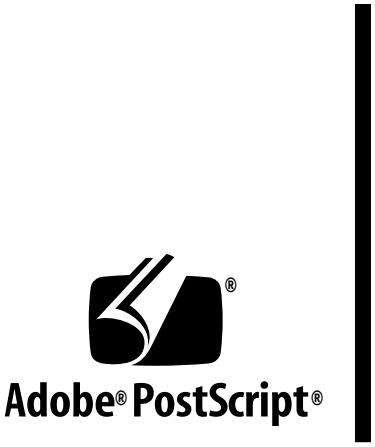

Font Naming Issues

Adobe Developer Support

Technical Note #5088

12 April 1993

Adobe Systems Incorporated

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

PN LPS5088

Contents

Font Naming Issues

1 Introduction

This document discusses the following aspects of font names:

- The PostScript® language **FontName**
- Font menu names
- Font file names
- The need for unique font names and the use of Vendor ID codes
- Font naming conventions for Latin, Japanese, and multiple master fonts
- Font names and document portability issues
- Font naming and compatibility with Adobe Type Reunion™ software

In addition, the following appendices are included:

Appendix A:	Font name style abbreviations
Appendix B:	Multiple master primary font axis labels
Appendix C:	Adobe PC font file name device codes
Appendix D:	Adobe PC font file name extensions
Appendix E:	Adobe PC font file style abbreviations
Appendix F:	Font naming bibliography

This information is intended to be of use to developers of fonts, font creation and manipulation software, and to systems and application software developers.

2 Font Names

This section discusses the three keywords used to specify a font name in a Type 1 font program: the **FamilyName**, **FullName**, and **FontName**. Although font programs from Adobe Systems contain all three entries, all are optional and software cannot rely on their presence.

From the font developer's perspective, font naming begins with consideration of the basic family name of the typeface being implemented. This name is used as the value for the **FamilyName** keyword. The complete name must

then be constructed, including elements denoting the style, weight, width, and character set. This complete name, which may include abbreviations but not spaces, is used for the keyword entry **FullName**. This must then be transformed into the **FontName**. Although this name may, theoretically, be as long as any PostScript language name — in practice, Adobe limits it to 29 characters (see section 2.2). This constraint may necessitate abbreviation of portions of some font names.

Generally, the **FontName** is the name used with the **definefont** operator in the font program to associate a name with a font dictionary in the *FontDirectory*. This is true for fonts in the Adobe Type Library™. Whatever name is used with **definefont** is the name that *must* be used with the **findfont** operator in the PostScript language code in a print job. Using **definefont** with a name already in the *FontDirectory* associates the name with the new font, with the result that the original font can no longer be referenced. Hence, it is essential that the name passed to **definefont** be unique for each font.

2.1 FamilyName and FullName

In addition to the PostScript **FontName**, font programs in the Adobe Type Library contain two name entries, **FamilyName** and **FullName**. Since the **FullName** is complete and not abbreviated, it is potentially the most useful form of the font name to present to users in a font menu. This is because in most environments the default font menu name is restricted in length. The **FamilyName** is also useful for the family portion of a hierarchical font menu, and can be extracted from the **FullName** to obtain the style attributes.

In environments where the font's Adobe Font Metrics (AFM) file can be accessed (such as in the MS-DOS® or Unix® environments), software can parse that file for the required font name keywords (see Adobe Technical Specification: "Adobe Font Metrics File Format Specification", Version 4.0).

The following are the font name components as specified in Adobe Type 1 font programs and their associated AFM files:

FamilyName

The **FamilyName** entry in a font or AFM file defines the family group to which the font belongs. For example, the typeface *Garamond Bold* belongs to a group of typefaces with the family name of *Garamond*. Some Unix, Display PostScript™, and other systems use the **FamilyName** to match against the **FullName** for sorting into family groups. Hence, this entry must match the corresponding portion of the **FullName**, and be suitable for display in font menus. All fonts that are stylistic variations of a unified design should share the same **FamilyName**.

FamilyName examples:

ITC Avant Garde Gothic
Berthold Garamond
Goudy Text MT
Minion

Note *For comparison purposes, the examples shown in this section for the* ***FamilyName****,* ***FullName****, and* ***FontName*** *are all taken from the same four typefaces.*

FullName

The **FullName** entry should show the complete name of a typeface in its "natural" form, including style and character set information (if any), and without abbreviations. This is the name that some systems and applications look at to determine full, unabbreviated font menu names.

The **FullName** begins with a copy of the **FamilyName** and is completed by adding style attributes — generally in this sequence: weight, width, slope, optical size (an example of optical size might be *10point* or *Display*), and special purpose attributes such as *Expert* (for the Adobe Expert character set fonts).

The *weight* portion of the name should correspond to the value associated with the **Weight** keyword in the FontInfo dictionary and in the AFM file. In general, the *weight* name should be included unless it is *Regular.* However, if neither a width nor a slope name exists for a face with weight *Regular*, the weight name should be included. This is suggested because a software program might extract the family name and expect to have a style name remaining to display in a sub-menu. For example, if a font's family and style is *Garamond Condensed Regular*, the *Regular* portion may be dropped. However, if the font name is *Garamond Regular*, the *Regular* should be retained.

In addition, a Vendor ID (see section 2.4) should be included if one is used for other fonts in the family. The **FullName** must be unique; numbers can be used in the name, but the name should not depend on them for uniqueness.

