-directional communication in many PC systems, the application cannot determine the format of resident fonts, only whether it needs to be downloaded or not. Pass-through applications are advised to use the AdobePS driver to handle font downloading and style manipulations, which makes it unnecessary for the application to determine the formats of printer-resident fonts.

4.3 AdobePS Driver Software for the Macintosh

The AdobePS printer driver for the Macintosh (previously called the Adobe PSPrinter driver), for CJK languages, is currently 8.5.1CJK, and the upcoming version will be 8.6 and fully international.

It is possible for applications to use the driver with any degree of interaction, and no distinct “modes” are supported as in the case of AdobePS for Windows.

Outline fonts must either be in printer ROM or on the printer’s hard disk. If a font is only resident on the host, the driver will download bitmap *images* for outline fonts that exist only on the host computer. This means that a bitmap is downloaded for each instance of the character, and there is no caching of the character bitmaps. If a character from an outline font is requested, the driver replies that it cannot handle the request, and WorldScript then requests the appropriate bitmaps from either ATM or the TrueType rasterizer, as appropriate. Hence, the driver only ever sees the image data, and never sees the character codes. In future versions, it is hoped that the driver will support downloading of subsetted CID and TrueType fonts (but not for OCF fonts), as well as performing incremental downloading for each page in the print job. Currently, there is no API call to query about fonts that have been permanently downloaded to a printer’s hard disk.

In general, the AdobePS 8.5.1 CJK printer driver does not currently support proportional CJK characters. The exception is that CID sfnt fonts (as defined in Adobe Technical Note #5180, *The Macintosh CID-Keyed sfnt Font File Format*) are able to support proportional CJK characters. Also, support for this will be an integral part of the OpenType format.

4.4 Font Downloading for the Macintosh

Font downloading with the AdobePS Macintosh printer driver can be accomplished by using the PRGeneral API call. Also, several applications use the "space hack" to make the driver download a font to a printer. The application does this by specifying text consisting of a single space character (usually a space or line-feed character), and drawing it outside the page boundary for the given page. The driver then downloads the font, and the application can use the font for other text in the print job.

In future versions of the driver, the driver will support an API call to subset and download CID and TrueType fonts. However, it is not known if the driver will continue to support the "space hack" approach to downloading (as is currently true for the AdobePS printer driver for Windows). The subsetting feature will allow applications to print to printers with too little memory for entire outline fonts to be downloaded, and performance will be much better than for downloading bitmap character images. Also, print jobs that use multiple point sizes of a single font will exhibit a significant performance increase, and the resulting PostScript language print file will be device independent since the document references or embeds outline fonts instead of device dependent bitmaps.

4.5 The Macintosh System and Vertical Writing

The Macintosh OS does not support an explicit vertical writing mode, so applications must handle all positioning and rotation of characters for both display and printing of vertical writing.

For example, to support the rotation of Roman characters from a CJK font for vertical writing (see Figure 1), the application must request rotated bitmap characters from ATM. For printing, the PostScript code must copy the font in VM, apply the rotation by adjusting the values of **FontMatrix**, and then position and show each character individually.

4.6 PPD Files

PostScript Printer Description (PPD) files specify the set of features that are supported by the printer they describe, as well as listing printer resident fonts. Macintosh and Windows applications will get the complete list of fonts if they query the driver, so only applications in other environments need to consult the PPD file. The PPD file specification is Adobe Technical Note #5003, *The PostScript Printer Description File Format Specification*.

PPD files should not be updated by font downloaders when additional fonts are downloaded permanently to the printer's hard disk. Currently, the AdobePS driver for Windows maintains a private database of downloaded fonts, and this information will be available to drivers and applications when the specification is published. The AdobePS printer driver for the Macintosh does not currently have this capability, but it will be added in an upcoming release.

5 Supporting Application Text Features

The following sections discuss those operations that applications must do to manage fonts for printing, such as font enumeration and downloading. In addition, the application may choose to improve the quality of support for vertical writing by rotating the Roman characters in the font. Other advanced text features include supporting stylistic and metric variation glyphs, and glyph substitution. These operations are described in the following sections.

5.1 Font Enumeration

Pass-through applications must determine which fonts are available and whether they are located on the host or printer. They also must determine the correct PostScript font name (which is different than the font menu name) so the fonts can be correctly referenced in the print job.

The application should first query the driver to determine which fonts are available. To determine which fonts are available only on the host for Windows, the application can query the driver again to ask whether a particular font is available in the printer. For the Macintosh, the application can query ATM to see what fonts are available on the host.

Applications that do not have resources such as the AdobePS driver and ATM available to them can access the PPD file to determine printer resident fonts.

