



# Macintosh FOND Resources

---

*Adobe Developer Support*

Technical Note #0091

31 March 1992

Adobe Systems Incorporated

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

PN LPS0091

Copyright ©1987, 1990, 1992 by Adobe Systems Incorporated. All rights reserved.

NOTICE: All information contained herein is the property of Adobe Systems Incorporated. No part of this publication (whether in hardcopy or electronic form) may be reproduced or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written consent of Adobe Systems Incorporated.

PostScript is a registered trademark of Adobe Systems Incorporated. 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.

Except as otherwise stated, any reference to a "PostScript printing device," "PostScript display device," or similar item refers to a printing device, display device or item (respectively) which contains PostScript technology created or licensed by Adobe Systems Incorporated and not to devices or items which purport to be merely compatible.

Adobe, AdobePS, Acrobat, Adobe Acrobat Distiller, Adobe InDesign, Extreme, PageMaker, PostScript and the PostScript logo are trademarks of Adobe Systems Incorporated. Macintosh is a trademark of Apple Computer, Inc., registered in the United States and other countries. OpenType, Windows and Windows NT are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. All other trademarks are the property of their respective owners.

*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

---

## **Macintosh FONDR esources 5**

- 1 Introduction 5
  - FOND Resource Overview 5
  - Fractional Data Representation 6
  - Font Style Code 6
- 2 FOND Resource Format 7
- 3 Family Style Property Table 9
- 4 Font Association Table 10
- 5 Width Table Format 11
  - Width Table Entry Format 11
- 6 Style Mapping Table Format 12
  - Style Encoding 13
  - Font Class 13
  - Style Indexes for Derived Fonts 14
  - How to Derive Printer Font Names—An Example 14
  - Font File Names 16
  - Encoding Table Format 16
- 7 Kerning Table Formats 17

## **Appendix: Changes Since Earlier Versions 19**

## **Index 21**



# Macintosh FOND Resources

---

## 1 Introduction

Several kinds of information are needed to successfully use screen and downloadable printer fonts on the Apple® Macintosh®. This information is stored in a FOND resource. Each Adobe™ typeface has an associated FOND resource that provides information for using its screen and printer fonts. This resource will be used by the Macintosh Plus Font Manager and versions 3.0 and later of the LaserWriter® driver.

### 1.1 FOND Resource Overview

Each FOND resource provides the following information for its typeface:

Global Data	Information pertaining to the typeface family as a whole and directory data for the rest of the FOND resource.
Font Association Table	List of FONT resources associated with this typeface.
Width Tables	Width tables for related fonts in a family. These are the widths of the characters in a one-point font. From this table, the widths of the characters at any point-size can be derived. Note that these widths have fractional values.
Kerning Tables	Kerning tables for related fonts in a family for a one-point font.
Style Mapping Table	Table providing the name of the printer font associated with each style (bold, and so on) available for that family.
Family Style Properties	Approximations of the character width changes needed for styles derived by QuickDraw™.

All FOND resources are of type FOND and have the same names as the associated FONT resources.

A FOND resource number is the same as the resource number of the zero point size FONT or NFNT resource for the plain style of the family. This will be a unique number in the range 1 to 32766, assigned by Apple Technical Support. Note that NFNT resources work only with the new font manager. Also note that FONT resources only support numbers 0 to 255.

## 1.2 Fractional Data Representation

Much of the data in a FOND resource is represented in fractional quantities. Non-integral values are represented as two-byte quantities with a four-bit integer part and a twelve-bit fractional part.

Negative values can appear as either one's complement or two's complement, depending on the version of the FOND. (See the information for Font Version Code in section 2.)

## 1.3 Font Style Code

The Macintosh and the FOND resource encode the style attributes of a font in a one-byte code, each bit of this code specifies the presence or absence of a particular style. Table 1 lists the code.

**Table 1** *Style attributes in one-byte code*

<i>Bit</i>	<i>Meaning</i>
0	Bold
1	Italic
2	Underline
3	Outline
4	Shadow
5	Condensed
6	Extend

More than one bit can be turned on to express a style.

## 2 FOND Resource Format

A Macintosh FOND resource starts with a set of global information pertaining to the font family. This information is followed by the offset values to the data tables; the offsets are followed by the tables themselves. Figure 1 shows the contents of a Macintosh FOND resource. The detailed explanations of the entries appear after the diagram and in the sections that follow.

