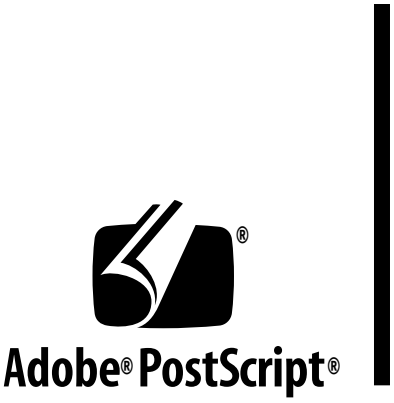


CID-Keyed Font Technology Overview

Adobe Developer Support

Technical Note #5092

12 September 1994

Adobe Systems Incorporated

Adobe Developer Technologies
345 Park Avenue
San Jose, CA 95110
http://partners.adobe.com/

PN LPS5092

Contents

CID-Keyed Font Technology Overview

1 Introduction

The CID-keyed font file format was designed for large character set fonts for use with PostScript™ language printing, Adobe Type Manager™ (ATM™) software, Configurable PostScript Interpreter (CPSI) software, and Display PostScript (DPS) software. It is the ideal format for Chinese, Japanese, or Korean fonts (referred to collectively as CJK fonts), and may also be used for Roman fonts with very large character sets. This document provides an overview of the format and describes the benefits and compatibility issues for both users and font developers.

The CID-keyed font file format is an open, published format which is specified in Adobe Technical Specification #5014, "The CMap and CID-Keyed Font Files Specification." This document specifies a new *file organization* that allows optimal flexibility and performance. The characters contained in a CID-keyed font are in the standard PostScript Type 1 format that is the industry standard for high-quality, cross-platform printing.

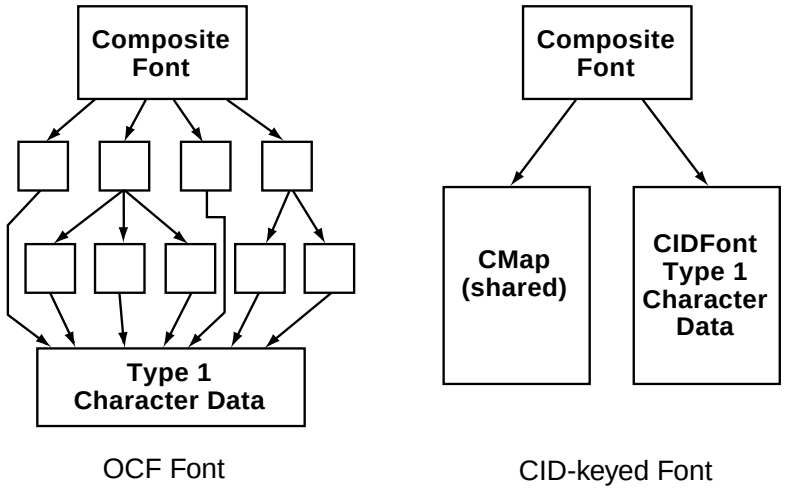
A major advantage of the CID-keyed font file format is the ease with which font developers can support a wide variety of character sets and encodings. It makes it easier for vendors to offer better products, and users benefit from the format's compatibility with a wide range of PostScript output devices. Performance will be considerably enhanced with future PostScript printers (version 2015 and greater) which will directly interpret CID-keyed fonts. Further performance enhancements will be seen with the new PostScript products for the Asian market that have hardware-assisted font rendering capability.

2 Background

Originally, a specification for Adobe's *Composite Font* format was published in the *PostScript Language Reference Manual, 2nd edition* (Addison Wesley, 1990). While this description explained the basic structure of a composite font, it did not give details of a complete font implementation. Adobe's implementation of this format, now known as the *Original Composite Font*

(OCF) format, used a complex font structure and disk file organization. Figure 1 illustrates the general structure of an OCF and a CID-keyed font, as they appear in the memory of a PostScript interpreter.