5.2 Determining Font Names

As with all PostScript language printing, it is essential that the PostScript language code reference all fonts by their PostScript font name, not by the font menu name generally used by applications. Because font menu names are not consistent across platforms, Adobe recommends that applications use the PostScript font name for font references in the application document as the best method to achieve cross-platform compatibility.

The PostScript font name must not contain spaces, and must use only characters in the standard ASCII range. This means that double-byte CJK characters cannot be used. PostScript font names are described more fully in Adobe Technical Note #5088, *Font Naming Issues*. That document also describes the naming convention for CID fonts, including how the family name is separated from the CMap name by a double-hyphen, and how the CMap file name specifies the encoding, character set, and writing direction.

For the Macintosh, font names for host-based fonts can be derived from the font's FOND resource for the Macintosh (see Technical Note #0091, *Macintosh FOND Resources*). For Windows, the application can use ATM API calls to get the PostScript font name, or parse the font's PFM file (see Technical Note #5178, *Building PFM Files for PostScript Language CJK Fonts*). For UNIX systems, the PostScript font name is specified in the font's Adobe Font Metrics (AFM) file (see Technical Note #5004, *Adobe Font Metrics File Specification*). PostScript font names for printer-resident fonts can be found in the printer's PPD file.

5.3 Font Downloading

Wherever possible, applications should use the AdobePS driver to download either outline or bitmap fonts to the PostScript printer. If applications are in an environment where the AdobePS driver is not available, they must decide whether to support font downloading, or to only support fonts already installed in the printer.

Some notes on downloading considerations for the main CJK font formats:

- OCF fonts are very difficult to download because of their complex structure and the variety of interpretations of reverse-engineered OCF fonts in the market. Downloading of OCF fonts is not supported by AdobePS drivers. Either an application or the driver can request bitmaps from ATM on the host for downloading. AdobePS for Windows will download Type 32 bitmap fonts if appropriate, or a pass-through application can generate and manage its own downloading of Type 32 fonts (for version 2016 or greater PostScript printers).

- CID fonts that are on the host, but not on the printer, can be downloaded as either outline or bitmap fonts. The obvious problem is the size of the font and the amount of memory available. It is much easier to dissect a CID font than an OCF font in order to subset it, but it must be done with considerable care to get all the necessary hint dictionaries and subroutines needed for a particular font. To download a CID outline font, the printer must be at least version 2015 or greater (see Note on page 24) to use the font in native mode, or have a CSL (CID Support Library; see section 6) resident in the printer to use the font correctly in compatibility mode (for versions before 2015).

- TrueType fonts can be downloaded by AdobePS driver as a Type 42 font (which is implemented as a CID font with Type 42 base fonts for large double-byte fonts). They can also be converted to a CID Type 32 bitmap font. (Note: Type 32 is relatively easy to implement; Type 42 is much more difficult). AdobePS for Windows subsets and incrementally downloads TrueType fonts (as either Type 42 or Type 32 fonts).

  There is a new Escape in Windows to subset TrueType fonts so that only the requested characters are downloaded. This is not yet necessary for PostScript fonts because downloading of host-based OCF or CID fonts is not yet supported, characters from those fonts must be downloaded as bitmap fonts.

  The AdobePS printer driver for Windows handles the downloading of TrueType fonts following the user setting of preferences and the capabilities of the printer. They can be downloaded as Type 42 fonts (a CID composite font with Type 42 base fonts) if the printer contains a TrueType rasterizer; as a Type 1 outline font if the number of characters needed is small; or as a Type 3 bitmap font, where the driver obtains bitmaps from ATM. This works in AdobePS 4.1 CJK, and in AdobePS 4.2. (AdobePS 4.1 CJK does have a bug regarding handling proportional kana characters, but that is fixed in version 4.2).

5.4 Vertical Writing

Typically, a vertical CJK font will have the correct orientation for glyphs in the double-byte range, but ideally those in the single-byte Roman range should be rotated by the application or driver. For Windows, ATM and the AdobePS driver do the correct thing for vertical writing, but applications for the Macintosh and other environments must handle the necessary glyph rotation themselves. This can be done either by using the ATM API to get bitmaps for display or printing, or by rotating the outline fonts in the printer.

With OpenType fonts, font developers will be able to specify the correct behavior in the font, so applications will not need to do extra work to get the correct results.

5.5 Stylistic Variants

For the Macintosh, the OS supports bold, italic, or outline versions of fonts. However, CJK font developers tend not to provide bold or oblique versions of their fonts because CJK fonts are so large, and traditionally the operating system has supported those extra styles using algorithmic transformations.