**Figure 1** *FOND resource format*

Bytes		Bytes	
2	Flag Word	4	Style Mapping Table Offset
2	Family ID Number	18	Family Style Property
2	First Character	4	Reserved for International Fonts
2	Last Character	2	Font Version Code
2	Ascent	2+6*(#fonts)	Font Association Table
2	Descent	typical 2000	Width Table
2	Leading	typical 100	Style Mapping Table
2	WidMax	variable	Kerning Tables
4	Width Table Offset		
4	Kerning Table Offset		

### Flag Word

Describes the type of spacing to be used for this font. Bits are assigned meanings as follows:

- 0000 Proportional spacing
- bit 15 on Mono spacing
- bit 14 on Do not use family widths (see Width Tables)
- bit 13 on Do not use Family Style extra widths (see Family Style Table)
- bit 12 on Use Family Style extra widths even if fractional widths are disabled (see Family Style Table)

### Family ID Number

A number from 1 to 32766 assigned by Apple Technical Support.

### First Character, Last Character

Character codes for the first and last characters in the font. For Apple fonts, these usually have values of 0 and 255, respectively.

### Ascent, Descent

The maximum height above the baseline and depth below the baseline reached by characters in this font. The depth is usually a negative number. These are fractional quantities.

Leading, WidMax	Maximum leading and maximum character width for the family. The leading value is usually set to 1.0. These are fractional quantities.
Offsets	4-byte offsets from the beginning of the FOND resource to the beginning of the table.
Family Style Property Table	Table of extra character widths for derived typeface styles.
Font Version Code	If the value of this word is 2 or greater, then negative numbers in the FOND resource are kept as 2's complement; font version 0 and 1 have 1's complement negative values.

### 3 Family Style Property Table

The Family Style Property Table (Figure 2) lists the amount of extra widths that should be added to the widths of the characters in the derived styles of a family. These extra widths are fractional quantities of a one-point font.

When a screen font for a particular style is derived from the Roman screen font, that style's character widths are calculated by multiplying the extra width for the style by that font's point size and adding the result to the Roman screen font width. Roman screen font widths are obtained from the Widths Table (see section 5).

*Note* If a printer width table exists for the font and style being used, then the widths in that table should be used rather than the extra widths calculated from the Family Style Property Table.

The Family Style Property Table has nine two-byte entries—one for each style. If styles are combined (as in bold italics), the extra widths will also be combined.

**Figure 2** Family Style Property Table

Bytes	
2	Unused—set to 0
2	Extra width for Bold text
2	Extra width for Italic text
2	Extra width for Underline text
2	Extra width for Outline text
2	Extra width for Shadow text
2	Extra width for Condensed text
2	Extra width for Extended text
2	Undefined

## 4 Font Association Table

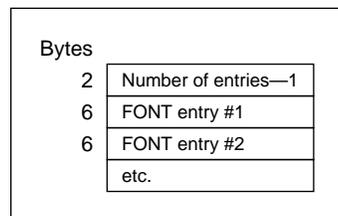
The Font Association Table is a list of all FONT or NFNT resources related to the font family being described by the FOND resource.

The table begins (see Figure 3) with a word containing the number of entries in the table minus one. This word is followed by the FONT entries themselves, one for each FONT resource associated with the FOND resource.

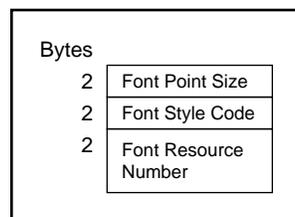
Each FONT entry (Figure 4) consists of three words:

- an integer representing the point size of the font
- the style of that font (expressed according to the standard Macintosh style code; see section 1)
- the resource number associated with that FONT resource

**Figure 3** *Font Association Table*



**Figure 4** *FONT Entry Format*



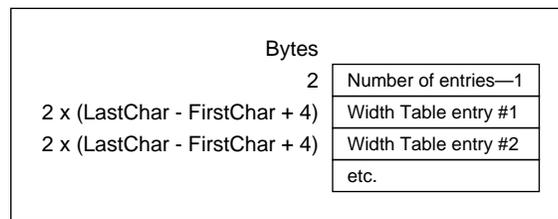
## 5 Width Table Format

The Width Table contains a series of printer font widths for each of the fonts associated with the FOND resource. Each entry in the Width Table is a list of character widths for one of the fonts in the typeface family. These widths are used with styles that have separate screen fonts. If the style does not have a corresponding width table, character widths can be determined from the Family Style Property Table (see section 3).

Widths are expressed as fractional quantities of a one-point font. As always, fractional numbers consist of a four-bit integer followed by a twelve-bit fraction.