Figure 1 *The structure of OCF and CID-keyed fonts*



The complex structure of OCF intermediate fonts was designed to support printing of multiple character sets from multiple operating systems. The intermediate “fonts” can be used to select different sets of characters for printing from a Windows™ system than those used for a Macintosh® system. With a CID-keyed font, an operating system–specific CMap file is easily selected to provide the required encoding and character set, without the complex structure. CID-keyed fonts also set up a composite font, but their much simpler structure means they use less memory, and the interpreter can retrieve and rasterize character outlines much faster. While OCF format fonts offer high quality and will continue to be used by many users of existing font products, the format does not offer the flexibility and simplicity that are key benefits of the CID-keyed format.

As Adobe learned more about the Asian font market over the years, the OCF font format continued to evolve. The format was not documented or supported because it continued to change, did not meet all the needs of the market, and was not as extensible as it needed to be. If software developers had written software that depended on the structure of the OCF format, it would have been very difficult to switch to a more appropriate format.

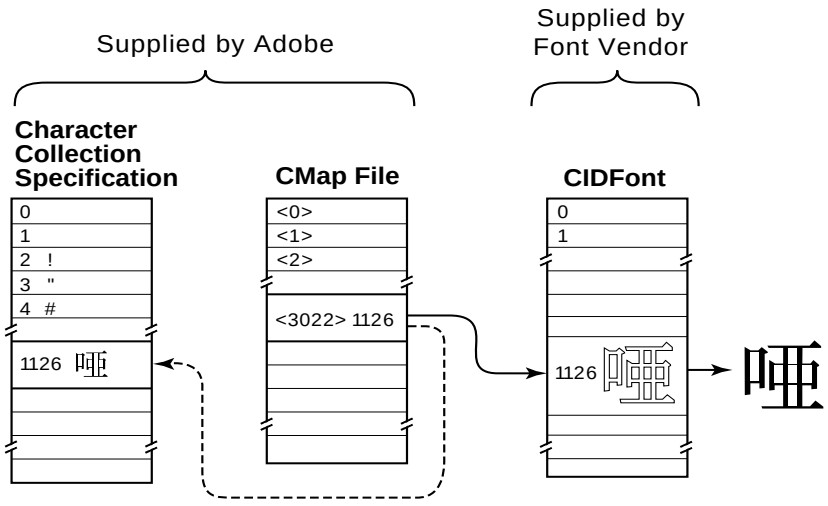
3 Overview of the CID-Keyed Font Architecture

The design of the CID-keyed font file format benefited from all that was learned by developing the OCF format. The CID-keyed font format does address the needs of the Asian market, and is designed to insulate font and

software developers from the inevitable evolution of the format that must occur to continue to meet industry needs. It is openly published and supported to encourage all font developers to use it.

Figure 2 illustrates the components of a CID-keyed font. The order of the components (left-to-right) also indicates the order of their development.

Figure 2 *Sequence and relationship of files used for CID-keyed fonts*



The sequence of development is as follows:

- ***Character collection****:* Adobe Systems develops a *character collection* document (see following section for more explanation) for a specific language. This document specifies all of the characters (and their CID numbers) needed for that language. Character collections go through a review process involving industry experts from the appropriate countries before being published for general use.

- ***CMap files****:* Adobe Systems develops one or more *CMap files* for each character collection. These files specify the correspondence between character codes and CID numbers for popular character sets and encodings. Other font developers may easily define their own character collections and CMap files, but it is expected that the ones supplied by Adobe will fit the needs of most font vendors.

- ***CIDFont files****:* If font developers use Adobe's published character collections and CMap files, they then only need to produce the *CIDFont* files that contain the character shapes. This can represent a significant savings in development resources.

This general architecture minimizes the amount of work a font developer has to do, yet still allows the necessary flexibility to use any desired non-standard character set or encoding.

