# Composed and Decomposed Names for Fonts

## 1 Question

*Does the Mac OS X use decomposed names for fonts and Linux/Windows use composed, and GUIs use both?*

## 2 Answer

Not quite. The composed/decomposed difference comes from the **filesystem**, not from the fonts or the operating system as a whole. GUIs do not choose a form either; they display both forms identically, so you cannot see which one a name uses.

### 2.1 Font names inside the font file

The family, full and PostScript names are stored in the OpenType `name` table. The font producer chooses their encoding, and the OS does not change it. In practice these names are almost always NFC (composed), and PostScript names are restricted to ASCII anyway.

### 2.2 File names on disk

| System | Behaviour |
|---|---|
| macOS, HFS+ (before 10.13) | Forces decomposed form (Apple’s NFD variant) on every file name it stores. |
| macOS, APFS (10.13 and later) | Stores the name exactly as given and compares names ignoring normalization. Names can be NFC or NFD. |
| Linux (ext4, btrfs, etc.) | Stores raw bytes with no normalization. Names are usually NFC because keyboards and input methods produce NFC. |
| Windows (NTFS) | Stores UTF-16 with no normalization. Names are usually NFC. |

On a current Mac you will often still find NFD names. They come from files created on HFS+ volumes, from older archives and installers, and from some Apple frameworks and tools that still decompose names when they write them.

### 2.3 Where it causes problems

A byte-level comparison fails when one side is NFC and the other is NFD. Examples are a font database built by a script, a name typed in a `.tex` file, and a file copied from a Mac to Linux. On Linux the two forms are even two *different* files that can sit side by side. On APFS they resolve to the same file.

### 2.4 Checking a name

```
import os, unicodedata
```

```
for n in os.listdir("."):
    if not unicodedata.is_normalized("NFC", n):
        print(repr(n), "-> not NFC")
```

The command `ls | xxd` also shows the difference. For example, ‘é’ is `c3 a9` in NFC and `65 cc 81` (e + combining acute) in NFD.

The safe practice is to normalize both sides to NFC before you compare or index names, and to rename font files to NFC (or plain ASCII) when you move them between machines.