The Width Table (see Figure 5) begins with a word that holds the number of entries minus 1. This is followed by the entries themselves, one for each screen font in the family.

**Figure 5** *Width Table Format*



### 5.1 Width Table Entry Format

Each Width Table entry (see Figure 6) describes the character widths for fonts that have coordinated printer fonts. Each entry begins with a word that holds the font's style code (see section 1).

The style word is followed by a list of fractional character widths, one for each character in the font. The last width is that of the missing character, the character used to denote an unprintable control character. (In Apple fonts, the missing character is often a square.)

Each Width Table entry ends with an unused word, set to zero.

**Figure 6** *Width Table Entry Format*

Bytes	
2	Style Word
2	Code 01 Character Width
2	Code 02 Character Width
	etc.
2	"Missing Character" Width
2	Unused Word

## 6 Style Mapping Table Format

Macintosh applications must associate a correct printer font with each font style. An application must be able to derive the name of the appropriate printer font from the font and style code. This is complicated by the fact that printer fonts have suffix names such as "Light" and "Demi," which are not available on the Style menu.

Printer font names consist of a base font name (such as Times) followed by one or more optional suffixes (such as Bold, Italic, and Condensed). Each of the 48 possible font style codes has associated with it a list of zero or more suffixes that, together with the base font name, make up the printer font name corresponding to that style.

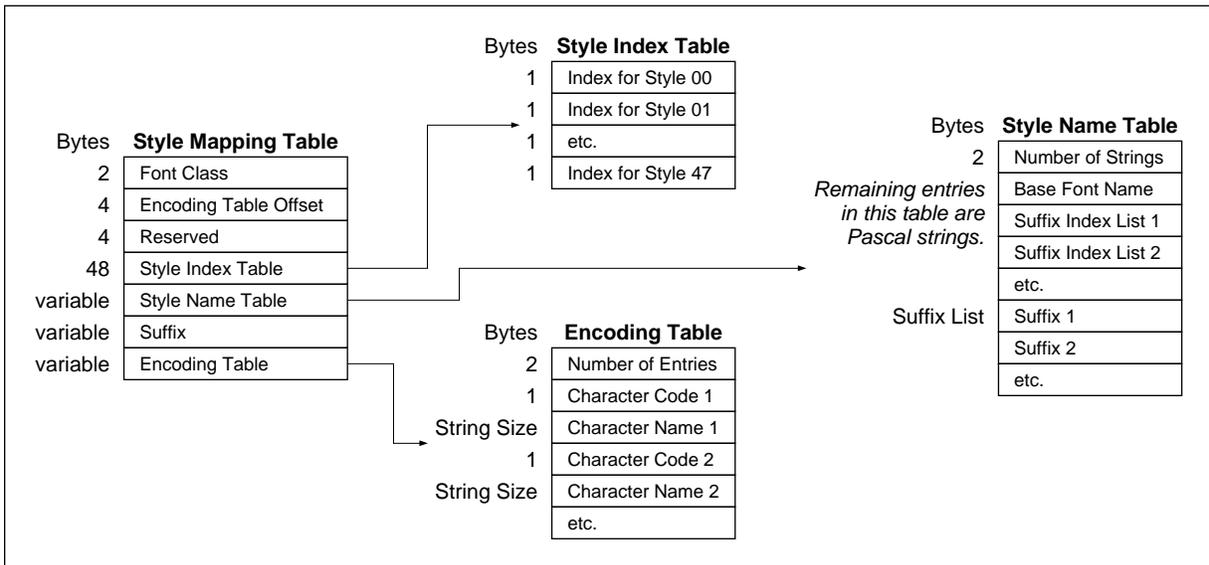
Three tables within the Style Mapping Table are used to determine printer font names. These tables (in reverse order of appearance in the FONDR resource) are

Suffix List	List of suffixes (Pascal strings) used in the typeface family. The Suffix List is part of the Style Name Table.
Style Name Table	Table of entries in the suffix list to be used with each style. Each is a Pascal string whose characters holds an index into the Suffix Table.
Style Index Table	Table with 48 entries (corresponding to style codes 0 through 47), each of which is an index into the Style Name Table. (See the note on style encoding.)

A fourth table, the Character Encoding Table, is discussed in section 6.6, "Encoding Table Format." It is not used to derive printer font names.

In Figure 7, note that the offset to the encoding table is relative to the beginning of the Style Mapping Table and not to the beginning of the entire FONDR resource.