Note *Software which deals with these name entries must be prepared to find some Adobe and possibly other third party fonts that do not follow the general rules described in this document. If possible, software used to build a custom font menu should have an alternate strategy such as using the* ***FontName*** *if the* ***FullName*** *and* ***FamilyName*** *entries do not conform to the rules.*

FullName examples:

ITC Avant Garde Gothic Bold Condensed Oblique
Berthold Garamond Medium Condensed Expert
Goudy Text MT Dfr
Minion Regular Small Caps & Oldstyle Figures

2.2 The PostScript Language FontName

The **FontName** generally consists of a family name (specifically, the one used for **FamilyName**), followed by a hyphen and style attributes in the same order as in the **FullName**. For compatibility with the earliest versions of PostScript interpreters and with the file systems in some operating systems, Adobe limits the number of characters in the **FontName** to 29 characters.

As with any PostScript language name, a valid **FontName** must not contain spaces, and may only use characters from the standard ASCII character set. If abbreviations are necessary to meet the 29 character limit, the abbreviations should be used for the entire family. In some cases, some additional characters may be included after the family name and before the hyphen, such as a Vendor ID (see section 2.4) or a language label such as *Greek* or *Cyrillic*.

FontName examples:

AvantGarde-BoldCnOblique
GaramondBE-MediumCn
(but regular weight is: GaramondBE-Condensed; BE specifies that it is from the Berthold Library)
GoudyTextMT-Dfr
Minion-RegularSC

As mentioned above, the **FontName** is usually the name by which a font is known to the PostScript language interpreter. In addition, some applications in MS-DOS or Unix environments use this name to embed as a platform independent font reference in the application document. The **FontName** is also a good candidate for use in font menus customized by an application, and is used by Adobe Type Reunion software to create font menus on the Macintosh (see section 7).

2.3 Font Style Attributes

The style attributes in a **FontName** or **FullName** specify the weight, width, and other style features of the face, such as *bold italic* or *light condensed*. If the font has a special character set, such as the Adobe Expert set, the typeface name might be *Adobe Garamond Bold Expert*. In order to express this full name as a PostScript language **FontName**, and to ensure maximum

compatibility for more than one operating system, system-specific name length constraints must be considered for all platforms on which the font might be used.

A hyphen should be used between the family name and the style attribute to help applications parse for that information. Also, it is recommended that the family name use both uppercase and lowercase characters, as some parsing algorithms look for the family portion of the name by finding the first uppercase character that is followed by lower case characters.

Traditionally, various operating system and applications software conventions have made it necessary to include the style properties such as *weight* and *slope* in the font's name. In addition, the character set may be specified (for example, for the Adobe Expert sets) and, in the case of Japanese language fonts, it is helpful to designate the encoding and writing direction information. While the **FontName** is fairly descriptive and provides an ideal cross-platform font reference, the above-mentioned constraints often dictate that the name must be abbreviated.

2.4 The Need for Unique Font Names

If a PostScript **FontName** is not unique (in the general case where it is the name used with **definefont**), it can cause problems. If a font is downloaded with the same name as a font already in the interpreter's **FontDirectory**, the first font can no longer be referenced.

On the Macintosh, for example, font file names are derived algorithmically from the **FontName** entry — if this name is not unique, the resulting file name will not be unique. If the font file name duplicates the name of another font in the system, the font cannot be installed in the system.

There are two options for achieving unique font program names. One method is to try using a distinctly unusual name for a font. Common nouns, adjectives, and place names should not be used without some other element to make them unique.

The second and preferred option is to use a two or three character *Vendor ID* in the **FontName**, details of which are described below. The final choice as to which approach, or what mix thereof, is to be used, belongs to the font software vendor — but the need for uniqueness must be given serious consideration.

2.5 Vendor ID Codes

A Vendor ID is an abbreviation or mnemonic code for the company that sells to end users. It generally consists of two or three uppercase characters positioned at the end of the family name portion of a PostScript language **FontName**. *Using a Vendor ID is the recommended method for achieving unique font names.*

There are several reasons why the Vendor ID should be placed after, not before, the family name. There is an established practice for companies who license type designs to a variety of vendors to put their initials (such as *ITC*, for the International Typeface Corporation) before the family name. If font vendors put their Vendor ID before the family name, it is likely to confuse the user. Secondly, if the Vendor ID comes before the family name, it may interfere with the alphabetizing of names for the font menu. Some vendors prefer to group their fonts this way, but the majority of users do not.

In addition, if the Vendor ID is put at the beginning of the font family name, the effect of the Macintosh 5:3:3 rule (see section 4.1) means that the part of a font file name which allows the most information to be included (5 characters) is used for the Vendor ID, which limits the displayed family name to 3 characters. In this case, the Vendor ID assures uniqueness, but the user may see less information for the family portion of the font menu name. Adobe recommends putting the Vendor ID after the family name and before any style attributes.

Although a font software vendor's primary concern in naming a font program will be related to design and marketing issues, from a PostScript language and user's perspective the essential consideration is that every font program name and font file name should be unique. In order to help vendors create unique font program names, Adobe Systems maintains a Vendor ID registry in addition to a Unique ID registry.

