## Storing a Font Name in a Document

One problem with identifying fonts by font family ID rather than by name is the plethora of font families for the Macintosh. Many share the same font family ID, and even though the font the user wants is present in the System file, another font with the same ID may appear in a font menu. Another problem is that one font family may have different IDs on different computer systems, so that when the application opens the document using this font family on a different computer system, it can't find the proper font, even though it is there, and substitutes another.

If you've stored the name of the font in the document, you can find its font family ID by calling **GetFNum**. However, if the font isn't present in the system software where the user opens the document, **GetFNum** returns 0 for the ID. Zero is also, you may remember, the system font ID. In this case you need to double-check the name of the font from the document against the name of the system font, as illustrated in the following program example:

```
// Checking a font family ID against the font name

#include<Fonts.h>
#include<OSUtils.h>

Boolean GetFontNumber(Str255 fontName, short *fontNum)

{
// GetFontNumber returns in the fontNum parameter the number for the font
// with the given font name. If there's no such font, it returns FALSE.

    Str255 systemFontName;

    GetFNum(fontName, fontNum);
    if (*fontNum == 0) {
        // Either the font was not found, or it is the system font.
        GetFontName(0, systemFontName);
        return EqualString(fontName, systemFontName, FALSE, FALSE);
    }
    else
        // If the font number was not 0, the font is available.
        return TRUE;
}
```

Storing a font's name rather than its font family ID is a more reliable method of finding a font, because the name, unlike the font family ID, does not change from one computer system to another. You may also want to store the checksum of a font with its name, to be sure that the version of the font is the same on different computer systems.