For Windows, a user might request a bold or italic version of a font. This means that the application must determine if those styles are available in the user's system or printer, and decide what to do if they are not. Additionally, a Windows application might decide to support the outline style either as an added feature, or for cross-platform compatibility with the Macintosh version of the application. Hence the application must either support those style variations itself, or rely on ATM and the printer driver to handle it.

The following are brief suggestions for how to handle stylistic variants:

- **Italic/Oblique**: one of the easier styles to support; the font dictionary can be copied and the **FontMatrix** values can be adjusted to slant the upright version to the desired angle.

- **Bold**: while algorithmically making a bold version of a plain font can produce typographically terrible results, it is sometimes necessary when no bold version exists. It is usually accomplished by over-printing the same font multiple times with a slight offset to make the font appear bolder.

- **Outline**: refers to a glyph whose outline is stroked and the interior area remains the color of the background rather than being filled with the color of the stroked outline.

- **Shadow**: can be made by printing or displaying a solid filled character beneath an offset version of the outline form of the character.

Other style variants such as underlined and strike-through text can be achieved by drawing an additional line through or under the text. Reduced-size characters, such as those used for superscript, subscript, and small caps for Roman characters, should be produced by an additional scaling in the x direction. The result is a slightly heavier weight character that looks better with the full size characters than they would if they were created with an even scaling in x and y.

5.6 Metric variants

The following are examples of situations where either alternate or proportional (rather than monospaced) metrics would be used.

Proportional CJK Glyphs

In the past, most CJK applications and operating systems have assumed that glyphs in a CJK font are either full- or half-width characters because of lack of support from the OS or the font formats. Metric variant glyphs such as proportionally spaced Japanese kana characters, were not supported.

Currently, the only CJK fonts that use proportionally spaced glyphs are Adobe’s CID sfnt fonts for the Macintosh. The only way for font developers to produce such fonts for cross-platform use will be in the upcoming OpenType format, where there are tables specifically for proportional spacing and kerning. Applications will be able to access the proportional widths through the OpenType Support Services Library to be supplied by Microsoft.

Glyph Substitution

Glyph substitution consists of specifying one or more glyphs in the font that will be substituted for one or more other glyphs. It is something that is specified in the font, by the font developer. This is a useful feature to support alternate glyphs for vertical writing, to support additional national character sets and encodings, and, for example, to enable a user to convert a document using a traditional Chinese font into one using a simplified Chinese font.

Glyph substitution is supported in the Macintosh sfnt font format, and will be supported in OpenType. In OpenType, applications will be able to enable substitution and access the metrics through the OpenType Services Library DLL, which will be supplied by Microsoft.

Kerning

Currently, few if any CJK fonts support kerning, mainly because OCF and the CID formats provide no support for specifying pair kerning data. Most CJK fonts only contain full- or half-width characters, for which no kerning is needed. However, fonts containing proportionally spaced CJK characters can benefit from kerning, and kerning will be very important for CID fonts for large Roman character sets.

The only font format that currently supports pair kerning is the Macintosh sfnt format (implemented as either a PostScript CID or TrueType font). For information on PostScript versions of sfnt fonts, see Technical Note #5180, *The CID-Keyed sfnt Font Format*. The ultimate, cross-platform solution for adding kerning data for CID fonts will be the OpenType format.

6 The CID Support Library (CSL)

The CSL is a collection of PostScript language files that enable CID fonts to be compatible with earlier PostScript printers (interpreter versions previous to 2015) and software applications. It does this by converting CID fonts into an OCF-like structure for PostScript interpreters previous to version 2015. It also patches various PostScript language operators so they will work in a CID-aware manner.

Note: *Although all Adobe PostScript printers, version 2015 or greater, do have native support for CID fonts and contain the CSL, there are a few version 2014 printers that do also. Rather than checking the version number, it is recommended that the printer be queried for a CID operator like* ***composefont****, or an appropriate resource type, as follows (that is, any one of the following lines of code could be used):*

```
systemdict /composefont known
9 /FMapType resourcestatus
9 /FontType resourcestatus
```

Newer PostScript printers, version 2015 or greater, have the CSL installed—generally in the printer’s ROM. It is needed to redefine operators and to have the option to convert CID fonts into an OCF structure in case this is needed for applications that are not CID-aware.

The CSL may also be downloaded to printers by font downloaders or installers. Installation must adhere to the procedure described in Adobe Technical Note #5174, Installing CID-Keyed Fonts in PostScript Language File Systems, to ensure that the correct version of each CSL component is used.

Typically, users should receive from their font vendor a copy of the CSL when they purchase a CID font package. The font installer utility can then check to see if the version of the CSL in the font package is more current than the one in the printer, and do any necessary updating.