Inquiries about Vendor ID registration or about obtaining **UniqueID** numbers should be addressed to:

Unique ID Coordinator
Adobe Developers Association
Adobe Systems, Inc.
345 Park Avenue
San Jose, CA 95110-2704
e-mail: fontdev-person@adobe.com

and in Japan:

Adobe Systems Japan
Adobe Systems Co., Ltd.
Yebisu Garden Place Tower
4-20-3 Ebisu, Shibuya-ku
Tokyo 150 Japan
Tel: +81-3-5423-8100
Fax: +81-3-5423-8209

It is important to address correspondence to the attention of the Unique ID Coordinator, Adobe Developers Association.

2.6 The Latin Font Naming Convention

Having considered the various components of a font name, the recommended form for a PostScript language **FontName** can be stated as:

< Family Name > < Vendor ID > - < Weight > < Width > < Slant >
< Character Set >

Spaces are not allowed. This naming convention is recommended so users will see a consistent order in font name elements. It may also aid in algorithmic interpretation of font names, but software must not depend entirely on the convention since many fonts do not comply. For examples of style abbreviations that are commonly used for the *Width*, *Weigth*, *Slant*, and *Character Set* fields of the name, see Appendix A.

3 Font Menu Names

How font names are defined for use in font menus varies with the operating environment, and in some cases, with the application. Font menu names are critical since they are what the user sees, and they are what applications usually embed in a document as a font reference. The user needs a clear, unambiguous representation of what fonts are installed in the system. From the point of view of a multi-platform application and for the sake of document portability, font menu names should be as similar as possible on each platform. Some early Adobe fonts do not strictly adhere to this rule.

Environments like the Macintosh, Microsoft Windows®, and NeXT™ have system-defined methods to display font menus, but an application can override the default menus if the intention is to present more complete font names and hierarchical menus.

The following section gives an overview of how font menu names are specified and used in major operating environments.

3.1 Macintosh Font Menu Names

The standard names used in Macintosh font menus come from the name of the FOND resource associated with a Type 1 outline font. FOND resource names are limited to 31 characters, and may contain spaces. Because of a bug in a major software application which limits font menu names to 30 characters, Adobe recommends staying within this lower limit. In addition, System 7 for the Macintosh requires names to be unique for the first 28 characters.

The font menu windows of some early Macintosh system software, as well as some major software applications, were fairly narrow in width. The result was that, for many fonts, the family name was visible but not the style attributes.

To compensate for this problem, Adobe put style attribute abbreviations at the beginning of font menu names. For example, the FOND resource name (and resulting default font menu entry) for *Helvetica Bold Oblique* would be *BI Helvetica BoldOblique*. This solved the most crucial user interface problem, but caused fonts to be inconveniently alphabetized in the menu. Adobe Type Reunion software solved the alphabetization problem for Macintosh users by using the PostScript language **FontName** for use in its custom font menu (see section 7). Adobe has since ceased producing menu names with style prefixes.

3.2 DOS Font Menu Names

Since there is no system-level support for fonts in the DOS operating system, each application defines its own conventions. Adobe Type 1 font packages for the PC include a *.INF file* which contains ASCII information including font menu names for major applications such as Ventura Publisher®, WordPerfect®, and Microsoft Word®.

The package also contains the Adobe Font Foundry software utility which converts Type 1 outline font programs into bitmaps for screen display or for printing to a non-PostScript-language printer. The resulting bitmap fonts, along with the font menu information in the .INF file, can be installed in major DOS applications.

Furthermore, there is also no standard form of font reference to embed in a document generated in the DOS environment. An application may embed either its own font menu name, the **FontName**, or some other version of the typeface name. The **FontName** is recommended as a general, cross-platform font reference solution because of its inherent compatibility with the platform- and device-independent PostScript language and the wide range of output devices supporting the language.

3.3 Windows Font Menu Names

In the Windows environment, font menu names are specified in the Printer Font Metrics (PFM) file. This file is roughly equivalent to an AFM file; the format is defined by Microsoft and the specification is available from Microsoft's developer support department.

With Windows 3.1, font families are no longer limited to the four standard styles of *Normal* (or *Plain*), *Bold, Italic*, and *Bold Italic*. Fonts for the Windows environment from the Adobe Type Library currently support four-member families for compatibility with earlier versions of Windows.

The limit for Windows font menu names is 31 characters. For compatibility with Macintosh applications, it is recommended that the names be limited to 30 characters.

3.4 NeXT Font Menu Names

Font menus in the NeXT system consist of full typeface names, arranged in hierarchical menus. The names are derived from the **FullName** entry in a font's AFM file. The significant advantage to this approach is that users see full, unabbreviated names in their font menus. Also, they see the actual style attribute name, such as *Medium* or *Extra Bold*, rather than the more generic system designation of just *Bold*. The hierarchical menu lets users focus on less information initially — just the family name, and then see the additional information for the style, once the family is selected.

3.5 Font Menu Names: Document Portability Issues

There is no standard method for translating font menu names from one platform to another. It is highly recommended that font developers consider it a top priority to name their fonts so that the menu names are the same on all platforms for which they provide fonts.

Some early font packages from Adobe Systems included different menu names in PC fonts than the ones used in Macintosh fonts. Some of these fonts have been updated and the problem corrected, while other fonts may be corrected in the future. In this case, cross-platform compatibility has been chosen as a priority over backward compatibility.

4 Font File Names

The font file name is the name of the file in which the outline font program is stored on a user's hard disk or on a file server. There are a number of issues to consider when choosing font file names. For example, in a Macintosh system, font file names are algorithmically derived from the **FontName**. This means that the **FontName** may be unique, but the shortened version of it may not be.