**Figure 7** *Style Mapping Table Format*



## 6.1 Style Encoding

The Style Mapping Table uses a slightly different mapping code from that used by other parts of the Macintosh font manager. The bit for underline is dropped to produce the code that appears in Table 2.

**Table 2** *Style Mapping bits*

<i>Bit</i>	<i>Meaning</i>
0	Bold
1	Italic
2	Outline
3	Shadow
4	Condensed
5	Extend

## 6.2 Font Class

The first entry in the Style Mapping Table is the Font Class of the fonts associated with the FOND resource. The Font Class is a sixteen-bit value whose bits define how various styles are to be rendered on the LaserWriter. The meaning of each bit is listed in Table 3.

**Table 3** *Font Class bits*

<i>Bit</i>	<i>Meaning</i>
0	Set if font name needs coordinating.
1	Set if font requires the Mac Vector re-encoding scheme.
2	Set if font is “outlineable” by changing <b>PaintType</b> to 2.
3	Reset to allow outlining to be simulated by a “smear and whiteout” technique.
4	Reset to allow emboldening by a “smear” technique.
5	Set if emboldening is simulated by point size increase.
6	Reset to allow obliquing for italic.
7	Reset to allow automatic simulation of condensed style.
8	Reset to allow automatic simulation of expanded style.
9	Set if font requires re-encoding other than the Mac encoding.
10 – 15	Reserved.

Fonts that provide their own faces are assigned classes that prevent style simulations from occurring.

### **6.3 Style Indexes for Derived Fonts**

If no FONT resource is available for a particular style, the corresponding entry in the Style Index Table might hold an index to the font face to be used for the derivation of that style, or it might also hold a zero, in which case the plain face will be used for style derivation.

### **6.4 How to Derive Printer Font Names—An Example**

To determine the name of the printer font associated with a given style in the current typeface, the style code (as described above) is used as an index into the Style Index Table. The number retrieved from this table is an index into the Style Name Table.

The entry obtained from the Style Name Table is a Pascal string. Each of the string’s characters holds an index into the Suffix List. These indexes point to the suffixes that should be concatenated onto the Base Font Name (in the Style Name Table) to produce the final printer font name. The total length of the name must be less than 255 characters in length.

As an example, let us say that an application is trying to determine the printer font associated with the BoldItalic style of the mythical Scriptus font, whose Style Name and Suffix tables appear in Figure 8.

**Figure 8** *Sample Style Name Table and suffix list for Scriptus font*

Index		
0	7	Number of Strings
1	"Scriptus"	Base Font Name
2	"\2\5\6"	Index lists for style suffixes
3	"\2\5\7"	(as Pascal strings; first number in each is character count)
4	"\3\5\6\7"	
5	"."	Suffix
6	"Demi"	Suffix
7	"Oblique"	Suffix

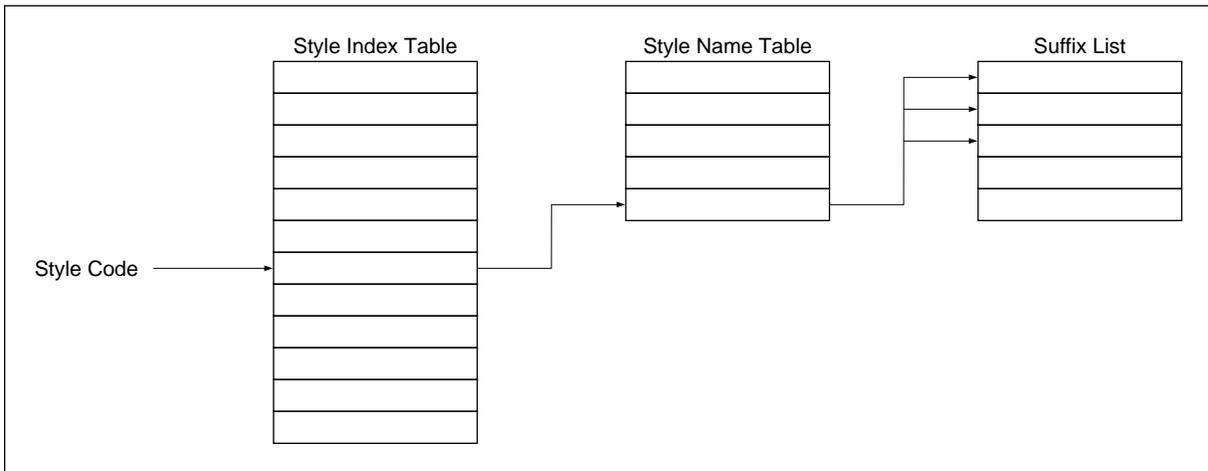
*Bold* and *Italic* correspond to bits 0 and 1 of the style code, respectively. Thus, the code for BoldItalic would be 3.