Table 3 shows the three CSL operating modes.

TABLE 3 CSL Operating Modes

MODE	POSTSCRIPT VERSION	OPERATION
Compatible	< 2015*	Converts CID font into OCF style structure.
Native	≥ 2015*	Creates CID Native structure for CID fonts; supplies necessary system procsets for use of OCF fonts.
Native -Compatible	≥ 2015*	Converts CID into OCF so it will be compatible with applications that assume an OCF structure; supplies necessary system procsets for use of OCF fonts.

* see Note on page 24.

The CSL mode affects the performance of CID fonts, and is controlled by the user and the print driver. Hence, it does not affect applications, other than it should be realized that even if an application is CID-aware, users may still experience lower performance because of how a printer’s CSL must be set to accommodate all applications being used on the user’s system.

The CSL contains the functional equivalent of the table shown in Table 4.

TABLE 4 CSL Encoding Table

FONT ENCODING	VALUE	ACTION TAKEN
/83pv-RKSJ-H	*true*	compatible mode; creates OCF VM structure
/H	*false*	native mode; creates flat CID font structure
/Ext-H	*true*	compatible mode; creates OCF VM structure

where a value of *true* results in the creation of an OCF-compatible structure in VM; and a value of *false* indicates that the CID font can be used in native mode. Applications should not attempt to change these settings because they may be necessary for other applications on the system. For compatibility mode, the driver generates code that uses the **findfont** operator; for native mode the **composefont** operator is used. If an application wants to force use of native mode, it can either use the AdobePS driver, which will use native mode wherever possible, or it can generate its own PostScript code and use **composefont**.

On the Macintosh, the PPD contains a *job patch* that sets all entries in the CSL encoding table to *false*, which allows use of CID fonts in native mode. The job patch in the PPD file is used by the driver and is inserted at top of each print job. In the next version of the driver, the user will be able to use the driver user interface to allow the job patch to be inserted before an individual page.

For Windows, the Adobe PS 4.1.1J print driver has a CID-aware mode, where it uses the **composefont** operator for a CID font and a version 2015 (see Note on page 24) or greater printer. That version of the driver will create a composite font structure in VM (which is not an OCF structure, just a composite font structure as described in the *PostScript Language Reference Manual, Second Edition*).

7 Generating PostScript Language Code

Generation of correct PostScript language code for CJK fonts requires supporting multiple font formats. Secondly, the application should support CID fonts for native-mode printing to take advantage of the benefits offered by that format (see section 8). The easiest method is to let the AdobePS driver handle the tasks of font downloading and generating style variations.

In order for an application to perform necessary font manipulations in a PostScript language print job, such as obliquing, emboldening, or creating an outline style, it is necessary to know the format of the font(s) involved. Generally the application cannot know the format of a printer resident font in advance, because of the unidirectional communication lines in the PC environment. Hence, the PostScript code must determine the format in the print job code, and use conditional code to handle each case.

If the application is generating its own code, and the font is only on the host and must be downloaded, the application can determine the format before creating the print job, and generate the correct code. But the safest way is to generate code that determines the format when the print job executes in the printer.

Font transformations can be made by copying the font and changing either the **FontMatrix** in the font dictionary, or the Current Transformation Matrix (CTM). Changing the **FontMatrix** has the advantage that it applies only to the font and not the graphics state; changing the CTM means that the transformation applies to the entire graphics state. TrueType fonts are handled in similar manner, since the Type 42 font dictionary also contains a **FontMatrix** array.

8 Supporting CID Fonts

There are significant advantages that can be realized when an application is “CID-aware” and supports the CID font format. These advantages include:

- Application and font developers benefit from the standardized character collection and CMap files that are available from the Adobe Developers Association for use with any CID fonts that conform to Adobe’s standard Character Collections. While font developers can define their own Character Collections and CMap files for special needs, those defined by Adobe provide significant value to developers by making it easier to support standard industry and national character set standards.

- Applications have more flexibility in supporting multiple encodings and character sets because of the easy interchangeability of CMap files for CID fonts. Once an application invests in the development time to support one CJK language, it is relatively easy to support additional encodings as well as other languages. OCF fonts are not as flexible, and TrueType fonts only support the *cmap* tables build into the font.

- The font rearrangement capability of the CID format can solve a number of problems that would not be easy to solve in any other way. For example, an application could allow users to add their own symbol glyphs to a font, which is an important consideration for the Asian market.

- Applications can use CMap files to migrate to Unicode by mapping between a Unicode CMap file and the CMap file for the other encoding. Applications interested in cross-platform compatibility can also CMap files to map characters between the standard character sets and encodings for the Macintosh and those for the Windows environment.