The conventions are dependent on the operating system and, for the DOS environment, on individual applications. Font developers who use commercial software products may have the file name derived for them from the font's family name, but it is important to understand how this works in order to minimize problems.

4.1 Macintosh Font File Names and the 5:3:3 Rule

For the Macintosh, the PostScript **FontName** is stored in the FOND resource in the *Style Name Table*. The font's family name and all style suffixes must correspond exactly to the font program's PostScript language **FontName**, including any hyphens. The printer font file name is derived from the name stored in this table using the *5:3:3 rule* (described below). Since Macintosh applications and drivers locate font files for downloading by applying this rule, it must obeyed.

To apply the 5:3:3 rule, the name must first be conceptually divided into "words", where each uppercase letter begins a new word. The font file name is then derived by concatenating the initial uppercase letter and up to 4 lowercase letters in the first word, and the initial uppercase letter and up to two lowercase letters of all subsequent words. A hyphen between the family and style portions of the name is dropped.

An example would be the font whose **FontName** is *Helvetica-BoldOblique*, — and the resulting file name is *HelveBolObl*. Another example would be the typeface Adobe Garamond Bold, whose **FontName** is *AGaramond-Bold* (in this case, the first word of the **FontName** was abbreviated by Adobe before the 5:3:3 rule was applied). The "words" are *A, Garamond,* and *Bold,* and the resulting file name would be *AGarBol*. In this case, the first part of the font file name is only a single character instead of its potential length of five.

The result of this abbreviation is that even though the **FontName** and FOND Resource file names are unique, the downloadable font file name might conflict with that from another vendor. Note that if another font were named *AGarlicBolivia,* it would have the same font file name as *A Garamond Bold*. The resulting font program could not be installed in the same file system with the font whose name was duplicated. Also, under System 7.1, the font file name produced by the 5:3:3 rule and the suitcase file name must not be identical.

A more detailed explanation of how Macintosh font name mapping is done is described in Adobe Technical Note #0091, "Macintosh FOND Resources." Additional information is available in *Inside Macintosh,* Addison Wesley, Volumes I and IV; and in *The Apple LaserWriter® Reference*, Addison Wesley, 1988.

4.2 PC Font File Names

For the PC environment, file names must consist of an eight-character name followed by a three-character extension. The first five characters of the printer font file, font metrics file, and bitmapped screen font file should be the same for a given typeface; their extensions are different. Details of the Adobe convention are given in the manual included with all Adobe PC font programs (see *Adobe Type Library: User Guide IBM PC Version*).

File names in the Adobe Type Library always consist of 11 characters: eight characters followed by a three-character extension. Adobe font file names have the following format:

ttsssppc.xxx

where:

tt	is the typeface
sss	is the typeface style code. (See the list of typeface style codes in Appendix E.)
pp	is the point size, if applicable. (Point size is specified only if the font is a single point size.)
c	is a device code for a bitmapped font. (See the list of device codes shown in Appendix C. If a device code is not required, an underscore is used in the eighth position.)
xxx	is the filename extension that identifies the file type. The extensions for Adobe fonts are listed in Appendix D.

Although this is the current Adobe PC file name standard, it cannot be guaranteed that this formula will be used in the future due to the limited number of family names which can be represented in this format. In cases where the font's identifying information does not use the entire eight character field, underscore (_) characters are substituted in the remaining positions (this is also required by some installers and is highly recommended).

For example, the following file names are font files for ITC Stone® Serif Bold Italic:

SRBI____.PFB	outline font software
SRBI____.AFM	Adobe Font Metrics (AFM) file

SRBI_____.FON	Microsoft Windows bitmapped font data containing more than one point size
SRBI_12C.FON	Microsoft Windows bitmapped font data for a specific point size and device (In this case, the “C” stands for VGA or MCGA)

4.3 NeXT Font Program File Names

The font program naming convention for the NeXT environment is explained in a document titled *Fonts in NeXTStep*™, available from NeXT, Inc. In addition to the naming convention, this document details the directory structure and installation procedure for font programs to be used on NeXT machines. The information may also be found in the Developers Library section of the NeXT operating system documentation.

5 Font Name Conventions for Multiple Master Fonts

This section discusses *font menu names* for multiple master fonts. While the The PostScript language **FontName** can be handled in the usual way, the menu names are more critical because of length constraints and the need to embed design coordinates in the menu name.

Multiple master font menu names for the Macintosh are specified as the name of the FOND resource. The axis labels differ for the two basic classes of multiple master fonts: *Primary fonts* are those that are pre-built by the font vendor and that generally correspond to the standard styles usually supplied for a given typeface family. Examples would include: *regular* (or *roman*), *italic*, *bold*, or *bold condensed*. *Custom fonts* are those created by the user.

Note *An explanation of multiple master font terminology and concepts can be found in Adobe Technical Note #5015, “Type 1 Font Format Supplement.”*

The following is the preferred naming convention for multiple master font menu names:

<FamilyNameMM> [Style/Char Set] _$Number_1$<$Label_1$>_…
$Number_n$<$Label_n$>

where items in brackets are optional. The components of the name are:

FamilyNameMM (Required)

FamilyNameMM specifies the family name of the typeface family, such as Minion™. The MM suffix is not a required part of the family name, but will be used by Adobe, at least initially, to identify multiple master typefaces. The

family name portion of the menu name for Adobe multiple master fonts is currently derived from the PostScript language **FontName** by using the 5:3:3 rule (in addition to using the rule to produce the font file name). However, doing this is not a requirement and may not be strictly followed in the future.