4 CID-Keyed Font Components

The name *CID-keyed Font* refers to the *Character ID* (CID) numbers that are used to index and access the characters in the font. This method is more efficient for large fonts than the method of accessing by character name, as is used for Type 1 Roman fonts.

A CID-keyed font consists of one or more *CMap* files and a *CIDFont* file. The Character ID numbers in the font are based on a predefined and named *character collection*, and a specific *ordering* of that collection. These components are discussed in the following sections.

Character Collections

A *character collection* consists of an ordered set of all characters needed to support one or more popular character sets for a particular language. The order of the characters in the Character Collection determines the CID number for each character. Each CID-keyed font must explicitly reference the character collection upon which its CID numbers are based.

Adobe Systems publishes character collections for Chinese, Japanese, and Korean fonts; other font developers can reference these character collections, or they can develop and name their own character collections.

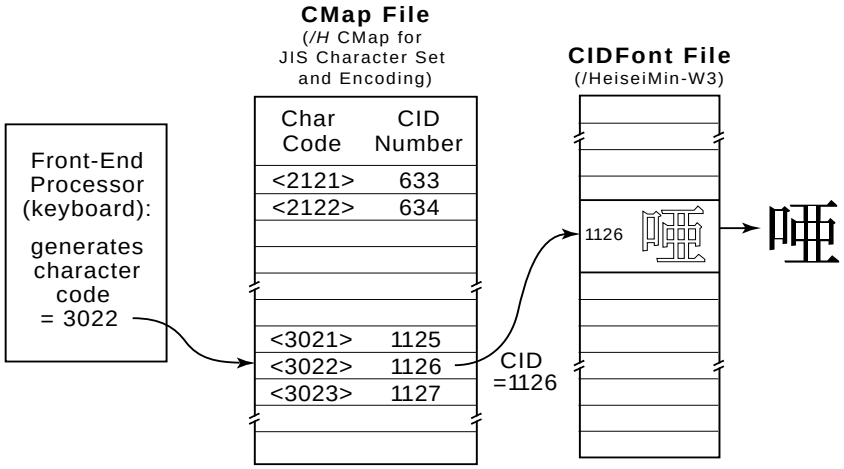
For example, the Adobe™ character collection for the Chinese simplified character set is *Adobe-GB1-0*, where *Adobe* is the registry name; *GB1* represents the first version of the ordering based on the Chinese GB 2312-80 character set, and 0 (zero) specifies that it is the base character collection to which additional supplements may be added. This naming convention allows for strict version and compatibility control, while also allowing the flexibility required by font developers.

CMap Files

CMap (Character Map) files specify the correspondence between a character code and the CID number used to access the character description in the CIDFont file. It is equivalent to the concept of an *encoding* as used in the Type 1 font format. Whereas a Type 1 font allows a maximum of 256 characters to be encoded and accessible at one time, users can access all of the thousands of characters in a very large CID-keyed font.

Figure 3 illustrates a CMap file and CIDFont file, and how character codes and character IDs are used to access characters in a CID-keyed font.

Figure 3 *Accessing characters with CMap and CIDFont files*



A CMap file can reference an entire character collection, or only a subset. It can also reference any other CMap file, without having to duplicate it, to provide extra font configurations (see section 5, “Rearranged Fonts”).

A common question is whether a font format can support the ISO 10646 and Unicode® character encoding standards. The CID format does provide for the double-byte character addressing that is required for Unicode fonts. The only other requirement is the addition of a CMap file, which Adobe provides, that specifies the correspondence between Unicode character code values and the CID numbers contained in the font. This inherent flexibility makes it very easy for developers to add Unicode support to their font products.

CIDFont Files

The CIDFont file contains the characters in the font, each of which is a computer language procedure that “draws” a given character shape for either display on the screen or for printing. The CIDFont file is the only component that most font developers will have to produce. The character procedures are in the same format as those in a Type 1 font, which is why CID-keyed fonts can be made compatible with many PostScript interpreters by adding, to the font package, a *compatibility module* that is installed in the user’s printer.