- CID fonts, with either sfnt wrappers for the Macintosh, or in the OpenType format, offer many advantages to font developers and users alike in terms of potential support for advanced composition and support for various language and writing direction features. The sooner applications prepare to support CID fonts in those formats, the further ahead they will be in offering better typographic features to their users.

- Better hinting will be possible because multiple hint dictionaries can be included in a CID font. This means that the Roman and ideographic glyphs can each have their own set of hints, which prevents the hints on one type of glyph from interfering with glyphs they were not intended for.

- Printing with OCF fonts can present a variety of problems due to the complexity of the format. Migrating to the CID format will benefit both the application and the end user.

- CID fonts are compatible with Adobe Acrobat and Adobe Extreme, thus giving applications better compatibility with those environments.

8.1 How Applications Can Be CID-Aware

The easiest way for an application to support CID fonts is to use the AdobePS printer drivers—version number 4.2.2 for Windows, and version 8.5.1 CJK for the Macintosh. If an application must generate its own code, it should:

- Determine the format of the fonts being used in a document. If possible, do this on the host. If it is not possible, the print job code must query the PostScript interpreter to determine the format of all fonts being used. All subsequent operations involving those fonts must be in accordance with the appropriate format.

- If CID fonts are being used, determine whether the version number of the PostScript printer is 2015 or greater (see Note on page 24). If the printer is at least version 2015, it is capable of supporting CID fonts in native mode.

- If the application wants to print CID fonts in native mode, it should use the **composefont** operator. If it needs to use the font in compatibility mode, it should use the **findfont** operator.

- When the CIDFont resource instance is the current font, only the **glyphshow** operator may be used.

Be aware that the job patch in the Macintosh PPD file may override your best efforts to print in native mode. This has to be controlled by the user, since they may have other “print criminal” applications that require that setting.

8.2 Supporting CID-Native Printing

To generate PostScript language code for using CID fonts in native mode on interpreters with version 2015 or greater, the **composefont** operator must be used. For more information on **composefont**, see Adobe Technical Note #5213, *PostScript Language Extensions for CID-keyed Fonts*, which is an extension of the *PostScript Language Supplement for Version 2015*.

CID fonts can only print in native mode if the interpreter version is 2015 or greater (see Note on page 24); and must be used in compatibility mode (requiring the CSL) for versions before 2015. The CSL is always resident in all version 2015 or greater printers. For versions before 2015, the CSL must have been installed on the printer’s hard disk, and CID fonts must be used in compatibility mode.

The Adobe PS driver determines the printer version and font type, and is able to create a print job that will utilize the correct mode. This works using conditional code in the PostScript language file, so that it will work correctly regardless how or where the job is spooled; for example, it may be spooled one day, or even archived, and routed to an alternate printer at a later date.

9 Acrobat PDF Compatibility Issues

PostScript code generated by applications and print drivers should be convertible to PDF, and the resulting text should be searchable. The goal should be to be sure that all character codes in a PostScript language print job can be identified by a program such as Adobe Acrobat Distiller, which converts PostScript language files into PDF files. In the case of a CID-keyed font, the character codes are interpreted relative to the CMap file referenced in the font. The CMap file contains a mapping from the input character code to the CID number used to access the character.

The way in which an application or printer driver might prevent the text from being searchable would be if a font were subsetted, and the character names or ID numbers used in the new font were specific to that file, rather than being interpretable relative to a known encoding and character set.

At this time, only a Japanese version of the Adobe Acrobat software is available, so full details of how searchable PDF files should be created for CJK language documents will be published when available.

APPENDIX A

Font Format Overview

A.1 Original Composite Format (OCF)

The Original Composite Format (OCF) was Adobe's original double-byte font format. While the few Japanese OCF fonts worked well, the format was not as flexible in terms of addressing a variety of character sets and encodings, as the market required.

While the basic composite font format (Type 0) is documented in the *PostScript Language Reference Manual, Second Edition*. Adobe's OCF fonts use a very complicated file structure and consist of over 80 files per font. OCF was intentionally not documented or supported because Adobe wanted to focus all development on the much more flexible CID font format. However, several companies have reverse-engineered OCF fonts, and there is now an installed base of OCF fonts for all CJK languages.

A.2 CID Font Format

The primary font format for optimal compatibility for PostScript printing is the CID font format. The CID format was designed for maximum flexibility and performance, and is fully documented and supported by Adobe Systems. See Appendix B for a list of CID-related Technical Notes.

Adobe sells CID fonts, as well as CID sfnt fonts for the Macintosh. In the future, Adobe will use the OpenType font format, in which CID fonts will be represented as a table within the OpenType wrapper.