Style/Char Set (Optional)

This specifies style characteristics such as *italic*, and/or character set information such as *Expert* (for the Expert character set). An example would be *MinioMM ItEp* for Italic Expert. The preferred form of the resulting menu name would have been MinioMM ItaExp, but this would have been too long by the time the design coordinates (for three axes) were added.

The advantage of using industry-standard abbreviations is that it will be more recognizable by both users and software. Appendix A: “Font Name Style Abbreviations,” shows a list of style abbreviations which were agreed upon by a consortium of major font vendors. It is recommended that these be used wherever possible.

_ (underscore character, Required)

The family name/style/char set portion of the name and the design coordinate/label portion must be separated by an underscore character. This character tells Adobe Type Manager™ software (ATM™) that the font may be a multiple master font, and tells it where to begin looking for design coordinates. However, ATM does not assume that any font name containing an underscore is a multiple master font. An application that finds an underscore in a font name can determine whether the font is a multiple master font by using an ATM backdoor call (see Adobe Technical Note #5072, “Adobe Type Manager Software API: Macintosh”).

Number (Required)

This represents the design coordinate number which has been selected from the range specified by the designer. Digits 0 through 9 are allowed (but not decimal points or other punctuation), and any number of digits are allowed since the range is specified in the font. For a font with *n* axes, there will be *n* design coordinates specified in the name. The design coordinate numbers must be separated by one or more non-numeric characters; text labels are recommended if space permits, otherwise the use of a space or underscore character is recommended.

Label (Required)

The axis *Label* field is required to separate the design coordinates in the name, though it may optionally consist of only a single non-numeric character. Descriptive abbreviation labels are highly recommended for the

sake of the user interface; however if space is limited, Adobe recommends that the minimal single separator character be a space or underscore character.

The axis label may consist of any number of non-numeric characters (though two are recommended); it may have leading, trailing, or embedded spaces.

Labels for Primary fonts, which should be all uppercase, are listed in Appendix B: “Multiple Master Primary Font Axis Labels.” Labels for Custom fonts should be all lowercase, and use the generic axis labels *wt*, *wd*, and *op* for the appropriate axes contained in the font (new axis labels may be registered with Adobe Systems). The label is strictly a comment, and is not used by ATM, except as a separator.

5.1 Example Multiple Master Font Menu Names

The following are examples of allowed font menu names:

MinioMM ItEp_367 RG 485 NO 11 OP	This example shows *Minion* truncated by the 5:3:3 rule, *MM* distinguishes it from the non-multiple master version of Minion; and *ItEp* is an abbreviation of Italic Expert. the labels are uppercase characters because it is a Primary font.
GaramMM Bol_700wt470wd24op	Legal, but separating spaces recommended (especially before numbers).
GaramMM Bol_700 wt 470 wd 24 op	Recommended.
BodoniMM Ital_730 540 9	Legal, but use of labels would improve legibility.
MinioMM It_670_540_14	Legal, but use of labels would improve legibility.

These examples illustrate various options for separating or not separating the coordinates and labels, as well as the use of lowercase, generic axis labels for the Garamond fonts because they are *Custom* as opposed to *Primary* fonts.

5.2 Multiple Master Font Menu Name Length Constraints

Although both Macintosh and Windows allow font menu names to be 31 characters (and limited to 30 characters for application compatibility), System 7 for the Macintosh requires names to be unique for the first 28 characters. If the last three characters of a name are a label (which need not

be unique), a font menu name could have 30 characters in its menu name and still keep documents portable between the two environments. For example, the name

Bodoni Reg_700 wt 560 wd 24 op

has 30 characters (including spaces). The “op” portion of the label field is allowable because it fits within the intended limit of 30 characters, and the name does not depend on it for uniqueness.

5.3 Design Coordinates

The range of design coordinates for a multiple master typeface is specified in the font itself by the font designer/developer. Design coordinates are put in a font name for two reasons. Traditional terminology for the font weight, such as *light*, *medium*, and *bold*, is not adequate to express the range and number of weight variations possible with multiple master fonts.

Secondly, the design coordinates are useful in making multiple master fonts backward compatible with existing systems and applications. This is important because most software applications and drivers only know about a basic family name, and do not know to download specific instances of a font. Each multiple master font contains a new definition for the **findfont** operator that checks to see if a referenced *font instance* exists, and if not, generates it. This can be done by using the design coordinates specified in the font reference to create a **WeightVecto**r array — which is then used as the argument to the **makeblendedfont** operator to create the font instance.

Typefaces are often chosen by designers and publishers for their visual weight — or the *typographic color* (that is, the shade of gray) which they create on the printed page. Color is affected primarily by the stem weight, but relative character width, stroke contrast, and serif weight can also be a factor. For example, a typeface with a light stem weight, condensed design, and tight spacing can create a darker color than one with a slightly heavier stem weight, but having an expanded design and more open spacing.

Adobe has attempted to choose design coordinate ranges for various typefaces so that different fonts with the same coordinates yield equivalent results when printed. This is not required, but it is important to users for the numbers to have the same meaning for all fonts.