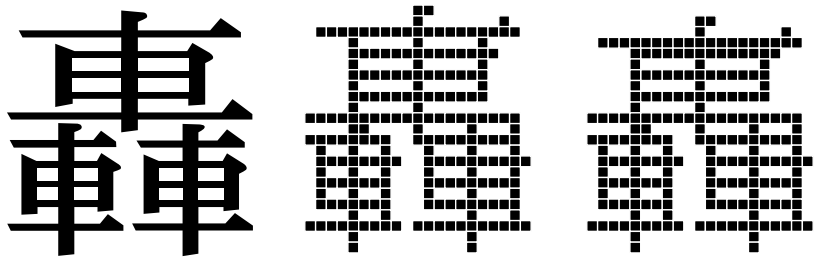
The CIDFont file also contains tables that help the interpreter locate the required characters and associated data, as well as additional information called *hints* which help the interpreter to create high-quality images of the characters at small sizes or at low resolutions.

New Counter Control Hints

One type of hint information that is particularly important for complex CJK characters is the *Counter Control* hint (*counters* are the white spaces between stems in a character). It helps insure that counters and overall proportions are rendered as accurately as possible, subject to the number of pixels available for a given size and resolution. This specialized type of hint information is not needed for ordinary Type 1 Roman fonts, but adds an essential degree of control for complex characters such as the one shown in Figure 4. Failure to add these hints can significantly hurt the quality of a font used with a printer containing the Type 1 Coprocessor, and the performance of a font used with other PostScript printers and ATM software.

Figure 4*a* shows the Chinese character that represents *rumble* (the sound), and *b* and *c* show the bitmap which is created for display at 24-point on a 72-dot-per-inch screen. The bitmap character in *b* shows the effect of having Counter Control hints in the font, and *c* shows how one of the counter spaces can collapse if these hints are not included.

Figure 4 *Effect of rendering with, and without, Counter Control hints*



a Original Character

b With counter control hints

c Without counter control hints

5 Rearranged Fonts

One of the most powerful features of the CID-keyed font file format is the ability to create a *rearranged* font. This feature allows font developers to create a font which consists of only a CMap file that references other fonts installed in the user's system. End users can utilize this feature by using the Adobe Type Composer™ software that allows users to create their own rearranged fonts.

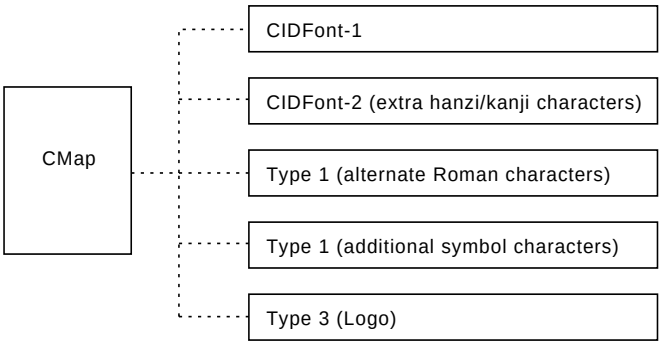
A rearranged font is created using a CMap file. It may reference one or more CID-keyed, OCF, Type 1, or Type 3 fonts. This feature provides great flexibility with minimal development effort and file storage requirements. There are no actual characters in the rearranged font; the CMap file serves

only as a template which describes the fonts from which characters are to be borrowed. It also contains a specification of how those characters correspond to the input codes.

The rearranged font behaves the same way as any other CID-keyed font: its name appears in font menus, it can be downloaded to a printer, and it can be used with ATM software. The advantage is that the size of the resulting CMap file is typically less than 30 kilobytes. Although the referenced font files must be in the user's system, this approach avoids the need for duplication of any of the component fonts.