CID fonts are interpreted in *native mode* (which yields the best performance) on version 2015 or greater of the PostScript interpreter. For pre-2015 printers, the CID Support Library (CSL) must be installed for the CID font to operate in *compatibility mode*. The CSL may be licensed without charge from the Adobe Developers Association.

CID fonts can be based on any of the Adobe Character Collections (that is, character sets) or CMap files (encodings) defined by Adobe (see Technical Note #5094, *Adobe CJK Character Collections and CMaps for CID-Keyed Fonts*). Font developers may also define their own Character Collections or CMap files. In addition, support for Unicode is easy; it is simply a matter of using a Unicode CMap file with the CIDFont file.

For more information on CID fonts, see the bibliography in Appendix B.

A.3 Macintosh CID sfnt Format

Macintosh CID sfnt fonts consist of an sfnt wrapper with a CID font represented as one table in the file. The format allows the addition of tables that support advanced features such as glyph substitution and proportionally spaced glyphs and kerning for CJK glyphs. The format is fairly similar to the OpenType format, but works only on the Macintosh. For more information, see Adobe Technical Note #5180, *The Macintosh CID sfnt Font Format*.

A.4 TrueType Font Format

The TrueType font format is the native font format in the Windows and Macintosh operating systems. TrueType fonts can only be used for PostScript printing by either converting the font to a Type 1 font, or for printing with PostScript printers that include a TrueType rasterizer (which is specified in the printer's PPD file).

TrueType fonts are supported for PostScript printing by the AdobePS driver; version 4.1 allows the user to select how their TrueType fonts should be handled for printing. If the PostScript printer contains a TrueType rasterizer, the font will be downloaded as a Type 42 font. Otherwise the TrueType font will be converted to a Type 1 outline font, or to Type 3 or Type 32 bitmap fonts, depending on the user's preferences. Also, version 4.2 of the AdobePS CJK printer driver supports font subsetting of TrueType fonts. where only the characters needed for the print job are downloaded.

A.5 Multiple Master

Multiple master fonts are Type 1 font programs that include two or more *master* fonts within a single font file. It allows users to interpolate many intermediate *instances* of the typeface. The fonts have one or more *axes* which might typically represent the weight, width, or optical size of the font. Thus the user can generate a very large number of variations from a single font, thus providing unprecedented flexibility. While the multiple master format is suitable for CJK language fonts, no products have yet been developed in that format.

The specifications for the multiple master font format are contained in Adobe Technical Note #5015, *Type 1 Font Format Supplement*. Much of the format's potential and the related design issues are discussed in Adobe Technical Note #5091: *Designing Multiple Master Typefaces*. While this document is addressed to type designers, portions of it can make interesting reading for software developers and others interested in the capabilities of the format.

A.6 CFF/Type 2

The Compact Font Format (CFF) is a compact form of a font file structure which was designed to significantly reduce the size of font files. It was designed primarily for use with the Type 2 charstring format, but it can be used as a file structure to include any of several possible *charstring* (character description procedures) types. The CFF format forms the basis for the Type 1 OpenType font format, and is used for embedding fonts in PDF files generated by Adobe Acrobat Distiller 3.0 software.

A.7 OpenType

The OpenType font format is being jointly developed by Microsoft and Adobe Systems. At this time, the specifications are not yet final, but drafts are available at both companies' web sites.

As with Macintosh sfnt fonts, PostScript CID fonts will be represented as a single CFF table in an OpenType wrapper. The font can either be a basic OpenType font with the minimum number of features, or it might have a number of additional tables included to support a variety of advanced features such as contextual glyph substitution, various types of kerning and tracking data, and alternate baseline and alignment parameters. The OpenType format provides a means to specify proportional widths and kerning values for double-byte CJK fonts, which is something not available previously with OCF or CID fonts. All OpenType fonts will use Unicode encodings.

A.8 Type 1

The font format for single-byte Roman fonts for use with Adobe Type Manager software and with PostScript printers. Type 1 fonts use a specialized subset of the PostScript language which is optimized for better performance and a more compact representation than is possible using the PostScript language to make Type 3 fonts. The Type 1 operator set includes hint information which helps font rasterizers create more accurate bitmaps for smaller sizes and lower resolutions. For more information, see *Adobe Type 1 Font Format*, published by Addison-Wesley (This book must be ordered from book sellers by the ISBN number, ISBN # 0-201-57044-0, to avoid confusion over which edition is desired), or the PDF version available in the Developer Support area of Adobe's web site.

A.9 Type 2

A charstring format that offers a compact representation of the character description procedures in an outline font file. The format is designed to be used with the Compact Font Format (CFF). The CFF/Type2 format is the basis for PostScript OpenType fonts, and is used for embedding fonts in PDF files created by Adobe Acrobat Distiller 3.0 PDF files.