6 Font Naming for Japanese Language Font Programs

Japanese language font programs can have variations of character set, encoding, writing mode, and different types of kana characters which are best organized by using a font naming convention. It is up to the font software vendor to put this information in the name because most operating

environments do not have any other mechanism for communicating this data. Since the most common value for each field is the default value which can be omitted, many font programs will not need to have names as long as the specification indicates.

The naming convention presented below illustrates how Adobe Systems currently does Japanese language font naming. The PostScript language **FontName** for a Japanese language font program should be in the following form:

< Family Name > < Vendor ID> - < Width > < Weight > -
< Kana/Prop > - < Charset > - < Encoding > - < WMode >

The components of the name are as follows:

Family Name

The *Family Name* is the typeface or base name by which a related group of fonts are to be known. With Japanese language typefaces, some family names may include attributes which refer to weight or relative width, and which have traditionally been considered part of the family name. Also, all members of a family may not have the family name in their name.

Vendor ID

The *Vendor ID* is an optional two or three character identifier for the company which sells or publishes the font program. The Vendor ID should be in all uppercase characters. Each vendor should register their ID with Adobe Systems to help ensure uniqueness.

Width

The *Width* is a style attribute which describes the design width of a font relative to the regular or normal design; examples include: *condensed*, *compressed*, or *expanded*.

Weight

Weight is a style attribute designating the weight or degree of boldness of a font. Standard weight terms such as *Light*, *Medium*, or *Bold* may be used.

Kana/Prop

Kana/Prop is the size and spacing (proportional or fixed-width) attributes of the kana portion of a font.

Kana/Prop values:

(null)	large kana, fixed
KS	small kana, fixed
KO	old kana, fixed
KR	regular kana, fixed
KM	medium kana, fixed
KLP	large kana, proportional
KSP	small kana, proportional
KOP	old kana, proportional
KRP	regular kana, proportional
KMP	medium kana, proportional

Although these values may represent only a small number of those currently in use in the industry, additional values may be registered with Adobe Systems.

Charset

Charset is the character set used; additional values should be registered with Adobe Systems.

Charset values:

(null)	JIS-83 character set
Ext	NEC PC character set
Add	Fujitsu character set
83pv	Macintosh character set
NWP	NEC word processor

Encoding

Encoding is the name of the encoding vector used; values should be registered with Adobe Systems to help assure uniqueness.

Encoding values:

(null)	JIS encoding
RKSJ	Romaji Kana Shift JIS
EUC	Extended Unix Code
SJ	Shift JIS

WMode

WMode indicates the writing mode (direction of writing).

Values:

H	Horizontal
V	Vertical

For example: MinchoABC-Light-H is a font program where *Mincho* is the family name, the Vendor ID is (a fictitious) *ABC*, the weight is *Light*, and the writing mode is *H* (horizontal). The width, Kana/Prop, character set, and encoding attributes are all the default values of: *standard width*, *large kana with fixed widths*, *JIS-83 character set*, and *JIS encoding*, respectively.

An example of a longer name would be: MinchoABC-Bold-83pv-RKSJ-V, where *Mincho* is the family name, *ABC* is the Vendor ID, *Bold* is the style (weight), *83pv* specifies the Macintosh character set, *RKSJ* specifies the Romaji Kana Shift JIS character set, and *V* indicates that the writing mode is vertical.

7 Adobe Type Reunion Software Compatibility

Adobe Type Reunion software is a utility program for the Macintosh which presents users with a hierarchical font menu. The advantage is that the user sees a shorter, more organized font menu than the default system menu. The first level menu shows only alphabetized family names. Selecting a family name with the cursor causes a second level menu to appear which shows the styles available for that family. This approach shortens the font menu and correctly groups styles of one family which otherwise would be incorrectly grouped in the default system font menu.

7.1 Historical Font Menu Issues

Adobe Type Reunion software attempts to address two historical problems with font menus in the Macintosh system. One is that menu windows in both system utilities and in some applications were often quite narrow. The result was that style attributes following the family name often could not be seen by the user. Adobe attempted to solve this problem by putting style codes before the family name in the font menu names: for example, *Helvetica Bold Oblique* would appear in the font menu as *BI Helvetica BoldOblique*. This ensured that the user could at least see the style abbreviation (for example, *BI*), but it caused the styles of a family to be alphabetized by the style code and hence not be grouped together by family name. It also did not solve the problem of long font menus since every style had its own FOND resource and hence its own entry in the font menu.

An additional problem is that font families on the Macintosh can be constructed so that there is only one FOND per family. In this case, one menu name must supply all of the information for an entire family, and style variations are selected from the system style menu (typically allowing regular, bold, italic, and bold italic styles). The result is that font menus are not always as clear and descriptive as they might be.

7.2 Adobe Type Reunion

Adobe Type Reunion software overcomes the limitations mentioned above by extracting the **FontName** from the *Style Name Table* in the FOND Resource located in the suitcase file. This table contains a Base Font Name as well as fields for style suffixes. For example, the Base Font Name field might contain *Janson*, while the style suffix field might contain "-Bold".

This approach mainly helps fonts which have a FOND for each font in the family (as do fonts from the Adobe Type Library). The entries in the Style Name Table are not restricted in length; as a result, fonts with longer names may have more style information available in this table than there is in the FOND resource name.