Figure 5 illustrates an example of a rearranged font which "borrows" characters from two CID-keyed fonts, references two Type 1 fonts for alternate Roman and symbol characters, and one Type 3 font that may contain one or more company logos (Type 3 fonts are useful for complex artwork that may be unsuitable as a Type 1 font).

Figure 5 *CMap file used to create a rearranged font*



6 Benefits of the CID-Keyed Font Format

The CID-keyed font file format offers a wide variety of benefits for both users and font developers. Users are concerned about a variety of issues related to price, performance, and compatibility. Font developers are concerned about production efficiency, flexibility, and the need to protect their typeface designs, but these issues are particularly important to CJK font developers because of the large investment necessary to design, produce, and test those fonts. Following are the benefits of CID-keyed fonts:

- ***Extensible design:*** Support for additional character sets and encodings may be easily added to existing fonts. Font developers may start with support for a limited character set and then add characters, in a modular fashion, to reach additional markets. This is probably the single-most valuable feature of the CID-keyed font format for font developers.

- ***Performance:*** The performance of CID-keyed fonts is comparable to OCF fonts in compatibility mode with existing printers, but performance will be up to 50 percent faster with future native-mode printers with PostScript interpreters of version 2015 and later (however, performance can vary significantly depending on the quality of the PostScript language code created by the printer driver).

  Complex CJK fonts can create a performance problem because of the large number of characters that must be imaged for an average print job. The Adobe Type 1 Coprocessor, an ASIC, significantly enhances the performance of CID-keyed fonts. The recent development of an inexpensive version of this coprocessor will ensure its wide availability in a variety of new printers for the Asian market.

  Table 1 shows the rendering speeds, at 12-point and at 300-dpi, of two types of characters in three formats, with a RISC-based controller and with the Type 1 Coprocessor. These numbers include the time required for parsing and decrypting the font program, as well as for rasterizing, caching, and transferring the characters to the frame buffer for printing. Performance times can vary significantly depending on the clock speed and the type of controller used.

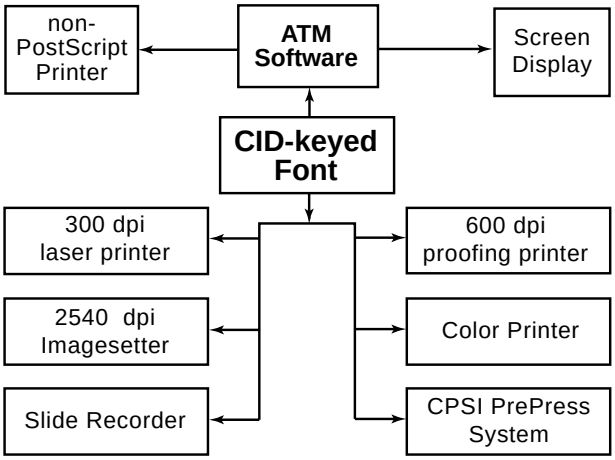
Table 1 *Rendering speeds for Type 1 Coprocessor and RISC Controller*

Character Type	**Format**	**RISC-based Controller** (characters-per-second)	**Type 1 Coprocessor** (characters-per-second)	**Percent Increase**
Roman	Type 1	94	370	293
Kanji	OCF	38	175	360
	CID	41	250	509

- ***Compatibility:*** Because the characters in a CID-keyed font are in the industry standard Type 1 format, they are compatible with a wide variety of operating systems, applications software, and output devices. For comparison, a TrueType™ font must be translated into a Type 1 font for it to be interpreted by PostScript imagesetters, and many service bureaus only accept jobs which use Type 1 fonts. Also, ATM software enables users to use CID-keyed fonts when printing to a wide variety of non-PostScript printers.

  Figure 6 illustrates the versatility and device independent quality of CID fonts by showing some of the output device choices available.