For more information, see Technical Note #5176, *The Compact Font Format Specification*, and #5177, *Type 2 Charstring Format.*

A.10 Type 3

Type 3 fonts can use the full range of PostScript language operators. This means they can express more effects such as shading, color, and fill patterns, but they are also slower and cannot be used with ATM because it is not a full PostScript interpreter. Type 3 fonts also do not have a method to specify hint information, which assist the rasterizer in producing higher quality bitmaps for lower resolutions.

Type 3 fonts use a general “graphics fill” rasterizer; the result is that imaged characters look slightly bolder than they would if expressed as a Type 1 font. Type 3 fonts are mainly useful only for special purpose or very complex fonts (such as complex logos or converted EPS art files). The format also provides a way to represent bitmap characters. For example, PostScript printer drivers for Windows can convert an outline font to a Type 3 bitmap font for downloading to a PostScript language printer.

A.11 Type 4

Type 4 is a format that was used to make fonts for printer font cartridges and for permanent storage on a printer's hard disk. Type 4 refers to the font file organization; the character descriptions are expressed in the Type 1 format. The format is proprietary and is not documented.

The advantage of the format is that PostScript interpreters are able to read into printer memory (VM), from the printer's hard disk, only those characters needed for the current print job, thus saving VM storage space. With all LanguageLevel 2 PostScript printers, the same capability exists for both Type 1 Roman fonts and for CID CJK fonts, thus making the Type 4 format essentially obsolete.

A.12 Type 5

Type 5 is similar to the Type 4 format but is used for fonts stored in the ROMs of a PostScript printer. Type 5 refers to the font file organization; the character descriptions are expressed in the Type 1 format.

A.13 Type 32

Used for downloading bitmap fonts to PostScript interpreters with version number 2016 or greater. The font is downloaded as a CID font containing Type 32 base fonts. The bitmap characters are transferred directly into the interpreter's font cache, thus saving space in the printer's memory. The format is specified in the *PostScript Language Reference Manual Supplement for Version 2016*.

The primary purpose of the Type 32 format is to ensure that a large document will print successfully on a printer with only a small amount of memory. Documents with a large number of complex CJK characters can often cause limitcheck errors on such printers because they run out of memory. With Type 32, it is possible to print complex CJK print jobs to Roman printers (i.e. printers with less memory than those usually targeted for the CJK market).

Successful use of Type 32 fonts requires that the driver manage the font cache. The interpreter (version 2016 or greater) allows bitmaps to be flushed from the font cache as needed. For this reason, it is ideal for use in direct printing to a local printer, rather than for network printing. It also works best where users have a relatively fast CPU on their host system.

Type 32 fonts may not be a good solution for high resolution devices, where multiple sizes of fonts are used, or for situations where the user requires device independence of the resulting print file, or for conversion of that file to PDF.

A.14 Type 42

Type 42 fonts consist of a PostScript language wrapper around a TrueType font. A Type 42 font is usually generated by a printer driver to download TrueType fonts to a PostScript printer that includes a TrueType rasterizer. By this method, the TrueType font is interpreted directly rather than being converted to a Type 1 font. See Adobe Technical Note #5012, *The Type 42 Font Format Specification*.

APPENDIX B

Bibliography

All format specifications and most font-related Adobe Technical Notes are available in PDF format at the Adobe web site, in the Developer Support area:

http://www.adobe.com/supportservice/devrelations/technotes.html

Additional Adobe Technical Notes on ATM software and Adobe PostScript print drivers are available to members of the Adobe Developers Association.

B.1 Font Format Specifications

Adobe Type 1 Font Format. Addison-Wesley, 1991; ISBN 0-201-57044-0.

Specification of the Type 1 font format. This document describes the platform-independent form of the format as interpreted by the PostScript interpreter, rather than the format used for installation and storage on Macintosh or Windows systems.

Technical Note #5015, *The Type 1 Font Format Supplement*

Contains all updates to the Type 1 format, including the specification of the multiple master font format, and the Counter Control hints that are important for CJK fonts.

Technical Note #5176, *The Compact Font Format Specification*

Specification for the Compact Font Format (CFF) used for PostScript OpenType fonts and PDF font embedding.

Technical Note #5177, *The Type 2 Charstring Format*

Specification for the compact charstring format used with CFF font files.

B.2 CID Font File Format

Technical Note #5014, *Adobe CMap and CIDFont Files Specification*

Specification and tutorials on the CID-keyed font technology which is used for multi-byte encodings for PostScript fonts.