Adobe Type Reunion considers everything after the first hyphen to be style attributes. In general, use of multiple hyphens is discouraged. However, the word *condensed* is special-cased (if it is the first word in the style string) to be included with the family name. The font program name should still be, for example, *Garamond-CondensedBold*, but it will be presented in the menu with a family name of *Garamond Condensed*, and have a style of *Bold* in the submenu.

When naming fonts in a font creation program, it is important to be accurate in entering only the family name as the Base Font Name, and only true style attributes for the "suffix" entries. Most problems with font names have resulted from either intentional or unintentional subverting of this model.

Also, some commercial font creation programs or utilities may make assumptions about names entered by users, and not allow all of the name components to be explicitly specified. For example, if a user names a font *GaramondNew*, the name may be divided at the second uppercase letter, and the resulting family name will be *Garamond* and the style attribute will be *New*. Ideally, the software should allow users to explicitly override any names derived algorithmically from other names.

7.3 Options with Font Menu Names

Font developers have two basic options when producing their fonts. One is to create a separate FOND for every outline font file, which is the best choice for compatibility with Adobe Type Reunion. With the default System 7 font menu, these fonts will have each family member listed separately. Adobe Type Reunion users will see the fonts grouped together under a single family name, and style names will appear in the sub-menu.

Generating a font family with separate FONDs has the disadvantage, initially, that there is a separate suitcase file for each style for the user to deal with. However, with System 7, the developer needs only to select all suitcase files but one and drag them to the unselected suitcase to have them all in one file.

With this approach, there will still be separate FOND resources for each style (unlike with the *FOND merge* approach discussed below), but they will all be included in a single file.

The other scenario is to use a *font harmonizer* or *style merger* utility. These generally have the effect of merging the actual FOND resources of each member of the family (not just the suitcase files) into a single FOND resource that contains references to multiple font family members. This shortens the font menu by displaying only a single family name for a typeface, and its style variations, in the font menu. Users must then access style variations from the keyboard or by choosing from the standard Mac font style menu. This approach works well for users who do not use Adobe Type Reunion software, and for families of four styles or less with standard style names. Larger families and ones with non-standard style names should probably not be merged.

Appendix A

Font Name Style Abbreviations

Following is a list of names and abbreviations for components of the *style* portion of font and menu names. In cases where there are constraints on name lengths, standardization of the abbreviations are expected to aid both users and software that needs to handle fonts in an intelligent manner. This list was originally compiled by Apple Computer; additional entries have been added by Bitstream and Adobe Systems.

Weight

Blk	Black	Md	Medium
Bd	Bold	Nd	Nord
Bk	Book	Po	Poster
Dm	Demi[1]	Rg	Regular
DS	Display	Su	Super
Hv	Heavy	Th	Thin
Lt	Light		

1. *Demi* is sometimes used as a weight, sometimes as a modifier.

Width

Cm	Compressed	Ex	Extended
Cn	Condensed	Nr	Narrow
Ct	Compact		

Slope

Ic	Inclined	Obl	Oblique
It	Italic	Up	Upright
Ks	Kursiv	Sl	Sloped, slanted

Miscellaneous (usually used to modify other styles)

Dm	Demi	Ult	Ultra
Sm	Semi	X	Extra

Design and/or "special purpose"

A	Alternate font[1]	Ou	Outline
Alt	Alternate[2]	Rm	Roman
Dfr	Deutsche Fraktur	Rd	Rounded
Exp	Expert	Scr	Script
In	Inline	SC	Small caps
MM	Multiple master‡	Sh	Shaded
Or	Ornaments	Sw	Swash
OS	Oldstyle, Old Style	Ti	Titling
OsF	Oldstyle figures	Typ	Typewriter

1. The abbreviation *A* typically distinguishes two fonts which differ only in the design of one or a few characters. For example, AmericanTypewriter-Light differs from AmericanTypewriter-LightA in the design of *R*, *e*, *&*, and *$*.
2. The abbreviation *Alt* designates a font whose character set is meant to supplement another font. The design characteristics of the two fonts are identical. For example, ACaslon-AltRegular contains long-s, ct, and st ligatures which match the design of ACaslon-Regular.

Appendix **B**

Multiple Master PrimaryFont Axis Labels

The following abbreviations are standardized for use as axis labels in primary font names. Custom font names for multiple master fonts will use the generic lower case labels such as *wt*, *wd*, and *op*. In either case, the label's purpose is to clarify a preceding value.

Weight axis

Long Name	Abbreviation	Long Name	Abbreviation
Extra Light	XL	Bold	BD
Light	LT	Black	BL
Regular	RG	Extra Black	XB
Semibold	SB		

Width axis

Long Name	Abbreviation	Long Name	Abbreviation
Extra Condensed	XC	Semi Extended	SE
Condensed	CN	Extended	EX
Semi Condensed	SC	Extra Extended	XE
Normal	NO		

Optical size axis

Long Name	Abbreviation
Optical Size	OP

Appendix C

Adobe PC Font File Name Device Codes

This appendix lists the device codes and file name extensions used by Adobe Systems for font file names for the PC platform. Font-handling software should be aware that other vendors may use other conventions.