Figure 6 *Compatibility of CID fonts with a variety of output devices*



- ***Simplicity:*** The CID-keyed font format reduces the number of files needed, from nearly 100 for OCF fonts, to only one for CID-keyed fonts (CMap files, while required to make use of the CID-keyed font, are shared among all CID-keyed fonts). The simplicity of the format results in approximately a 15 to 20 percent savings in file size (see Table 2), and thus the font uses less of the printer's memory. This simplicity also means that fonts are easier to build, install, and verify. Even a 15 percent savings per font can be very significant when multiplied by the number of fonts handled by a user or font developer.

Table 2 *Sample file sizes for OCF and CID fonts (file sizes in megabytes)*

	OCF	CID	CID: % smaller
Ryumin-Light	5.463 MB	4.618 MB	15%
GothicBBB-Medium	4.125 MB	3.223 MB	20%

Note: The CID versions of fonts evaluated for this example contain approximately 150 more characters than their OCF counterparts.

- ***Quality:*** The new format continues to allow the addition of standard Type 1 hint information, which greatly improves the quality for small point sizes and low resolutions (see Figure 4 for an example of the effect of hint information on rendering quality). Also, the addition of Counter Control hints allows the extra control needed for complex CJK fonts.

- ***Portability:*** A single font file will work with ATM software (version 3.8), PostScript printers or imagesetters, CPSI software, and Display PostScript applications for the Macintosh, Windows, or UNIX® environments. This means a significant savings in development and packaging costs for the font developer, and enhanced ease-of-use for the end user.

- ***Elimination of redundancy:*** Once a CMap file is in memory, it can be shared across all fonts, whether they are full or subset fonts.

- ***Font Protection:*** CID-keyed fonts are computer programs, and therefore may be copyrighted, in most countries, like any software product. However, developers of large CJK fonts usually want some form of physical protection since each font is a much larger investment than that needed for a Roman-alphabet font. Physical copy protection mechanisms are as easy with CID-keyed fonts as with OCF fonts.

- ***Support:*** The OCF format is Adobe proprietary, while the CID format is published and fully supported by Adobe Systems.

In summary, these features offer developers the best potential to provide high-quality fonts with the kind of flexibility needed to meet market demands.

7 Compatibility Issues

The CID-keyed font format was designed for maximum compatibility with a wide range of system and applications software, as well as with most PostScript interpreters. Adobe Systems tests CID-keyed fonts with a wide variety of software applications that support CJK language fonts, and actively works to resolve any incompatibilities which are found.

With respect to output devices, CID-keyed fonts will be interpreted directly, in native mode, by version 2015 or later of the PostScript interpreter. For printing with pre-2015 versions, a *CID Support Library* module (available from the Adobe Developers Association) must be included in the font. This module makes CID-keyed fonts backwards compatible with existing PostScript Level 2 interpreters (version 2011 or later), and with Level 1 PostScript Japanese printers.

In order for CID-keyed fonts to be installed and used with popular operating systems, an operating system-specific font metrics file must accompany the CIDFont file. This includes a suitcase file for the Macintosh, a PFM (Printer Font Metrics) file for Microsoft® Windows, or an AFM (Adobe Font Metrics) file for UNIX systems.

The CID-keyed format is compatible with ATM 3.8 software for the Macintosh. It will also be supported by the next major release of ATM (subsequent to the current 2.5.1J) for the Japanese version of Windows. ATM 3.8 software for the Macintosh will also allow the use of CID-keyed fonts with the Apple® QuickDraw™ GX operating system that will ship with Macintosh System 7.5™.

8 Conclusion

The CID-keyed font file format offers major advantages over the OCF format, and font developers should definitely choose to develop CID-keyed fonts for the best quality, performance, and compatibility with PostScript printing. The relative simplicity and flexibility of CID-keyed fonts will enable font developers to more easily reach more markets with better products, and end users should benefit from price reductions that result from easier development and more competition.

Adobe Systems encourages font developers to use the CID-keyed font file format, and provides information and materials to assist in the development of CID-keyed fonts through the Adobe Developers Association. Software tools for developing CID-keyed fonts are available from third-party suppliers.

Index

T

U