The style code directs us to entry number 3 in the Style Index Table, whose value is, say, 4. Fetching the fourth item from the Style Name Table returns a string with three characters containing character codes 5, 6, and 7.

*Note* Because the Suffix List is part of the Style Name Table, these indexes count from the beginning of the Style Name Table. In particular, note that the number of strings counts as an entry and has index value 0.

We now have the pieces that make up the printer font name associated with the style and typeface family. Stringing the suffixes located at positions 5, 6, and 7 in the Style Name Table onto the end of the body font name, we get Scriptus-DemiOblique as the name of our printer font. Figure 9 presents a schematic overview of this process.

**Figure 9** Overview of name derivation



## 6.5 Font File Names

The file name of the downloadable font that is to be sent to the LaserWriter is derived by using the “5:3:3” rule. Initially, the name must be conceptually divided up into “words”, where each uppercase letter begins a new word. The font file name is found 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 parts of the name is dropped.

For example, the file in which the *Scriptus-DemiOblique* from our example resides would be named *ScripDemObl*. The file in which the PostScript™ language font *NewCenturySchlbk-Bold* resides is *NewCenSchBol*.

## 6.6 Encoding Table Format

The encoding table (see Figure 7) is a list of character codes and names to be overlaid on the printer font encoding. It is a list of changes to be made to the PostScript language character encoding for the font. This is used for printer fonts that have unencoded characters.

The table starts with a word holding the number of entries. Each entry in the table consists of a one-byte character code followed by the PostScript language name of the character that is to be assigned that code. The character names are in the form of Pascal strings.

For example, if the encoding table contained the following:

```

2
168
diamond
169
heart

```

the characters named *diamond* and *heart* would be assigned character codes 168 and 169, respectively. The characters already assigned to these codes would become unavailable.

## 7 Kerning Table Formats

Kerning is the altering of intercharacter spacing to achieve a more pleasing appearance. Many characters look too far apart or close together when placed normally beside each other because of their shapes. To compensate for this, character pairs are moved closer together when they appear in print. This is referred to as pair kerning, since the amount that the right-hand character is moved, or kerned, depends on the character with which it is paired.

Each entry in the Kerning Table (see Figure 10) is a table of character pairs and kern amounts for a particular font style. This kern amount is the amount of space that should be added to or removed from the space between a particular pair of characters. Each table begins with a style word, identifying the style to which the table applies, the number of entries in that table, and then the entries themselves.

**Figure 10** *Kern table formats*

Bytes	<b>Kerning Table</b>	Bytes	<b>Kerning Table Entry</b>
2	Number of entries—1	21	Style Code
variable	First Entry	2	Number of Kern Pairs
	Second Entry	4	Kern Pair Entry #1
	etc.	4	Kern Pair Entry #2
			etc.

Each kerning pair entry (Figure 11) has the one-byte character codes of the first and second characters in a pair, and then the fractional amount by which the pair's intercharacter spacing should be altered when the two characters

appear together in print. This kerning distance is given for a one-point font. The actual amount by which the spacing should be changed is obtained by multiplying the kerning amount in the table by the point size of the font.

**Figure 11** *Kern pair entry format*

Bytes	
1	1st Character in Pair
1	2nd Character in Pair
2	Kern Amount

# Appendix: Changes Since Earlier Versions

---

## **Changes since August 13, 1991 version**

- Document was reformatted in the new document layout and minor editorial changes made throughout.
- Paragraph 6.5, Font File Names, was changed to more accurately describe how font file names are derived.

## **Changes since July 19, 1990 version**

- Updated the addresses on the cover page.

## **Changes since June 14, 1987 version**

- The document has been reformatted to conform to the Adobe Technical Note document layout. Content has not changed.





# Index

---

## F

### FOND resource

- Family Style Property Table 9
- Font Association Table 10
- font style code 6
- format 7
- fractional data representation 6
- Kerning Table formats 17
- overview 5–6
- style attributes
  - one-byte code 6
- Style Mapping Table 12–17
  - deriving printer font names 14
  - encoding table format 16
- Font Class 13
- font file names 16
- Style Encoding 13
- Style Index Table 12
- style indexes for derived fonts
  - 14
- Style Name Table 12
- Suffix List 12
- Width Table format 11

## M

### Macintosh

- FOND resource 5–18