Device Codes (c)

The device codes for bitmapped printer fonts are as follows:

H	Portrait (300 dpi)
@	Landscape (300 dpi)

The device codes for Microsoft Windows bitmapped screen fonts are as follows:

A	EGA Low Resolution, CGA
B	EGA High Resolution, Hercules Adapter
C	VGA, MCGA
E	IBM 8514/A
F	User-defined

The device codes for Ventura Publisher bitmapped screen fonts are as follows:

I	IBM CGA or compatible, IBM EGA Low Resolution or compatible
J	IBM EGA High Resolution or compatible, Hercules Card or compatible, Xerox 6065/ AT&T 6300

K	IBM VGA or compatible, IBM PS2/30 MCGA or compatible, MDS Genius Full Page, Wyse WY-700, STB VGA Extra/ EM, AST-VGA, AST-VGA PLUS, ATI VGA WONDER, ATI VGA WONDER-16, Genoa SuperVGA, SuperVGA High Resolution, Paradise VGA PLUS, PLUS16, PROF.

Appendix D

Adobe PC Font File Name Extensions

This appendix lists the file name extensions used by Adobe Systems for font file names for the PC platform. Font-handling software should be aware that other vendors may use other conventions.

Font File Name Extensions

The following list shows the file name extensions used in the naming convention discussed in section 4.2.

File name extension	Description	Created by
.INF	Font information used by installer and the Font Foundry program	Supplied in font package
.PFB	PostScript outline font and data for the Font Foundry program	Supplied in font package
.AFM	PostScript font metrics and data for the Font Foundry program	Supplied in font package
.SFP or .SFL	PCL bitmapped printer font Portrait or Landscape	Font Foundry
.FON	Microsoft Windows bitmapped screen font	Font Foundry
.PFM	Microsoft Windows font metrics	PostScript — Installer PCL — Font Foundry
.VGA, .EGA, or .CGA	Ventura Publisher bitmapped screen font	Font Foundry
.WID	Ventura Publisher font metrics	PostScript — Installer PCL — Font Foundry

File name extension	Description	Created by
.ALL	WordPerfect font metrics	PostScript — Installer PCL — Font Foundry
.PRD	Microsoft Word font metrics	PostScript — Installer PCL — Font Foundry

Appendix **E**

Adobe PC Font File Style Abbreviations

This table contains a sampling of the abbreviations used in Adobe PC font file names. Appendix A contains style abbreviations agreed upon by major font vendors for general use and where the file name may not necessarily limited to 8 characters as with PC fonts. The table in this appendix is mainly for information purposes; because of evolving conventions from various vendors, it is not possible for software to parse PC font file names and get reliable results.

BL	Black	D	Demi	NB	Narrow-Bold
BLI	BlackItalic	DI	DemiItalic	NBO	Narrow-BoldOblique
BLO	BlackOblique	DO	DemiOblique	NO	Narrow-Oblique
B	Bold	EB	ExtraBold	O	Oblique
BI	BoldItalic	EBO	ExtraBoldOblique	PS	Poster
BO	BoldOblique	EK	ExtraCompressed	RG	Regular
BSL	BoldSlanted	H	Heavy	R	Roman
W	Book	HI	HeavyItalic	S	SemiBold
WI	BookItalic	HO	HeavyOblique	SI	SemiBoldItalic
WO	BookOblique	I	Italic	SB	Semibold
K	Compressed	KB	KursivBold	SBI	SemiboldItalic
C	Condensed	KX	KursivRegular	SL	Slanted
CB	CondensedBold	L	Light	T	Thin
CBO	CondensedBoldOblique	LI	LightItalic	TI	ThinItalic
CEB	CondensedExtraBold	LO	LightOblique	U	Ultra
CL	CondensedLight	M	Medium	UK	UltraCompressed
CLO	CondensedLightOblique	MI	MediumItalic	UL	UltraLight
CO	CondenseOblique	N	Narrow	ULI	UltraLightItalic

Appendix **F**

Font Naming Bibliography

- Adobe Technical Note #0091, “Macintosh FOND Resources.” Presents details of the FOND resource data structure. It gives an example of how PostScript language font names are represented in a FOND resource and how the font file name is derived using the 5:3:3 rule.

- Adobe Technical Note #5072, “Adobe Type Manager Software API: Macintosh.” This document explains how an application can use ATM back door calls to determine whether a font is a multiple master font, given the font menu name.

- Adobe Technical Note #5015, “Type 1 Font Format Supplement” Specifies all extensions to the Type 1 format since publication of the “ Adobe Type 1 Font Format” book, including those for multiple master fonts.

- Adobe Technical Note #5087, “Multiple Master Font Programs for the Macintosh.” Describes the BLND resource that is required to make multiple master fonts work in the Macintosh environment. Font menu name conventions for multiple master fonts are discussed.

- “Printer Font Metrics Files Specification.” Microsoft Windows Device Development Kit, Volume: Printers and Fonts Kit. Available from Microsoft Corporation, this document specifies the format for the PFM files required for font installation in the Windows environment. It explains how the font family name and style attributes are specified for the Windows font menu.

- Adobe Type Library: User Guide IBM PC Version. This manual is supplied in all PC fonts from the Adobe Type Library. The information on PC font file names in this document is taken from the manual.

- *Inside Macintosh*, Volumes I and IV, Addison Wesley, 1987. These books describe the basic architecture of Macintosh fonts, including where and how font menu and font file names are handled.

- The Apple LaserWriter Reference, Addison Wesley, 1988. Basic reference for information on FOND resources — including the Style Name Table, and a variety of font related issues for the Macintosh environment.

Index