Technical Note #5092, *CID-Keyed Font Technology Overview*

Gives an overview of the CID font technology, its relationship to the Original Composite Format (OCF), and what the benefits are for both users and font developers.

Technical Note #5174, *CID-Keyed Font Installation for PostScript File Systems*

Describes how to install CID-keyed fonts and the CID Support Library (CSL) files into PostScript language file systems.

Technical Note #5175, *CID-Keyed Font Compatibility with ATM Software*

Describes how to install CID fonts for use with ATM software for both Macintosh and Windows.

Technical Note #5178, *Building PFM Files for PostScript Language CJK Fonts*

Explains how to build Printer Font Metrics (PFM) files for PostScript language CJK fonts (both OCF and CID-keyed). Not a file format specification (which is available from Microsoft), and it does not specifically address building PFM files for Roman fonts.

Technical Note #5180, *CID-Keyed sfnt Font File Format for the Macintosh*

Specification for the CID-keyed Type 1 fonts that use the sfnt font file organization. CID sfnt fonts contain *extra tables to support additional encodings and characters for vertical writing.*

Technical Note #5213, *PostScript Language Extensions for CID-Keyed Fonts*

Specifies the PostScript language extensions for CID fonts that were added in PostScript language version 2015. Describes additional CIDFontTypes to those specified in the CID specification.

PostScript Language Reference Manual Supplement for Version 2015

Specifies CIDFont and CMap resource categories, extensions to existing operators, and new operators, related to CID fonts. Also information on Type 42 font dictionaries.

PostScript Language Reference Manual Supplement for Version 2016

Specifies the new Type 32 font format; additional information on Type 42; and additional information on the CID font file format.

SupplementPostScript Language Reference Manual, LanguageLevel 3 Specification and Adobe PostScript 3 Version 3010 Product Supplement

B.3 CID Font Character Collections

Technical Note #5078, *Adobe-Japan1-2 Character Collection for CID-Keyed Fonts*

Technical Note #5079*, Adobe-GB1-0 Character Collection for CID-Keyed Fonts*

Adobe-GB1-0 provides support for the GB 2312-80, GB 12345-90, and GB 1988-89 character set standards. Supported encodings include 7-bit and EUC.

Technical Note #5080, *Adobe-CNS1-0 Character Collection for CID-Keyed Fonts*

The Adobe-CNS1-0 Character Collection supports the Big Five and CNS 11643-1992 character set standards. Supported encodings include 7-bit, EUC, and Big Five.

Technical Note #5093*, Adobe-Korea1-0 Character Collection for CID-Keyed Fonts*

Adobe-Korea1-0 provides support for the KS C 5601-1992 and KS C 5636 character set standards, and select corporate variations thereof. Supported encodings include 7-bit and EUC encodings.

Technical Note #5094*, Adobe CJK Character Collections and CMaps for CID-Keyed Fonts*

Describes the national character set and encoding standards supported by each of the Adobe CJK character collection documents.

Technical Note #5097, *Adobe-Japan2-0 Character Collection for CID-Keyed Fonts*

Provides support for the JIS X 0212-1990 character set standard. Supported encodings include JIS (7-bit) and EUC (code set 3).

B.4 Adobe Type Manager Technical Notes

Technical Note #5070, *Benefits of Supporting the Adobe Type Manager Software API*

Explains the benefits, for both applications and end users, of using ATM as a system resource. Although mainly for software developers, this document also discusses why Type 1 fonts and the ATM API should be of interest to software developers.

B.5 Miscellaneous

Technical Note #0091, *Macintosh FOND Resources*

Describes the structure and use of Macintosh FOND resources, with examples of how PostScript font names are extracted and how the font file namecan be derived.

Technical Note #5003, *PostScript Printer Description File Format Specification Version 4.3*

Technical Note #5004, *Adobe Font Metrics File Format Specification, Version 4.1*

Specifies the AFM file format, which is an ASCII file that specifies both font-wide and character-specific font metrics and information. Version 4.1 contains information on CID-ordered AFM files for CID-keyed fonts.

Technical Note #5040, *Supporting Downloadable PostScript Fonts*

Explains how software applications and print drivers can handle Macintosh and Windows format Type 1 fonts for downloading.

Technical Note #5075, *Supporting Fonts in the PostScript Language Environment*

An overview discussion of how application software can do a better job of supporting fonts for PostScript printing. Published in 1992, this document does not discuss CJK issues, but contains other useful information.

Technical Note #5088, *Font Naming Issues*

Explains the technical differences between the PostScript font name, the font menu name, and the font file name for Macintosh, Windows, and UNIX systems. Includes naming conventions for CID-keyed and multiple master fonts.